POLITECNICO DI TORINO

Master's Degree in
Communications and Computer Networks Engineering

Master's Degree Thesis

Integration of navigation sensors based on advanced Bayesian estimation methods

Supervisors

Prof. Fabio DOVIS

Dr. Alex MINETTO

Dr. Gianluca FALCO

Candidate

Oliviero VOUCH

A.Y. 2020/2021

Abstract

Global Navigation Satellite Systems (GNSSs) represent a ubiquitous radio-navigation technology offering reliable and long-term stable localization services with global coverage. In the framework of outdoor positioning and navigation, there are some critical applications, such as autonomous driving for land-vehicles, whose strict requirements, in terms of ranging-accuracy and service availability, cannot be met by a standalone GNSS receiver. A typical approach to overcome GNSS limitations and to pursue high-precision Positioning, Navigation and Timing (PNT) involves fusing the former system with an Inertial Navigation System (INS). A INS/GNSS integrated navigation unit leverage the complementary characteristics of the two navigation sensors in order to enhance the accuracy and robustness in the localization solution. Depending on the degree of information sharing between the coupled units, multiple hybridization strategies are available. Among them, a Tightly Coupled (TC) architecture builds on a centralized processor, the navigation filter, which exploits low-rate GNSS noisy measurements (pseudorange and Doppler shift) to correct high-rate INS estimates to the inertial position, velocity and attitude states of a moving target. The navigation filter, in the flavour of a Bayesian estimator, must rely on some discrete-system state-space formulation, which involves the definition of a state-vector, the actual object of the estimation process, and the identification of convenient models to characterize state and measurement dynamics. On top of that, statistical models for both state and measurement noises are needed too.

The state-of-art solution for fusing INS and GNSS is represented by the Extended Kalman Filter (EKF), where the aforementioned models are handled in a linearised fashion. While the process model, describing the evolution of inertial dynamics, is mostly linear thanks to the high INS-rate, the measurement model, which stems from satellite-based multilateration, is inherently non-linear. Based on these premises, this thesis targets the study of more advanced Bayesian estimation approaches better fitting with non-linearities. A first methodology, still confined within the class of Kalman filters, deals with the Unscented Kalman Filter (UKF), which relies on the Unscented Transform (UT) function to avoid model

linearisation. However, all Kalman-based strategies might be strongly penalized by their underlying assumption of Gaussian-distributed input measurements. In fact, when travelling multipath environments, such as down town roads, noisy GNSS-observables might not be normally distributed and this would reflect on a performance decay due to mismodelling of their probability densities. The former issue steers the research towards Sequential Monte-Carlo (SMC) estimation and, specifically, towards the Particle Filter (PF). This estimator, besides preserving non-linearities, offers enough flexibility to accommodate multiple density models for the description of noise statistics. The joint synthesis between the UT-concept, at the basis of the UKF, and the Sequential Importance Sampling (SIS) idea, the operational trademark of any PF, gives rise to the Unscented Particle Filter (UPF), which sets as the innovative and high-complexity proposal aimed at enhancing the integrated system capability.

A crucial requirement to achieve accurate state estimation lies on the appropriate representation of noise statistics. In a dynamic system which fast evolves in dynamic environment, measurement noise frequently behaves as a non-stationary and non-ergodic stochastic process. As such, static (or semi-static) Bayesian formulations might fail to track it properly. Therefore, as a first optimization pattern, it is explored the application of adaptive integration schemes able to promptly mirror environmental changes on the filter statistical information. Starting from low-complexity innovation-based (IAE) and residual-based (RAE) approaches, a more sophisticated strategy exploiting redundant measurements is developed. Furthermore multipath, which identifies the harshest error source affecting GNSS ranging observables, is addressed too; its mitigation is pursued through the elaboration of a portable, low-complexity and self-contained signal-processing module allowing to relax bias injections on the navigation solution.

To pursue a performance validation, the proposed Bayesian algorithms, considered both in legacy and adaptive implementations, are embedded in a software environment engineered to simulate a TC-architecture which fuses real INS and Global Positioning System (GPS) data. For testing purposes, it is employed a dataset with measurements collected during a on-field campaign. The hybrid filters' estimation accuracy is assessed by comparing the experimental trajectories against a sub-centimetre accurate ground-truth. Then, a Root-Mean-Square Error (RMSE) metric is chosen to mathematically quantify the loss budget w.r.t. the ideal solution. In light of the above, the ultimate proposal is to verify whether the expected accuracy gain, theoretically entailed by the investigated filters, truly justifies the increase of complexity over a plain EKF approach.

Table of Contents

List of Tables

List of Figures

Acronyms

GNSS Global Navigation Satellite System

GPS Global Positioning System

GPST GPS-Time

UERE User Equivalent Range Error

GLONASS Globalnaya navigatsionnaya sputnikovaya sistema

BDS BeiDou Navigation Satellite System

EU European Union

ESA European Space Agency

U.S. United States

DoD Department of Defence

ToA Time-of-Arrival

MEO Medium-Earth Orbit

GEO Geostationary-Earth Orbit

IGSO Inclined Geosynchronous Orbit

CDMA Code Division Multiple Access

FDMA Frequency Division Multiple Access

SIS Signal-in-Space

PRN Pseudo Random Noise

INS Inertial Navigation System

IMU Inertial Measurement Unit

MEMS Micro Electro-Mechanical System

ECEF Earth-Centered Earth-Fixed

UTM Universal Transverse Mercator

CDF Cumulative Density Function

DCM Direction Cosine Matrix

RF Radio Frequency

IF Intermediate Frequency

ADC Analogue-to-Digital Conversion

AGC Automatic Gain Control

ML Maximum Likelihood

CAF Cross-Ambiguity Function

PLL Phase-Lock Loop

DLL Delay-Lock Loop

LC Loosely Coupled

TC Tightly Coupled

LOS Line-of-sight

NLOS Non-Line-of-sight

C/N_0 Carrier-to-Noise-density ratio

PVT Position, Velocity, Timing

PNT Positioning, Navigation and Timing

LS Least-Squares

WLS Weighted Least-Squares

FSM Finite-State Machine

r.v. random variable

p.d.f. Probability Density Function

MMSE Minimum-Mean-Square Error

RMSE Root-Mean-Square Error

RMS Root-Mean-Square

HMM Hidden Markov Model

MAP Maximum a-Posteriori

KF Kalman Filter

EKF Extended Kalman Filter

UKF Unscented Kalman Filter

UT Unscented Transform

MC Monte-Carlo

SMC Sequential Monte-Carlo

PF Particle Filter

IS Importance Sampling

SIS Sequential Importance Sampling

SISR Sequential Importance Sampling Resampling

UPF Unscented Particle Filter

AUPF Adaptive Unscented Particle Filter

GEV Generalized Extreme Value

IAE Innovation-based Adaptive Estimation

RAIM Receiver Autonomous Integrity Monitoring

RAE Residual-based Adaptive Estimation

RMNCE Redundant Measurement Noise Covariance Estimation

FOSD First-Order Self-Difference

SOMD Second-Order Mutual Difference

RTK Real-Time Kinematic

DSP Digital Signal Processing

TTFF Time To First Fix

Chapter 1

Radio Navigation and Inertial Navigation systems

1.1 Global Navigation Satellite Systems

Global Navigation Satellite System (GNSS) is a general concept used to identify a broad family of space-based networks which allow to determine positioning and timing information, with respect to an absolute reference frame, by relying on a constellation of satellites. To supply an exhaustive definition, it is due saying that a satellite-based navigation system is a passive (listen-only) radio-ranging system continuously broadcasting synchronized Radio Frequency (RF) signals to terrestrial users. Such waveforms, then, allow for the evaluation of Position, Velocity, Timing (PVT) information using one-way ranging from a set of spacecrafts orbiting the Earth [1]. Despite the great popularity and the wide spectrum of nowadays applications involving the use of satellites, space-based navigation is not such an old concept. However, the underlying radionavigation paradigm takes its roots much earlier, after World War II.

1.1.1 The principle of satellite-based radionavigation

The conventional approach for GNSS-based localization, also referred to as Time-of-Arrival (ToA) positioning, involves two main steps: the collection of ranging measurements from a set of visible reference transmitters and the resolution of a multilateration problem.
As regards the first point, the distance between the receiver, hereinafter referred to as user, and each satellite, acting as reference point, is estimated through a passive ToA measurement, which requires accurate synchronization between the two communicating endpoints. Being infeasible, for cost-complexity reasons, to

have the user clock perfectly aligned to that of the transmitter, the reciprocal temporal misalignment is typically considered as an additional unknown in the positioning context. In other words, a conventional receiver is involved with the solution of a four-dimensional problem: estimation of the three components of its spatial position plus the bias of its clock. Once distances are computed from a minimum of four space-trackers (as many as the number of involved unknowns) and considering, for sake of simplicity, a bi-dimensional set-up, a geometrical position can be unambiguously retrieved from the intersection of three circles (i.e. trilateration) as represented in Figure 1.1.

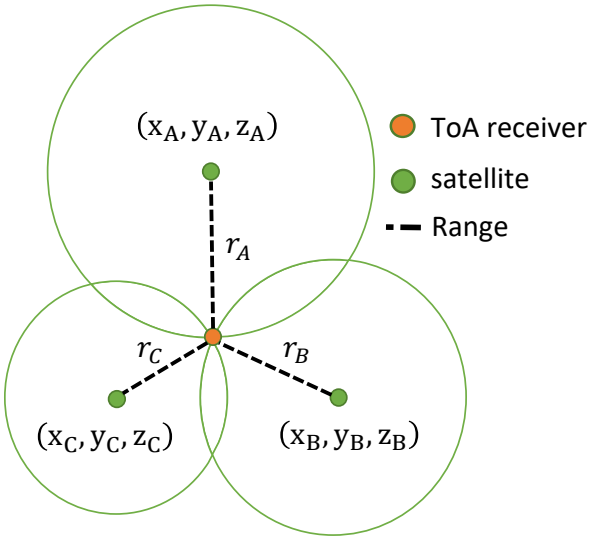


Figure 1.1: 2-D trilateration based on Time-of-Arrival (ToA) measurements with synchronized transmitters and receiver.

Extension to 3-D positioning, in the aim of estimating the third spatial component (i.e. height), requires to account for spheres in place of circles. The range measurement from a first reference satellite S1 localizes a terrestrial observer on a spherical surface of positions centred in S1. The retrievement of another ranging measurement from a second satellite S2 restricts the unknown receiver location on a circular line of positions. Further, a third distance estimate from S3, reduces the former circle to two points. Eventually, a fourth reference transmitter positions the receiver unequivocally.

1.1.2 Navigation Satellite Systems

Nowadays, multiple satellite systems are actively operated both for military and civilian applications. In the following, a short description of the most popular ones providing global coverage is offered, with the primary intent of reviewing their fundamental characteristics in terms of architectural organization and signal planning. Among them, special attention is devoted to the Global Positioning System (GPS) system, since all the GNSS ranging data managed within this thesis pertain to the latter constellation.

GPS

Within the class of GNSSs, the GPS, originally Navstar GPS, represents the oldest space-based navigation project that was developed, for military purposes, by the U.S. Navy and the Air Force in the late 1960s. Later, in 1973, the Department of Defence (DoD) approved the basic architecture and the first satellite was launched in 1978. The nominal GPS constellation involved a fleet of 24 Medium-Earth Orbit (MEO) satellites circling the Earth twice a day (celestial time) at an altitude of about 20200 Km. Satellites were nearly-uniformly arranged on six equally-spaced orbital planes with a 55° inclination relative to the equatorial plane and a mutual separation of 60° between the corresponding ascending nodes. Each plane consisted of a minimum of four slots hosting baseline satellites. In this sense, the original 24-slots arrangement was conceived to grant users with at least four visible satellites from nearly any point on the Earth. Over the years, the original constellation has been modified by expanding the slots to include extra-satellites, in the aim of improving global coverage and enhancing system availability. At the time of writing, the number of operational GPS satellites amounts to 31 [2, 3]. As regards frequency planning, the L-band (1 GHz - 2 GHz) was selected to accommodate two carrier frequencies: L1 at 1575.42 MHz and L2 at 1227.60 MHz. Each carrier, then, identified the central frequency of a 20 MHz bandwidth which was carrying the BPSK modulated spectrum of a Pseudo Random Noise (PRN) ranging code, useful to implement Code Division Multiple Access (CDMA) scheme to grant orthogonality between transmissions from different satellites. Furthermore, two signals were transmitted over the L1-band, one for civil-users and the other for DoD-authorized users (military-service). On the contrary, the L2 band hosted the military signal only [1].
Looking at the GPS architecture, but this is true for any GNSS, it organizes across three operational segments: *space segment*, *control segment* and *user segment*. The space segment consists of the whole satellites' constellation whose major function is the downlink transmission of radionavigation signals, including storage and re-transmission of the navigation message sent by the control segment. As regards the latter, it consists of a global network of ground facilities that track the GPS

satellites, monitor their transmissions, perform analyses and send commands and data to the space constellation. Finally, the user-segment includes GPS-receiver equipment that physically acquires the RF signals broadcast by the satellites and extracts useful information to supply end-users with coordinates and time [4].

Other global satellite systems

Although the final assessment primarily addresses GPS signals and their corresponding measurements, the research carried out in this thesis applies to all satellite systems and the explored methodologies can even be tailored to multi-constellation signal processing. Hence, it is the right time to briefly address other global space-based navigation systems which have become operational after the GPS advent.

Galileo, a joint initiative of the European Union (EU) and the European Space Agency (ESA), is a GNSS system designed for civilian use and planned as an open, global system, fully interoperable with GPS and GLONASS, but independent from them [1]. As scheduled to be hit by the end of 2020, the fully deployed Galileo system consists of 24 operational satellites plus six in-orbit spares, positioned in three circular MEO-planes at an altitude of 23222 Km above the Earth [5]. Galileo satellites permanently transmit three independent CDMA signals: E1 (1575.420 MHz), E5 (1191.795 MHz) and E6 (1278.750 MHz); E5 signal is further sub-divided into two signals, E5a (1176.450) and E5b (1207.140) [6]. Differently from GPS signals, a Galileo payload, on top of the carrier modulation, bears a sub-carrier modulation (Binary-offset carrier - BOC) meant to ensure interoperability with GPS system and to reduce interference from other systems transmitting over the same bands.

In the late 1970s, while GPS system was under development, the Sovietic Union launched a similar system, the **Globalnaya navigatsionnaya sputnikovaya sistema (GLONASS)**. It is a space-based navigation technology providing reliable positioning, navigation and timing services to users worldwide. Similarly to GPS, it was primarily designed for military purposes, with the first satellite launched in 1982 and a full constellation established in 1996. Legacy GLONASS-M satellites transmit Frequency Division Multiple Access (FDMA) signals, implying that the same PRN is broadcast by the same satellite over different RF-carriers. Next generation GLONASS-K satellites are meant to transmit CDMA signals for both restricted and civil services [1].

The **BeiDou Navigation Satellite System (BDS)**, also known as BeiDou-2, is China's second-generation satellite navigation system, conceived as a global evolution of the previous regional navigation system (BeiDou-1). The official approval by the Chinese Government for the development and deployment of BeiDou-2 was done in 2006 and the operability of the global positioning service was planned

by 2020. At the date of writing, the nominal space constellation consists of 3 Geostationary-Earth Orbit (GEO) satellites, 3 Inclined Geosynchronous Orbit (IGSO) satellites, and 24 MEO satellites.

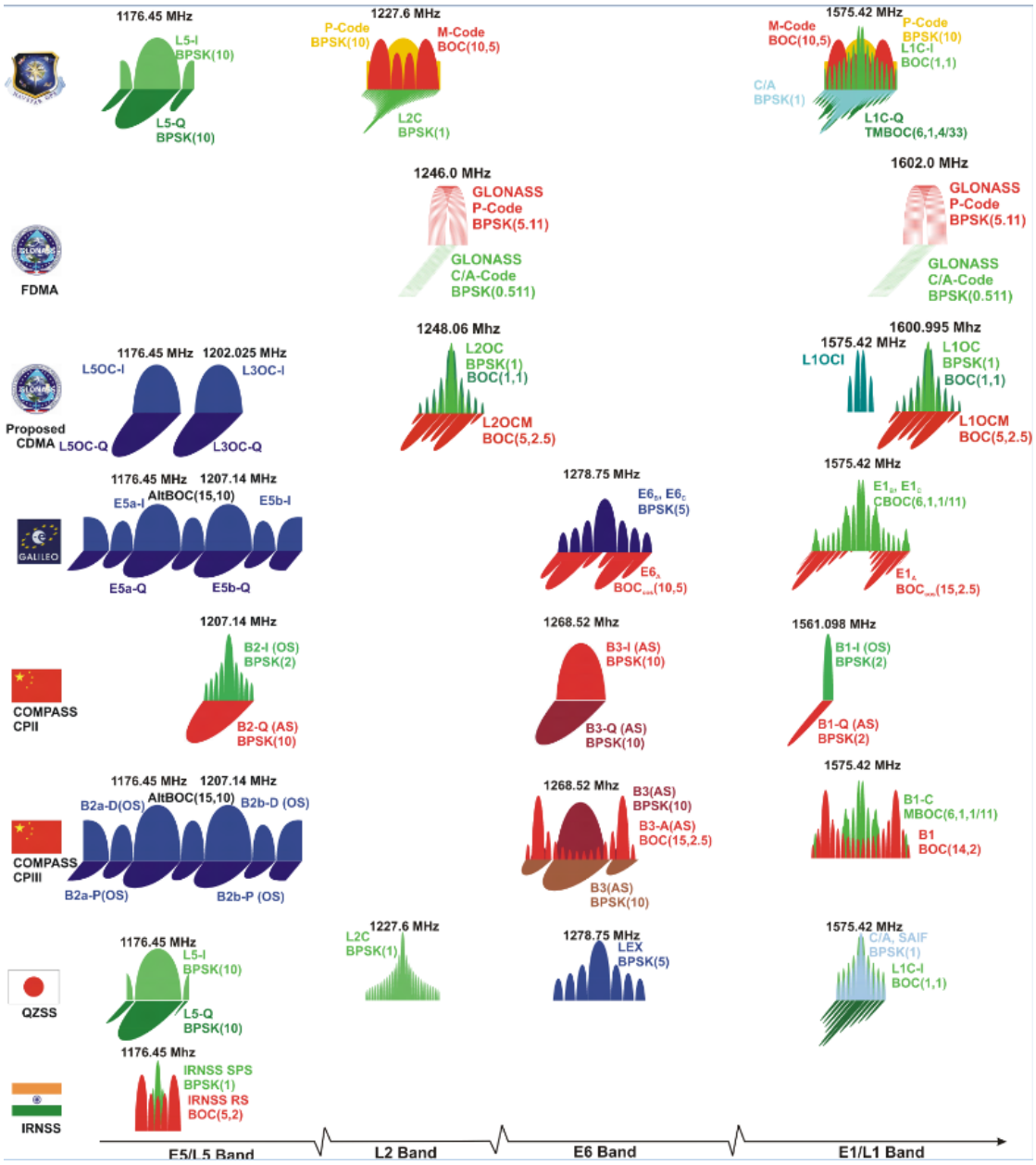


Figure 1.2: Frequency bands of the various GNSSs [7].

1.2 GNSS receiver: from ranging signals to position estimation

A GNSS receiver represents the central node of the user segment and sets as an interface between the navigation system and a generic target end-user. Considering the most popular application of the involved navigation technology oriented to localization purposes, the receiver accomplishes all those operations which are necessary to retrieve positioning and timing information. Given a group of visible satellites belonging to a generic GNSS-constellation, it acquires the involved RF-signals which, travelling huge distances and crossing both ionospheric and tropospheric layers, are typically characterized by a very low power budget. From them, the receiver should extract temporal misalignment information, leading to pseudorange measurements, and frequency-offset information, leading to Doppler-shift measurements. The combination of these observables, then, is exploited within a processing unit to produce a PVT solution (Section 1.3).
In the most common architecture, a GNSS receiver allocates one processing channel for each visible satellite and, in order to ensure uninterrupted signal tracking, two quantities are continuously estimated and corrected: code delay and carrier frequency offset. For low-cost receiver equipment in the mass-market sector, code delay is used to evaluate an unambiguous but noisy pseudorange measurement, while the received signal frequency is employed to detect a Doppler-shift value, which carries on information about the relative user-to-satellite velocity. On the contrary, high-cost professional receivers use phase-information for range measurements too, gaining precision at the price of integer-cycle ambiguity [8].
With this in mind, the current section offers a short overview of the GNSS receiver radio-frequency front-end and, beyond analogue-to-digital conversion, of the early signal-processing stages.

1.2.1 The Signal-in-Space structure

GNSS satellites continuously transmit synchronous navigation signals, conventionally named Signal-in-Space (SIS), towards the Earth. As a matter of fact, these waveforms are properly constructed to contain ranging codes and navigation data, thus allowing any receiver to compute, at every epoch, both the satellite coordinates and the user-to-satellite temporal misalignment, in the view of constructing an estimate of the pseudorange (i.e. noisy and biased user-to-satellite range) [7]. Additionally, the SIS should be tolerant to some level of multipath reflections, to reasonable amounts of intentional and unintentional interference and should provide ionospheric delay measurements.
The multiple components of a navigation signal are generated by the satellite payload and, once mixed, they give rise to a baseband navigation signal. The

latter is further modulated on a predefined carrier at RF and transmitted by the front-end. The frequency planning among the various navigation systems is shown in Figure 1.2.
It is important to remark that the SIS, which is the beamed waveform at RF, may potentially multiplex several baseband components (channels). Anyway, each component must always include:

- **Spreading code**, a binary PRN ranging sequence which grants the orthogonality between signals belonging to the same constellation (Multiple Access scheme) and implements spectral spreading (DSSS paradigm) to mitigate inter-system interference.
- **Navigation message**, a binary sequence which brings about relevant information including ephemeris, satellite-clock corrections, and all synchronization parameters exploited by the receiver to retrieve a pseudorange estimate.

To give an example, Figure 1.3 shows the three (including the carrier) fundamental layers of a single GPS-channel broadcast over L1 and L2 bands.

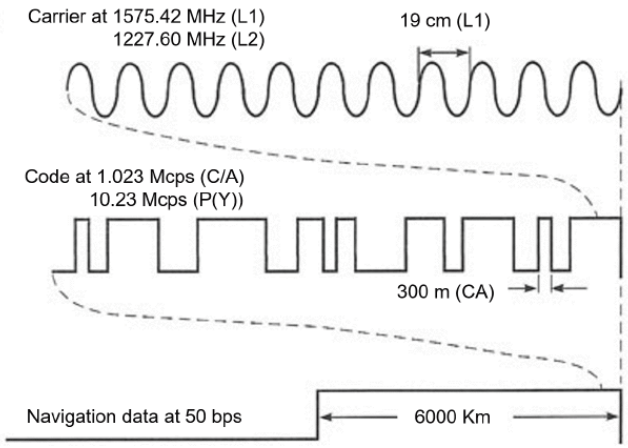


Figure 1.3: Transmitted GPS channel with its constituent components: carrier (top), code (middle), navigation message (bottom) [1]. The picture is not in scale.

1.2.2 The received signal and the front-end

The RF-SIS broadcast by a given satellite travels distances of thousands of kilometers and crosses the upper (ionosphere) and lower (troposphere) layers of the atmosphere, thus being impaired by interference effects (RFI), both of intentional (e.g. jamming, spoofing etc.) and unintentional (rising from other GNSS

systems transmitting over adjacent bands) natures, and ionospheric scintillation effects, which primarily involve fluctuations in the amplitude and phase dimensions. Consequently, the GNSS receiving antenna is involved in capturing a radio signal which, besides heavily compromised in its integrity, is also characterized by very low intensity.
Considering a single constellation and recalling the intrinsically shared nature of the physical propagation medium, the received waveform results in the superposition of the contributions of the various signals, transmitted over a fixed bandwidth, by all the satellites in view. Assuming N_s satellites in view, the typically adopted model is the following:

$$z_{RF}(t) = \sum_{i=1}^{N_s} \tilde{z}_{RF,i}(t) + \eta(t) \tag{1.1}$$

where $\tilde{z}_{RF,i}(t)$ identifies the i-th satellite SIS-contribution, affected by both delay and Doppler in general, and $\eta(t)$ represents an overall additive noise random process, which is considered white, zero-mean and normally distributed (AWGN-channel assumption).
Looking at the i-th satellite term $\tilde{z}_{RF,i}(t)$, it generally multiplexes M different channels (independent baseband components) and, by accounting for the m-th fundamental component, it can be formally expressed as:

$$\tilde{z}_{RF,im}(t) = \sqrt{2P_{rx}}c(t-\tau)\,\bar{c}(t-\tau)\,d(t-\tau)\cos(2\pi(f_c+f_D)t+\phi) \tag{1.2}$$

where τ represents the unknown code-delay, f_D the unknown doppler frequency and ϕ the unknown carrier phase-offset.
Once the GNSS signal has been captured by the receiver antenna, it is fed to the front-end section, which determines the analogue part of the radio-receiver. The front-end (Figure 1.4) is responsible for preparing and conditioning the received tiny waveform such that it becomes suitable for further signal processing. In particular, the main operations can be summarized as follows [9]:

- Filtering (to ensure low-noise and out-of-band rejection) and amplification (to raise-up the intensity of the weak received signal).
- Down-conversion of the spectrum from RF to IF, thus leaving a residual carrier modulation on top of the unknown Doppler modulation.
- Sampling and Quantization, which are the standard steps to achieve Analogue-to-Digital Conversion (ADC).
- Automatic Gain Control (AGC), which adjusts the gain of the front-end section to benefit from the full dynamic range.

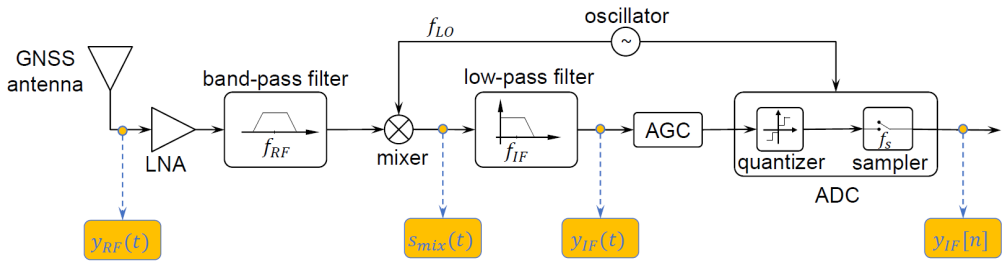


Figure 1.4: GNSS receiver analogue front-end. Picture taken from [10].

1.2.3 Signal Acquisition and Tracking

At the receiver front-end output, the ADC definitely identifies the frontier between analogue and digital domains. In fact, it supplies the downstream signal processing stages with a sequence of noisy IF samples $y_{IF}[n]$ generated from the properly-conditioned received RF-signal. On the digitalized signal, then, the cascaded acquisition and tracking stages target to estimate delay and Doppler-shift values to align the incoming signal with a local replica of the PRN ranging sequence. In particular, the acquisition stage, starting from the noisy IF-samples, is in charge of identifying which satellites are visible by correlating multiple local replicas (code and carrier) and by extracting coarse estimates of the delay and Doppler-shift. Then, the tracking process refines the local replica generation in order to retrieve accurate delay and Doppler estimates, which are eventually exploited to construct pseudorange and Doppler measurements [1].

Acquisition stage

The acquisition stage is adopted by a GNSS receiver to estimate the arrival time τ, which contains basic range and timing information required to compute the receiver position and the clock bias, and the Doppler shift f_{dop}, which contains pseudorange rate information useful to compute the user velocity and the clock drift [1]. It follows that, the sequence of IF-samples $y_{IF}[n]$ contains two unknowns which can be collected in a vector $\mathbf{p} = \{\tau, f_{dop}\}$ (here, we are neglecting the carrier phase offset). Acquisition, then, requires the identification of visible satellites to produce an estimate $\hat{\mathbf{p}} = \{\hat{\tau}, \hat{f}_{dop}\}$ of the unknown couple $\mathbf{p}$. Adopting Maximum Likelihood (ML) approach, the ML estimate of the code-delay and the Doppler-shift can be obtained as:

$$\hat{\mathbf{p}}_{ML} = arg \max_{\{\tau, f_{dop}\}} \left| \frac{1}{S} \sum_{s=0}^{S-1} y_{IF}[s]\, \hat{l}_{IF}[s] \right|^2 \tag{1.3}$$

where S identifies the number of samples employed to process the received IF sequence. Evidently, the acquisition problem consists in the maximization of a 2-D cross-correlation (Cross-Ambiguity Function (CAF)) between $y_{IF}[n]$ and a local signal replica $\hat{l}_{IF}[n]$ defined as:

$$\hat{l}_{IF}[n] = c[n - \hat{\tau}]\, e^{j2\pi\left(f_{IF}+\hat{f}_{dop}\right)n} \tag{1.4}$$

where $\hat{\tau}$ is a test value for the delay and $\hat{f}_{dop}$ is a test value for the Doppler, both extracted as a couple $\{\hat{\tau}, \hat{f}_{dop}\}$ from a pre-defined 2-D discrete Search-Space. From (1.4), it is clear that the local carrier involves both a In-phase and a Quadrature components, which implies that both the I/Q branches of the received sequence take part in the correlation defined in (1.3). The latter aspect, besides appearing redundant, is fundamental for the stage to estimate $\hat{f}_{dop}$ without needing the knowledge of the carrier phase-offset ϕ [1].

Tracking stage

The chief objective of the tracking stage is the refinement of the code-delay estimate $\hat{\tau}$ and the Doppler-shift estimate $\hat{f}_{dop}$, starting from the error-bounded coarse predictions produced at the output of the acquisition stage. In this sense, the tracking stage must guarantee, with the highest possible level of accuracy, the preservation of the alignment between the received code and the local code replica, in such a way that the receiver can continuously retrieve pseudorange and Doppler measurements. Structurally, this stage defines an outer closed feedback control loop which, inside, contains other two coupled feedback loops, namely the Delay-Lock Loop (DLL) and the Phase-Lock Loop (PLL) (see Figure 1.5). The DLL implements code-tracking by continuously adjusting the local code replica to keep it aligned with the code-layer in the received signal sequence. When the two codes are aligned, the PRN sequence is removed from $y_{IF}[n]$ (*code wipe-off*), leaving the carrier which still is modulated by the low-rate navigation data message. The DLL output signal, then, is input to the PLL carrier tracker. The latter, by relying on a sinusoidal signal generated through a local oscillator (LO) unit, synchronizes to the frequency and phase of the incoming carrier, in order to accomplish navigation data demodulation and to refine the Doppler-shift estimate (*carrier wipe-off*). It is important to remark that, at the PLL output, the received code is at baseband since, through Doppler refinement, full signal demodulation is possible. Furthermore, the two loops are initialized by the outputs of the acquisition stage $\{\hat{\tau}_A, \hat{f}_{dop,A}\}$ [1].

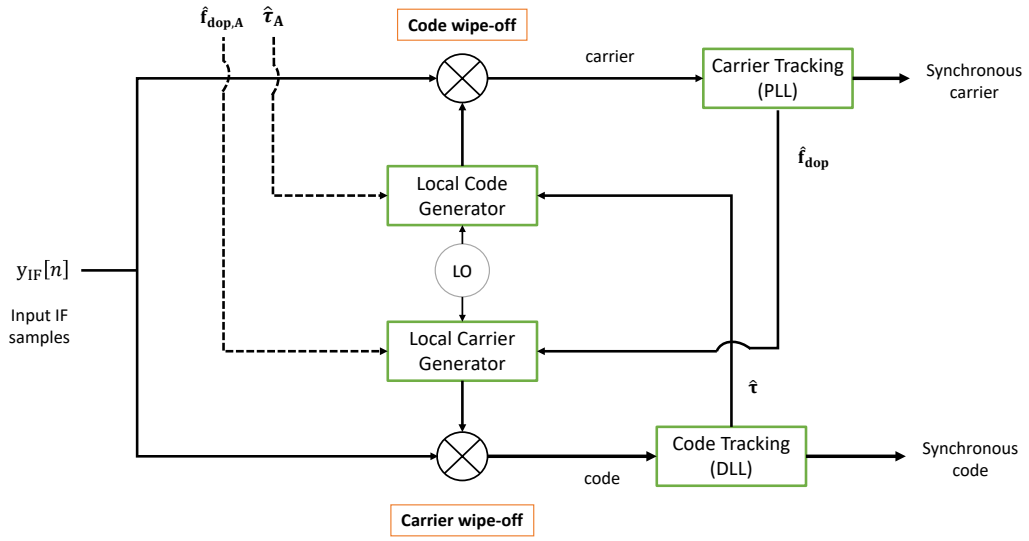


Figure 1.5: High-level tracking loop (PLL+DLL) architecture.

1.3 PVT computation

When both the carrier (PLL) and code (DLL) tracking loops are locked, the GNSS receiver can estimate the apparent transit time of the signal from each satellite, computed as the difference between the signal reception time (ToA measurement), as defined by the receiver clock, and the signal transmission time, as marked on the signal navigation message (demodulated after carrier synchronization). This measurement is intrinsically biased due to the fact that the two interacting endpoints keep time independently and, therefore, are not synchronized. The product of each biased transit time by the speed of light, then, allows to retrieve an estimate of the biased user-to-satellite range (ie. pseudorange) [1]. As anticipated in Section 1.1.1, given the temporal misalignment between receiver and transmitter, the positioning problem involves four unknowns: the components of the position w.r.t. an absolute 3-D reference frame, and the receiver clock bias w.r.t. a common GNSS time-scale. Consequently, a minimum of four pseudorange measurements are required to pursue localization. Furthermore, using the same set of satellites, but independently from the foregoing pseudorange estimation pattern, the receiver carrier tracking loop supplies Doppler-shift measurements based on the variation of the received ranging signals' frequency.
In light of the above, the current section is devoted to the analysis of the fundamental models and mathematical principles allowing to get a real-time estimate of the receiver position, velocity and time (PVT) based on pseudorange and pseudorange-rate (i.e. Doppler-shift) measurements.

1.3.1 Pseudorange measurement model

The basic pseudorange measurement equation from the k-th satellite at epoch t can be written as:

$$\tilde{\rho}_k(t) = r_k(t, t-\tau) + c\left[\delta t_u(t) - \delta t_s(t-\tau)\right] + I_k(t) + T_k(t) + \epsilon_k^{\rho}(t) \tag{1.5}$$

where $r_k(t, t-\tau)$ is the true (unknown) distance between the receiver antenna at signal reception time t and the satellite antenna at transmission time $t-\tau$ (τ being the transit time); $\delta t_u(t)$ and $\delta t_s(t-\tau)$ are the receiver clock offset and the satellite clock-offset w.r.t. GNSS time-scale, respectively; $I_k(t)$ and $T_k(t)$ are the ionospheric and tropospheric induced delays, respectively; finally, $\epsilon_k^{\rho}(t)$ accounts both for modelling errors (satellite clock mismodelling or errors in orbital predictions) and unmodelled effects (receiver noise, multipath etc.).
Based on the parameters extracted from the demodulated navigation binary stream, a standalone GNSS receiver can correct the known bias-error components including, among the others, the satellite clock offset, relativist effects and the ionospheric

and tropospheric induced delays.
The corrected pseudorange $\rho_k(t)$, after deterministic-error compensation, can be re-written as:

$$\rho_k(t) = r_k(t, t-\tau) + c \cdot \delta t_u(t) + \tilde{\epsilon}_k^{\rho}(t) \tag{1.6}$$

where $\tilde{\epsilon}_k^{\rho}(t)$ identifies the User Equivalent Range Error (UERE), which includes all residual error sources.
Defining $\mathbf{x}_u = (x_u, y_u, z_u)$ the vector collecting the components of the receiver spatial position and $\mathbf{x}_k = (x_k, y_k, z_k)$ the vector collecting the components of the k-th satellite spatial position, the true user-to-satellite geometric range is expressed as:

$$r_k = \sqrt{(x_k - x_u)^2 + (y_k - y_u)^2 + (z_k - z_u)^2} = ||\mathbf{x}_k - \mathbf{x}_u|| \tag{1.7}$$

where the time-index is dropped without loss of generality and $\|\cdot\|$ is the norm-operator.
Substituting (1.7) into (1.6):

$$\rho_k = ||\mathbf{x}_k - \mathbf{x}_u|| + b_u + \tilde{\epsilon}_k^{\rho} \tag{1.8}$$

where the receiver clock-bias term $c \cdot \delta t_u$ has been replaced by its range-equivalent b_u, in units of meter.
It is worth remarking that, in the computation of the geometric range, the positions of both the user and the satellite must be expressed w.r.t. a common absolute 3-D reference frame. For convenience, the Earth-Centered Earth-Fixed (ECEF)-frame [11], together with a Cartesian coordinate system, is adopted, and the positioning solution discussed in the following section is determined in such frame [1].

1.3.2 Position computation using pseudorange measurements

Given that (1.8) entails four unknowns, then a minimum of four pseudorange equations, constructed from four distinct satellites, are strictly necessary to solve for the instantaneous user position and clock-bias states. Neglecting the error term $\tilde{\epsilon}_k^{\rho}$ affecting each pseudorange measurement, a mathematical formalization of the solution to the trilateration problem can be provided in the following algebraic system:

$$\begin{cases} \rho_1 &= \sqrt{(x_1 - x_u)^2 + (y_1 - y_u)^2 + (z_1 - z_u)^2} + b_u \\ \rho_2 &= \sqrt{(x_2 - x_u)^2 + (y_2 - y_u)^2 + (z_2 - z_u)^2} + b_u \\ \rho_3 &= \sqrt{(x_3 - x_u)^2 + (y_3 - y_u)^2 + (z_3 - z_u)^2} + b_u \\ \rho_4 &= \sqrt{(x_4 - x_u)^2 + (y_4 - y_u)^2 + (z_4 - z_u)^2} + b_u \end{cases} \tag{1.9}$$

Clearly, (1.9) is a non-linear system of equations and this evidences the intrinsic non-linearity characterizing trilateration. A typical approach to solve the pseudorange equations involved in (1.9) is to linearize them about an approximate user position, and to solve iteratively. More precisely, the idea is to start with rough estimates of the user position and bias states, identifying an initial approximation state $\hat{\mathbf{x}}_u = \left(\hat{x}_u, \hat{y}_u, \hat{z}_u, \hat{b}_u\right)$, and to iteratively determine correction states $\boldsymbol{\delta}\mathbf{x}_u = (\delta x_u, \delta y_u, \delta z_u, \delta b_u)$ until the estimated states do not fit the measurements better (Newton-Raphson method) [1].
In light of the above, the true position and clock bias can be expressed as:

$$\begin{aligned} x_u &= \hat{x}_u + \delta x_u \\ y_u &= \hat{y}_u + \delta y_u \\ z_u &= \hat{z}_u + \delta z_u \\ b_u &= \hat{b}_u + \delta b_u \end{aligned}$$

and, accounting for a generic satellite k, the approximated pseudorange equation can be written in the following manner:

$$\hat{\rho}_k = \sqrt{(x_k - \hat{x}_u)^2 + (y_k - \hat{y}_u)^2 + (z_k - \hat{z}_u)^2} + \hat{b}_u \tag{1.10}$$

At this point, knowing the pseudorange $\hat{\rho}_k$ at the approximation point $\hat{\mathbf{x}}_u$, it is possible to re-formulate the original non-linear problem into a linearized approximation where the unknowns become the correction states collected in $\boldsymbol{\delta}\mathbf{x}_u$ [1]:

$$\begin{aligned} \delta\rho_k &= \rho_k - \hat{\rho}_k \\ &= ||\mathbf{x}_k - (\hat{\mathbf{x}}_u + \boldsymbol{\delta}\mathbf{x}_u)|| - ||\mathbf{x}_k - \hat{\mathbf{x}}_u|| + \left(b_u - \hat{b}_u\right) \\ &\approx -a_{x,k}\delta x_u - a_{y,k}\delta y_u - a_{z,k}\delta z_u + \delta b_u \end{aligned} \tag{1.11}$$

where the linearization is the result of a first-order Taylor series expansion applied to the norm-operator. In particular, the coefficients $a_{x,k}$, $a_{y,k}$ and $a_{z,k}$ are calculated as:

$$
\begin{aligned}
a_{x,k} &= \frac{x_k - \hat{x}_u}{||x_k - \hat{x}_u||} \\
a_{y,k} &= \frac{y_k - \hat{y}_u}{||y_k - \hat{y}_u||} \\
a_{z,k} &= \frac{z_k - \hat{z}_u}{||z_k - \hat{z}_u||}
\end{aligned}
\tag{1.12}
$$

and they identify the Cartesian components of a unit steering vector $\mathbf{a}_k$ pointing from the approximation point $(\hat{x}_u, \hat{y}_u, \hat{z}_u)$ to the k-th satellite.
Considering the approximate pseudorange equation (1.11), the linearized trilateration (more properly, multilateration) problem can be reformulated using the following matrix notation:

$$
\boldsymbol{\delta\rho} = \begin{bmatrix} \delta\rho_1 \\ \delta\rho_2 \\ \delta\rho_3 \\ \delta\rho_4 \end{bmatrix} = \underbrace{\begin{bmatrix} -a_{x,1} & -a_{y,1} & -a_{z,1} & 1 \\ -a_{x,2} & -a_{y,2} & -a_{z,2} & 1 \\ -a_{x,3} & -a_{y,3} & -a_{z,3} & 1 \\ -a_{x,4} & -a_{y,4} & -a_{z,4} & 1 \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} \delta x_u \\ \delta y_u \\ \delta z_u \\ \delta b_u \end{bmatrix}}_{\boldsymbol{\delta}\mathbf{x}_u}
\tag{1.13}
$$

which admits an equivalent compact notation:

$$
\boldsymbol{\delta\rho} = \mathbf{H}\, \boldsymbol{\delta x}_u
\tag{1.14}
$$

where matrix $\mathbf{H}$ characterizes the user-satellite geometry [1].
Eventually, the correction-state vector $\boldsymbol{\delta x}_u$ can be computed:

$$
\boldsymbol{\delta x}_u = \mathbf{H}^{-1}\, \boldsymbol{\delta\rho}
\tag{1.15}
$$

The solution proposed so far, which considers the minimum required number of pseudorange equations, is not always achievable. In fact, if matrix $\mathbf{H}$ is not full-rank, which physically means that the elevation angles of at least two satellites measured from the user position are the same (bad satellite geometry in the sky), then more than four pseudorange equations are necessary to retrieve a trilaterated position and a clock-bias estimate [1]. That said, modern GNSS constellations are designed to guarantee more than four satellites visible at nearly all terrestrial latitudes. Assuming to have a number $K > 4$ of pseudorange measurements, the Least-Squares (LS) criterion is typically adopted to solve the linearized problem

[1]:

$$\min \|\boldsymbol{\delta\rho} - \mathbf{H}\,\boldsymbol{\delta}\hat{\mathbf{x}}_u\|^2 \tag{1.16}$$

and the LS-solution for the correction estimate to be applied to the initial approximation takes the form:

$$\boldsymbol{\delta}\hat{\mathbf{x}}_u^{LS} = \left(\mathbf{H}^T\,\mathbf{H}\right)^{-1}\,\mathbf{H}^T\,\boldsymbol{\delta\rho} \tag{1.17}$$

Recalling the initial assumption $\mathbf{x}_u = \hat{\mathbf{x}}_u + \boldsymbol{\delta}\mathbf{x}_u$, the solution in (1.17) can be applied iteratively until $\boldsymbol{\delta x}_u$ becomes acceptably small according to a pre-defined threshold.

Approaching the end of this section, two important remarks are still to be done. First of all, the LS-solution in (1.17) implicitly assumes that all the pseudorange measurements have equal quality. In fact, this is never true; pseudorange measurements from low-elevation satellites are, most of times, affected by larger errors than those from high-elevation satellites. Given so, the typical way to handle measurements with different quality is to weight them based on some goodness criterion (e.g. satellite elevation). Calling $\mathbf{W} = diag\,(w_1, w_2, ..., w_K)$ a weighting diagonal matrix, the Weighted Least-Squares (WLS) solution of (1.13) is defined [1]:

$$\boldsymbol{\delta}\hat{\mathbf{x}}_u^{WLS} = \left(\mathbf{H}^T\,\mathbf{W}\,\mathbf{H}\right)^{-1}\,\mathbf{H}^T\,\mathbf{W}\,\boldsymbol{\delta\rho} \tag{1.18}$$

Furthermore, in many GNSS-based positioning systems (e.g. integrated navigation systems), the multilateration problem in (1.9) is solved by means of Bayesian estimators (e.g. Kalman Filters or Particle Filters), which allow for a PVT solution at every epoch irrespective of the constraint $K \geq 4$ on the number of visible satellites [12]. In such framework, then, a WLS is run just to initialize the solution or in case it is necessary to re-set the system.

1.3.3 Doppler measurement and velocity determination

The relative motion between the satellite and the GNSS-receiver reflects in changes of the observed frequency of the received signal. This Doppler-shift w.r.t. the nominal carrier frequency is measured routinely in the carrier tracking loop (PLL) and, when combined with the information about the satellite velocity provided through ephemerides, it allows to estimate the user velocity [1, 12]. Despite the possibility to estimate user velocity through an approximate derivative of the user

position, a better approach involves processing carrier-phase measurements to obtain a refined estimate of the received Doppler frequency [12].
According to a Doppler-model, the received frequency from k-th satellite is expressed as:

$$f_u^k = f_c \left(1 - \frac{\mathbf{v}_u^k \cdot \mathbf{a}_k}{c}\right) \tag{1.19}$$

where f_c is the nominal carrier frequency of the transmitted SIS, $\mathbf{v}_u^k$ is the user-to-satellite relative velocity in 3-D ECEF components, $\mathbf{a}_k$ is the unit steering vector pointing from the user position towards the k-th satellite, and c is the speed of light. The dot-product $\mathbf{v}_u^k \cdot \mathbf{a}_k$ simply computes the projection of the relative velocity vector on the LoS to the satellite.
The relative user-to-satellite velocity $\mathbf{v}_u^k = \left(v_{u,x}^k, v_{u,y}^k, v_{u,z}^k\right)$ is given as:

$$\mathbf{v}_u^k = \mathbf{v}_k - \mathbf{v}_u \tag{1.20}$$

where $\mathbf{v}_k = (v_{k,x}, v_{k,y}, v_{k,z})$ is the k-th satellite velocity and $\mathbf{v}_u = (v_{u,x}, v_{u,y}, v_{u,z})$ is the receiver velocity, both referenced to a common ECEF-frame.
Then, the Doppler-shift δf_u^k can be computed:

$$\delta f_u^k = f_u^k - f_c^k = -f_c^k \frac{(\mathbf{v}_k - \mathbf{v}_u) \cdot \mathbf{a}_k}{c} \tag{1.21}$$

where f_c^k is the actual transmitted satellite frequency, which does not coincide with the nominal carrier frequency f_c. In fact, satellites generate frequencies based on a high-accuracy on-board atomic clock which is offset w.r.t. the system time. This offset is specified in the navigation message and, after data demodulation, is applied by the receiver to the nominal carrier frequency to obtain the transmitted frequency [12].
From (1.21), the Doppler equation for k-th satellite in (1.19) can be re-written as:

$$f_u^k = f_c^k \left\{1 - \frac{1}{c}\left[(\mathbf{v}_k - \mathbf{v}_u) \cdot \mathbf{a}_k\right]\right\} \tag{1.22}$$

Moreover, the GNSS receiver measures a received frequency f_k which is different from f_u^k by a frequency bias offset. The latter can be related to the receiver clock-drift $\delta\dot{t}_u$ relative to the GNSS time-scale [12]. It follows that this relation holds:

$$f_u^k = f_k \left(1 + \delta\dot{t}_u\right) \tag{1.23}$$

In light of the above, substituting (1.23) into (1.22), results in:

$$\frac{c\,\left(f_c^k - f_k\right)}{f_c^k} - \mathbf{v}_k \cdot \mathbf{a}_k = -\mathbf{v}_u \cdot \mathbf{a}_k + \frac{c\,f_k\,\delta\dot{t}_u}{f_c^k} \tag{1.24}$$

and, expanding the inner product terms, it is obtained:

$$\begin{aligned} &\frac{c\,\left(f_c^k - f_k\right)}{f_c^k} - v_{k,x}\,a_{x,k} - v_{k,y}\,a_{y,k} - v_{k,z}\,a_{z,k} = \\ &\quad = -v_{u,x}\,a_{x,k} - v_{u,y}\,a_{y,k} - v_{u,z}\,a_{z,k} + \frac{c\,f_k\,\delta\dot{t}_u}{f_c^k} \end{aligned} \tag{1.25}$$

The left-hand side terms of (1.25) are known; the satellite velocity components $\mathbf{v}_k = (v_{k,x}, v_{k,y}, v_{k,z})$ are estimated from ephemerides, while the unit steering vector components $\mathbf{a}_k = (a_{k,x}, a_{k,y}, a_{k,z})$ are inherited from the earlier solution to the multilateration problem in (1.13). Furthermore, without inducing any relevant error, a simplification can be done in the right-hand side by assuming that $\frac{f_k}{f_c^k} \simeq 1$. Introducing a dummy variable d_k to label the left-side, (1.25) reduces to:

$$d_k = -\,a_{x,k}v_{u,x} - \,a_{y,k}v_{u,y} - \,a_{z,k}v_{u,z} + \underbrace{c\,\delta\dot{t}_u}_{\dot{b}_u} \tag{1.26}$$

In fact, (1.26) defines the generic Doppler-measurement equation for k-th satellite and it is function of four unknown quantities: the 3-D components of the user velocity in ECEF-frame and the clock-drift. By defining $\dot{\mathbf{x}}_u = \left(v_{u,x}, v_{u,y}, v_{u,z}, \dot{b}_u\right)$ the unknown vector and considering a generic number $K > 4$ of Doppler-measurements, the following matrix-equation holds:

$$\mathbf{d} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_K \end{bmatrix} = \underbrace{\begin{bmatrix} -a_{x,1} & -a_{y,1} & -a_{z,1} & 1 \\ -a_{x,2} & -a_{y,2} & -a_{z,2} & 1 \\ \vdots & \vdots & \vdots & 1 \\ -a_{x,K} & -a_{y,K} & -a_{z,K} & 1 \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} v_{u,x} \\ v_{u,y} \\ v_{u,z} \\ \dot{b}_u \end{bmatrix}}_{\dot{\mathbf{x}}_u} \tag{1.27}$$

Employing the LS-method, provided that the geometric matrix $\mathbf{H}$ is invertible, the solution is retrieved as:

$$\hat{\dot{\mathbf{x}}}_u^{LS} = \left(\mathbf{H}^T\,\mathbf{H}\right)^{-1}\,\mathbf{H}^T\,\mathbf{d} \tag{1.28}$$

which simplifies to $\hat{\dot{\mathbf{x}}}_u = \mathbf{H}^{-1}\,\mathbf{d}$ when $K = 4$ and $\mathbf{H}$ is full-rank.

Similarly to the approach described in Section 1.3.2, it is possible to re-formulate the velocity estimation problem using Newton-Raphson method. In particular, it is possible to start from coarse estimates of the user velocity and drift states, collected in the approximation state $\hat{\dot{\mathbf{x}}}_u = \left(\hat{v}_{u,x}, \hat{v}_{u,y}, \hat{v}_{u,z}, \hat{\dot{b}}_u\right)$, and to iteratively retrieve correction states $\boldsymbol{\delta}\dot{\mathbf{x}}_u = \left(\delta v_{u,x}, \delta v_{u,y}, \delta v_{u,z}, \delta\dot{b}_u\right)$ until proper fitting with the Doppler measurements.
Again, the true velocity and clock-drift are expressed as:

$$\begin{aligned} v_{u,x} &= \hat{v}_{u,x} + \delta v_{u,x} \\ v_{u,y} &= \hat{v}_{u,y} + \delta v_{u,y} \\ v_{u,z} &= \hat{v}_{u,z} + \delta v_{u,z} \\ \dot{b}_u &= \hat{\dot{b}}_u + \delta\dot{b}_u \end{aligned}$$

Given so, considering satellite k, the true Doppler measurement and the approximate Doppler measurement can be expressed as:

$$\begin{aligned} d_k &= -\,\mathbf{a}_k \cdot (\hat{\mathbf{v}}_u + \boldsymbol{\delta}\mathbf{v}_u) + \hat{\dot{b}}_u + \delta\dot{b}_u \\ \hat{d}_k &= -\,\mathbf{a}_k \cdot \hat{\mathbf{v}}_u + \hat{\dot{b}}_u \end{aligned} \tag{1.29}$$

Moreover, d_k and $\hat{d}_k$ can be related to the measured Doppler-offset $\dot{\rho}_k$ and the approximate Doppler-offset $\hat{\dot{\rho}}_k$ using the following relations:

$$\begin{aligned} \dot{\rho}_k &= d_k + \mathbf{a}_k \cdot \mathbf{v}_k \\ \hat{\dot{\rho}}_k &= \hat{d}_k + \mathbf{a}_k \cdot \mathbf{v}_k \end{aligned} \tag{1.30}$$

Eventually, it is possible to reformulate a Doppler model where the unknowns are the correction states collected in $\boldsymbol{\delta}\dot{\mathbf{x}}_u$:

$$\delta\dot{\rho}_k = \dot{\rho}_k - \hat{\dot{\rho}}_k = -a_{x,k}\delta v_{u,x} - a_{y,k}\delta v_{u,y} - a_{z,k}\delta v_{u,z} + \delta\dot{b}_u \tag{1.31}$$

and extending to a generic number $K > 4$ of Doppler measurements, the following matrix equation holds:

$$\boldsymbol{\delta\dot{\rho}} = \begin{bmatrix} \delta\dot{\rho}_1 \\ \delta\dot{\rho}_2 \\ \vdots \\ \delta\dot{\rho}_K \end{bmatrix} = \underbrace{\begin{bmatrix} -a_{x,1} & -a_{y,1} & -a_{z,1} & 1 \\ -a_{x,2} & -a_{y,2} & -a_{z,2} & 1 \\ \vdots & \vdots & \vdots & 1 \\ -a_{x,K} & -a_{y,K} & -a_{z,K} & 1 \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} \delta v_{u,x} \\ \delta v_{u,y} \\ \delta v_{u,z} \\ \delta\dot{b}_u \end{bmatrix}}_{\boldsymbol{\delta\dot{\mathbf{x}}}_u} \tag{1.32}$$

Being (1.32) equivalent to the formulation in (1.13), just in terms of velocity and clock-drift correction terms in place of position and clock bias components, the overall WLS for the PVT solution can be solved by combining the multiple factors into a bigger composite problem.

1.4 Inertial Navigation System

Historically, the first application of inertial navigation can be dated back to the guidance system in the V-2 rocket engineered by Germans in 1942. However, a steady and fruitful research on inertial technology sparked off, after World War II, in the United States, which produced inertial guidance units for ballistic missiles. Then, with the advent of the Space Age in the 1960's, inertial systems found their way in space missions, to be then deployed in aeronautics, both for military and commercial purposes [13].
This section offers an overview on inertial navigation technology, from the architectural and functional basics to the mathematical models allowing to solve for position, velocity and attitude states.

1.4.1 Principles and Structure

Inertial navigation is based on laws of inertia and gravity defined by Newtonian physic and it involves a blend of inertial measurements, mathematics, control system design and geodesy [14]. Differently from other navigational aids, an Inertial Navigation System (INS) does not rely on external measurements. Instead, it exploits the inertial properties of sensors to supply self-contained, non-radiating, non-jammable and possibly accurate estimation of instantaneous navigation states [15], including position, velocity and attitude. It is important to remark that, differently from any GNSS, INSs can only retrieve relative positioning information; thus, some initial a-priori information, together with an initial frame, must be known, in order to provide absolute navigation solutions (i.e. dead reckoning).
Two types of measurements are involved, namely, specific forces (i.e. inertial accelerations according to Newton's laws), as measured by accelerometers, and angular rotations, as measured by gyroscopes [15]. More specifically, looking at the accelerometer, it measures the total acceleration, caused by the application of external forces on the sensor, encountered by the object. After integration in time, and knowing some initial conditions, it is possible to determine variations of body velocity and position. However, these sensors cannot distinguish between the accelerations provoked by gravity and those induced by inertial motion. Therefore, when using an accelerometer, it is necessary to compensate for the specific force caused by gravity [16]. To this end, the tilt of the platform, accelerometers are mounted on, must be known. This motivates the need for gyroscopes.
Gyroscopes are used to retrieve relative attitude information, based on the spinning rate they are subjected to w.r.t. a local sensitivity axis; given a set of measured angular rates plus some initial conditions being set up, it can be determined the tilt of the platform w.r.t. an external reference frame (i.e. the body attitude in ECEF frame).

The combination of inertial sensors (i.e. accelerometers and gyroscopes) defines an Inertial Measurement Unit (IMU). Figure 1.6 shows a typical strapdown inertial sensors assembly for an IMU. By the term *strapdown*, it is meant an IMU-configuration in which inertial sensors are directly strapped to the vehicle and rotate with the body platform [15].

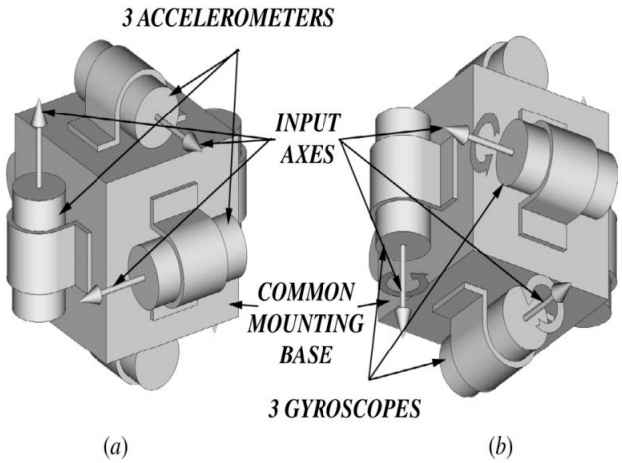


Figure 1.6: IMU assembly for a triad of strapdown inertial sensors [17].

There are different categories of IMUs based on their accuracy performance (*grade*). In this thesis, we deal with a low-cost Micro Electro-Mechanical System (MEMS) IMU, which can be included in the automotive-grade and that is typically employed in the mass-market sector. MEMS-technology involves small sensors able to realize low-performance IMUs [18, 10]. A conventional IMU (Figure 1.6) consists of three gyroscopes for measuring angular rates and three accelerometers for measuring accelerations. They are mounted in triads with the sensitive axes of sensors mutually orthogonal, setting up a Cartesian reference frame. The IMU frame is aligned with the local reference frame of the platform where these sensors are mounted. The body frame is the coordinate system that is aligned with the body of the sensor [19]. On an vehicle, the sensor x-axis often points to the forward direction, the y-axis points to the lateral direction, and the z-axis points to the vertical down direction, thus forming a right-handed orthogonal coordinate (see Figure 1.7) [16].

1.4.2 Strapdown Mechanization

A INS mainly involves two cooperating blocks: an IMU, which is defined by the triads of strapdown inertial sensors (whose assembly is mounted on the vehicle platform), and a strapdown processing unit, which elaborates raw sensor measurements to calculate by dead reckoning navigation solutions.

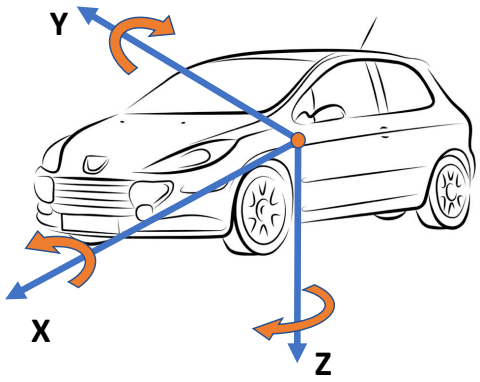


Figure 1.7: IMU axes aligned with the vehicle body frame.

Before moving to the details of the inertial equations allowing to compute the instantaneous position, velocity and attitude states of a moving object, it is essential the choice of a convenient reference frame. Theoretically, both an inertial reference frame and a Earth-fixed frame would be equally acceptable choices, despite leading to different resolution processes. However, when considering navigation over the Earth surface, a Earth-fixed frame (which rotates with the Earth) is usually selected. Hereafter, we consider a ECEF frame [11].
With this in mind, starting from the available inertial sensor measurements, the system of equations used to compute the inertial position, velocity and attitude, in the selected reference frame, is called *mechanization* [10].

Resolution of accelerometer measurements and the Coriolis theorem

Accelerometers usually provide a measurement of specific force, denoted by $\mathbf{f}^b$, in a body-fixed axis set. In order to navigate, it is necessary to resolve the components of specific force in the selected reference frame [20]. Given the choice of an Earth-fixed frame, the following operation is simply required:

$$\mathbf{f}^e = \mathbf{C}_b^e \mathbf{f}^b \tag{1.33}$$

where $\mathbf{f}^e$ is the specific-force vector in the ECEF-frame and $\mathbf{C}_b^e$ identifies the rotation Direction Cosine Matrix (DCM) from the body to the Earth frames.
When considering a situation where the vehicle navigates w.r.t. an inertial frame (fixed, non-rotating set of axis), the measured components of specific force (collected in vector $\mathbf{f}^b$) and the estimates of the gravitational field $\mathbf{g}$ are summed to jointly determine the components of body acceleration in a space-fixed reference frame [20].

Temporal integration of these estimates, then, allows to retrieve inertial position and velocity. However, when considering navigation over the surface of the Earth with respect to a rotating frame, some additional apparent forces, due to the frame motion itself, must be accounted for. In this context, to estimate the body ground speed w.r.t. the Earth, the *Coriolis theorem* can be exploited [20]:

$$\mathbf{v}^e = \dot{\mathbf{p}}^e = \mathbf{v}^i - \boldsymbol{\omega}_{ie} \times \mathbf{p}^i \tag{1.34}$$

where superscripts b,e and i indicate that the involved quantities are expressed in the body, inertial, Earth-fixed reference frames respectively. Clearly, the term $\mathbf{v}^e$ identifies the vehicle ground velocity, while ω_{ie} represents the rotation rate of the ECEF frame with respect to an inertial frame. Differentiating (1.34), the body ground acceleration is obtained as [20]:

$$\dot{\mathbf{v}}^e = \ddot{\mathbf{p}}^e - \boldsymbol{\omega}_{ie} \times \mathbf{v}^e - \boldsymbol{\omega}_{ie} \times \left[\boldsymbol{\omega}_{ie} \times \mathbf{p}^i\right] \tag{1.35}$$

where the term $\boldsymbol{\omega}_{ie} \times \mathbf{v}^e$ identifies the Coriolis acceleration due to the vehicle speed over a rotating surface, and the term $\boldsymbol{\omega}_{ie} \times [\boldsymbol{\omega}_{ie} \times \mathbf{p}^i]$ is the body centripetal acceleration.

Attitude computation

Given the low-cost MEMS strapdown inertial sensor measurements, the first inertial navigation state which can be resolved is the attitude, which describes the body orientation in the selected coordinate frame. There are different mathematical representations of the attitude information w.r.t. a reference frame. Among them, we mention DCM and Euler angles [20]. While the former approach involves 3×3 matrices and is suitable when operating vector transformation between different sets of reference axes, the latter notation is certainly more physically intuitive. In terms of Euler angles, frame transformation is accomplished through three successive ordered rotations about three different orthogonal axes, where the Euler angles of rotations are termed *yaw* (ψ), *pitch* (θ) and *roll* (ϕ) [20]. Beyond that, it is quite easy to construct 3×3 DCMs from Euler angles. For the following discussion on mechanization, DCM-based attitude representation is adopted.

To update the body attitude information, it is necessary to compute the rotation rate vector of the body w.r.t. the Earth frame, expressed in body coordinates, according to the following relationship [20]:

$$\boldsymbol{\omega}_{eb}^b = \boldsymbol{\omega}^b - \mathbf{C}_b^{eT} \boldsymbol{\omega}_{ie}^e \tag{1.36}$$

in which $\boldsymbol{\omega}^b$ is the gyroscopes measured angular rate vector, $\boldsymbol{\omega}_{ie}^e$ is the estimated Earth's rotation rate and $\mathbf{C}_b^e$ is the body frame to Earth-fixed frame rotation DCM matrix. The latter is computed as [20]:

$$\dot{\mathbf{C}}_b^e = \mathbf{C}_b^e \boldsymbol{\Omega}_{eb}^b \tag{1.37}$$

with $\boldsymbol{\Omega}_{eb}^b$ being the skew-symmetric form of $\boldsymbol{\omega}_{eb}^b$.
Matrix $\mathbf{C}_b^e$ being estimated, then, it is possible to derive Euler angles to represent the change in the body orientation between successive updates of the attitude information.

Velocity computation

Computing $\mathbf{C}_b^e$ as in (1.37), the accelerometer specific force measurements in the Earth-fixed frame (collected in vector $\mathbf{f}^e$) can be resolved according to (1.33). At this point, everything is set-up to allow for the evaluation of the body ground acceleration (rate of change of the vehicle ground speed w.r.t. Earth axes) using [20]:

$$\mathbf{a}_e^e = \dot{\mathbf{v}}_e^e = \mathbf{C}_b^e \mathbf{f}^b - 2\boldsymbol{\omega}_{ie}^e \times \mathbf{v}_e^e + \mathbf{g}_l^e \tag{1.38}$$

where the Coriolis term $(\boldsymbol{\omega}_{ie}^e \times \mathbf{v}_e^e)$ is subtracted, and the components of mass gravity attraction (accelerometers sense the reaction to gravity attraction on the body) and centripetal acceleration are compensated for and collected in the term $\mathbf{g}_l^e$.
Integrating the ground acceleration over time, the vehicle ground speed can be estimated and updated.

Position computation

Eventually, the body position in Earth-fixed frame axes can be retrieved by performing a second temporal integration on the estimated ground speed exploiting the well known relation (1.39), where the dot-symbol on top of a vector indicates that a partial derivative is taken on each axis component.

$$\dot{\mathbf{p}}_e^e = \mathbf{v}_e^e \tag{1.39}$$

Chapter 2

Integration of Inertial Navigation and Satellite-based positioning

Different sensors which are employed for positioning and navigation, depending on their nature and intrinsic properties, are characterized by both weaknesses and strengths. From a general perspective, under some assumptions and proper conditions, a single navigation sensor (e.g. a standalone Global Navigation Satellite System (GNSS) receiver) may be sufficient to estimate and track over time the position, velocity and timing information of a self-navigating body. Considering an Inertial Navigation System (INS), choosing a proper reference frame and being some a-priori conditions set up, it is possible to derive position, velocity and attitude solutions by processing the raw angular rate and specific force measurements from the Inertial Measurement Unit (IMU). Such system is robust, self-contained and immune to radio interference effects; moreover it provides updates to the positioning information at high rate (e.g. 100 Hz). However, real inertial sensors are intrinsically affected by deterministic errors, such as bias turn-on or scale-factor errors (related to the sensor sensitivity) and stochastic errors, such as bias drift or sensor noise [21]. As regards the latter, it is generally moderate w.r.t. the amount of noise affecting GNSS code-based ranging measurements. Nevertheless, despite a stochastic modelling of these foregoing errors is feasible, they cannot be completely removed. Consequently, they are integrated within the inertial mechanization and affect the accuracy level in the navigation solution over time.
In contrast to an INS, GNSS offers both good long-term stability, with positioning errors limited to few meters, and the availability of low-cost user hardware in the mass-market sector [16, 21]. However, the system rate is typically lower (some high-rate GNSS receivers are indeed available on the market), if compared to that of an

inertial system, and the localization performance is strongly affected by the external environment. In fact, at least four satellites in visibility are needed to supply a Weighted Least-Squares (WLS) navigation solution, and the Radio Frequency (RF) signals broadcast by a constellation of satellites suffer from reflections and diffractions due to the presence of obstacles along the travelled path, degradation caused by the crossing of ionospheric and tropospheric layers, induced multipath effects and several other impairments [22]. For sake of completeness, it is due remarking here that differential GNSS receivers implementing carrier-phase based ranging (e.g. Real-Time Kinematic (RTK)) can achieve sub-metre accurate Positioning, Navigation and Timing (PNT). By the way, they belong to the professional market segment and have higher costs, hence being out of the scope of this research.
Capitalizing on the complementary characteristics of the two systems, their synergistic integration allows to overcome individual drawbacks in the view of enhancing the accuracy and robustness of the navigation solution both in the short term and in the long term [21]. In such context, while GNSS can update the inertial navigation solution to prevent it from drifting, the INS can bridge during GNSS-system outages or in case of ranging signal losses. Regarding the latter aspects, the short-term accuracy and the high resolution of an INS allow to finely track the dynamics of motion between consecutive GNSS-epochs. This is perfectly suitable to support vehicle-tracking applications with strict requirements in terms of accuracy, continuity and reliability, but at affordable costs.

2.1 Integration strategies

Various strategies are proposed in literature to implement hybridization of GNSS and INS systems. These approaches mainly differ in the way the navigation units interact each other and in the kind of information which is shared throughout the integration process. The following strategies are the most common [23]:

- *uncoupled* integration
- *loose* integration
- *tight* integration
- *ultra-tight* integration

In the uncoupled architecture, the degree of integration is extremely low and the two systems are mainly acting separately. Once properly initialized, the INS solves by dead-reckoning the mechanization to compute its high-rate inertial navigation solution, which is adopted as the integrated system output between consecutive GNSS epochs. Clearly, depending on the IMU grade, such solution might drift more

or less rapidly. When a new set of GNSS measurements becomes available, the GNSS-receiver supplies its standalone Position, Velocity, Timing (PVT) solution, which is selected as system output for that epoch and, furthermore, is used to re-set the INS solution. Nonetheless, differently from the remainder hybridization schemes, the uncoupled strategy does not exploit effectively all the advantages brought about by integrating the two systems, primarily because no control is put on the temporal accumulation of sensor drift errors over consecutive GNSS epochs [24, 25].
In the ultra-tight architecture, instead, there is a strict cooperation between the two systems with the deepest possible level of integration. In fact, the GNSS tracking module is directly inserted within the integration filter and, by deriving the Doppler estimate from the INS, it is possible to maintain the tracking loops during GNSS signal outages or when navigating in harsh environments with severe motion dynamics. However, this deep-level coupling requires access to the receiver hardware and is not limited to pure DSP. [10, 26, 25].

In the following sections, a detailed description is provided of the integration architectures which are most often implemented: loosely coupled and tightly coupled.

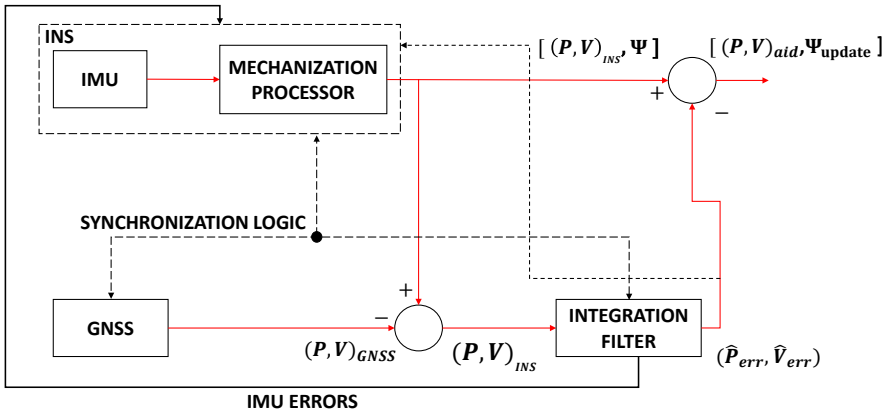


Figure 2.1: Block-diagram for INS/GNSS loosely-coupled architecture.

2.1.1 Loosely-coupled architecture

In the Loosely Coupled (LC) hybridization strategy, the GNSS receiver and the INS act as two interacting but independent navigation units. At each epoch, the absolute position and velocity estimates computed by the GNSS receiver are fed as input observations to an integration filter (also referred to as navigation filter in the following). In parallel, at higher rate, the INS can provide the updates to

the position, velocity and attitude states through the mechanization of raw inertial sensors data. It is clear that, in this architecture, the GNSS measurements from satellites signals (code delay, phase delay and Doppler shift) are processed by the GNSS-receiver prior to be fed to the navigation filter. The latter, then, blends the relative positioning information from the INS with the absolute position and velocity estimates from the GNSS system, in order to retrieve the final positioning solution which is output from the whole navigation unit. Moreover, through the information about the body attitude update, the biases affecting inertial sensors are estimated by the navigation filter and fed-back to the INS. In this way, IMU sensor errors can be somehow mitigated and the inertial trajectory refined [10, 27]. A schematic representation of the discussed LC approach is detailed in Figure 2.1.
The first advantage of this architecture is that it is quite easily implementable since integration is achieved at high-level and there are no a-priori constraints on the choice of the GNSS receiver. In addition to that, the number of navigation states which have to be estimated in such configuration is smaller than that of a traditional Tightly Coupled (TC) architecture, thus lowering the computational complexity and shortening the processing time.
However, few drawbacks are present. First of all, since the navigation filter is independent from the GNSS-unit, the former is unaware of the statistics, in terms of variances and correlations, characterizing the navigation states. Thus, if such information is not externally provided to the filter, performance drops are experienced [10]. Secondly, the GNSS receiver is still acting as an independent module and retains its intrinsic flaws. For instance, it might happen the GNSS filter to loose the locking condition in the tracking loop. In such case, without some aiding information, the receiver is forced to repeat the acquisition of satellites (*cold start*) and this considerably impacts on the Time To First Fix (TTFF). A possible solution to mitigate the latter problem could be that of providing the GNSS receiver with the necessary information about the inertial position and velocity states, so that to allow it undergoing a *warm start* [23].
To summarize, a LC-architecture implements a *decentralized filtering* scheme where the raw data measurements are first processed by two local filters (GNSS receiver and INS-mechanization) working in parallel (at different rates) and producing "partial" navigation solutions which, later on, are further mixed in a main navigation filter. The resulting structure has a reduced computational load but the overall architecture is not robust and performances are inferior than those achievable with a TC-scheme [28].

2.1.2 Tightly-coupled architecture

The main characteristic of a TC-architecture is that the noisy ranging observables extracted by the GNSS-receiver, in terms of pseudorange and Doppler shift

measurements, are directly passed to the navigation filter without being processed first by an independent GNSS filter. The raw GNSS measurements, then, are fused with the position, velocity and attitude high-rate updates coming from the INS mechanization. In other words, in a TC-approach, the inertial system acts as a self-contained unit (INS filter) which predicts the nominal position, velocity and attitude of the navigating body with high-rate. In addition, the inertial predictions are used themselves to predict the nominal pseudorange and Doppler shift measurements related to a set of tracked satellites. At each GNSS-epoch, then, the observed pseudorange and pseudorange-rate measurements represent the navigational aid allowing the integration filter to evaluate the errors on the inertial solution as well as on the estimate of sensor biases, and to refine the system position, velocity and attitude states. Eventually, the refined solution is used as the output of the whole integrated navigation unit [10, 27, 23].
The detailed block-diagram for a TC architecture is provided in Figure 2.2.
The first advantage of the TC-strategy, compared to the LC-scheme, involves the availability of the navigation states' covariance information directly inside the integration filter. As discussed in Section 2.1.1, the LC-configuration requires the independent GNSS processing unit to furnish, as part of its output, some precious information about the statistical correlation among different states. Contrarily, in the currently discussed integration approach, the statistical characterization of noise sources and the modelling of the correlations between different states, are directly elaborated inside the navigation filter. This eventually results in the knowledge of better statistics and, consequently, in a higher accuracy of the integrated solution [10, 28].
A second strength of the TC-scheme lies on the fact that it is not computed any GNSS standalone WLS solution, which would have required a minimum of four satellites in Line-of-sight (LOS) visibility. Thus, if less than four satellites are tracked at a given epoch, the small set of raw GNSS measurements are anyway exploited in the fusion routine to generate the refined integrated solution. The latter aspect represents a major gain to enhance system continuity and robustness in harsh navigation environments (e.g. urban canyons). Furthermore, as discussed in [23], process noise is only added in the integration filter and this results in a better elaboration of GNSS measurements.
Likewise the LC-scheme, the TC-configuration presents some disadvantages. For instance, the number of states which have to be resolved by the integration filter is larger since, for instance, the timing information, in terms of receiver clock bias and drift, is no longer estimated separately by a standalone GNSS processing unit. Hence, the computational load and the system complexity are enhanced.

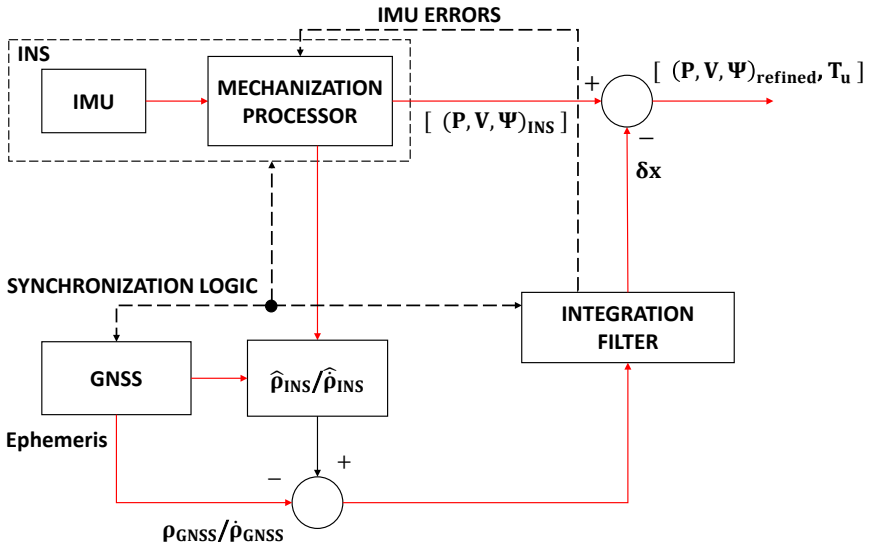


Figure 2.2: Block-diagram for INS/GNSS tightly-coupled architecture.

2.2 Tight integration module: details and implementation

In the framework of this thesis, an integrated INS/GNSS unit is implemented in the TC-configuration, hence focusing on such hybridization paradigm hereinafter. The present section covers the formulation of a discrete state-space representation for the involved integrated navigation system, meant to characterize the evolution of inertial dynamics as well as to relate GNSS measurements with the system state. In the final part, specific details about the developed TC-architecture are provided.

2.2.1 INS/GNSS state-space model

As discussed in Section 2.1.2, a INS/GNSS TC-architecture builds on a *centralized* filtering stage where the raw GNSS-receiver measurements are processed and blended with the information about position, velocity and attitude retrieved from the mechanization of inertial sensors. The rigorous formulation of a state-space model for the system under analysis is pivotal to the development of a navigation filter fusing data from heterogeneous sources. It constitutes a substrate layer which fixes the physical and mathematical means allowing to construct a navigation solution at each time instant. It includes the description and modelling of the inertial motion dynamics of the navigating body which is tracked over time, as well as a mathematical model which, by handling and processing the raw measurements from an external sensor (GNSS in our context), converts them into a useful aid to update and refine the estimate of the system state. In literature, depending on the state-estimation procedure and the filtering system model, we can distinguish between two separate navigation modes for implementing an integrated INS/GNSS unit [29]:

- *direct* navigation mode, which is based on the standard inertial navigation equations described in Section 1.4.2, and where the integration filter estimates the absolute body position, velocity and attitude quantities together with the biases affecting INS sensors
- *indirect* navigation mode, which relies on inertial error-propagation equations and where the navigation filter estimates error-states useful to correct and refine earlier predictions of the associated absolute positioning quantities

Hereinafter, the thesis will address the development of a filtering system model agreeing with the indirect (or error-based) mode.

Given a INS/GNSS TC navigation system with indirect configuration, the system state-vector, which identifies the multi-dimensional and unknown output of the

navigation filter at every integration epoch, does not contain total states but rather corrections. In other words, as anticipated earlier, the integration filter, by relying on the set of ranging and ranging-rate measurements available at every GNSS-epoch, is involved in estimating correcting factors (or error components) which are meant to rectify the standalone INS high-rate predictions to the overall navigation states (i.e. body position, velocity and attitude, plus INS-sensors' biases). As such, the state-vector is more properly regarded to as an *error-state vector*. However, throughout the course of this thesis, the two terms are often confused, since it is always implicit the reference to an indirect architecture. Furthermore, as regards notation, the symbol δ indicates a correction to the quantity it is preposed to. For the integrated system under analysis, by selecting Earth-Centered Earth-Fixed (ECEF) frame coordinates, the error-state vector counts 17 elements (i.e. dimensions) and it is defined as follows [23]:

$$\begin{aligned}\boldsymbol{\delta}\mathbf{x}^e = \Big[\; & \delta r_x^e \; \delta r_y^e \; \delta r_z^e, \; \delta v_x^e \; \delta v_y^e \; \delta v_z^e, \; \delta \epsilon_x^e \; \delta \epsilon_y^e \; \delta \epsilon_z^e \\ & \delta b_{a,x}^b \; \delta b_{a,y}^b \; \delta b_{a,z}^b, \; \delta b_{g,x}^b \; \delta b_{g,y}^b \; \delta b_{g,z}^b, \; \delta t_u \; \delta \dot{t}_u \; \Big]\end{aligned} \tag{2.1}$$

where superscripts e and b refer to the Earth-fixed frame and the local body-frame respectively. From (2.1) we can identify the following terms:

- $\boldsymbol{\delta}\mathbf{r}^e = [\; \delta r_x^e \; \delta r_y^e \; \delta r_z^e \;]$ is the vector of position errors along the three Cartesian axes of the ECEF-frame
- $\boldsymbol{\delta}\mathbf{v}^e = [\; \delta v_x^e \; \delta v_y^e \; \delta v_z^e \;]$ is the vector of velocity errors along the three Cartesian axes of the ECEF-frame
- $\boldsymbol{\delta}\epsilon^e = [\; \delta \epsilon_x^e \; \delta \epsilon_y^e \; \delta \epsilon_z^e \;]$ is the vector of misalignment angles along each axis, expressed in ECEF coordinates
- $\boldsymbol{\delta}\mathbf{b}_a^b = [\; \delta b_{a,x}^b \; \delta b_{a,y}^b \; \delta b_{a,z}^b \;]$ is the vector collecting the estimated corrections to the biases of the accelerometers along the tri-axial orthogonal body-frame reference
- $\boldsymbol{\delta}\mathbf{b}_g^b = [\; \delta b_{g,x}^b \; \delta b_{g,y}^b \; \delta b_{g,z}^b \;]$ is the vector collecting the estimated corrections to the biases of the gyroscopes along the tri-axial orthogonal body-frame reference
- $\boldsymbol{\delta}t = [\; \delta t_u \; \delta \dot{t}_u \;]$ is the vector collecting the estimated receiver clock-bias and clock-drift corrections

Evidently, error-states related to position and velocity are represented in the ECEF-frame; attitude errors (which are linked to misalignment errors) and sensor biases are expressed in the local body frame.

2.2.2 System process model

This section is devoted to the discussion of the process (system) model, included its formalisation in state-space form, which is essential to mathematically describe the temporal evolution of the states (collected in (2.1)) for the tracked navigating body. For the TC-architecture, the dynamic system model is set from the mechanization of inertial navigation equations. However, according to the indirect paradigm, the resolution of the navigation equations as discussed in Section 1.4.2 is not sufficient to provide information about the errors affecting the total system states and the way such errors evolve and propagate over time. In other words, the standard equations of inertial motion blindly process data received from the IMU to obtain updated navigation parameters without regard to the veracity of these parameters [23].
That said, in the developed integrated framework, it is required to derive a mechanization involving inertial error propagation equations. This can be achieved through a perturbation of the intrinsically non-linear system of inertial equations, and the result is summarized in the following matrix equation [23]:

$$\begin{bmatrix} \dot{\delta\mathbf{r}}^{e} \\ \dot{\delta\mathbf{v}}^{e} \\ \dot{\boldsymbol{\epsilon}}^{e} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\delta}\mathbf{v}^{e} \\ -\mathbf{F}^{e}\boldsymbol{\epsilon}^{e} + \mathbf{N}^{e}\boldsymbol{\delta}\mathbf{r}^{e} - 2\boldsymbol{\Omega}_{ie}^{e}\boldsymbol{\delta}\mathbf{v}^{e} \\ -\boldsymbol{\Omega}_{ie}^{e}\boldsymbol{\epsilon}^{e} \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{R}_{b}^{e}\boldsymbol{\delta}\mathbf{f}^{b} \\ \mathbf{R}_{b}^{e}\boldsymbol{\delta}\boldsymbol{\omega}_{ib}^{b} \end{bmatrix} \tag{2.2}$$

where we observe that

- $\mathbf{F}^{e}$ is the skew-symmetric form of accelerometer measurements expressed in the ECEF-frame
- $\mathbf{N}$ is the tensor form of the gradient operator applied to the gravity vector $\mathbf{g}$
- $\boldsymbol{\delta}\mathbf{f}^{b}$ collects the body-frame triaxial error components on accelerometer measurements
- $\boldsymbol{\delta}\boldsymbol{\omega}_{ib}^{b}$ collects the body-frame triaxial error components on gyroscope measurements

However, the linearized error mechanization model in (2.2) would fit with the process model (error-propagation model) for an INS/GNSS integrated system only in case the inertial sensor error vectors, specifically $\boldsymbol{\delta}\mathbf{f}^{b}$ and $\boldsymbol{\delta}\boldsymbol{\omega}_{ib}^{b}$, agreed to a zero-mean multi-variate normal distribution. Unfortunately, the aforementioned conditions do not apply due to the presence of deterministic sensor bias components that move the average of such distribution [23]. Therefore, in [23], a simple but effective model (first-order linearization) for the inertial sensor errors is proposed. They can be written as

$$\boldsymbol{\delta}\mathbf{f}^b = \mathbf{b}_a + \boldsymbol{\omega}_f$$

$$\boldsymbol{\delta}\boldsymbol{\omega}_{ib}^b = \mathbf{b}_g + \boldsymbol{\omega}_\omega$$

where each sensor error is constituted by a bias component (with some hidden temporal variability), which embodies all time-dependent effects that are not explicitly addressed in such reduced and simplified error model, plus an additive noise component. Let's underline that the errors related to the bias components ($\boldsymbol{\delta}\mathbf{b}_a^b$ and $\boldsymbol{\delta}\mathbf{b}_g^b$) are elements of the error-state vector defined in (2.1).
Given this simplified sensor bias error-model, the resulting INS mechanization, in terms of error states, can be expressed in state-space form as follows [23]:

$$\begin{bmatrix} \dot{\boldsymbol{\delta}\mathbf{r}}^e \\ \dot{\boldsymbol{\delta}\mathbf{v}}^e \\ \dot{\boldsymbol{\epsilon}}^e \\ \dot{\boldsymbol{\delta}\mathbf{b}}_a^b \\ \dot{\boldsymbol{\delta}\mathbf{b}}_g^b \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{N}^e & -2\boldsymbol{\Omega}_{ie}^e & -\mathbf{F}^e & \mathbf{R}_b^e & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\boldsymbol{\Omega}_{ie}^e & \mathbf{0} & \mathbf{R}_b^e \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -diag(\boldsymbol{\alpha}_i) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -diag(\boldsymbol{\beta}_i) \end{bmatrix}}_{\boldsymbol{\Phi}} \begin{bmatrix} \boldsymbol{\delta}\mathbf{r}^e \\ \boldsymbol{\delta}\mathbf{v}^e \\ \boldsymbol{\epsilon}^e \\ \boldsymbol{\delta}\mathbf{b}_a^b \\ \boldsymbol{\delta}\mathbf{b}_g^b \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{R}_b^e & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_b^e & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}}_{\mathbf{Q}} \begin{bmatrix} \boldsymbol{\omega}_f \\ \boldsymbol{\omega}_\omega \\ \boldsymbol{\omega}_{b_a} \\ \boldsymbol{\omega}_{b_g} \end{bmatrix} \tag{2.3}$$

which is clearly a linear model, additive in noise, and time-dependent. Moreover, the following terms need to be specified:

- $diag\,(\boldsymbol{\alpha}_i)$ is a diagonal matrix for modelling the accelerometer bias states are first-order Gauss-Markov processes
- $diag\,(\boldsymbol{\beta}_i)$ is a diagonal matrix for modelling the gyroscope bias states are first-order Gauss-Markov processes
- $\boldsymbol{\omega}_{b_a}$ is the noise related to accelerometer bias components
- $\boldsymbol{\omega}_{b_g}$ is the noise related to gyroscope bias components

Finally, as highlighted in (2.3) through curly under-bracing, $\boldsymbol{\Phi}$ is a 17×17 *state-transition* matrix which determines the corresponding state-transition model (**process model**) in the integrated navigation system state-space representation. In other words, $\boldsymbol{\Phi}$ is the mathematical formalization of the dynamic model for the time-evolution of the errors affecting the inertial states. Furthermore, matrix $\mathbf{Q}$ identifies the 17×17 **process-noise covariance** matrix which statistically characterizes, in terms of variances and correlations, the noise components on sensor errors and on sensor bias errors, together with any mismodelling in the state-representation.

Before moving ahead with the discussion about the observation model, it is important to remark that the dynamic inertial error-model illustrated in (2.3), despite being linear, is still poorly handleable and practically not feasible in the context of a real-time implementation of an INS/GNSS fusion algorithm; thus, a further numerical approximation is required.
Using a Taylor series expansion, matrix $\mathbf{\Phi}$ can be formulated as [23]:

$$\mathbf{\Phi} = \mathbf{I} + \mathbf{F}\Delta t + \frac{(\mathbf{F}\Delta t)^2}{2!} + ... \tag{2.4}$$

and, truncating at the first-order term, it is eventually obtained the approximated error-dynamics matrix $\mathbf{F}$. Moreover, since $\mathbf{F}$ contains terms involving specific-force measurements, it is by nature time-dependent ($\mathbf{F}_k$) and is updated at the INS rate. An analogous procedure of numerical approximation is operated on the process noise covariance matrix $\mathbf{Q}_k$ which, in the end, is discretized and approximated in terms of matrix $\mathbf{G}_k$.

To conclude, the approximate linearized inertial error dynamic model can be summarized in the following equation [23]:

$$\boldsymbol{\delta}\mathbf{x}_{k+1} = \mathbf{F}_k\boldsymbol{\delta}\mathbf{x}_k + \mathbf{G}_k\mathbf{v}_k \tag{2.5}$$

The simplified linear model in (2.5) can be employed as process model in the framework of real-time implementation of an INS/GNSS TC-integrated system provided that, at system bootstrap, a calibration process is carefully conduced on inertial sensors [30].

2.2.3 System observation model

As briefly mentioned at the beginning of Chapter 2, the observation model, once mathematically formalized, represents the tool allowing the navigation filter to combine and process the new set of raw input measurements, at every integration epoch, in the view of constructing a refined update of the error-state vector estimate. In this sense, the same model sets up and specifies the actual relationship existing between the observations and the states. In the TC-scheme with indirect configuration, the measurement vector supplied at the integration filter input is obtained from the difference between the raw GNSS measurements and the INS-predicted measurements. The former measurements correspond to the estimates of pseudorange and Doppler-shift obtained from a GNSS-receiver exploiting ephemeris data of tracked satellites, while the latter measurements involve a prediction of pseudoranges and pseudorange-rates by combining the position and

velocity of satellites with the INS-predicted position and velocity states [21]. In this context, the difference between the two groups of measurements embodies the truly innovative information brought, as an external aid, to the navigation system [16].
Being N_{sat} the number of tracked satellites, the **observation vector** in the measurement model for a TC-architecture can be formalized as follows [27]:

$$\mathbf{z}_k = \boldsymbol{\zeta}_{sat,k} - \hat{\boldsymbol{\zeta}}_{ins,k} \tag{2.6}$$

where the following terms are specified:

- $\boldsymbol{\zeta}_{sat,k} = [\ \boldsymbol{\rho}_{sat,k},\ \dot{\boldsymbol{\rho}}_{sat,k}\ ]^T$ is a column vector of size $2N_{sat} \times 1$ collecting the raw GNSS pseudorange ($\boldsymbol{\rho}_k$) and Doppler-shift measurements ($\dot{\boldsymbol{\rho}}_k$) from visible satellites at k-th epoch
- $\hat{\boldsymbol{\zeta}}_{ins,k} = [\ \hat{\boldsymbol{\rho}}_{ins,k},\ \dot{\hat{\boldsymbol{\rho}}}_{ins,k}\ ]^T$ is a column vector of size $2N_{sat} \times 1$ collecting the pseudorange and pseudorange-rate predictions (related to the same set of satellites) from the current INS-estimate of the body trajectory

Recalling Section (1.3.2), it has been provided an extensive derivation of the Least-Squares (LS) method, applied to the non linear PVT system of equations, to retrieve a standalone GNSS-based PVT solution starting from a set of raw pseudorange measurements and given an initial approximation point. The linearized pseudorange model is specified in (1.13). Similarly, in Section (1.3.3), it has been described the linearized model, in state-space format, for the determination of the velocity and clock-drift navigation states starting from the raw GNSS-receiver Doppler-shift measurements. Such model is summarized in (1.32).
The combination of the aforementioned models, leads to the formulation of the following (linear) INS/GNSS observation model [16]:

$$\begin{bmatrix} \rho_{1,k} - \hat{\rho}_{1,k} \\ \vdots \\ \rho_{N_{sat},k} - \hat{\rho}_{N_{sat},k} \\ \hline \dot{\rho}_{1,k} - \hat{\dot{\rho}}_{1,k} \\ \vdots \\ \dot{\rho}_{N_{sat},k} - \hat{\dot{\rho}}_{N_{sat},k} \end{bmatrix} = \underbrace{\left[\begin{array}{ccccccc} \mathbf{u}_{1,k}^T & \mathbf{0}_{3\times1} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{u}_{N_{sat},k}^T & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 1 & 0 \\ \hline \mathbf{0}_{1\times3} & \mathbf{u}_{1,k}^T & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_{1\times3} & \mathbf{u}_{N_{sat},k}^T & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 0 & 1 \end{array}\right]}_{\mathbf{H}_k} \underbrace{\begin{bmatrix} \boldsymbol{\delta}\mathbf{r}^e \\ \boldsymbol{\delta}\mathbf{v}^e \\ \boldsymbol{\epsilon}^e \\ \boldsymbol{\delta}\mathbf{b}_a^b \\ \boldsymbol{\delta}\mathbf{b}_g^b \\ c\delta t_u \\ c\dot{\delta}t_u \end{bmatrix}}_{\boldsymbol{\delta}\mathbf{x}^e} + \boldsymbol{\epsilon}_k \tag{2.7}$$

where $\mathbf{u}_{j,k} = \frac{1}{R_j}[\ x_u - x_j,\ y_u - y_j,\ z_u - z_j\ ]$ is the line-of-sight unit vector, in ECEF-frame Cartesian coordinates, pointing from the approximation point, built

with the most recent updates to the INS position and velocity, to the j-th satellite at integration epoch k. As highlighted in (2.7) through curly under-bracing, $\mathbf{H}_k$ represents the system **observation matrix** having size $2N_{sat} \times 17$. It is worth remarking that $\mathbf{H}_k$, after a proper re-framing, includes the Jacobian matrix of the non-linear relationship between the user's position and clock and the set of N_{sat} pseudoranges. Moreover, observing the size of the measurement matrix, a relevant difference between the LC-scheme and the TC-scheme pops up. In fact, in the former, $\mathbf{H}_k$ has fixed size while, in the latter approach, it depends on the number of tracked satellites, that reasonably changes over time depending on the navigated environment. Finally, $\boldsymbol{\epsilon}_k$ collects the residual stochastic errors affecting, for each tracked satellite, the measurement vector $\mathbf{z}_k$ both in the pseudorange and in the Doppler components. The stochastic characterization, in terms of variances, for such errors is collected in the **observation-noise covariance** matrix $\mathbf{R}_k$.
From (2.7), the linearity of such model is clear. However, in the INS/GNSS integration context, the state of the system includes some quantities, among others, which are exclusively associated to the INS and its IMU-error model, while, on top of that, the GNSS noisy observations are coming from a source completely external to the INS. Thus, reasonably, there should exist an indirect relationship between the observations and the state-variable, at least up to some components. As a matter of fact, a linear dependency can be constructed if, instead of considering the raw GNSS-measurements directly, it is accounted for their difference w.r.t. a prediction of the same noisy observables based on the system inertial state (which is the idea behind the observation vector definition (2.6)). The latter difference, then, can be regarded as an error that, likely, is directly related to the system error-state variable [24].

2.2.4 The INS/GNSS tightly-coupled architecture

In previous sections, the theoretical framework for the state-space definition of a INS/GNSS TC system in the indirect configuration has been set-up. Given so, this section can be devoted to the analysis of the software implementation of the hybridized architecture which has been used for this thesis work.
First of all, the integrated navigation module is administrated by a Finite-State Machine (FSM) which acts as a scheduling unit to properly interleave the operations between the inertial navigation unit and the integration filter and to avoid conflicts. Simultaneously, the FSM has to manage the flow of input sensor data involving raw accelerometer specific-force measurements and gyroscopes angular-rate measurements from the IMU, and raw GNSS-observables produced as output of a GNSS-processing algorithm. As a matter of fact, the latter task is not straightforward since the two integrated systems, having complementary natures, operate at consistently different rates. On one hand, the GNSS processing unit can

supply Time-of-Arrival (ToA) based pseudorange and Doppler-shift measurements at low-rate, which is usually upper-bounded to 1 Hz (this depends on the data-rate of the navigation message). On the other hand, the INS-sensors produce raw specific force and angular-rate measurements at much higher rate and, correspondingly, the overall INS-filter elaborates such data at a convenient rate to keep up with the IMU pace.
In light of the above, to solve the synchronization problem between the two coupled units, the FSM assigns timestamps separately to both inertial data and GNSS observables. Consequently, scheduling is operated following the timestamps' order and it is not shared any common clock.
In the developed integration software, which should mimic a realistic operating scenario, the GNSS-rate is set to 1 Hz while the INS-rate is fixed to 10 Hz with an IMU-rate of 100 Hz. A schematic representation of the integration software is provided in Figure 2.3. Every 10 ms, a new set of inertial sensors measurements is produced. For each group of raw data, the inertial processing unit solves the *coning* and *sculling* integrals. Such integrals are related to coning and sculling motions which, in turn, are related to the non-linear evolution of the navigating body attitude. Due to non linearities of inertial motion, it is fundamental to solve these integrals at high rate. However, this approach is beneficial in terms of complexity. In fact, through coning and sculling, the change of the target orientation and velocity is finely tracked with high temporal granularity and the update to the inertial states, provided by the resolution of the mechanization, can be done at a lower rate of 10 Hz, that is every 100 ms. All these operations, clearly, are executed within the INS subsystem.
Instead, the timing for the integration filter operations is dictated by the arrival instants of groups of GNSS-observables. Any time a new set of pseudorange and Doppler-shift measurements is available, the integration filter applies the observation model to generate an update, in the flavour of a set of corrections, to the inertial predictions.
However, in such framework, there is a considerable difference between the two integrated systems: while inertial sensor data are produced with a fixed periodicity being the IMU completely self-contained, the arrival frequency of GNSS data is not stable since it is subjected to external world effects. Therefore, the minimum temporal width between consecutive updates to the navigation solution is one second, but it could be potentially wider. The latter consideration emphasizes the importance of the INS as an interpolator device between consecutive GNSS-epochs, which lets to enhance the resolution in the tracking process of the motion dynamics for a given target [24].
For sake of completeness, it might be useful to detail more on the estimation process involving navigation states.

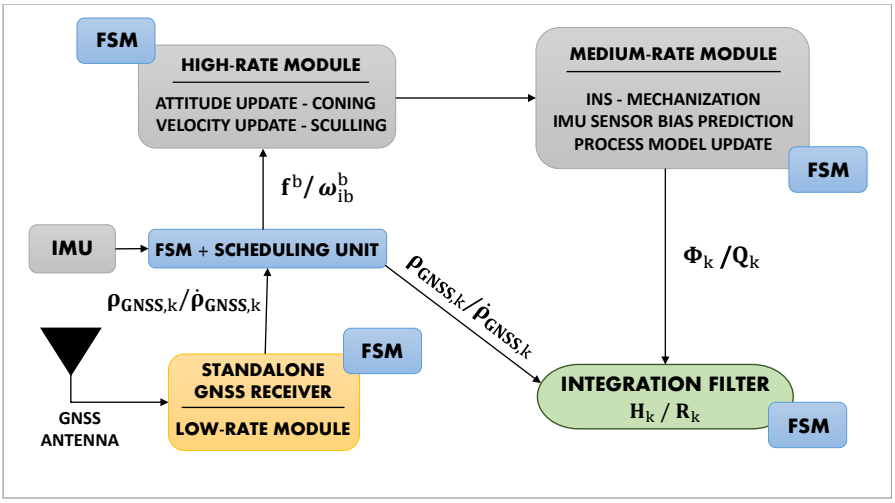


Figure 2.3: Flow-diagram of the integration software for the real-time simulation of an INS/GNSS TC-architecture

The medium-rate processing unit inside the INS, solving the mechanization equations, produces medium-rate predictions to the absolute position, velocity and attitude information related to the tracked vehicular target. At the same time, knowing the change in the navigation states with respect to the previous prediction, the INS can update the dynamic model for the evolution of the error affecting the inertial states of the moving body. As already discussed in Section 2.2.2, this model is characterized in terms of the linearized transition matrix $\boldsymbol{\Phi}_{k-1}$ (or its approximation $\mathbf{F}_{k-1}$) and of the process noise covariance $\mathbf{Q}_k$ (or its approximation $\mathbf{G}_k$). The updated state-transition error model, then, can be used to upgrade the prediction of the errors on accelerometers and gyroscopes bias states.
Whenever a new set of GNSS noisy observables becomes available, the integration filter steps in the integration chain and, by relying on the inertial error dynamic model from the INS and by constructing the matrices involved in the observation model (Section 2.2.3), produces an estimate $\hat{\boldsymbol{\delta}}\mathbf{x}^e$ of the error-state vector defined in (2.1). In the end, such (low-rate) estimate is used to correct and refine the most recent and prior INS-predictions to the vehicle trajectory and to the inertial sensor biases.
It is worth noting that, by its nature, a INS can only measure relative navigation states, unless some initialization is set-up. To this end, at integration software bootstrap, after the inertial calibration process, the first PVT-solution is obtained by running a WLS on the first incoming set of GNSS-observables.
As a final remark, the timing information inside the integrated system state-vector cannot be estimated from inertial data. Therefore, a new estimate of the GNSS-receiver clock-bias and drift can only coincide with an update operation executed

(at low rate) by the integration filter, and no intermediate prediction is possible.

Latency management

When implementing an integrated navigation software which can faithfully simulate a real-time system, latency issues should be accounted for. As a matter of fact, GNSS observables do not manifest a regular behaviour in their arrival frequency, for different reasons: GNSS-outages, ranging signal losses etc. When integrating two systems, however, such delays must be properly managed. Among the options presented in [23], the navigation software for this thesis is developed according to the replay-buffer solution, which is shown in Figure 2.4.
IMU data coming from inertial sensors are immediately stored and processed as they are available, thus being never subjected to delays (real-time processing). If a new set of GNSS-observables is expected but it has some unknown delay, the INS goes on processing the incoming inertial measurements but also buffers them. Then, when the latent GNSS-observables arrive, the buffered data are restored and the GNSS-update is accomplished.
Moreover, in case the delay interval becomes too large, a GNSS-outage is declared and the corresponding update is skipped.

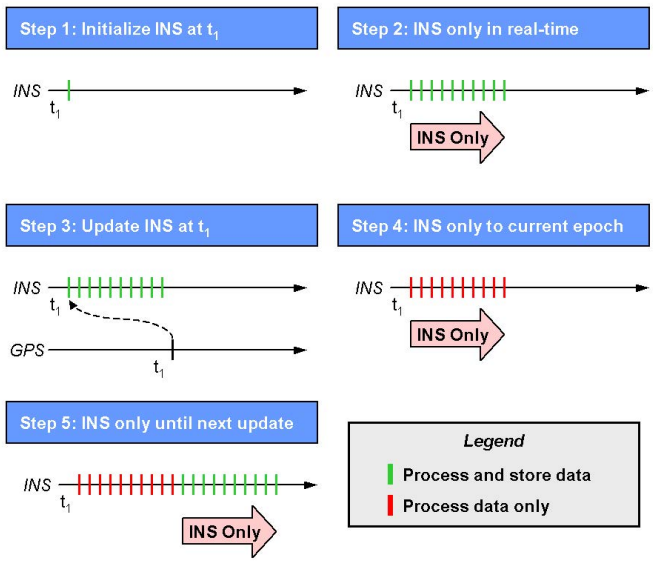


Figure 2.4: Managing of a replay-buffer to control latency in GNSS-data. Picture taken from [25].

Chapter 3

Advanced Bayesian signal processing for INS/GNSS hybridization

Chapter 2 has presented the major architectural solutions for the construction of a Tightly Coupled (TC) Inertial Navigation System (INS)/Global Navigation Satellite System (GNSS) integrated navigation unit. In particular, a stronger focus has been oriented to an *indirect* (or *error-state*) configuration, and the details about the definition of a discrete state-space representation, suitable for the accurate modelling of the dynamic motion of a generic body-target, have been specified. The latter model is meant to describe, with the highest possible degree of resolution, the evolution of the dynamics of both the system-state and the measurements, jointly with the associated noise statistics. However, at the same time, such model must be simple enough to limit the complexity of the filtering algorithm [31].
The current chapter, then, zooms on the most relevant module of the whole TC-scheme (refer to Figure 2.3), the *integration* (or *navigation*) filter. The latter can be more properly regarded to as a mathematical estimator that, by relying on the aforementioned dynamic system model, blends the information brought about by INS and GNSS measurements, in order to produce an estimate of the error-state vector (2.1).
Given the intrinsic complexity and time-dependence of both the system state-transition model and the measurement model, non-linear time-variant estimators are considered. Despite multiple approaches can be followed for implementing a discrete mathematical estimator, in the context of sensor fusion for high-accuracy positioning in vehicle-tracking applications, the *Bayesian approach* [32] is usually preferred. According to the Bayesian method, the estimation problem becomes statistical. On one hand, the stochastic integration filter relies on some a-priori

(i.e. deterministic) information about the state temporal evolution, which is encompassed inside the state-transition model; on the other hand, the system state-vector is treated as a random variable (r.v.) whose most likely realization, given prior data and input measurements, defines the output of the navigation filter [32].

In the Bayesian framework, the state-of-art solution for filtering applications over a generic non-linear discrete state-space model is represented by the Extended Kalman Filter (EKF), thanks to its strikingly low complexity coupled with a consistent robustness throughout the whole estimation routine. The latter filter defines a sub-optimal implementation, based on a first-order state-space linearization, of the Kalman Filter (KF), which represents the target ideal Bayesian estimator, given its capability of pursuing Maximum a-Posteriori (MAP) state-estimation. Unfortunately, the KF is optimal only under a set of restricting assumptions related both to the state-space characteristics and to the measurement noise modelling, thus resulting not feasible for real fusion algorithms.
Starting from the critical review of a pre-existent Kalman-based hybridization structure, the proposal of this thesis consists on the implementation and performance assessment of more advanced and innovative Bayesian algorithms, acting as integration filters in the framework of the INS/GNSS TC-architecture presented throughout Chapter 2. These filters are presented and developed according to an incremental path in terms of architectural and computational complexities, with the goals of improving the accuracy-level of the state-estimate and, more important, of identifying which solution determines the best compromise in terms of accuracy-complexity trade-off.

This chapter is organized as follows. Section 3.1 presents the general framework of a recursive Bayesian estimator which can be applied to any generic non-linear non-Gaussian state-estimation problem. Moreover, an optimal algorithm implementing recursive Bayesian estimation is discussed.
Section 3.2, then, presents the EKF as the state-of-art sub-optimal Bayesian solution for fusing INS and GNSS. Later, in Section 3.3, the Unscented Kalman Filter (UKF) architecture is treated, which consists of a more-refined non-linear filtering solution in the class of Kalman filters.
After that, in Section 3.4, the Particle Filter (PF) approach is discussed, as an alternative class of non-linear Bayesian estimators implementing Monte-Carlo approximation; in such context, the strengths and weaknesses of this approach, also compared to Kalman-based solution, are investigated.
Finally, Section 3.5 presents an innovative Unscented Particle Filter (UPF)-architecture which hybridizes the two classes of Kalman filters and Particle filters.

3.1 Recursive Bayesian state-estimator

In this section, the general concept of Bayesian estimation is presented tailored to the framework of target-tracking applications under non-linear and non-Gaussian system state-space characterizations. To avoid loss of generality, we consider an unspecified discrete-time state-space representation, both for the process and the observation models.
Recursive Bayesian estimation involves a probabilistic approach to estimate, recursively over time, an unknown Probability Density Function (p.d.f.), by using a set of external noisy measurements together with a pre-defined process model. In turns, this p.d.f. is used to estimate the dynamic state of a system; the latter, is mathematically formalized through the state vector $\mathbf{x}$, which includes position and velocity states plus other states (in the case of an INS/GNSS TC-integrated system, the state vector takes the form in (2.1), accounting for an indirect configuration) [33].
In particular, given a set of available incoming observations at epoch k, collected in the *measurement vector* $\mathbf{z_k} \in \mathbb{R}^{N_z}$, and a sequence of past measurements $\mathbf{z}_{1:k-1} = [\mathbf{z}_1, ..., \mathbf{z}_{k-1}]$, the Bayesian filter estimates the system state-vector $\mathbf{x}_k \in \mathbb{R}^{N_x}$ at same epoch k, as the most probable realization of an unknown random vector $\mathbf{X}_k$ with known a-priori distribution $p(\mathbf{x})$ [33].
At the basis of Bayesian recursion, it lies the assumption that the system state agrees to a first-order Markov model, and more precisely to a Hidden Markov Model (HMM). According to such paradigm, the same system can be viewed as a stochastic process whose states are not directly observable (i.e. hidden) and can be inferred from measurements which, instead, are available at system input [32]. In light of the previous statement, the Bayesian algorithm has to solve the following problem in the end: given an initial state $\mathbf{x}_0$ with known initial distribution $p(\mathbf{x}_0)$, the sequence of hidden states $\mathbf{x}_{0:k} = [\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_k]$ has to be estimated, given the knowledge of the observation sequence $\mathbf{z}_{1:k} = [\mathbf{z}_1, ..., \mathbf{z}_k]$ and the specification of a discrete state-space model. The latter, according to the HMM assumption, can be formalized as follows [16, 32]:

$$\mathbf{x}_k = f_{k-1}(\mathbf{x}_{k-1},\ \mathbf{u}_{k-1},\ \mathbf{v}_{k-1}) \tag{3.1}$$

$$\mathbf{z}_k = h_k(\mathbf{x}_k,\ \mathbf{u}_k,\ \mathbf{w}_k) \tag{3.2}$$

where (3.1) is defined as *process/state-transition* equation and models the dynamic evolution of the system states as a discrete-time stochastic process (it represents the process model), while (3.2) is referred to as *measurement equation* and characterizes the relationship between the states and the measurements (it represents the

observation model). Specifically, $\mathbf{f}_{k-1}$ and $\mathbf{h}_k$ are the *state-transition* (or *process*) *function* and the *measurement* (or *observation*) *function*, respectively, which can be either linear or non-linear in general. In practice, in the framework of the INS/GNSS TC-architecture treated in Chapter 2, the process function $\mathbf{f}_{k-1}$ is identified by the approximated error-dynamics matrix $\mathbf{F}_{k-1}$ in (2.5), while the observation function $\mathbf{h}_k$ is known (and simple) for GNSS (refer to (2.7)). Moreover, $\mathbf{v}_{k-1}$ identifies the *process noise* vector, which includes any mismodelling in the state-space characterization and additional unmodelled effects, while $\mathbf{w}_k$ defines the *observation noise* vector and gathers all impairments affecting the sequence of input measurements. Both noises are assumed to be statistically independent from the state-variables at all times, mutually independent each other and with known statistics. Finally, $\mathbf{u}_k$ is a vector collecting the set of input deterministic signals to the system (for example, in a vehicle, they might correspond to output signals generated by a closed-loop feedback controller with actuators or signals from temperature and stability sensors).
Observing the process equation in (3.1), the effect of the Markovian assumption is clear: the current system state vector $\mathbf{x}_k$ only depends, through the state-transition model, on the previous state realization $\mathbf{x}_{k-1}$ at estimation epoch $k-1$, and it is not affected at all by the past history of system states $\{\mathbf{x}_{k-1}, ..., \mathbf{x}_0\}$.
The discrete system state-space representation can be equivalently reformulated in terms of probability distributions as [32]:

$$\mathbf{x}_k \sim p\left(\mathbf{x}_k|\mathbf{x}_{k-1}\right) \tag{3.3}$$

$$\mathbf{z}_k \sim p\left(\mathbf{z}_k|\mathbf{x}_k\right) \tag{3.4}$$

where (3.3) is simply a reformulation of (3.1) and $p\left(\mathbf{x}_k|\mathbf{x}_{k-1}\right)$ is named *state-transition density* (or *motion model*), which tells where the system is expected to move at time k, given the prior knowledge of the previous state $\mathbf{x}_{k-1}$ and the availability of a model for the description of the dynamic state evolution. On the other side, (3.4) is a reformulation of (3.2) and $p\left(\mathbf{z}_k|\mathbf{x}_k\right)$ is typically called *likelihood function* (or *perception model* [33]). For sake of meticulousness, $p\left(\mathbf{z}_k|\mathbf{x}_k\right)$ is a probability distribution and not a likelihood. The difference is subtle but hides the fundamental paradigm of Bayesian inference. In fact, using the likelihood, the input measurements would represent the fixed deterministic conditioning information and the hypothesis on the system-state (i.e. $\mathbf{x}_k$) would identify the variable quantity (a Maximum-Likelihood estimator attempts at maximizing the hypothesis). On the contrary, the Bayes theorem, which lies at the basis of any Bayesian estimator, overturns the prior conditioning order allowing to move from a likelihood to a conditional distribution. The latter, then, sets up the model allowing to exploit the

innovation from input measurements to derive an a-posteriori update to the state variable at time k. That said, the term "likelihood" is indeed employed throughout the chapter, but it is important to be aware of the committed conceptual inaccuracy.

According to the Bayesian perspective, all the information necessary to retrieve the estimate of $\mathbf{x}_{0:k}$ given the observations sequence $\mathbf{z}_{1:k}$, is included in the joint a-posteriori conditional state-distribution $p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$. However, thanks to the Markovian assumption, at epoch k, the marginal posterior distribution $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ is fully sufficient to produce an estimate of $\mathbf{x}_k$ [32].
$p(\mathbf{x}_k|\mathbf{z}_{1:k})$ is called *Bayesian belief* [33] and Bayesian filters aim at sequentially estimating it over the state-space, given the information coming from external sensors. At every estimation epoch, the belief can be obtained through two steps: **prediction** and **update**.
In the prediction step, the last estimated belief at epoch $k-1$, that is $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$, is propagated to the present epoch k using the motion model (3.3), hence leading to the determination of the *a-priori density* [32]

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})\, p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})\; d\mathbf{x} \tag{3.5}$$

Then, in the update step, when the input measurement vector $\mathbf{z}_k$ is available, the conditional *a-posteriori density* is obtained from Bayes' rule [32]:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)\, p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \tag{3.6}$$

Being the density $p(\mathbf{z}_k|\mathbf{z}_{1:k-1})$ at denominator the integral of the numerator over the state-space, we can consider it as a normalization factor c, and (3.6) can be re-written as [16]:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = c \cdot p(\mathbf{z}_k|\mathbf{x}_k) \cdot p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) \tag{3.7}$$

From (3.7), a recursive pattern appears: the Bayesian belief at k-th epoch is proportionally related to the a-priori density $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ which, in turn, depends on the estimate of the Bayesian belief at previous epoch $k-1$.

The last step of the general Bayesian algorithm requires to perform a *Bayesian inference* [16], corresponding to the extrapolation of an estimate $\hat{\mathbf{x}}_k$ of the true state-vector $\mathbf{x}_k$ from the a-posteriori density $p(\mathbf{x}_k|\mathbf{z}_{1:k})$. As a goodness principle, the MAP-criterion is typically adopted: the selected output state-vector estimate coincides with the realization of $\mathbf{x}_k$ maximizing the Bayesian belief $p(\mathbf{x}_k|\mathbf{z}_{1:k})$.

Specifically [32]:

$$\hat{\mathbf{x}}_{k,MAP} = arg \max_{\mathbf{x}_k} \left[p\left(\mathbf{x}_k | \mathbf{z}_{1:k}\right) \right] \tag{3.8}$$

3.1.1 Optimal Bayesian estimation: the Kalman Filter

Before entering the details of the KF algorithm, it is worth specifying some notation, concerned with the system state-vector, which is going to be adopted in the remainder of this chapter:

- $\mathbf{x}_{k-1|k-1}$ identifies the system a-posteriori state at epoch $k-1$. In particular, the subscript $k-1|k-1$ specifies that the estimate at $k-1$ is conditioned upon the information brought about by the measurements at time $k-1$.
- $\mathbf{x}_{k|k-1}$ identifies the system a-priori state at epoch k. In particular, the subscript $k|k-1$ specifies that the state-hypothesis at k is conditioned upon the a-priori information that the system owns on the basis of the old measurements observed at time $k-1$.
- $\mathbf{x}_{k|k}$ identifies the system a-posteriori state at epoch k. In particular, the subscript $k|k$ specifies that the state-hypothesis at k is refined, w.r.t. $\mathbf{x}_{k|k-1}$, based on the innovative information carried by the new set of measurements at time k.

Previous Section 3.1 introduced a high-level perspective on the conceptual solution for Bayesian filtering algorithms applied to discrete-time state-space models. In reality, such solution can be only treated analytically. In fact, in most of the cases, it is not possible to characterize (3.3) and (3.4) with enough statistical resolution [32], and the integrals involved in the computation of the a-priori (3.5) and a-posteriori (3.7) state densities cannot be solved in closed-form [16].
However, if the discrete state-space model satisfies the following conditions:

- the state-transition model f_{k-1} is a known linear function of $\mathbf{x}_{k-1}$, $\mathbf{u}_{k-1}$ and $\mathbf{v}_{k-1}$
- the observation model h_k is a known linear function of $\mathbf{x}_k$ and $\mathbf{w}_k$
- the process noise $\mathbf{v}_k$ and the observation noise $\mathbf{w}_k$ are zero-mean additive and with known Gaussian statistics

then, the KF is found to be an optimal recursive Bayesian filter, since it can achieve MAP state-estimation. In the following, the general KF-algorithm is presented.
The constrained additive linear and Gaussian state-space model, under which the optimality of this filter is guaranteed, can be expressed through the following

process (3.9) and measurement (3.10) equations [32]:

$$\mathbf{x}_k = \boldsymbol{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{v}_k \tag{3.9}$$

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{w}_k \tag{3.10}$$

where [32] $\boldsymbol{F}_{k-1}$ is the linear state-transition matrix and $\mathbf{H}_k$ represents the linear observation matrix. Moreover, justified by the assumptions, the process noise vector $\mathbf{v}_k$ and the measurement noise vector $\mathbf{w}_k$ are considered to be drawn from a zero-mean Gaussian distribution with noise covariances $\mathbf{Q}_k$ and $\mathbf{R}_k$ respectively.
The fundamental aspect which characterizes the KF algorithm is that of considering the Bayesian belief as normally distributed, implying that a full statistical characterization is possible by simply knowing its mean vector and its covariance matrix. Looking at the posterior density at epoch $k-1$, it has the following statistical description:

$$p\left(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}\right) \sim \mathcal{N}\left(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}\right) \tag{3.11}$$

where $\hat{\mathbf{x}}_{k-1|k-1}$ is the last system state-estimate and $\mathbf{P}_{k-1|k-1}$ the related state-covariance matrix. As such, given the recursive pattern in (3.7), the a-priori state density, the likelihood density and the a-posteriori density functions at the current epoch k can be regarded as normally distributed too.
Similarly to the general Bayesian algorithm discussed in Section 3.1, the KF-routine implements the prediction and update steps. In the **prediction step**, the a-priori state estimate $\hat{\mathbf{x}}_{k|k-1}$ is obtained by propagating the last a-posteriori state-estimate through the linear process model. Formally [32]:

$$\hat{\mathbf{x}}_{k|k-1} = \boldsymbol{F}_{k-1}\hat{\mathbf{x}}_{k-1|k-1} \tag{3.12}$$

$$\mathbf{P}_{k|k-1} = \mathbf{Q}_k + \boldsymbol{F}_{k-1}\mathbf{P}_{k-1|k-1}\boldsymbol{F}_{k-1}^T \tag{3.13}$$

In the **update step**, then, when a new input measurement sequence $\mathbf{z}_k$ is available, the innovation vector $\mathbf{s}_k$ is computed as:

$$\mathbf{s}_k = \mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1} \tag{3.14}$$

From (3.14), it is clear that the innovation vector, being the difference between the observed measurements and the predicted measurements using the observation

matrix, includes the truly new information on the system dynamic behaviour that external observables have brought about. Thus, the filter itself must exploit such new information to update the state-prediction optimally.
Given so, the update step can be summarized through the following equations [32]:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \left(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1} \right) \tag{3.15}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^T \tag{3.16}$$

with $\mathbf{S}_k$ being the *innovation sequence covariance* matrix and $\mathbf{K}_k$ being the *Kalman Gain* matrix. The latter quantities are computed as follows [32]:

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \tag{3.17}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \tag{3.18}$$

In particular, the Kalman gain $\mathbf{K}_k$ has an outstanding importance in the economy of the algorithm since, depending on the values of its entries, the filter tunes the impact of external observations in correcting the a-priori state-estimate as a result of the update step.

3.2 Extended Kalman Filter

The optimality conditions for the KF state-estimate are in general too restrictive and have a poor bearing on applications involving real navigation scenarios, such a INS/GNSS fusion. In most cases, the system process and observation models might be non-linear, regardless of the specific degree of such non-linearity and, even in case they were linear, the system and measurement noises could be characterized by unknown non-Gaussian multivariate distributions which, furthermore, might not be centred in zero [32, 33]. For these applications, the optimal state-estimate in Bayesian-sense cannot be achieved. Thus, it is necessary to resort to sub-optimal Bayesian filtering approaches that can better fit with real scenarios. In such framework, as already hinted at in the introduction to this Chapter, the highly-popular state-of-art solution is represented by the EKF.

Considering a non-linear non-Gaussian set-up, the fundamental idea at the basis of the EKF-algorithm is to operate a local linearization of the discrete system state-space model while preserving the Gaussian constraint on the stochastic densities [32]. That said, it becomes clear that the sub-optimality of such filter does not reside in the filter itself, but rather on the way the filter handles the model. In fact, in the EKF-context, the original non-linear problem is transformed into its linear approximation which is solved optimally (MAP) [28]. Consequently, the EKF-performance heavily depends on how much the linearized problem is accurately modelled.

3.2.1 General EKF algorithm

The discrete-time state-space representation for non-linear dynamic state-space models can be written as [32]:

$$\mathbf{x}_k = f_{k-1}\left(\mathbf{x}_{k-1}\right) + \mathbf{v}_k \tag{3.19}$$

$$\mathbf{z}_k = h_k\left(\mathbf{x}_k\right) + \mathbf{w}_k \tag{3.20}$$

where f_{k-1} is a known non-linear process function and h_k is a known non-linear measurement function. Similarly to the KF-case, both the process noise $\mathbf{v}_k$ and the observation noise $\mathbf{w}_k$ are mutually independent random vectors, statistically characterized by zero-mean multivariate Gaussian distributions with covariance matrices $\mathbf{Q}_k$ and $\mathbf{R}_k$, respectively.
Moreover, retaining the assumption on the Gaussianity of the a-posteriori p.d.f., the state estimate is obtained through the two-stage process of prediction and

update.
In the **prediction step**, a linearization of the state-transition function is operated locally, around the updated state-estimate $\hat{\mathbf{x}}_{k-1|k-1}$ at previous epoch, by applying the gradient operator to $f_{k-1}(\mathbf{x}_{k-1})$. The linearized function can be expressed in the following manner [32]:

$$\hat{\mathbf{F}}_{k-1} = \left[\nabla_{\mathbf{x}_{k-1}} f_{k-1}(\mathbf{x}_{k-1})\right]_{|\ \mathbf{x}_{k-1}=\hat{\mathbf{x}}_{k-1|k-1}} \tag{3.21}$$

Given $\hat{\mathbf{F}}_{k-1}$, the a-priori predicted state estimate $\hat{\mathbf{x}}_{k|k-1}$ and its covariance $\mathbf{P}_{k|k-1}$ can be obtained as [32]:

$$\hat{\mathbf{x}}_{k|k-1} = \hat{\mathbf{F}}_{k-1}\hat{\mathbf{x}}_{k-1|k-1} \tag{3.22}$$

$$\mathbf{P}_{k|k-1} = \mathbf{Q}_k + \hat{\mathbf{F}}_{k-1}\mathbf{P}_{k-1|k-1}\hat{\mathbf{F}}_{k-1}^T \tag{3.23}$$

Later, in the **update step**, similarly to the above, a local linearization of the non-linear observation function h_k is operated around the a-priori predicted state-estimate at current epoch [32]:

$$\hat{\mathbf{H}}_k = \left[\nabla_{\mathbf{x}_k} h_k(\mathbf{x}_k)\right]_{|\ \mathbf{x}_k=\hat{\mathbf{x}}_{k|k-1}} \tag{3.24}$$

At this point, given the linearized observation function $\hat{\mathbf{H}}_k$ and the input measurement sequence $\mathbf{z}_k$, the innovation (or residual) sequence $\mathbf{s}_k$ can be obtained:

$$\mathbf{s}_k = \mathbf{z}_k - \hat{\mathbf{H}}_k\hat{\mathbf{x}}_{k|k-1} \tag{3.25}$$

In (3.25), the innovation sequence is computed as the residual between the observed measurement vector $\mathbf{z}_k$ and the predicted measurements using the approximate linearized observation function $\hat{\mathbf{H}}_k$.
Next, the Kalman Gain is computed [32]:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\,\hat{\mathbf{H}}_k^T\,\mathbf{S}_k^{-1} \tag{3.26}$$

where $\mathbf{S}_k = \hat{\mathbf{H}}_k\,\mathbf{P}_{k|k-1}\,\hat{\mathbf{H}}_k^T + \mathbf{R}_k$ is the residual sequence covariance matrix.
In the end, the a-posteriori state estimate $\hat{\mathbf{x}}_{k|k}$ and the a-posteriori state-covariance $\mathbf{P}_{k|k}$ are computed as [32]:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{s}_k) \tag{3.27}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \, \mathbf{S}_k \, \mathbf{K}_k^T \tag{3.28}$$

The general EKF-routine is summarized in Algorithm 1.

Algorithm 1 Summary of EKF algorithm

Assumptions: Non-linear state-transition function f_{k-1} and observation function h_k + zero-mean Gaussian distributed process and observation noises.
Require: knowledge of the initial state $\mathbf{x}_0$ and state-covariance $\mathbf{P}_0$.

1: **for** $k = 1 : \inf$ **do**

Prediction step

2: Estimate the a-priori state-vector $\hat{\mathbf{x}}_{k|k-1}$ as:

$$\hat{\mathbf{x}}_{k|k-1} = \hat{\mathbf{F}}_{k-1}\hat{\mathbf{x}}_{k-1|k-1}$$

where $\hat{\mathbf{F}}_{k-1}$ identifies the linearized system model. The latter is obtained using the gradient operator applied locally, around the a-posteriori estimated state at $k-1$, that is $\hat{\mathbf{x}}_{k-1|k-1}$,to the non-linear process function:

$$\hat{\mathbf{F}}_{k-1} = \left[\nabla_{\mathbf{x}_{k-1}} f_{k-1}\left(\mathbf{x}_{k-1}\right)\right]_{|\ \mathbf{x}_{k-1}=\hat{\mathbf{x}}_{k-1|k-1}}$$

3: Estimate the a-priori state covariance $\mathbf{P}_{k|k-1}$ related to the a-priori predicted state-vector $\hat{\mathbf{x}}_{k|k-1}$:

$$\mathbf{P}_{k|k-1} = \mathbf{Q}_k + \hat{F}_{k-1}\mathbf{P}_{k-1|k-1}\hat{F}_{k-1}^T$$

Update step

4: Estimate the innovation sequence $\mathbf{s}_k$ as the residual between the observed measurement vector $\mathbf{z}_k$ and the predicted observation sequence from the measurement model:

$$\mathbf{s}_k = \mathbf{z}_k - \hat{\mathbf{H}}_k\hat{\mathbf{x}}_{k|k-1}$$

where $\hat{\mathbf{H}}_k$ is the local linear approximation, using the gradient operator, of h_k.

5: Estimate the Kalman Gain matrix $\mathbf{K}_k$ as:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \cdot \hat{\mathbf{H}}_k^T \cdot \mathbf{S}_k^{-1}$$

with $\mathbf{S}_k = \hat{\mathbf{H}}_k\mathbf{P}_{k|k-1}\hat{\mathbf{H}}_k^T + \mathbf{R}_k$ being the residual sequence covariance.

6: Compute the a-posteriori state-estimate and its covariance:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\left(\mathbf{z}_k - \mathbf{s}_k\right)$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T$$

7: **end for**

3.2.2 INS/GNSS EKF-based integration filter

After the establishment of a purely theoretical framework for the description of the general EKF-algorithm for non-linear and Gaussian state-estimation problems, the application of such estimator as a navigation filter is now investigated in the framework of a INS/GNSS TC-architecture.
First of all, it is reminded that, implementing the sensor-fusion algorithm according to an indirect (error-based) approach, the state vector $\mathbf{x}_k$ is indeed an error-state vector $\boldsymbol{\delta}\mathbf{x}_k$, which contains corrections (incremental states) to the fundamental 3D position, velocity and attitude states of the navigating vehicular target, estimates of the inertial sensors bias-errors (according to a simplified modelling presented in Section 2.2.2) and clock-bias and clock-drift terms (timing information). The mathematical representation of the system error-state vector was given in (2.1). Moreover, for the same integrated system, the discrete state-space model is that formalized in Sections 2.2.2 and 2.2.3. As long as the process model is concerned, it coincides with the approximated dynamic inertial error-mechanization model in (2.3). On the other hand, the measurement model is embodied in the formulation of (2.7). Clearly, both models are linear, and this is a remarkable condition in light of implementing an EKF-based hybridization filter.
Moving the attention to noises, both the system process noise and the system observation noise are considered as mutually independent and additive. However, based on the available information, the degree of accuracy in their statistical characterization is different. For the process noise, the accuracy resolution in the modelling is quite high. Usually, any realization of the process noise sequence is gathered from a zero-mean multi-variate Gaussian distribution. The zero-mean assumption is valid since all the deterministic bias components affecting the system state-variable (bias turn-on, temperature variability, inertial body-axes alignment and others) are compensated within the INS, partly through an initial calibration process on inertial sensors, and partly through a de-noising (with temperature compensation) operated on raw inertial sensors data. Consequently, the residual noise affecting the states can be reasonably considered to have null bias. Concerning the state covariance matrix $\mathbf{Q}_k$, its elements are initially estimated at calibration-time and then, any time the INS solves the mechanization at medium-rate, updated through $\mathbf{C}_b^e$ (refer to 1.33).
Moving the attention towards noises affecting GNSS-measurements, we can consider them to have zero-mean in ideal conditions, but not in general. In fact, the GNSS-unit inside the navigation software, compensates for all the deterministic (i.e. a-priori known) bias components, including ionospheric and tropospheric induced delays, satellite-clock error, relativistic effects etc. Then, ideally, the stochastic residual-error would be unbiased. However, in real scenarios, there might be unpredictable and uncontrollable error sources (e.g. multipath) injecting some

extra-bias on ranging and ranging-rate measurements, thus violating the foregoing zero-bias condition. As regards the statistical distribution for such error, if using the EKF as sub-optimal Bayesian estimator implementing the fusion filter, the standard Gaussian model is selected. The normal distribution is the preferred choice since, in most of cases, gets closest to the true unknown distribution of the error. Moreover, being the involved noises directly modelled inside the covariance matrix $\mathbf{R}_k$, it is forced to employ the same statistical characterization for all the measurements. However, when navigating in highly-reflective environments (e.g. urban canyons) some pseudorange measurements might be affected by multipath and other major impairments. In such case, the corresponding residual errors would be more suitably modelled through other distributions [34].

Given the presented framework, the EKF-algorithm in the INS/GNSS integrated system architecture is implemented inside the navigation filter module (Figure 2.3) and agrees with the flow presented in former Section 3.2.1.

The **prediction** step relies on the most recent INS update to the dynamic error-state transitional model, embodied by matrices $\mathbf{\Phi}$ and $\mathbf{Q}_k$. It must be noticed that $\mathbf{\Phi}$ indicates the general continuous-time linear and time-variant process model, but the integrated system under analysis is inherently discrete. In such context, similarly to the general EKF-algorithm, the adopted state-transition model is identified by the approximated error-dynamics matrix $\mathbf{F}_k$ defined in (2.4). The latter matrix is a discrete approximation of the linear process matrix $\mathbf{\Phi}$ up to a 2^{nd}-order Taylor series expansion. Given such comments, it follows that the actual process model used in the prediction step matches with (2.5).
After a prediction $\hat{\boldsymbol{\delta}\mathbf{x}}_{k|k-1}$ of the error-state vector is formulated, all ingredients are set-up to proceed with the **update step**. Given the collected input raw pseudorange and pseudorange-rate measurements from the set of tracked satellites, after compensating for the most recent estimates of the receiver clock-bias and clock-drift errors, the innovations are constructed. For each satellite j, the innovation $\mathbf{s}_j$ is computed as the difference between the raw pseudorange and Doppler observables and the nominal measurements obtained from the lately inertial updates to the body trajectory. Additionally, the linear observation model, in terms of the observation matrix $\mathbf{H}_k$, is constructed as in (2.7).
After that, the Kalman Gain matrix $\mathbf{K}_k$ is retrieved according to (3.26).
Eventually, the a-posteriori updated error-state vector estimate $\hat{\boldsymbol{\delta}\mathbf{x}}_{k|k}$ and its relative error-state covariance matrix $\mathbf{P}_{err,k|k}$ are evaluated using (3.27) and (3.28), respectively.

3.3 Unscented Kalman Filter

As thoroughly discussed in Section 3.1.1, the fundamental trademark of Kalman filters is the propagation of the state, seen as a Gaussian-distributed random variable, through the system dynamic process and observation models. In the EKF algorithm, the posterior state-density is characterized according to a multidimensional Gaussian model and all relevant noise densities are assumed to be additive, zero-mean and normally distributed. The unknown system state, then, is propagated though a first-order linearization of both process and measurement models [16]. However, for a wide variety of real applications, the system state is strongly non-linear and, even more, its distribution might be other than Gaussian. If this happens, or in case of inaccurate filter initialization, the EKF estimate may diverge and the accuracy performance of the whole integrated unit deteriorates.
Given so, as part of the contribution to this thesis work, more advanced and possibly robust Bayesian architectures are investigated.
By standing within the class of Kalman estimators, an evolution of the EKF is represented by the UKF [32]. As any other KF-based algorithm, the UKF adopts a multidimensional Gaussian model for the a-posteriori state distribution. As such, to fully characterize it from a statistical point of view, the knowledge of the mean-vector and the covariance matrix are enough. These parameters become the output of the Bayesian estimation routine over a non-linear state-space model. Thus, up to this point, no big difference exists w.r.t. the EKF. However, the UKF does not estimate the a-posteriori statistics of the state by first linearizing the discrete state-space model, but it makes use of the mathematical concept of the Unscented Transform (UT) [32].
The fundamental idea behind the UT is that it is much easier to approximate a Gaussian distribution rather than a non-linear model [35]. The UT-process involves the generation of a finite set of deterministically and properly selected points, named *sigma-points*, from the domain of the distribution to be estimated. These sample-points are artificially selected in such a way that they can completely represent the mean and covariance of the distribution they are drawn from. Then, these points are propagated through the non-linear models. Finally, the set of transformed points are exploited to construct an estimate of the mean and covariance parameters of the target (a-posteriori) state-distribution [35].
A simplified and intuitive 2-D representation of the UT-process involving sigma-points for state-estimation is provided in Figure 3.1.

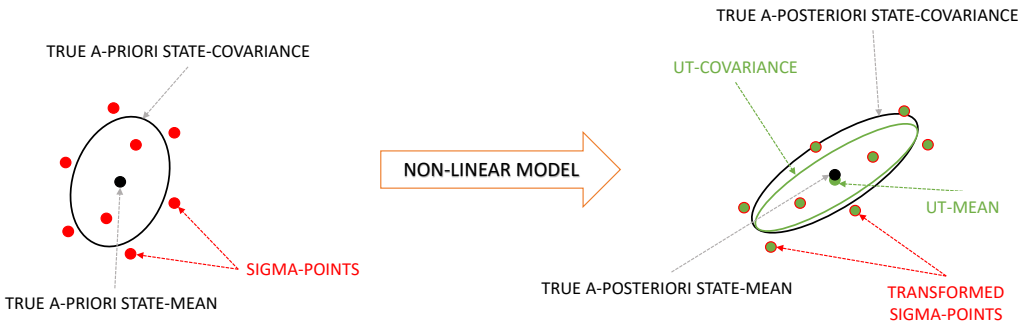


Figure 3.1: 2-D representation of the UT-concept and sigma-points over a non-linear system model.

3.3.1 General UKF algorithm

The discrete state-space formulation of the non-linear dynamic system model is the same as that formally expressed in (3.19) and (3.20) for the EKF-algorithm (Section 3.2.1). The state-transition function f_{k-1} and the observation function h_k are considered known and possibly non-linear, while the system process noise $\mathbf{v}_{k-1}$ and the measurement noise $\mathbf{w}_k$ are still additive, zero-mean and Gaussian-distributed. Moreover, identically to the EKF-case, all involved state-distributions are Gaussian and the knowledge of an initial distribution $p(\mathbf{x}_0)$ is assumed.
In light of the above, looking at the k-th estimation epoch, the general UKF algorithm starts by considering the last estimated a-posteriori state-distribution $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$, characterized by the posterior estimated mean-vector $\hat{\mathbf{x}}_{k-1|k-1}$ and its corresponding covariance matrix $\mathbf{P}_{k-1|k-1}$. Given these parameters, the UT is applied and a set of $2n+1$ sample-points, n being the state-vector dimension, is deterministically generated according to the following relations [32, 35]:

$$\begin{aligned}
\boldsymbol{\chi}_{k-1}^{0} &= \hat{\mathbf{x}}_{k-1|k-1} \\
\boldsymbol{\chi}_{k-1}^{i} &= \hat{\mathbf{x}}_{k-1|k-1} + \sqrt{(n+\kappa)\left[\mathbf{P}_{k-1|k-1}\right]_i} \quad , \quad i = 1, ..., n \\
\boldsymbol{\chi}_{k-1}^{i+n} &= \hat{\mathbf{x}}_{k-1|k-1} - \sqrt{(n+\kappa)\left[\mathbf{P}_{k-1|k-1}\right]_i} \quad , \quad i = 1, ..., n
\end{aligned} \tag{3.29}$$

where $\left[\mathbf{P}_{k-1|k-1}\right]_i$ identifies the i-th row of $\mathbf{P}_{k-1|k-1}$, while κ is a tuning factor typically chosen as $\kappa = n(\kappa_1^2 - 1)$, κ_1 being a positive scaling scalar that can be made arbitrarily low. Specifically, κ_1 determines the spread of the sigma points

around $\hat{\mathbf{x}}_{k-1|k-1}$ and κ_2 is a secondary parameter usually set to 0 [32].
It is interesting to point out here that, for the construction of the sigma-points, the term $\sqrt{(n+\kappa)\left[\mathbf{P}_{k-1|k-1}\right]}$ can be obtained by applying a Cholesky factorization to $(n+\kappa)\,\mathbf{P}_{k-1|k-1}$ [16].
The set of sigma-points is then assigned a corresponding set of weights [32]:

$$\begin{aligned} w_{k-1}^{0} &= \frac{\kappa}{n+\kappa} \\ w_{k-1}^{i} &= \frac{\kappa}{2\cdot(n+\kappa)} \quad , \quad i=1,...,n \\ w_{k-1}^{i+n} &= \frac{\kappa}{2\cdot(n+\kappa)} \quad , \quad i=1,...,n \end{aligned} \tag{3.30}$$

satisfying the normalization condition $\sum_{i=0}^{2n} w_{k-1}^{i} = 1$.
At this point, the **prediction step** can be accomplished. The set of deterministically selected samples is directly propagated through the non-linear state-transition function to obtain the transformed points [32]:

$$\boldsymbol{\chi}_{k|k-1}^{i} = f_{k-1}\left(\boldsymbol{\chi}_{k-1}^{i}\right) \tag{3.31}$$

The set of transformed points $\{\boldsymbol{\chi}_{k|k-1}^{i}\}_{i=1}^{2n+1}$, along with the weights' set $\{w_{k-1}^{i}\}_{i=1}^{2n+1}$, is then used to construct the mean-vector $\hat{\mathbf{x}}_{k|k-1}$ and the covariance matrix $\mathbf{P}_{k|k-1}$ estimates of the a-priori state distribution, where $\hat{\mathbf{x}}_{k|k-1}$ coincides with the a-priori prediction to the state-vector variable. Formally [32]:

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{i=1}^{2n+1} w_{k-1}^{i}\boldsymbol{\chi}_{k|k-1}^{i} \tag{3.32}$$

$$\mathbf{P}_{k|k-1} = \mathbf{Q}_{k} + \sum_{i=1}^{2n+1} w_{k-1}^{i}\left(\boldsymbol{\chi}_{k|k-1}^{i} - \hat{\mathbf{x}}_{k|k-1}\right)\left(\boldsymbol{\chi}_{k|k-1}^{i} - \hat{\mathbf{x}}_{k|k-1}\right)^{T} \tag{3.33}$$

Moving to the **update step**, the standard Kalman filter equations are used to refine and update the state-prediction. Specifically, given the input observation sequence $\mathbf{z}_k$, the innovation sequence $\mathbf{s}_k$ can be retrieved [32]:

$$\mathbf{s}_k = \mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1} \tag{3.34}$$

where the predicted measurements, collected in $\hat{\mathbf{z}}_{k|k-1}$, are obtained by propagating the transformed set of sample-points $\{\boldsymbol{\chi}_{k|k-1}^{i}\}_{i=1}^{2n+1}$ through the possibly non-linear

observation function h_k in the following way [32]:

$$\hat{\mathbf{z}}_{k|k-1} = \sum_{i=1}^{2n+1} w_{k-1}^i h_k \left(\boldsymbol{\chi}_{k|k-1}^i \right)$$

Defining the measurements auto-covariance matrix $\mathbf{P_{zz}}$ and the state-measurement cross-covariance matrix $\mathbf{P_{xz}}$:

$$\mathbf{P_{zz}} = \sum_{i=1}^{2n+1} w_{k-1}^i \left(h_k \left(\boldsymbol{\chi}_{k|k-1}^i \right) - \hat{\mathbf{z}}_{k|k-1} \right) \left(h_k \left(\boldsymbol{\chi}_{k|k-1}^i \right) - \hat{\mathbf{z}}_{k|k-1} \right)^T$$

$$\mathbf{P_{xz}} = \sum_{i=1}^{2n+1} w_{k-1}^i \left(\boldsymbol{\chi}_{k|k-1}^i - \hat{\mathbf{x}}_{k|k-1} \right) \left(h_k \left(\boldsymbol{\chi}_{k|k-1}^i \right) - \hat{\mathbf{z}}_{k|k-1} \right)^T$$

all ingredients are made available for the computation of the Kalman Gain matrix $\mathbf{K}_k$ [32]:

$$\mathbf{K}_k = \mathbf{P_{xz}} \mathbf{S}_k^{-1} \tag{3.35}$$

with $\mathbf{S}_k = \mathbf{P_{zz}} + \mathbf{R}_k$.
Eventually, the a-posteriori statistics are constructed [32]:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \left(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1} \right) \tag{3.36}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^T \tag{3.37}$$

The general UKF-routine is summarized in Algorithm 2.

Algorithm 2 Summary of UKF algorithm

Assumptions: Non-linear state-transition function f_{k-1} and observation function h_k + zero-mean Gaussian distributed process and observation noises
Require: knowledge of the initial state $\mathbf{x}_0$ and state-covariance $\mathbf{P}_0$

1: **for** $k = 1 : \inf$ **do**

Prediction step

2: Given n the state-vector dimension, generate a set $\{\boldsymbol{\chi}_{k-1}^i\}_{i=1}^{2n+1}$ of deterministic sigma-points

$$\begin{aligned}
\boldsymbol{\chi}_{k-1}^0 &= \hat{\mathbf{x}}_{k-1|k-1} \\
\boldsymbol{\chi}_{k-1}^i &= \hat{\mathbf{x}}_{k-1|k-1} + \sqrt{(n+\kappa)\left[\mathbf{P}_{k-1|k-1}\right]_i} \quad , \quad i = 1, ..., n \\
\boldsymbol{\chi}_{k-1}^{i+n} &= \hat{\mathbf{x}}_{k-1|k-1} - \sqrt{(n+\kappa)\left[\mathbf{P}_{k-1|k-1}\right]_i} \quad , \quad i = 1, ..., n
\end{aligned}$$

with its corresponding set of scalar normalized weights $\{w_{k-1}^i\}_{i=1}^{2n+1}$:

$$\begin{aligned}
w_{k-1}^0 &= \frac{\kappa}{n+\kappa} \\
w_{k-1}^i &= \frac{\kappa}{2 \cdot (n+\kappa)} \quad , \quad i = 1, ..., n \\
w_{k-1}^{i+n} &= \frac{\kappa}{2 \cdot (n+\kappa)} \quad , \quad i = 1, ..., n
\end{aligned}$$

3: Propagate $\{\boldsymbol{\chi}_{k-1}^i\}_{i=1}^{2n+1}$ directly through the possibly non-linear state-transition function f_{k-1}

$$\boldsymbol{\chi}_{k|k-1}^i = f_{k-1}\left(\boldsymbol{\chi}_{k-1}^i\right)$$

obtaining the transformed set $\{\boldsymbol{\chi}_{k|k-1}^i\}_{i=1}^{2n+1}$.

4: Estimate the a-priori state-vector $\hat{\mathbf{x}}_{k|k-1}$ and its covariance:

$$\begin{aligned}
\hat{\mathbf{x}}_{k|k-1} &= \sum_{i=1}^{2n+1} w_{k-1}^i \boldsymbol{\chi}_{k|k-1}^i \\
\mathbf{P}_{k|k-1} &= \mathbf{Q}_k + \sum_{i=1}^{2n+1} w_{k-1}^i \left(\boldsymbol{\chi}_{k|k-1}^i - \hat{\mathbf{x}}_{k|k-1}\right)\left(\boldsymbol{\chi}_{k|k-1}^i - \hat{\mathbf{x}}_{k|k-1}\right)^T
\end{aligned}$$

Update step

5: Estimate the innovation sequence $\mathbf{s}_k$ as the residual between the observed measurement vector $\mathbf{z}_k$ and the predicted observation sequence from the measurement model:

$$\mathbf{s}_k = \mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}$$

where $\hat{\mathbf{z}}_{k|k-1} = \sum_{i=1}^{2n+1} w_{k-1}^{i} h_k\left(\boldsymbol{\chi}_{k|k-1}^{i}\right)$ is obtained from propagation of transformed sigma-points through the non-linear observation function h_k.

6: Estimate the Kalman Gain matrix $\mathbf{K}_k$ as:

$$\mathbf{K}_k = \mathbf{P}_{\mathbf{xz}}\mathbf{S}_k^{-1}$$

with $\mathbf{S}_k = \mathbf{P}_{\mathbf{zz}} + \mathbf{R}_k$ is the residual sequence covariance and

$$\mathbf{P}_{\mathbf{zz}} = \sum_{i=1}^{2n+1} w_{k-1}^{i}\left(h_k\left(\boldsymbol{\chi}_{k|k-1}^{i}\right) - \hat{\mathbf{z}}_{k|k-1}\right)\left(h_k\left(\boldsymbol{\chi}_{k|k-1}^{i}\right) - \hat{\mathbf{z}}_{k|k-1}\right)^T$$
$$\mathbf{P}_{\mathbf{xz}} = \sum_{i=1}^{2n+1} w_{k-1}^{i}\left(\boldsymbol{\chi}_{k|k-1}^{i} - \hat{\mathbf{x}}_{k|k-1}\right)\left(h_k\left(\boldsymbol{\chi}_{k|k-1}^{i}\right) - \hat{\mathbf{z}}_{k|k-1}\right)^T$$

7: Estimate the a-posteriori state-vector $\hat{\mathbf{x}}_{k|k}$ and its covariance:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\left(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}\right)$$
$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T$$

8: **end for**

3.3.2 UKF-based INS/GNSS integration filter

Similarly to the approach followed in Section 3.2.2, the application of the UKF estimator in the INS/GNSS TC-integrated system context is explored.
Regarding the definition and characterization of the discrete-time system state-space model, the statistical description of the density distributions for both process and measurement noises, and the mathematical formalization for the system state-vector in terms of error-states, no significant differences are highlighted with respect to the EKF-based solution (the reader is invited to refer to Section 3.2.2 to investigate further on these aspects).
Moving the attention on the development of the fusion algorithm according to the UKF filtering scheme, the flow of operations presented in Section 3.3.1 is maintained to a large extent, but slight modifications are introduced anyway and are discussed here.
First of all, it is important to recall that, by relying on an indirect state-space representation, the a-posteriori error-state vector at previous epoch is considered to have all-zero error-components. This choice is motivated by the fact that, between consecutive integration instants, the INS-unit, by solving the inertial system mechanization, continuously updates the overall navigation states of the system, thus

allowing to track with higher resolution the evolution in the motion dynamics of the target body. Moreover, the medium-rate module updates the state-transition model too, and the prediction of the error components related to the inertial sensor biases can be corrected. As such, the error-state estimate from the navigation filter at a previous GNSS-epoch becomes completely obsolete and the integration algorithm cannot rely on it as an initialization.

As far as the **prediction step** is concerned, a first set of $2n$ (n being the number of error-states collected in the error-state vector, that is 17) sample points are deterministically generated according to (3.29). Referring back to Section 3.3.1, the tuning factor κ is set to a null value and all sigma-points are assigned the same weight $w_{k-1}^{i} = \frac{1}{n}$.
After that, sigma-points are transformed by applying them to the state-transition function. The latter, similarly to the EKF-framework, agrees to the simplified linear process model (2.3) and, after discretization, it coincides with matrix $\mathbf{F}_k$. Thus, in the context of INS/GNSS integrated system, the transformation of the set of sigma-points is a linear operation involving matrix multiplication. The transformed points, then, are used to build-up an a-priori prediction $\hat{\boldsymbol{\delta}\mathbf{x}}_{k|k-1}$ of the error-state vector and of its corresponding covariance $\mathbf{P}_{err,k|k-1}$ according to (3.32) and (3.33).

For the implementation of the **update step**, the a-priori predictions on the state-variables are used to generate, in the same exact way as for the prediction step, a new set of $2n$ sigma-points. Using some terminology, the former set of sample-points can be regarded as *a-priori* samples, while the latter set can be considered *a-posteriori* sigma points. Despite these minor lexical aspects, the algorithm applies these samples to the aforementioned observation matrix $\mathbf{H}_k$. The resulting points, properly combined, define the predictions (indicated as $\hat{\mathbf{z}}_{k|k-1}$ in Section 3.3.1) to the errors on the pseudorange and Dopper-shift input measurements. The latter predictions, then, are compared to the measured observation error-vector, which is computed as the residual between the raw input GNSS-measurements and the same observables predicted through the most recent INS-solution.
Blending the measured and predicted observation-error sequences, the innovation-error sequence is obtained and, later on, the Kalman Gain matrix can be set-up using (3.35).
Eventually, after deriving the correlations $\mathbf{P_{zz}}$ and $\mathbf{P_{xz}}$ between the predicted error measurement vector and the a-priori predicted error-state vector, the updated a-posteriori error-sate estimate $\hat{\boldsymbol{\delta}\mathbf{x}}_{k|k}$ and error-covariance estimate $\mathbf{P}_{err,k|k}$ can be generated ((3.36) and (3.37)).

Before concluding, there is a subtle aspect which deserves some care. In the above description and characterization of the update-step, the linearized (and,

hence, approximated) observation model, in terms of the measurement matrix $\mathbf{H}_k$, has been used. However, one strength of the UKF, compared to the EKF, is its capability of directly handling non-linear models without requiring any preliminary linearization. Tailoring the former sentence to our context, it means that, in principle, the UKF could directly use the non-linear pseudorange model (1.8) and the non-linear Doppler model (1.26) when propagating sigma-points to retrieve the predictions to the errors on the input raw pseudorange and Doppler-shift measurements.
That said, it might seem erroneous, or at least contradictory, the use of the linearized observation model when implementing an UKF as navigation filter for an integrated INS/GNSS system. In fact, this choice has a strong motivation; for GNSS-based positioning applications (at least for those which do not involve hybridization with terrestrial range-measurements), due to the huge distances separating terrestrial users from satellites, the approximation of the non-linear pseudorange and Doppler equations through a first-order Taylor series expansion is rather optimal, with an induced accuracy penalty in the positioning solution sufficiently small (few millimetres) to be mistaken with noise.
To further support the previous assertions, a simulation has been run on a real vehicular trajectory using the integration software described in Section 2.2.4 with an UKF-based integration filter relying on both models. To assess their performance, the horizontal components (Easting and Northing in Universal Transverse Mercator (UTM)-coordinate) of the vehicle ground position along the trajectory are compared to a ground-truth. The result is shown in Figure 3.2, where the Cumulative Density Functions (CDFs) of the Root-Mean-Square Error (RMSE) for both models are considered. Evidently, the two curves are almost matching with differences of, at most, few millimetres. Moreover, Figure 3.3 zooms in for horizontal errors below 2 m, thus proving that, neither in the high-accuracy, there is an apparent gain in using directly the non-linear measurement model.

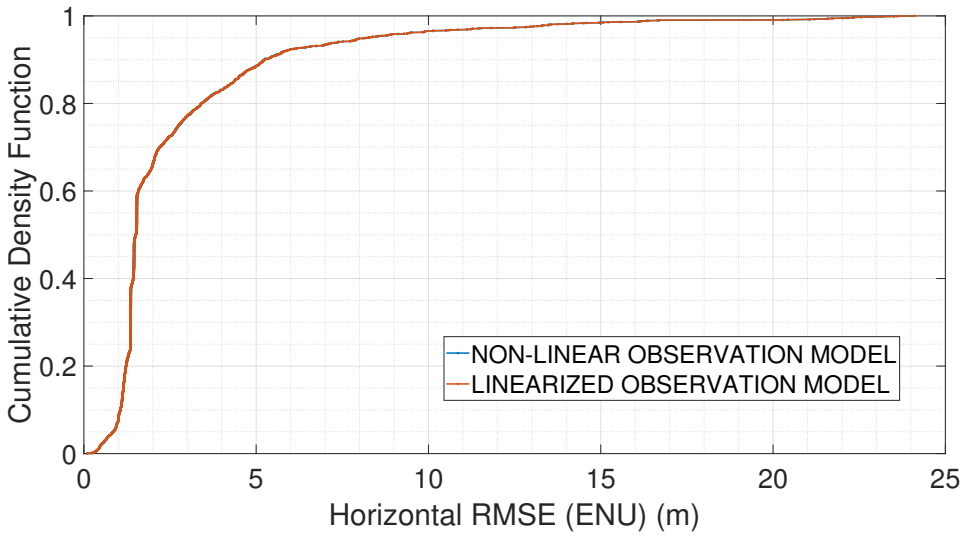


Figure 3.2: Empirical CDF of the RMSE on the horizontal component in UTM-coordinates. UKF accuracy performance when using both the linearized and the non-linear observation models.

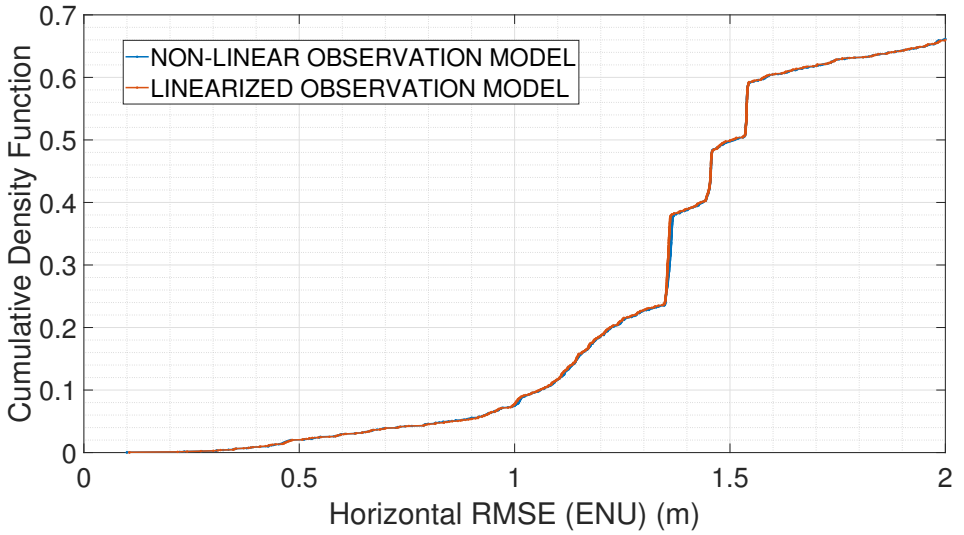


Figure 3.3: Zoom of Figure 3.2 for horizontal RMSE below 2 m.

3.4 Particle Filter

In the framework of Bayesian estimation applied to general non-linear non-Gaussian discrete state-space models, Sequential Monte-Carlo (SMC) methods identify an alternative class, considerably different from Kalman estimators, of simulation-based filtering methods. In the context of this research, the focus is oriented towards the Particle Filter (PF) [32]. PFs own the remarkable advantage of being able to represent any arbitrary statistical distribution, not necessarily in agreement with a Gaussian model, as well as the higher flexibility in handling possibly non-linear and non-Gaussian measurement models [32]. The latter feature, which highlights a breakthrough with respect to Kalman-based methodologies, turns out to be very useful in a variety of GNSS applications, since it offers the possibility of optimally tuning the statistical modelling of measurements depending on the characteristics of the external environment.

3.4.1 Monte-Carlo integration and Sequential Importance Sampling

The general purpose of SMC methods is the representation of a target density distribution through a set of randomly drawn samples, with associated weights, and the computation of estimates based on these samples and their relative weights [36]. More specifically, exactly as for KFs, the PF aims at recursively estimating the a-posteriori distribution $p\left(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}\right)$, assuming the knowledge of an initial distribution $p\left(\mathbf{x}_0, \mathbf{P}_0\right)$. Based on the HMM-assumption, at k-th estimation epoch, the posterior distribution can be fully specified in terms of its marginal distribution $p\left(\mathbf{x}_k|\mathbf{z}_{1:k}\right)$. Given so, the problem that the PF proposes to solve involves the evaluation of the expectation of a generic function $g\left(\cdot\right)$ of the state-vector $\mathbf{x}_k$, conditional upon the set of input observations up to time k, that is $\mathbf{z}_{1:k}$. Formally [32]:

$$I\left(g_k\right) = \mathbb{E}\{g_k\left(\mathbf{x}_k\right)|\mathbf{z}_{1:k}\} = \int g_k\left(\mathbf{x}_k\right) p\left(\mathbf{x}_k|\mathbf{z}_{1:k}\right)\, d\mathbf{x}_k \tag{3.38}$$

Clearly, the above integral must be evaluated over a pre-defined generic discrete state-space whose representation is formulated in (3.1) and in (3.2).
In order to approximately solve (3.38), a random set $\{\boldsymbol{\xi}_k^i\}_{i=1}^{N_s}$ of N_s identically distributed and statistically independent particles, with associated positive scalar weights $\{\psi_k^i\}_{i=1}^{N_s}$, are drawn from the domain of $p\left(\mathbf{x}_k|\mathbf{z}_{1:k}\right)$; then, the same distribution, together with the former expectation, are approximated as follows [32]:

$$\begin{aligned} \hat{p}\left(\mathbf{x}_k|\mathbf{z}_{1:k}\right) &= \sum_{i=1}^{N_s} \psi_k^i \delta\left(\mathbf{x}_k - \boldsymbol{\xi}_k^i\right) \\ \hat{I}\left(g_k\right) &= \sum_{i=1}^{N_s} \psi_k^i g_k\left(\boldsymbol{\xi}_k^i\right) \end{aligned} \tag{3.39}$$

where δ is the Dirac's Delta function. Analyzing (3.39), we capture the true meaning of Monte-Carlo (MC) integration: the original expectation integral over a p.d.f. is transformed into its discrete-approximation using a random set of weighted points ensembling the original marginal density (discrete-time sampling effect of δ) [16]. It is worth remarking that the proposed expectation defines an unbiased estimate which, for a large number of particles, should converge to the true value [33]. In practice, the latter statement opens up a big convergence issue for PFs (especially baseline versions), the curse of dimensionality. In fact, whilst the unbiased filter state-estimate approaches the true system-state with a rate which is proportional to $1/\sqrt{N_s}$ (Monte-Carlo rate), the estimation error grows exponentially with the number of dimensions n of the state-space. In other words, the rate of increase of the estimation error as a function of the system dimension is faster than the rate of convergence of the same estimate to the true value, thus making the filter infeasible for high-dimensional applications [37, 38]. By the way, in the studied framework where a TC-integrated INS/GNSS navigation unit is involved, the state-space dimension is limited to $n = 17$ and the foregoing convergence problem is rather marginal. More qualitatively speaking, provided that it is generated a sufficiently large number of particles properly fitting with the considered state-space dimension, the strength of the PF is in the ability of focusing its resources (i.e. particles) in high probability regions of the state-space so that the posterior statistics can be captured with high resolution.

Unfortunately, it is usually impossible to directly draw samples from the a-posteriori state-density $p\left(\mathbf{x}_k|\mathbf{z}_{1:k}\right)$, since it is often a complex multi-variate non-Gaussian distribution. Thus, most of times, it is necessary to resort to the Importance Sampling (IS) concept. It involves drawing the random set of samples $\{\boldsymbol{\xi}_k^i\}_{i=1}^{N_s}$ from another density function $\pi\left(\mathbf{x}_k|\mathbf{z}_{1:k}\right)$, called *importance/proposal density* function, which is expected to be proportional to the original unknown density $p\left(\mathbf{x}_k|\mathbf{z}_{1:k}\right)$ at every particle, with a different scalar weight for each sample-point [16, 32]. The validity of such approach is motivated by the chance of retrieving the particle set from a simpler and known distribution; then, by knowing a set of scalar weights (provided that they exist) and multiplying each particle for its weight, it is possible to infer the target set of particles which would be generated from the true, but unknown, a-posteriori density. To mathematically formalize the previous sentence,

if we denote as $\{\boldsymbol{\xi}_k^i\}_{i=1}^{N_s}$ the ideal set of particles which could be drawn from the posterior density $p(\mathbf{x}_k|\mathbf{z}_{1:k})$, and as $\{\boldsymbol{\chi}_k^i\}_{i=1}^{N_s}$ the set of samples from the importance density $\pi(\mathbf{x}_k|\mathbf{z}_{1:k})$, then we have [16]:

$$p\left(\mathbf{x}_k = \boldsymbol{\xi}_k^i|\mathbf{z}_{1:k}\right) = \tilde{w}_k^i \cdot \pi\left(\mathbf{x}_k = \boldsymbol{\chi}_k^i|\mathbf{z}_{1:k}\right) \tag{3.40}$$

where $\tilde{w}_k^i$ is the so called *importance weight* associated to particle i drawn from the importance distribution.
Provided that the set of importance weights exists, the approximated expectation $\hat{I}(g_k)$ formulated in (3.39) can be reframed as:

$$\hat{I}(g_k) = \sum_{i=1}^{N_s} w_k^i g_k\left(\mathbf{x}_k^i\right) \tag{3.41}$$

where, henceforth, the set of particles drawn from the importance density are labelled as $\{\mathbf{x}_k^i\}_{i=1}^{N_s}$, and w_k^i identifies the *normalized importance weight* associated to particle i. The latter is defined as [16, 32]:

$$w_k^i = \frac{\tilde{w}_k^i}{\frac{1}{N_s}\sum_{j=1}^{N_s}\tilde{w}_k^j} \tag{3.42}$$

The IS-approach presented so far hides a huge limitation. In fact, at every estimation epoch k, in order to produce a discrete-approximation $\hat{p}(\mathbf{x}_k|\mathbf{z}_{1:k})$ of the posterior density as in (3.39), and in order to retrieve an estimate of the expectation as in (3.41), it is necessary to employ all the history of input observables provided at the filter input up to k. Consequently, as the epochs accumulate, the computational burden introduced inside the algorithm grows unbounded, since weights must be always recomputed considering the update to the whole trajectory. In light of the previous comment, it is compulsory the introduction of some recursive pattern inside the IS-method, thus leading to the Sequential Importance Sampling (SIS) algorithm [32].
Assuming to know an initial state-distribution $p(\mathbf{x}_0)$ and relying on the HMM-assumption, the discrete approximation $\hat{p}(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ of the past-epoch $(k-1)$ a-posteriori density can be considered available at present epoch k.
Given so, as a new measurement sequence $\mathbf{z}_k$ is received, the algorithm should proceed with the generation of a new particles set. Factorizing the importance distribution [32]:

$$\pi(\mathbf{x}_k|\mathbf{z}_{1:k}) = \pi(\mathbf{x}_0)\prod_{j=1}^{k}\pi(\mathbf{x}_j|\mathbf{z}_{0:j-1}, \mathbf{z}_{1:j}) \tag{3.43}$$

it can be immediately understood that the new set of particles at epoch k can be retrieved by just expanding the already existing set at previous epoch $k-1$.
Moreover, the a-posteriori distribution can be factorized too [32]:

$$p\left(\mathbf{x}_k|\mathbf{z}_{1:k}\right) = p\left(\mathbf{z}_k|\mathbf{x}_k\right) \cdot p\left(\mathbf{x}_k|\mathbf{x}_{k-1}\right) \cdot p\left(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}\right) \tag{3.44}$$

where $p\left(\mathbf{z}_k|\mathbf{x}_k\right)$ corresponds to the likelihood function (refer back to (3.4) for a deeper insight) and $p\left(\mathbf{x}_k|\mathbf{x}_{k-1}\right)$ is the state-transition density (i.e. motion model (3.3)).
Reached this point, by combining (3.43) and (3.44) in (3.40), the recursive equation for the importance weights computation is obtained [32]:

$$\tilde{w}_k^i \quad \propto \quad \tilde{w}_{k-1}^i \frac{p\left(\mathbf{z}_k|\mathbf{x}_k^i\right) p\left(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i\right)}{\pi\left(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{z}_{1:k}\right)} \tag{3.45}$$

Once the computed weights are normalized, the necessary output estimates can be retrieved:

$$\hat{p}\left(\mathbf{x}_k|\mathbf{z}_{1:k}\right) = \sum_{i=1}^{N_s} w_k^i \delta\left(\mathbf{x}_k - \mathbf{x}_k^i\right) \tag{3.46}$$

$$\hat{x}_k^{MAP} = arg\max_{\mathbf{x}_k^i}\{w_k^i\} \tag{3.47}$$

$$\mathbf{P}_k = \sum_{i=1}^{N_s} w_k^i \left(\mathbf{x}_k^i - \hat{x}_k^{MAP}\right)\left(\mathbf{x}_k^i - \hat{x}_k^{MAP}\right)^T \tag{3.48}$$

The baseline SIS PF-scheme proposed and described in this section, although being applicable to any state-tracking problem, has two big flaws. First of all, it has a large sensitivity on the state initialization, meaning that the filter would diverge in case the initial state mean was far from its true value. Secondly, up to now, input external measurements have been used inside the likelihood density only, but no update step involving such observables is present [16].

3.4.2 Degeneracy phenomenon and Resampling

In previous Section 3.4.1, the general mathematical rule for computing particles' importance weights has been rigorously covered (3.45). A relevant weakness of the SIS-PF algorithm, and more in general of any SMC-based filter, reveals in the so-called *degeneracy phenomenon*, which, from a high-level point of view, gives rise

to an increase in the variance of importance weights over successive estimation epochs [16]. The evidence of such degeneracy effect is in that, after few recursions, the majority of particles tend to assume negligible normalized importance weights values (that, in the limit, tend to zero), except for one or few particles which take on high-valued weights (that, in the limit, tend to 1). Moreover, it has been proved in [39] that the variance of importance weights can only increase over time, implying that such degeneracy in unavoidable [36]. The big side-drawback which is brought about by particles degeneracy is that, within the algorithm, a large part of the computational load is put on particles that, being low-weighted, are practically not contributing to the discrete approximation of the a-posteriori density function. A convenient indicator which is typically used to quantify the degree of degeneracy inside the PF-algorithm is the *effective sample size* N_{eff}, which is defined as follows [32]:

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_s} \left(w_k^i\right)^2} \tag{3.49}$$

and the inequality $1 \leq \hat{N}_{eff} \leq N_s$ holds. Then, values of $\hat{N}_{eff}$ increasingly small and closer to the lower bound, indicate an increasingly higher degeneracy inside the algorithm.
In the view of counteracting degeneracy, a brute force approach could be that of increasing N_s, that is using more particles to approximate the posterior distribution and its statistical moments. Unfortunately, such method is useless and deplorable mainly for two reasons: first, the generation of a higher number of particles injects more complexity inside the algorithm; secondly, the approach itself is just trying to circumvent degeneracy without handling it directly, thus making it occurring after few iterations anyway [16].
In literature, there exist two major strategies which can be pursued to act against degeneracy:

- Appropriate choice of the importance density $\pi\left(\mathbf{x}_k|\mathbf{z}_{1:k}\right)$
- Resampling operation

As for the first approach, an appropriate choice for $\pi\left(\mathbf{x}_k|\mathbf{z}_{1:k}\right)$ is truly a critical issue. As a general key idea, being the degeneracy phenomenon an effect of increase in the variance of the importance weights, it is reasonable to consider that a good importance density would minimize the variance of these weights [36]. Needles to say, the optimal importance distribution matches to the target a-posteriori distribution $p\left(\mathbf{x}_k|\mathbf{z}_{1:k}\right)$, which by the way is not accessible. Consequently, it is necessary to resort to clever but sub-optimal choices. A common straightforward proposal is to employ the state transition density (3.3), thus making the filter implementation simple in general and with low complexity. Unfortunately, if no

additional operations (e.g. resampling, measurement-update etc.) are introduced in the algorithm, the estimator performance proves to be quite ineffective. The reason is that, many times, the paramount importance of the likelihood $p(\mathbf{z}_k|\mathbf{x}_k)$ (more correctly, it is a probability distribution), which indicates the state-space region where the true unknown system state is most likely to be, happens at the tails of the state-transition density and, accordingly, many particles are needed to let the filter capture the correct state after weights are assigned [16]. Moreover, a scarcity of diversity among particles onsets. This occurs because the latest available information brought by the input observations is not used to propose new values of the state.
To have a more qualitative understanding of the previous comments, the reader is invited to take a look at Figure 3.4, which considers a simplified one-dimensional state-estimation problem.

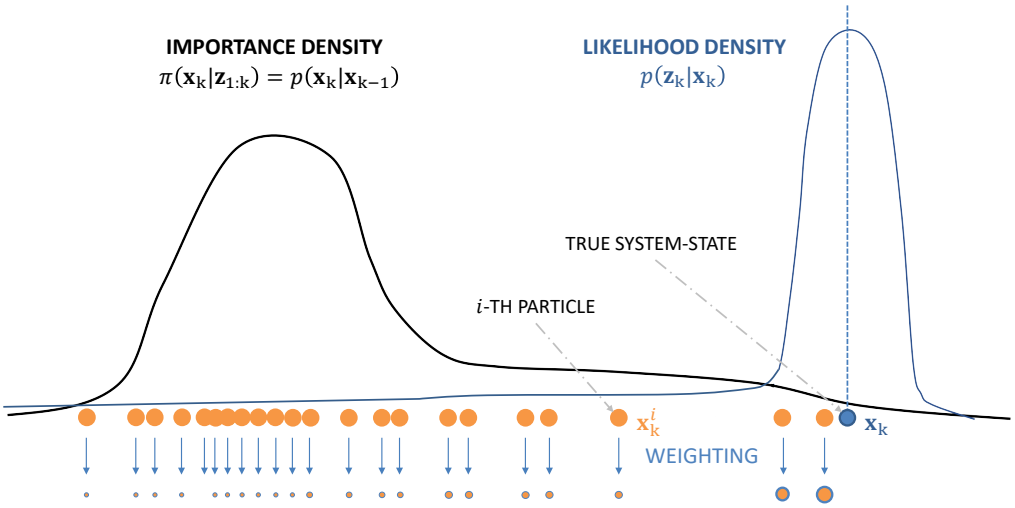


Figure 3.4: Weights distribution in SIS-PF when the state-transition density is employed as importance density for drawing particles.

In literature, alternative solutions for the selection of the importance distribution are available, promising better performance w.r.t. the state-transition density at the price of higher complexity.
In the last part of this Chapter, one of these solutions, which relies on the use of the UT, will be treated extensively and an innovative Bayesian architecture will be proposed.

The second strategy useful to mitigate the effects of degeneracy involves the use of a resampling procedure whenever a significant level of degeneracy is triggered (usually this happens when $\hat{N}_{eff} \leq N_{th}$, N_{th} being commonly set equal to $\frac{2}{5}N_s$). The basic idea of resampling is to eliminate particles that have small weights and to concentrate on particles with large weight, keeping fixed to N_s the size of the particles' set [36]. For further clearness, the rationale behind the resampling operation is summarized in the following relation [36]:

$$\mathbb{P}\left(\mathbf{x}_k^{i\star} = \mathbf{x}_k^j\right) = w_k^j \tag{3.50}$$

where $\mathbf{x}_k^{i\star}$ indicates the i-th particle in the resampled set, and $\mathbf{x}_k^j$ indicates the j-th particle in the original set. The meaning of (3.50) is the following: the probability that the i-th particle in the resampled set coincides with the j-th original particle, equals the weight of the latter particle. Therefore, the high-weight particles are going to be selected many times for resampling, while the low-weight particles are either eliminated or resampled few times.
The aforementioned approach, despite effectively counteracting the degeneracy effect, gives rise to another unwanted phenomenon: *samples impoverishment*. Indeed, after resampling, the total number of particles is the same but the actual information spread in the resampled set may come from few significantly weighted particles belonging to the initial set, leading to a loss of diversity among particles [16].
Before concluding, one aspect is still to be remarked. After resampling, a new set of N_s particles is generated and the weights are reset to $w_k^i = \frac{1}{N_s}$ for all resampled particles. It follows that, in terms of scalar weights, resampled particles are unweighted. This is justified by the fact that, after resampling, particle weights are hidden in the frequency of appearance of each particle within the resampled set: the more a particle is repeated , the higher is its weight and viceversa. To conclude, different resampling strategies are available in literature. Among them, *stratified resampling* is often found enhancing the performance of the implemented PF-based Bayesian estimation algorithm at a reasonable complexity (for a deeper insight on resampling strategies, the reader is invited to review [40]).

3.4.3 Sequential Importance Sampling Resampling (SISR) algorithm

The general Sequential Importance Sampling Resampling (SISR) PF-algorithm requires a set of baseline assumptions:

- Knowledge of an initial-state distribution $p(\mathbf{x}_0)$

- Knowledge of the set of a-posteriori particles $\{\mathbf{x}_{k-1}^i\}_{i=1}^{N_s}$ at previous estimation epoch $k-1$
- Adoption of the state-transition density as importance distribution, that is $\pi\left(\mathbf{x}_k|\mathbf{z}_{1:k}\right) = p\left(\mathbf{x}_k|\mathbf{x}_{k-1}\right)$
- Execution of resampling at every recursive step

Given these premises, the estimation process develops through five steps.
In the **initialization** step, a set of N_s particles $\{\mathbf{x}_k^i\}_{i=1}^{N_s}$ is created using the state-transition density as importance distribution. It follows that particle i is computed according to [32]:

$$\mathbf{x}_k^i = p\left(\mathbf{x}_k|\mathbf{x}_{k-1} = \mathbf{x}_{k-1}^i\right), \quad \mathbf{x}_{k-1}^i \in \{\mathbf{x}_{k-1}^i\}_{i=1}^{N_s} \tag{3.51}$$

After that, the **prediction step** is dealt with by propagating particles through the system process model, which is characterized in terms of the state-transition function f_{k-1} with its relative process noise sequence $\mathbf{v}_k$. Formally [32]:

$$\mathbf{x}_{k,pred}^i = f_{k-1}\left(\mathbf{x}_k^i\ ,\ \mathbf{v}_{k-1}\right) \tag{3.52}$$

Then, the algorithm proceeds with the **importance weights computation** by generating a set $\{\tilde{w}_k^i\}_{i=1}^{N_s}$ using the likelihood density $p\left(\mathbf{z}_k|\mathbf{x}_k\right)$. In particular, for particle i, the importance weight is evaluated as [32]:

$$\tilde{w}_k^i \propto p\left(\mathbf{z}_k|\mathbf{x}_k^i\right) \tag{3.53}$$

Furthermore, the resulting weights are also normalized according to (3.42), thus obtaining the normalized importance weights set $\{w_k^i\}_{i=1}^{N_s}$.
Examining (3.53), it can be noticed that w_k^i does not depend on w_{k-1}^i, something we would have instead expected according to (3.45). However, this result is not surprising because, according to the SISR-PF algorithm, resampling is executed at each recursion and, as a consequence, importance weights are always reset.
The following step in the PF-routine involves **resampling**, which recombines the original set of particles according to their associated weights. In the resampled set $\{\mathbf{x}_{k,res}^i\}_{i=1}^{N_s}$, then, particles are unweighted, meaning that they are assigned all-equal weights $w_{k,res}^i = \frac{1}{N_s}$.
In the end, using resampled particles, **Bayesian inference** is achieved by generating an a-posteriori refined estimate of the system state-vector variable $\hat{\mathbf{x}}_k$ and of its

covariance matrix $\mathbf{P}_k$ [32]:

$$\hat{\mathbf{x}}_k = \sum_{i=1}^{N_s} \mathbf{x}_{k,res}^i \tag{3.54}$$

$$\mathbf{P}_k = \sum_{i=1}^{N_s} \left(\mathbf{x}_{k,res}^i - \hat{x}_k\right) \left(\mathbf{x}_{k,res}^i - \hat{x}_k\right)^T \tag{3.55}$$

3.5 Unscented Particle Filter

Scrolling through former sections, it emerges a fundamental difference between Kalman-based solutions (EKF,UKF) and SMC-based algorithms (PF): while in the former class of Bayesian estimators, even in the more advanced Unscented version, both state and measurement noises are always treated as zero-mean, additive Gaussian processes, in PF architectures no assumptions are put neither on the linearity of the system-state models nor on noise statistics. Consequently, it is possible to assert that any PF, regardless of the specific implementation details, delivers a non-parametric estimation over a pre-defined discrete state-space model [30].
Undoubtedly, the hallmarks of a standard PF can be summarized in the following two points:

- **Monte-Carlo integration**, allowing to get a Delta Dirac's based discrete approximation of the target system a-posteriori state distribution.
- **Importance Sampling assumption**, that makes possible the generation of particles from a simpler and known importance distribution.

These two conditions must be tightly met to guarantee an acceptable estimation performance.
Focusing the attention on the importance distribution, it has been already pointed out, in Section 3.4.2, that a proper choice of such density is critical, because it has a direct impact on the variance of the generated particles and, accordingly, on the frequency of occurrence of the degeneration phenomenon. Assuming to use the state-transition density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ as importance distribution and observing Figure 3.4, it may happens that the two curves poorly overlap, implying that particles are not intrinsically generated in high-likelihood regions. In literature, different solutions to mitigate this problem are proposed, such as prior editing [41] and rejection methods [42], which are not discussed here since they are out of the scope of this thesis.
An interesting alternative, in light of the above, is represented by the UPF.

The true reason which has driven the choice of implementing the UPF-architecture takes its roots from the possibility of improving the design of a convenient importance distribution based on some goodness criterion. As regards the latter, it was stated in Section 3.4.2 that the optimal choice for the importance density would coincide with the state-posterior density which, besides being the object of the MC-approximation, is often unknown. Hence, in such context, it may seem appealing the idea of exploiting a Bayesian estimator to produce an acceptably accurate approximation of the true a-posteriori state density to be used as importance density in the PF afterwards.

Among Kalman-based solutions, there is a consistent gap between the EKF and UKF in terms of accuracy on the estimate of the state a-posteriori statistics (in the Gaussian assumption, for statistics it is enough to consider the state a-posteriori mean and covariance). Such difference is highlighted in figure 3.5.

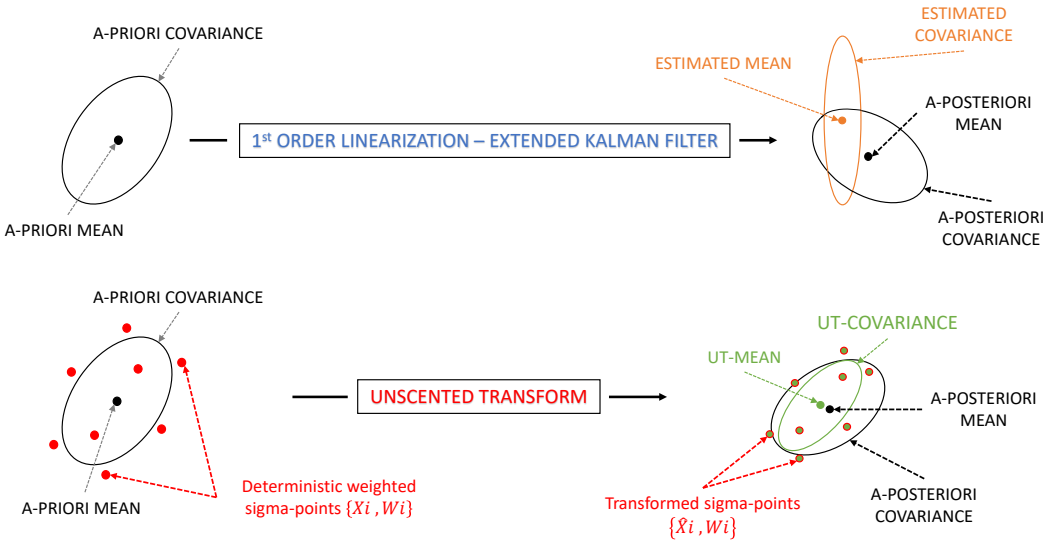


Figure 3.5: Difference between the effects of linearization (EKF) and unscented transformation (UKF) on the mean and covariance statistics in a simplified two-dimensional problem.

In fact, the EKF implements a first-order approximation of the discrete system state-space model and the estimates on the state statistics result being biased, as a consequence of linearization. On the contrary, the UKF is able to capture the first and second moments of the a-posteriori state variable with an accuracy up to the 3^{rd}-order, thanks to the propagation of the set of weighted deterministic sigma-points directly through the process and observation models without any linearization. This makes the UKF a better candidate for accurate importance distribution generation within the PF-framework [43].

The new UPF-architecture consists of a baseline PF which exploits, as a feeding stage, an UKF to construct an importance distribution which should be as much as possible matching with the optimal state a-posteriori distribution.
At each estimation epoch, given the UKF a-posteriori mean and covariance estimates, the PF importance distribution is specified and a new set of particles is produced. After computing the importance weights associated to these particles,

the estimates to the posterior state-mean and state-covariance are refined [30].
It is important to notice that, since the importance distribution is redefined at each algorithm iteration and, correspondingly, a new particle set is created, the resampling step becomes useless in such Bayesian architecture. This is a big advantage because resampling has, on average, a remarkable impact on the computational load of the estimation routine.

3.5.1 Unscented Particle Filter algorithm

First of all, the UPF algorithm must be developed according to a pre-defined discrete state-space representation. To this end, we consider a high-level formulation such that f_{k-1}, as part of the process model, is a generic linear/non-linear state-transition function and h_k, as part of the observation model, is a generic linear/non-linear observation function. Mathematically [30]:

$$\begin{aligned} \mathbf{x}_k &= f_{k-1}\left(\mathbf{x}_{k-1}\right) + \mathbf{v}_{k-1} \\ \mathbf{z}_k &= h_k\left(\mathbf{x}_k\right) + \mathbf{w}_k \end{aligned} \tag{3.56}$$

where both the process noise sequence $\mathbf{v}_{k-1}$ and the measurement noise vector $\mathbf{w}_k$ are considered zero-mean, additive, and with covariance matrices $\mathbf{Q}_{k-1}$ and $\mathbf{R}_k$, respectively. As a matter of fact, in order for the feeding UKF to produce accurate a-posteriori state estimates, a good knowledge of the system noise statistics is strictly necessary, otherwise the filter is prone to divergence and the overall system performance hugely degrades [16].
Given the aforementioned model and assuming the knowledge of an initial state $\mathbf{x}_0$ with initial distribution $p\left(\mathbf{x}_0\right)$, thanks to the HMM-assumption, the state a-posteriori mean estimate $\hat{\mathbf{x}}_{k-1|k-1}$ and the state covariance matrix estimate $\mathbf{P}_{k-1|k-1}$ at previous epoch $k-1$ are available, and can be exploited at the current epoch k. From these parameters, the UT is applied and a set of $2n$ sigma-points, n being the state-vector dimension, is deterministically generated according to the following relations [30]:

$$\begin{aligned} \boldsymbol{\chi}^{i}_{k-1|k-1} &= \hat{\mathbf{x}}_{k-1|k-1} + \sqrt{n\left[\mathbf{P}_{k-1|k-1}\right]_i} \quad , \quad i = 1, ..., n \\ \boldsymbol{\chi}^{i+n}_{k-1|k-1} &= \hat{\mathbf{x}}_{k-1|k-1} - \sqrt{n\left[\mathbf{P}_{k-1|k-1}\right]_i} \quad , \quad i = 1, ..., n \end{aligned} \tag{3.57}$$

where $\left[\mathbf{P}_{k-1|k-1}\right]_i$ identifies the i-th row of $\mathbf{P}_{k-1|k-1}$, while the term $\sqrt{n\left[\mathbf{P}_{k-1|k-1}\right]}$ can be obtained by applying the Cholesky factorization to $n\mathbf{P}_{k-1|k-1}$ [30].
Then, the **prediction step** can be accomplished. The set of deterministically

selected samples $\{\boldsymbol{\chi}^i_{k-1|k-1}\}^{2n}_{i=1}$ is directly propagated through the possibly non-linear state-transition function f_{k-1} to obtain the transformed samples [30]:

$$\boldsymbol{\chi}^i_{k|k-1} = f_{k-1}\left(\boldsymbol{\chi}^i_{k-1|k-1}\right) \tag{3.58}$$

At this point, the set of transformed samples $\{\boldsymbol{\chi}^i_{k|k-1}\}^{2n}_{i=1}$ are used to construct the mean-vector $\hat{\mathbf{x}}_{k|k-1}$ and covariance matrix $\mathbf{P}_{k|k-1}$ of the a-priori state distribution, where $\hat{\mathbf{x}}_{k|k-1}$ coincides with the UKF a-priori prediction to the state-vector variable. Formally [30]:

$$\hat{\mathbf{x}}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n}\boldsymbol{\chi}^i_{k|k-1} \tag{3.59}$$

$$\mathbf{P}_{k|k-1} = \mathbf{Q}_{k-1} + \frac{1}{2n}\sum_{i=1}^{2n}\left(\boldsymbol{\chi}^i_{k|k-1} - \hat{\mathbf{x}}_{k|k-1}\right)\left(\boldsymbol{\chi}^i_{k|k-1} - \hat{\mathbf{x}}_{k|k-1}\right)^T \tag{3.60}$$

Moving to the **update step**, a new set $\{\boldsymbol{\chi}^i_{k|k}\}^{2n}_{i=1}$ of $2n$ sample-points are deterministically generated based on the UKF a-priori state mean and covariance estimates [30]:

$$\begin{aligned}\boldsymbol{\chi}^i_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \sqrt{n\left[\mathbf{P}_{k|k-1}\right]_i} \quad , \quad i = 1, ..., n \\ \boldsymbol{\chi}^{i+n}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} - \sqrt{n\left[\mathbf{P}_{k|k-1}\right]_i} \quad , \quad i = 1, ..., n\end{aligned} \tag{3.61}$$

After that, given the input observation sequence $\mathbf{z}_k$, the innovation sequence $\mathbf{s}_k$ can be computed as:

$$\mathbf{s}_k = \mathbf{z}_k - \hat{\mathbf{z}}_k \tag{3.62}$$

where the predicted measurements, collected in $\hat{\mathbf{z}}_k$, are obtained by propagating the latter new set of sigma-points $\{\boldsymbol{\chi}^i_{k|k}\}^{2n}_{i=1}$ through the (possibly non-linear) observation function h_k:

$$\hat{\mathbf{z}}_k = \frac{1}{2n}\sum_{i=1}^{2n} h_k\left(\boldsymbol{\chi}^i_{k|k}\right)$$

Defining, at epoch k, the measurements auto-covariance matrix $\mathbf{P}^{\mathbf{zz}}_k$ and the state-measurement cross-covariance matrix $\mathbf{P}^{\mathbf{xz}}_k$ [30]:

$$\begin{aligned}\mathbf{P}_k^{\mathbf{zz}} &= \frac{1}{2n}\sum_{i=1}^{2n}\left(h_k\left(\boldsymbol{\chi}_{k|k}^i\right)-\hat{\mathbf{z}}_k\right)\left(h_k\left(\boldsymbol{\chi}_{k|k}^i\right)-\hat{\mathbf{z}}_k\right)^T+\mathbf{R}_k \\ \mathbf{P}_k^{\mathbf{xz}} &= \frac{1}{2n}\sum_{i=1}^{2n}\left(\boldsymbol{\chi}_{k|k}^i-\hat{\mathbf{x}}_{k|k-1}\right)\left(h_k\left(\boldsymbol{\chi}_{k|k}^i\right)-\hat{\mathbf{z}}_k\right)^T\end{aligned} \tag{3.63}$$

all the necessary quantities allowing for the computation of the Kalman Gain matrix $\mathbf{K}_k$ are set-up. The latter matrix is constructed in the following manner [30]:

$$\mathbf{K}_k = \mathbf{P}_k^{\mathbf{xz}}\left(\mathbf{P}_k^{\mathbf{zz}}\right)^{-1} \tag{3.64}$$

Eventually, the feeding UKF can construct a discrete-approximation $\hat{p}\left(\mathbf{x}_k|\mathbf{z}_{1:k}\right)$ of the true posterior state density $p\left(\mathbf{x}_k|\mathbf{z}_{1:k}\right)$ using the Delta Dirac's function:

$$\hat{p}\left(\mathbf{x}_k|\mathbf{z}_{1:k}\right) = \frac{1}{2n}\sum_{i=1}^{2n}\delta\left(\mathbf{x}_k-\boldsymbol{\chi}_{k|k}^i\right) \tag{3.65}$$

Moreover, the a-posteriori state statistics, in terms of mean and covariance, are retrieved [30]:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1}+\mathbf{K}_k\left(\mathbf{z}_k-\hat{\mathbf{z}}_k\right) \tag{3.66}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1}-\mathbf{K}_k\mathbf{P}_k^{\mathbf{zz}}\mathbf{K}_k^T \tag{3.67}$$

In the UPF, the first two moments (mean-vector and covariance) of the a-posteriori state-statistics estimated by the UKF, are used to shape a Gaussian importance distribution $\mathcal{N}\left(\hat{\mathbf{x}}_{k|k},\mathbf{P}_{k|k}\right)$ for randomly generating a set $\{\mathbf{x}_k^i\}_{i=1}^{N_s}$ of N_s particles [30]:

$$\mathbf{x}_k^i \sim \mathcal{N}\left(\hat{\mathbf{x}}_{k|k},\mathbf{P}_{k|k}\right) \quad i=1,...,N_s \tag{3.68}$$

together with a set $\{w_k^i\}_{i=1}^{N_s}$ of associated importance weights such that, the i-th particle weight $\tilde{w}_k^i$, is found according to [30]:

$$\tilde{w}_k^i = \frac{p\left(\mathbf{z}_k|\mathbf{x}_k^i\right)\cdot\mathcal{N}\left(\mathbf{x}_k^i|\hat{\mathbf{x}}_{k|k-1},\mathbf{P}_{k|k-1}\right)}{\mathcal{N}\left(\mathbf{x}_k^i|\hat{\mathbf{x}}_{k|k},\mathbf{P}_{k|k}\right)} \tag{3.69}$$

In (3.69), the term $\mathcal{N}\left(\mathbf{x}_k^i|\hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}\right)$ simply indicates that a multivariate Gaussian distribution with mean-vector $\hat{\mathbf{x}}_{k|k-1}$ and covariance matrix $\mathbf{P}_{k|k-1}$ is sampled at a point of its support which is coincident with the i-th particle $\mathbf{x}_k^i$.
Furthermore, the importance weights are normalized [30]:

$$w_k^i = \frac{\tilde{w}_k^i}{\sum_{i=1}^{N_s} \tilde{w}_k^i} \quad , \quad \sum_{i=1}^{N_s} w_k^i = 1 \tag{3.70}$$

Finally, the set of particles $\{\mathbf{x}_k^i\}_{i=1}^{N_s}$ and the set of normalized importance weights $\{w_k^i\}_{i=1}^{N_s}$ are used to refine the a-posteriori mean and covariance estimates [30]:

$$\hat{\mathbf{x}}_k = \sum_{i=1}^{N_s} w_k^i \mathbf{x}_k^i \tag{3.71}$$

$$\mathbf{P}_k = \sum_{i=1}^{N_s} w_k^i \left(\mathbf{x}_k^i - \hat{\mathbf{x}}_k\right)\left(\mathbf{x}_k^i - \hat{\mathbf{x}}_k\right)^T \tag{3.72}$$

From (3.71), it is clear that the refined a-posteriori state mean estimate is calculated as a weighted average over the set of particles.
A summary of the UPF algorithm is shown in Figure 3.6.

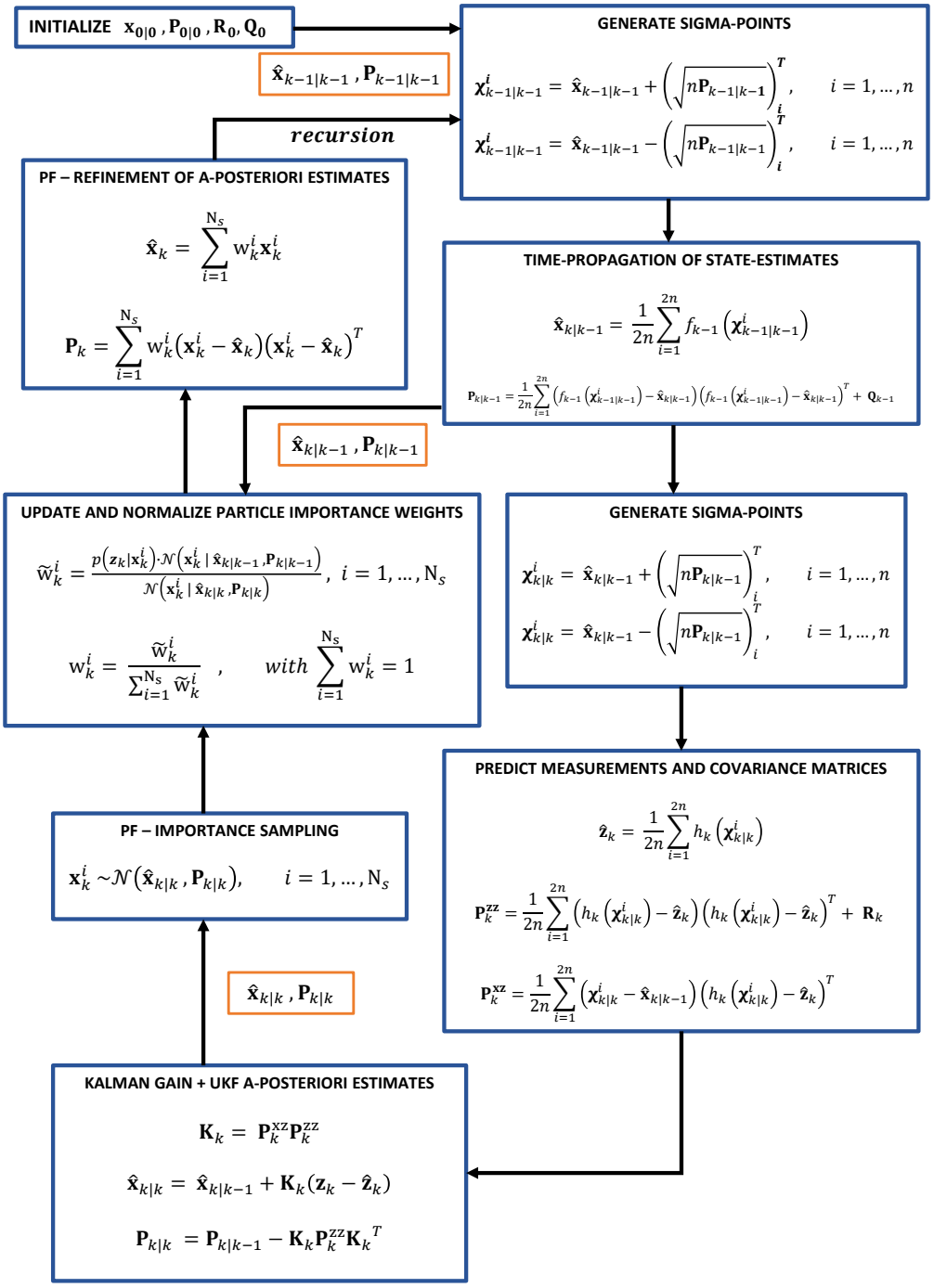


Figure 3.6: Unscented Particle Filter - Flowchart of the algorithm [30].

3.5.2 UPF: implementation and details

Having characterized the general anatomy of the UPF, it is now the due time to discuss its operation and employment within the INS/GNSS TC system context. As for the definition and characterization of the discrete-time system state-space model, both in terms of process model and in terms of measurement model, the linear approximations formalized in (2.3) and in (2.7) are adopted. In particular, referring to an error-based formulation of the system state-variable (2.1), the discretized error-state transition model, at k-th epoch, is embodied in the transitional matrix $\mathbf{F}_k$ (2.4), while the observation model is reflected on the observation matrix $\mathbf{H}_k$. Concerning the statistical characterization of process and measurement noises, the assumptions of additivity, mutual independence and zero-bias are kept the same as discussed in previous Sections 3.2.2 and 3.3.2. However, compared to the framework of Kalman-based estimation applied to INS/GNSS integrated navigation, it emerges a remarkable difference on handling the statistics of noise affecting input measurements. In fact, looking back at the class of Kalman filters, both in the EKF and in the UKF, the residual error components affecting the input raw GNSS pseudorange and Doppler-shift measurements are directly modelled within the observation noise covariance matrix $\mathbf{R}_k$ by relying on the hypothesis of Gaussianity. On the contrary, in the UPF, but more broadly in any PF-based scheme, the statistical modelling of the error on satellite-based measurements is achieved through the construction of the likelihood distribution $p\left(\mathbf{z}_k|\mathbf{x}_k\right)$ which, in turns, determines the importance weight that each particle gets assigned. In this sense, the latter statement is at the basis of the huge success of PFs for hybrid GNSS-based positioning applications.

To better grasp the outstanding relevance of the likelihood and the way it reflects on the description of the error statistics, let's consider a generic group of M visible satellites providing, at epoch k, a set $\{\rho_i\}_{i=1}^{M}$ of noisy pseudorange measurements and a set $\{\dot{\rho}_i\}_{i=1}^{M}$ of noisy Doppler measurements. Knowing, at the same epoch, the most recently updated inertial solutions to the position and velocity states of the tracked body-target, it is possible to construct nominal predictions $\{\hat{\rho}_i\}_{i=1}^{M}$ of the satellite-to-user ranges and predictions $\{\hat{\dot{\rho}}_i\}_{i=1}^{M}$ of the satellite-to-user range rates. Given so, an estimate of the error components affecting both types of measurements is formed as the residual between the observed and the predicted quantities. Hereinafter, $\{\zeta_i\}_{i=1}^{M}$ will denote the set of estimates of the pseudorange-measurement noise samples for the different satellites, while $\{\dot{\zeta}_i\}_{i=1}^{M}$ will indicate the set of estimates of the Doppler-measurement errors.
Identifying as $p_{i,\rho}$ the p.d.f. of the i-th satellite pseudorande error and as $p_{i,\dot{\rho}}$ the p.d.f. of the i-th satellite Doppler-shift error, and assuming statistical independence among the measurements, the likelihood associated to a generic particle n can be

calculated as the product of sampled p.d.f.s associated to each single measurement [44]:

$$p\left(\mathbf{z}_k|\mathbf{x}_k^n\right) = \prod_{i=1}^{M} p_{i,\rho}\left(\zeta_i\right) \cdot \prod_{i=1}^{M} p_{i,\dot{\rho}}\left(\dot{\zeta}_i\right) \tag{3.73}$$

From (3.73), the absolute flexibility and superiority of the PF in handling the statistics of noise affecting input observables is clear; in fact, no assumption is placed on the specific type of distribution characterizing each pseudorange and Doppler-shift measurement, thus allowing, at least in principle, to use different models for different types of measurements (pseudorange or Doppler) and, even inside the same class of measurements, to potentially employ different statistics for different satellites. As a matter of fact, such unconstrained variability in dealing with heterogeneous statistics is not feasible in the Kalman framework. Of course, the only non-negligible condition which must be met is the knowledge of well-suited models for the measurements. The latter comment, despite appearing straightforward, pictures out a pivotal point for all those real-time GNSS-based positioning applications where the noise statistics on measurements are time-variant and non-ergodic.

In the UPF-algorithm developed as part of this thesis work, the feeding UKF-stage, which is meant to produce an accurate estimate of the state a-posteriori distribution to be used later as importance density inside the PF, is implemented following exactly the same format as that treated in Section 3.3.2. Moreover, within the underlying PF-stage, given a set of particles drawn from the importance distribution, the likelihood for each particle is computed using (3.73); once the likelihoods for the different particles are retrieved, the corresponding importance weights are directly available since the set of particles is re-framed at each iteration and, consequently, there is no recursive pattern between weights belonging to consecutive epochs.

3.5.3 The State-Splitting concept

This final section aims at discussing on an innovative architectural paradigm involving PFs which, as a further contribution to this research, has been implemented and tested in the UPF framework.
In a legacy PF-algorithm, the set of raw GNSS input noisy measurements from each tracked satellite (pseudorange and Doppler-shift) are jointly combined as in (3.73) in order to weight the set of randomly generated particles. These samples, then, are averaged to produce an a-posteriori estimate of the state-variable. That said, there is an intrinsic fault in the way particle weights are assigned. In fact, for each particle, they are obtained by constructing a likelihood which mixes

the information coming from both pseudorange and Doppler-shift measurements. However, in the context of GNSS-based positioning, pseudorange measurements only provide information about position and receiver clock-bias, but neither on the velocity nor on the receiver clock-drift. Oppositely, Doppler-shift measurements can only supply information about velocity and clock-drift.
In view of the above, it would appear more correct and reasonable that particles, whose weights are obtained by blending pseudorange measurements, can only contribute to the determination of position and clock-bias states and, similarly, that particles, whose weights are computed from Doppler measurements, can only contribute to estimate the velocity and clock-drift states. Furthermore, in the framework of a INS/GNSS TC-integrated architecture with indirect configuration, GNSS-based measurements do not provide any information at all about the inertial components (inertial sensors errors) included inside the error-state vector.
The idea of the state-splitting method applied to the UPF-algorithm, in the framework of an integrated INS/GNSS navigation system, would then consist in the creation of a partition of the error state-vector $\boldsymbol{\delta}\mathbf{x}$ into three components:

- Component including errors on the three-dimensional body position and on the receiver clock-bias.
- Component including errors on the three-dimensional body velocity and on the receiver clock-drift.
- Component including any inertial-error state (misalignment errors and three-axial error terms associated to accelerometers and gyroscopes).

The above partition, clearly, is also propagated onto the set of particles, as they identify possible realizations of the state-vector itself.
To give some formalism, the three involved segments of the error-state vector can be indicated as follows:

$$\begin{aligned} \boldsymbol{\delta}\mathbf{x}_{\rho}^{e} &= \left[\begin{array}{cccc} \delta r_x^e & \delta r_y^e & \delta r_z^e & \delta t_u \end{array} \right] \\ \boldsymbol{\delta}\mathbf{x}_{\dot{\rho}}^{e} &= \left[\begin{array}{cccc} \delta v_x^e & \delta v_y^e & \delta v_z^e & \delta \dot{t}_u \end{array} \right] \\ \boldsymbol{\delta}\mathbf{x}_{INS}^{e,b} &= \left[\begin{array}{ccccccccccc} \delta \epsilon_x^e & \delta \epsilon_y^e & \delta \epsilon_z^e, & \delta b_{a,x}^b & \delta b_{a,y}^b & \delta b_{a,z}^b, & \delta b_{g,x}^b & \delta b_{g,y}^b & \delta b_{g,z}^b \end{array} \right] \end{aligned} \tag{3.74}$$

Obtained the above subdivision, the algorithm handles the three components differently. First of all, since GNSS measurements do not provide any useful inertial information, $\boldsymbol{\delta}\mathbf{x}_{INS}^{e,b}$ is just propagated from the UKF-output to the underlying PF-output without any further refinement. On the contrary, starting from a common set of particles $\{\boldsymbol{\delta}\mathbf{x}^i\}_{i=1}^{N_s}$, two distinct sets of weights $\{w_{\rho}^i\}_{i=1}^{N_s}$ and $\{w_{\dot{\rho}}^i\}_{i=1}^{N_s}$ are generated, the former built upon the set of pseudorange measurements and the

latter constructed from the set of Doppler-shift measurements. At this point, the algorithm splits into two separate but parallel branches.
Along one branch, the pseudorange-based partition $\{\boldsymbol{\delta}\mathbf{x}_{\rho}^{e,i}\}_{i=1}^{N_s}$ of the full set of particles is propagated and, if needed, resampled according to the group of weights $\{w_{\rho}^{i}\}_{i=1}^{N_s}$. Then, the error-state vector component $\boldsymbol{\delta}\mathbf{x}_{\rho}^{e}$ is estimated by taking a weighted average over $\{\boldsymbol{\delta}\mathbf{x}_{\rho}^{e,i}\}_{i=1}^{N_s}$. Along the other branch, instead, the Doppler-based partition $\{\boldsymbol{\delta}\mathbf{x}_{\dot{\rho}}^{e,i}\}_{i=1}^{N_s}$ of the full set of particles is propagated and, if needed, resampled according to the group of weights $\{w_{\dot{\rho}}^{i}\}_{i=1}^{N_s}$. Then, the error-state vector component $\boldsymbol{\delta}\mathbf{x}_{\dot{\rho}}^{e}$ is estimated by taking a weighted average over $\{\boldsymbol{\delta}\mathbf{x}_{\dot{\rho}}^{e,i}\}_{i=1}^{N_s}$.
Finally, the three estimated partitions are recombined to produce the refined a-posteriori error-state estimate.
The pseudo-code for the state-splitting strategy is provided in Algorithm 3.

Algorithm 3 State-Splitting Technique - PseudoCode

Assumptions: N_s particles, $N_{th} = \frac{2}{5}N_s$, M tracked satellites
Require: knowledge of pseudorange-measurement noise samples $\{\zeta_i\}_{i=1}^{M}$ and of Doppler-measurement errors $\{\dot{\zeta}_i\}_{i=1}^{M}$

for $k = 1 : \inf$ **do**

Generate N_s particles: $\boldsymbol{\delta}\mathbf{x}_k^j \sim \mathcal{N}\left(\boldsymbol{\delta}\mathbf{x}_{k|k,UKF}, \mathbf{P}_{k|k,UKF}\right)$

for $j = 1 : N_s$ **do**
Split particle j in three components: $\{\boldsymbol{\delta}\mathbf{x}_{\rho,k}^j,\ \boldsymbol{\delta}\mathbf{x}_{\dot{\rho},k}^j,\ \boldsymbol{\delta}\mathbf{x}_{INS,k}^j\}$
Get Likelihood on pseudoranges: $p\left(\mathbf{z}_{\rho,k}|\boldsymbol{\delta}\mathbf{x}_{\rho,k}^j\right) = \prod_{i=1}^{M} p_{\rho,k}\left(\zeta_i\right)$
Compute importance weight on pseudoranges: $\tilde{w}_{\rho,k}^j = p\left(\mathbf{z}_{\rho,k}|\boldsymbol{\delta}\mathbf{x}_{\rho,k}^j\right)$
Get Likelihood on Doppler: $p\left(\mathbf{z}_{\dot{\rho},k}|\boldsymbol{\delta}\mathbf{x}_{\dot{\rho},k}^j\right) = \prod_{i=1}^{M} p_{\dot{\rho},k}\left(\dot{\zeta}_i\right)$
Compute importance weight on Doppler: $\tilde{w}_{\dot{\rho},k}^j = p\left(\mathbf{z}_{\dot{\rho},k}|\boldsymbol{\delta}\mathbf{x}_{\dot{\rho},k}^j\right)$
end for

Compute normalized importance weights on pseudorange: $w_{\rho,k}^j = \frac{\tilde{w}_{\rho,k}^j}{\sum_{j=1}^{N_s}\tilde{w}_{\rho,k}^j}$
Compute normalized importance weights on Doppler: $w_{\dot{\rho},k}^j = \frac{\tilde{w}_{\dot{\rho},k}^j}{\sum_{j=1}^{N_s}\tilde{w}_{\dot{\rho},k}^j}$
Compute effective sample size on pseudoranges : $N_{eff,\rho} = \frac{1}{\sum_{j=1}^{N_s}\left(w_{\rho,k}^j\right)^2}$
Compute effective sample size on Doppler : $N_{eff,\dot{\rho}} = \frac{1}{\sum_{j=1}^{N_s}\left(w_{\dot{\rho},k}^j\right)^2}$

if $N_{eff,\rho} \leq N_{th}$ **then**
$\{\boldsymbol{\delta}\mathbf{x}_{\rho,k}^j\}_{j=1}^{N_s} = Resampling\left(\{\boldsymbol{\delta}\mathbf{x}_{\rho,k}^j\}_{j=1}^{N_s}, \{w_{\rho,k}^j\}_{j=1}^{N_s}\right)$
end if

if $N_{eff,\dot{\rho}} \leq N_{th}$ **then**
$\{\boldsymbol{\delta}\mathbf{x}_{\dot{\rho},k}^j\}_{j=1}^{N_s} = Resampling\left(\{\boldsymbol{\delta}\mathbf{x}_{\dot{\rho},k}^j\}_{j=1}^{N_s}, \{w_{\dot{\rho},k}^j\}_{j=1}^{N_s}\right)$
end if

A-posteriori estimate of pseudorange partition: $\hat{\boldsymbol{\delta}\mathbf{x}}_{\rho,k} = \sum_{j=1}^{N_s} w_{\rho,k}^j \boldsymbol{\delta}\mathbf{x}_{\rho,k}^j$
A-posteriori estimate of Doppler partition: $\hat{\boldsymbol{\delta}\mathbf{x}}_{\dot{\rho},k} = \sum_{j=1}^{N_s} w_{\dot{\rho},k}^j \boldsymbol{\delta}\mathbf{x}_{\dot{\rho},k}^j$

Reconstruction of the a-posteriori error-state estimate:

$$\hat{\boldsymbol{\delta}\mathbf{x}}_k = \{\hat{\boldsymbol{\delta}\mathbf{x}}_{\rho,k},\ \hat{\boldsymbol{\delta}\mathbf{x}}_{\dot{\rho},k},\ \boldsymbol{\delta}\mathbf{x}_{INS,k}^j\}$$

end for

Chapter 4

Adaptive measurement noise modelling and pseudorange pre-processing

4.1 Variance models for GNSS-based observations

The Bayesian filtering methods discussed in Chapter 3, considered in their application to discrete non-linear and non-Gaussian state-space models, provide reliable and highly accurate a-posteriori estimates of the state variable and of its covariance only in case the prior statistics of both process noise $\mathbf{v}_{k-1}$ and measurement noise $\mathbf{w}_k$ are well known [30]. In other words, the performance of any filtering scheme is not only strongly affected by the degree of truthfulness in the state-space formulation, which involves the modelling of both the dynamic system-state evolution and of the relationship between the input observations and the same state, but it is also strongly conditioned by the statistical characterization of state and measurement noises.
The latter sentence can be caught thinking about the concept of modelling in the context of an evolving system; by itself, a dynamic model is a summarized and simplified representation, in the flavour of a set of mathematical equations, of a real-world system (e.g. a self-navigating vehicle whose trajectory should be tracked over time), which is intended to mimic its essential features and the way such properties modify over time. For instance, the linearized Inertial Navigation System (INS) mechanization in (2.3) summarizes the time-evolution of the errors affecting the integrated system inertial states. Therefore, a model, how accurate it is, can only describe predictable features of the studied system behaviour and,

besides, it is often necessary to resort to approximations in order to trade some complexity.
In light of the above, noises become fundamental since they embed most of those unpredictable aspects that are unknown to the model, as they depend on external affections, or those system unessential features which are neglected in the model formulation. Therefore, the more the delineation of noise statistics is precise and faithful, the higher the accuracy in the estimate of the system states which, in a positioning context, embed all the variables useful to determine the trajectory of a moving target.
Based on these premises, the current chapter steers the focus on the noise modelling task in the context of a Tightly Coupled (TC) INS/Global Navigation Satellite System (GNSS) navigation unit, putting a stronger effort in the review of state-of-the-art variance estimation strategies for GNSS-based ranging and ranging-rate observables. In particular, after a short overview to highlight analogies and differences in the statistical characterization of both process and measurement noises inside the integration filter, it is treated a simple but effective parametric approach for measurements' covariance estimation. Later on, two adaptive noise-covariance modelling methods are proposed, seeking for a more flexible Bayesian paradigm able to promptly mirror the instantaneous changes in the navigated scenario. Eventually, it is put forward a self-contained, low-complexity pre-processing stage which acts by automatically reducing unpredictable multipath-related bias injections, hence reinforcing filter stability in harsh environments. Remarkably important, all the discussed methodologies operate directly on raw input GNSS measurements and, as such, are transversally fitting to any Bayesian architecture.

4.1.1 Statistical characterization of process and observation noises in INS/GNSS system

As already mentioned back in Section 3.2.2, in the framework on an INS/GNSS TC navigation system, both process and observation noises are usually considered mutually independent and additive. Nevertheless, there are substantial dissimilarities in the specific description of their statistics, which certainly depend on the available a-priori information inside the navigation filter.
As far as the process-noise sequence $\mathbf{v}_{k-1}$ (refer to (3.56)) is concerned, a typical approach is resorting to a zero-mean multi-variate Gaussian characterization. The zero-mean assumption is sensible because, at each integration epoch, all the deterministic effects biasing the inertial components of the system state variable are compensated for, both through a calibration process operated on inertial sensors and through a de-noising routine acting on raw accelerometer and gyroscope measurements. Consequently, the residual inertial error components are reasonably unbiased. As regards the state residual-noise covariance matrix $\mathbf{Q}_k$, its elements

are retrieved through inertial sensors' calibration and self-alignment processes, and are continuously updated within the medium-rate INS module (refer to Figure 2.3) [30]. Hence, $\mathbf{Q}_k$ gets directly constructed from processing raw inertial sensor measurements and there is no much refinement that can be done on it.
Turning the attention on GNSS measurements, which get engaged in the observation model (2.7), they are strongly subjected to the environmental effects encountered along the travelled path. For the bias-term in the observation error, the zero-mean assumption is still applicable considering that predictable error-sources (satellite errors, receiver errors, relativistic errors etc.), by agreeing with a certain model, can be effectively mitigated and, sometimes, even compensated for. However, as an example, accounting for the problem of estimating and tracking in real-time the trajectory of a moving vehicular target, there is plenty of impairments whose induced effects are unpredictable and cannot be handled through the observation model: multipath errors with uncontrollable bias injections, errors due to interfering signals, intentionally induced errors (e.g. jamming or spoofing), unpredictable signal propagation errors etc. [22]. Given so, it turns out that a proper statistical characterization of measurement-noise $\mathbf{w}_k$, besides necessary, is truly more challenging.
The latter issue could be worked out along two main paths: the identification of a known statistical distribution which can faithfully approximate the true error-density, and the formulation of a model for the characterization of the variance of the error.
As for the first strategy, it is commonly hard to deduce the distribution of noise affecting GNSS observables, since the related error pattern typically behaves as a time-variant, non-stationary and non-ergodic stochastic process. Moreover, by restricting the sight on code-based ranging only, multiple pseudoranges are measured from different RF-signals, broadcast by different satellites, which pursue disjoint physical paths and are impaired by heterogeneous error-sources. Therefore, it is likely that the best statistical fit, in terms of noise-distribution, changes among the various measurements; in such context, only Particle Filters (PFs) can offer enough flexibility to accommodate multiple density models.
On the other hand, the second strategy is by far more attainable and usually relies on a simpler Gaussian modelling for the Probability Density Function (p.d.f.) of the error on both pseudorange and Doppler measurements. Moreover, employing a common and unified density representation under the zero-mean assumption, the only parameter which is left out is the measurements' variance, and it could be directly featured inside the noise covariance matrix $\mathbf{R}_k$. Consequently, this technique is truly versatile and there are no limitations for its applicability in the framework of the various Bayesian architectures presented in Chapter 3.
Before going ahead, it is worth pointing out that the characterization of the variance of input observations is managed well differently between Kalman-based estimators

and PFs. Indeed, in the former class, variance modelling is operated within the measurement noise covariance matrix which, in turns, has an impact on the estimation of the Kalman gain matrix $\mathbf{K}_k$; on the contrary, the latter class keeps the noise characterization implicit in the definition of the likelihood distribution $p(\mathbf{z}_k|\mathbf{x}_k)$ which, overall, has a direct contribution in the weighting of the different particles. In light of the above, the different strategies which are going to be investigated in the following sections are all oriented to the construction of matrix $\mathbf{R}_k$. In particular, the forthcoming discussion presents the multiple approaches according to an ascending order in terms of computational load added to the integration algorithm.

4.1.2 Measurement weighting Based on Carrier-to-Noise-density ratio

The modelling and characterization, directly inside the covariance matrix $\mathbf{R}_k$, of the variances of the errors affecting GNSS-measurements, can be conceptually visualized as a weighting scheme applied to the input observations. The underlying theory supporting the latter statement reminds to the Weighted Least-Squares (WLS) strategy for an unbiased system state-estimate (Section 1.3.2), where variances become primary indicators useful to discriminate the quality of input observables. Structurally, $\mathbf{R}_k$ is typically a diagonal matrix with switching size over consecutive GNSS-epochs. The variability in size is due to the change in the number of tracked satellites over time, which depends on the instantaneous characteristics of the external environment. With regards to the diagonality property, it is a simplifying assumption that drastically neglects the existing correlations between pseudorange errors from different satellites [1]. Nonetheless, especially for code-based ranging, it is a well-established and commonly accepted hypothesis which, in GNSS-based positioning, eases the measurement-error characterization. Given so, on the main diagonal of $\mathbf{R}_k$, all the estimated variances of the input observables are placed. It follows that, when applying such matrix in the update step of a Unscented Particle Filter (UPF) routine where the measurements auto-covariance $\mathbf{P}_k^{\mathbf{zz}}$ computation (refer to (3.63)) is involved, the diagonal elements act as weights on the corresponding measurements.
The previous remarks are pivotal to drive the rationale in the formulation of a variance model. In fact, the latter must rely on some quantities that, by their values, can soundly reveal the expected quality of a given incoming pseudorange or Doppler-shift measurement. In other words, it is necessary to look for some parameters that could allow the algorithm to infer, even roughly, the potential impairments that may have impacted, more or less heavily, on the various signals measurements have been extracted from.

For high-precision PNT, satellite elevation dependent weighting represents a valuable solution, as the impact of error-sources is inversely proportional to the satellite elevation [45]. Thinking about a satellite at the Zenith (90° elevation), for example, the broadcast signal travels the shortest distance to reach a terrestrial user and it is highly probable for the GNSS-receiver tracking-loop to lock on the direct ray. Oppositely, considering a low-elevation satellite in deep urban navigation conditions, it is well likely the onset of strong multipath with potential tracking of Non-Line-of-sight (NLOS) signal reflections.
An alternative valuable indicator is represented by the Carrier-to-Noise-density ratio (C/N_0), which expresses the ratio between the received carrier power and the noise power, normalized with respect to frequency. In general, it can provide a bandwidth-independent assessment of the quality of the received signals from each of the different tracking channels.
Given that it seems to exist a higher correlation between the spread of the observation error and the C/N_0 values, at least for code-based ranging measurements, then [45] suggests a variance model for weighting GNSS observables based on such parameter. Considering, by now, only pseudorange measurements, the model, in terms of the pseudorange-noise covariance matrix $\mathbf{R}_{\rho,k}$, can be formalized as follows [45]:

$$\mathbf{R}_{\rho,k} = \begin{bmatrix} \sigma^2_{\rho_1} & 0 & 0 & \dots & 0 \\ 0 & \sigma^2_{\rho_2} & 0 & \dots & 0 \\ 0 & 0 & \sigma^2_{\rho_3} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \sigma^2_{\rho_{N_{sat}}} \end{bmatrix} \tag{4.1}$$

where the i-th diagonal element, representing the variance weight associated to the pseudorange measurement from the i-th tracked satellite, can be computed as [45]:

$$\sigma^2_{\rho_i} = a + b \cdot 10^{\frac{-\frac{C}{N_0}}{10}} \tag{4.2}$$

It is important to highlight that the aforementioned variance characterization is valid as long as GNSS range-measurements can be considered statistically independent, zero-mean and normally distributed. Indeed, these are common underlying assumptions at the basis of the observation models adopted in the framework of Kalman-based estimation.
As a matter of fact, (4.2) defines a parametric model with a very low complexity, since it suffices to determine suitable values of two scalar factors, a and b, in order to achieve a full statistical description of the errors on the noisy measurements. In particular, the magnitudes of these model parameters are strongly related to the

external environment and vary depending on the degree of corruption in the received ranging signals. For navigation in open-sky areas or even in mild urban scenarios, multipath effects and other major impairments are expected to be weak and to slightly degrade the quality of the received signal. Consequently, the standard deviation of the unbiased pseudorange observation should be small, and satisfactory values for the two constants are found to be: $a = 10\ m^2$ and $b = 150^2\ m^2Hz$ [45]. On the contrary, for navigation in harsh environments (e.g. canyons) within a highly-reflective context, the average signal attenuation and degradation are higher and it is necessary to use different parameter values to preserve a good fitting of the model to the standard deviation of the observation error. Good values are found to be: $a = 500\ m^2$ and $b = 10^6\ m^2Hz$ [45].
So far, only pseudorange variance modelling has been accounted for and Doppler measurements have not been examined yet.
In general, the variance model (4.2) can be extended, without major forcing, to cover the characterization of the statistics of pseudorange-rate measurements, as well. In this sense, the only requirement which is put on Doppler measurement noise involves the agreement with zero-bias, additivity and Gaussianity assumptions.
Following a similar approach, it is possible to lay out a Doppler-error covariance matrix $\mathbf{R}_{\dot{\rho},k}$ [45]:

$$\mathbf{R}_{\dot{\rho},k} = \begin{bmatrix} \sigma^2_{\dot{\rho}_1} & 0 & 0 & \dots & 0 \\ 0 & \sigma^2_{\dot{\rho}_2} & 0 & \dots & 0 \\ 0 & 0 & \sigma^2_{\dot{\rho}_3} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \sigma^2_{\dot{\rho}_{N_{sat}}} \end{bmatrix} \tag{4.3}$$

where $\sigma^2_{\dot{\rho}_i} = a + b \cdot 10^{\frac{-\frac{C}{N_0}}{10}}$ is computed identically to the former model (4.2). That being stated, a change in the type of measurement which is modelled, reasonably requires a change in the parametrization. For lightly degraded environments, acceptable values for the scalars are: $a = 0.01\ m^2/s^2$ and $b = 25\ m^2Hz/s^2$ [45]. Instead, when moving to environments involving a strong signal quality deterioration, the values modify to: $a = 0.001\ m^2/s^2$ and $b = 40\ m^2Hz/s^2$ [45].
Finally, the combination of $\mathbf{R}_{\rho,k}$ and $\mathbf{R}_{\dot{\rho},k}$ leads to the construction of the usual measurement-error covariance matrix $\mathbf{R}_k$ which is involved in the observation model definition:

$$\mathbf{R}_k = \begin{bmatrix} \mathbf{R}_{\rho,k} & \mathbf{0}_{N_{sat} \times N_{sat}} \\ \mathbf{0}_{N_{sat} \times N_{sat}} & \mathbf{R}_{\dot{\rho},k} \end{bmatrix} \tag{4.4}$$

Reached this point, it has been understood that two important indicators can

be exploited to drive a proper statistical weighting for the errors affecting GNSS-measurements: satellites elevation and C/N_0 density ratio.
Thus, an appealing idea would be that of developing a slightly more sophisticated variance model that can jointly combine both metrics. The latter task is readily achieved by expanding, without altering the parametric structure, the C/N_0-based model, in order to include the information on satellite elevation. The resulting observation weighting, presented in [27], is the following:

$$\sigma^2_{\rho_i} = \frac{a + b \cdot 10^{\frac{-\frac{C}{N_0}}{10}}}{\sin(\gamma_i)} \tag{4.5}$$

where γ_i corresponds to the i-th tracked-satellite elevation angle, in degrees.

4.1.3 Residual-based Adaptive Covariance estimation

In the Bayesian framework applied to an INS/GNSS integrated positioning system, the navigation filter estimation performance is mainly concerned with the degree of accuracy in the approximation of the true travelled trajectory and of the involved sensors errors, by relying on some information about the system dynamics and the measurements. Indeed, in order for the filter to provide reliable a-posteriori estimates of the quantities involved in the system state-vector, a precise-enough knowledge of the a-priori statistics involving both process and observation noises is strictly demanded [30]. As regards the system state noise, according to what has been discussed in Section 4.1.1, its stochastic characterization can be supposed known both in terms of density distribution and in terms of variance description. By the way, focusing on observation noise, the description of its statistics is much more challenging because they may change depending on the specific type of application and, in general, being influenced from the instantaneous properties of the external environment, they do not satisfy neither the stationarity nor the ergodicity assumptions. Given so, insufficiently known prior statistics on both noises would have the effect of worsening the accuracy on the state-estimate with the injection of some bias, and would even lead to potential filter divergence [46].
In light of previous comments, integration algorithms employing static or semi-static variance models (Section 4.1.2) may end up being ineffective in estimating poorly-observable states (e.g. attitude) or may suffer from abrupt changes in the physical environment.

An adaptive filter weakly relies on a-priori statistical information and aims at enhancing the estimation performance by exploiting the learning process based on the sequence of input innovations [46]. Moreover, thanks to a continuous and dynamic adaptation of the statistical information, any modification of the external

environment directly reflects on noise and on its stochastic properties. In the following, considering a standard Extended Kalman Filter (EKF) implementing the fusion algorithm, an innovation-based adaptive scheme is proposed and, in the final part of this section, its implementation within the UPF (Section 3.5) is explored. In the Innovation-based Adaptive Estimation (IAE) approach, the adaptation is operated directly on the observation-error covariance matrix $\mathbf{R}_k$ [46]. Considering a generic estimation epoch k , the innovation sequence $\mathbf{v}_k$ is, by definition, the residual between the set of input noisy observations, collected in the measurement vector $\mathbf{z}_k$, and the set of a-priori predicted measurements, collected in the prior observation sequence $\hat{\mathbf{z}}_k$. Formally:

$$\mathbf{v}_k = \mathbf{z}_k - \hat{\mathbf{z}}_k = \mathbf{z}_k - h_k\left(\hat{\mathbf{x}}_{k|k-1}\right) \tag{4.6}$$

where h_k identifies a non-specific observation model as part of a discrete state-space formulation, while $\hat{\mathbf{x}}_{k|k-1}$ corresponds to the EKF a-priori (predicted) state estimate.
Starting from the evaluation of the innovation sequence $\mathbf{v}_k$, the statistical information stored within the observation error covariance matrix $\mathbf{R}_k$ can be adjusted as [46]:

$$\hat{\mathbf{R}}_k = \hat{\mathbf{C}}_{\mathbf{v}_k} - \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T \tag{4.7}$$

where $\mathbf{H}_k$ identifies the observation matrix coming from the EKF linearization of the observation model h_k, $\mathbf{P}_{k|k-1}$ corresponds to the estimated a-priori state covariance matrix, and $\hat{\mathbf{C}}_{\mathbf{v}_k}$ is the estimated *innovation covariance* matrix [30]. The latter is simply defined as [46]:

$$\hat{\mathbf{C}}_{\mathbf{v}_k} = \frac{1}{N} \sum_{j=j_0}^{k} \mathbf{v}_j \mathbf{v}_j^T \tag{4.8}$$

and is retrieved by averaging, over a sliding window of length N, the autocorrelations of the various innovations, from the last enclosed past epoch at $k-N+1$ up to the present epoch k.

The presented IAE method, although its mathematical validity, turns out to be quite unsuitable when selected for adaptive estimation in a INS/GNSS integrated system. In fact, focusing on (4.7), the subtraction operation may cause some numerical instability. More precisely, if the updated estimate of the measurement-noise covariance $\mathbf{R}_k$ ever contained some negative diagonal elements, then the integration filter would diverge.

The straightforward solution to this problem involves the substitution of the innovation-sequence $\mathbf{v}_k$ by the residual-sequence, leading to the formulation of a Residual-based Adaptive Estimation (RAE)-method.

In RAE scheme, the residual sequence $\mathbf{r}_k$ is defined according to:

$$\mathbf{r}_k = \mathbf{z}_k - \hat{\mathbf{z}}_k = \mathbf{z}_k - h_k\left(\hat{\mathbf{x}}_{k|k}\right) \tag{4.9}$$

In practice, $\mathbf{r}_k$ coincides with the difference-sequence between the input raw measurements, collected in $\mathbf{z}_k$, and a prediction $\hat{\mathbf{z}}_k$ of the observables based on the a-posteriori (updated) state-estimate $\left(\hat{\mathbf{x}}_{k|k}\right)$ from the EKF. Considering residuals in place of innovations, the mathematical model for the measurement-noise covariance update becomes [46]:

$$\hat{\mathbf{R}}_k = \hat{\mathbf{C}}_{\mathbf{r}_k} + \mathbf{H}_k \mathbf{P}_{k|k} \mathbf{H}_k^T \tag{4.10}$$

where the only difference with respect to the IAE approach is represented by the use of the a-posteriori estimated state-covariance $\mathbf{P}_{k|k}$ to replace the a-priori covariance $\mathbf{P}_{k|k-1}$. Moreover, in this context, $\hat{\mathbf{C}}_{\mathbf{r}_k}$ represents the residual error covariance and it is computed identically as in (4.8), by just substituting innovations with residuals.
Analyzing (4.10), it is plain to see that the use of residuals solves the instability issue inside the filter by transforming the subtraction operation into an addition, hence making mathematically impossible to have negative diagonal entries in $\hat{\mathbf{R}}_k$. However, a second minor issue arises. By looking at the covariance update in the RAE-scheme, there is a dependence on the a-posteriori state covariance $\mathbf{P}_{k|k}$. The problem is that, according to the equations involved in the update-step of the general EKF algorithm (Section 3.2.1), the derivation of $\mathbf{P}_{k|k}$ indeed requires the availability of $\mathbf{R}_k$. To overwhelm this obstacle, the idea is to use the most-recent past estimated covariance $\hat{\mathbf{R}}_{k-1}$ to approximate the true observation covariance $\mathbf{R}_k$ for retrieving $\mathbf{P}_{k|k}$. Such approximation is not restricting; $\hat{\mathbf{R}}_k$ tends to vary slowly over time since it depends on $\hat{\mathbf{C}}_{\mathbf{r}_k}$ which, in turns, comes from an average over a temporally sliding window [30].

An application of the RAE-method can be found in the UPF (Section 3.5.1), in the aim of introducing an adaptive estimation of the measurement error covariance $\mathbf{R}_k$ which is employed in the update step of the feeding Unscented Kalman Filter (UKF) stage. The resulting Bayesian architecture, whose algorithmic flowchart is presented in Figure 4.1, is referred to as an Adaptive Unscented Particle Filter (AUPF).

It is important to remark that the covariance update equation (4.10) has been derived in the EKF framework. For the UKF, a modification is required [30]:

$$\hat{\mathbf{R}}_k = \hat{\mathbf{C}}_{\mathbf{r}_k} + \mathbf{P}_k^{\mathbf{zz}+} \tag{4.11}$$

where $\mathbf{P}_k^{\mathbf{zz}+}$ is defined as [30]:

$$\mathbf{P}_k^{\mathbf{zz}+} = \frac{1}{2n} \sum_{i=1}^{2n} \left(h_k \left(\chi_{k+|k}^i \right) - \hat{\mathbf{z}}_{k+} \right) \left(h_k \left(\chi_{k+|k}^i \right) - \hat{\mathbf{z}}_{k+} \right)^T \tag{4.12}$$

and $\hat{\mathbf{z}}_{k+} = \frac{1}{2n} \sum_{i=1}^{2n} h_k \left(\chi_{k+|k}^i \right)$, with $\chi_{k+|k}^i$ being the i-th sigma-point drawn from the UKF a-posteriori estimates $\{\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}\}$ (refer to (3.66) and (3.67)).

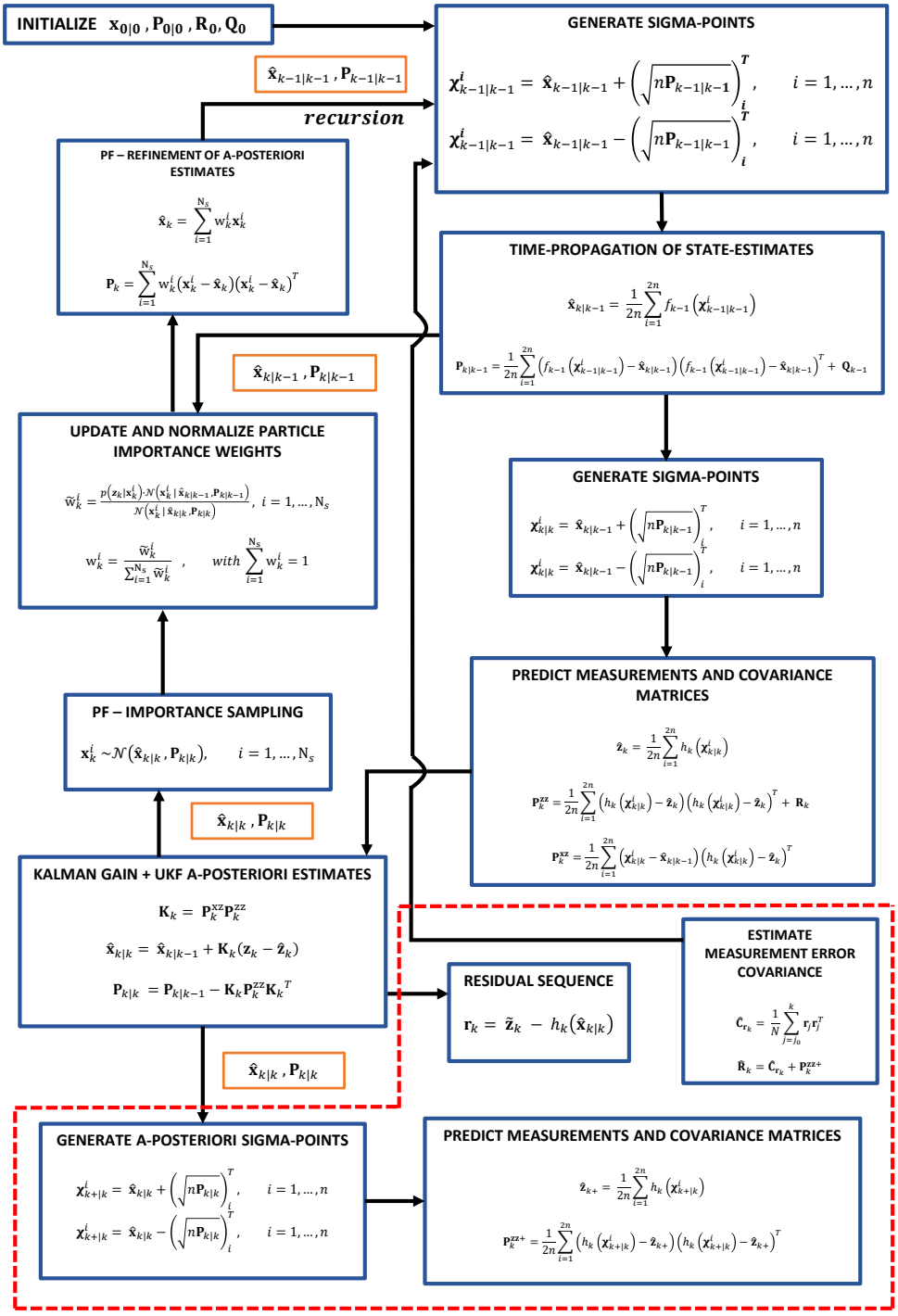


Figure 4.1: Adaptive Unscented Particle Filter (AUPF) - Flowchart of the algorithm with the adaptive part surrounded by a red-dashed line [30].

4.1.4 Adaptive Noise model based on Redundant Measurements

In the framework of INS/GNSS TC navigation systems, adaptive filtering methods which dynamically tune and update the error statistics characterizing input observables, are potentially useful for improving system performance, both in terms of state-estimation accuracy and in terms of system robustness. Indeed, an adaptive measurement-error estimation allows to finely track the evolution of the signals quality for different satellites and to enhance the effectiveness in coupling the two navigation sensors. The previous sentence well depicts the scenarios which are encountered in real navigation applications. When navigating in challenging contexts, such as downtown roads, satellites suffer from blockages and the GNSS-receiver might easily loose their tracking. When this happens, it is commonly observed, throughout the epochs preceeding the satellite loss, an average increase in the error affecting the code-based pseudorange and Doppler-shift measurements retrieved from that satellite. Moreover, this increase in the observation noise variance typically reflects onto the other visible satellites too [47]. Consequently, error-dynamics are apparently unstable as the result of the environmental instability, and on-line covariance estimation identifies one smart approach to guarantee flexibility when modelling observation noise inside a filter.
The IAE and RAE methods, presented in former Section 4.1.3, represent the most popular solutions to adaptively tune the noise covariance $\hat{\mathbf{R}}_k$. Despite promising good performances, these approaches hide a relevant pitfall. In fact, the mathematical model allowing to update the variances of GNSS-based observables depends on the state-vector and, in particular, on the estimates of its first two moments (mean and covariance). Thus, if the latter are biased, this may reflect on noises' characterization and, correspondingly, the filter estimation performance would be degraded.
To overcome the aforementioned weakness, this section presents an alternative Redundant Measurement Noise Covariance Estimation (RMNCE) scheme, which adaptively tunes observation noise statistics by relying on redundant information obtained from two independent measurement systems [48]. The main advantage of the RMNCE approach is in that the noise variance modelling is strictly based on measurements themselves and is fully independent from the system states and their related errors [48].

The idea at the basis of the RMNCE-method is the following: if there exist two redundant measurements for the same signal (the same quantity) such that they have uncorrelated zero-mean noises, the variances of such noises can be estimated based of the difference between the two measurements [48].

Specifically, let's consider $Z_1(k)$ and $Z_2(k)$ two independent redundant measurements, coming from two different systems, of the same quantity Z_k. These observables can be expressed as follows [47]:

$$\begin{aligned} Z_1(k) &= Z(k) + S_1(k) + V_1(k) \\ Z_2(k) &= Z(k) + S_2(k) + V_2(k) \end{aligned} \tag{4.13}$$

where $S_1(k)$ and $S_2(k)$ represent the unknown systems noises, while $V_1(k)$ and $V_2(k)$ identify the independent zero-bias noises affecting the measurements.
For both systems' measurements, it is possible to define [48] *First-Order Self-Differences (FOSDs)* $\Delta Z_1(k)$ and $\Delta Z_2(k)$ according to the following relations:

$$\begin{aligned} \Delta Z_1(k) &= Z_1(k) - Z_1(k-1) \\ \Delta Z_2(k) &= Z_2(k) - Z_2(k-1) \end{aligned} \tag{4.14}$$

and, by assuming that $S_i(k) - S_i(k-1) \simeq 0$, it is possible to write:

$$\begin{aligned} Z_1(k) - Z_1(k-1) &= [Z(k) - Z(k-1)] + [V_1(k) - V_1(k-1)] \\ Z_2(k) - Z_2(k-1) &= [Z(k) - Z(k-1)] + [V_2(k) - V_2(k-1)] \end{aligned}$$

Observing (4.14), it can be understood that the FOSD is nothing else than a discrete approximation of the first-order derivative of each redundant measurement with respect to epoch-time k. Hence, $\Delta Z_1(k)$ and $\Delta Z_2(k)$ incorporate information about the change of each observable between consecutive epochs.
Moreover, a *Second-Order Mutual Difference (SOMD)* $\Delta Z_{1,2}(k)$ can be defined [48]:

$$\Delta Z_{1,2}(k) = \Delta Z_1(k) - \Delta Z_2(k) \tag{4.15}$$

which only depends on the noises affecting the redundant measurements and expresses the residual between their discrete derivatives.

Given the above relations, the variances of redundant measurements from both systems can be obtained simultaneously and are given by [48]:

$$\begin{aligned} \sigma_1^2 &= \frac{\mathbb{E}\left(\Delta Z_{1,2}(k)\,\Delta Z_{1,2}^T(k)\right) + \mathbb{E}\left(\Delta Z_1(k)\,\Delta Z_1^T(k)\right) - \mathbb{E}\left(\Delta Z_2(k)\,\Delta Z_2^T(k)\right)}{4} \\ \sigma_2^2 &= \frac{\mathbb{E}\left(\Delta Z_{1,2}(k)\,\Delta Z_{1,2}^T(k)\right) - \mathbb{E}\left(\Delta Z_1(k)\,\Delta Z_1^T(k)\right) + \mathbb{E}\left(\Delta Z_2(k)\,\Delta Z_2^T(k)\right)}{4} \end{aligned} \tag{4.16}$$

where $\mathbb{E}\left(\Delta Z_{1,2}(k)\,\Delta Z_{1,2}^T(k)\right)$, $\mathbb{E}\left(\Delta Z_1(k)\,\Delta Z_1^T(k)\right)$ and $\mathbb{E}\left(\Delta Z_2(k)\,\Delta Z_2^T(k)\right)$ are the discrete auto-correlations, at k-th epoch, of $\Delta Z_{1,2}(k)$, $\Delta Z_1(k)$ and $\Delta Z_2(k)$ respectively.

From (4.16), the absolute gain of the RMNCE approach over the traditional IAE and RAE methods pops up: for both systems, the mathematical model for the estimation of observable variances depend solely on measurements and is totally independent from the system-state. Thus, any bias or inaccuracy in the estimate of the system state-variable does not propagate in the statistical characterization of the observation error. Furthermore, the variance estimate is immune to measurement-system errors and can be retrieved from FOSD and SOMD [48].
The preliminary assumption behind the RMNCE-method requires the existence of two independent systems which can guarantee redundancy of measurements, that is two systems providing simultaneously two measurements of the same target quantity having stochastically independent noises. That said, a TC-integrated INS/GNSS system seems to be perfectly tailored to such premise. In fact, on one hand, GNSS can directly supply noisy pseudorange and Doppler observables to the fusion algorithm while, on the other hand, based on the most recently updated INS estimates to the body inertial position and velocity, it is possible to indirectly retrieve estimates of the same quantities. Thus, GNSS and INS can be assigned the two aforementioned systems.
The method presented so far can already be applied inside the integration filter for a tightly-coupled architecture in order to estimate the variances of pseudorange and pseudorange-rate noises. However, few further refinements can still be achieved. First of all, considering the variability of the external environment along the travelled trajectory, it is very likely that, for the same measurements obtained from the same satellite signal, the noise distribution varies over time. Given so, it would be mindful to average the estimation of the autocorrelations involved in (4.16) over an observation window which can be made sliding along the encountered epochs. The use of a sliding window allows to track the real-time noises more accurately and to mitigate the effects of historical information [47].
Additionally, provided that a windowing scheme is adopted, it would be relevant to monitor the stability behaviour of each tracked satellite inside the same window. In this sense, a highly fluctuating satellite whose tracking is lost several times, should not contribute to the noise characterization since, with high probability, it would induce an overestimation of the measurement variance. Conversely, a stable satellite through the entire window length would represent a good candidate for the noise variance estimation. Finally, when the RMNCE-method is applied, assuming a number N_{sat} of tracked satellites at k-th epoch, the estimated covariance $\hat{\mathbf{R}}_k$ on pseudorange and pseudorange-rate measurements noise can be constructed as in (4.4).

4.2 Code-range pre-processing for multipath mitigation

Previous sections have been devoted to the analysis and characterization of multiple techniques which can be effectively used, in the context of a tightly-integrated INS/GNSS system, to implement a statistical weighting, directly inside the navigation filter, of the set of input noisy GNSS-observables. These methods, with different degrees of complexity, address on-line tracking of measurement-noise statistics' dynamic evolution in the view of achieving an adaptation of the stochastic information inside the filter such that it can promptly mirror the changes in the external environment. Anyway, however sophisticated and state-independent they are, such techniques are still rather weak to ensure filter robustness and high-accuracy positioning when navigation is pursued in challenging environments strongly affected by multipath.

Multipath propagation is very common when travelling dense urban areas with poor satellite visibility, low elevation angles and heavy RF-signals' obstructions; it involves the reception, by the GNSS receiver front-end, of several NLOS replicas of the direct signal rising from scattering phenomena caused by the same waveform impinging on buildings and trees (more in general, physical obstacles located in the environment). These reflected components, especially those arising from nearby objects and coming to the receiver with nanosecond delays w.r.t. the direct path (near echoes), are likely to induce a distortion of the correlation peak between the composite signal (Line-of-sight (LOS) part plus multipaths) and the locally generated replica within the receiver. Consequently, the true peak gets masked, thus leading to the injection of a considerable bias-error term in the measured pseudorange. On top of that, when the shadowing effect is stronger on the direct path, the received power of scattered contributions could even be higher than that of the LOS signal, hence misleading the tracking loop which locks on multipaths. Furthermore, the interfering multipath contributions have different phases and, after they combine at the receiver front-end, the phase of the composite signal (inherently related to its delay) reflects on the correlator whose code-based range measurement might be higher or smaller with respect to the true range [49]. It follows that, considering the problem of estimating and tracking the trajectory of a body which is continuously in motion, multipath certainly represents the harshest error-source for GNSS-based positioning, and its mitigation is quite difficult as a consequence of the uncontrollability of the phenomenon and the unpredictability of its effects.
As a contribution to the research carried on throughout this thesis project, a novel pseudorange pre-processing technique has been conceived in the framework of

the considered TC navigation system, seeking for an effective and low-complexity strategy allowing to counteract multipath and, correspondingly, to enhance the accuracy performance in the estimation process. In particular, this method arises from some empirical assessment done on a dataset involving real INS and Global Positioning System (GPS) data, where the integrated navigation software detailed in Section 2.2.4, running an EKF-based fusion routine, has been tested in the trajectory estimation for a moving vehicle. Moreover, for the same dataset, a highly accurate ground-truth is available. The latter has been obtained from a multi-constellation GNSS receiver, belonging to the Novatel OEM7-family, which combines a tactical-grade Inertial Measurement Unit (IMU) with Real-Time Kinematic (RTK) technique in order to supply real-time PNT at sub-centimetre accuracy level.

The basic idea is to start from the set of noisy GNSS pseudorange measurements which are supplied to the navigation filter input at each integration epoch. For sake of precision, it is remarked here that only code-based range measurements are considered, which greatly suffer from multipath effects. Referring to an error-based formulation of the hybridized architecture (Section 2.2) and assuming M tracked satellites at epoch k, the Bayesian integration filter does not directly plug the raw pseudorange measurements inside the observation model (2.7), but priorly retrieves a set of pseudorange error-samples $\{\zeta_{i,k}\}_{i=1}^{M}$ using, as references, the last INS-estimates to the position and velocity states of the moving target. To give some mathematical formalism, the following quantities are identified:

- $\rho_{i,k}^{GNSS}$, which represents the GNSS code-based pseudorange measurement from i-th satellite at epoch k.
- $\rho_{i,k}^{INS}$, which represents the INS prediction to the range between user and satellite i at epoch k.

Then, for satellite i, the measured pseudorange error $\zeta_{i,k}$ is obtained through the following difference:

$$\zeta_{i,k} = \rho_{i,k}^{GNSS} - \rho_{i,k}^{INS} \tag{4.17}$$

Moreover, thanks to the availability of a sub-centimetre accurate ground-truth position fix at every GNSS epoch, it is possible to simultaneously construct, for each visible satellite, the true user-to-satellite range. In particular, accounting for satellite i, it is calculated as:

$$R_{i,k} = \sqrt{\left(x_{i,k}^{true} - x_{i,k}\right)^2 + \left(y_{i,k}^{true} - y_{i,k}\right)^2 + \left(z_{i,k}^{true} - z_{i,k}\right)^2} \tag{4.18}$$

where $\left(x_{i,k}^{true}, y_{i,k}^{true}, z_{i,k}^{true}\right)$ represents the ground-truth 3-D position estimate and

$(x_{i,k}, y_{i,k}, z_{i,k})$ the i-th satellite position, both expressed in Earth-Centered Earth-Fixed (ECEF) coordinates at epoch k. Given so, it can be extracted a set of true pseudorange error-samples $\{\bar{\zeta}_{i,k}\}_{i=1}^{M}$ such that, for satellite i, the following relation holds:

$$\bar{\zeta}_{i,k} = R_{i,k} - \rho_{i,k}^{INS} \tag{4.19}$$

where the INS-based range is meant again as the reference observable.
Thus, to summarize, at each integration epoch and for each tracked satellite, the INS range is used as center of gravity to generate both pseudorange-error estimates $\{\zeta_{i,k}\}_{i=1}^{M}$ using raw GNSS-observables and nearly-exact error-samples $\{\bar{\zeta}_{i,k}\}_{i=1}^{M}$ exploiting ground-truth fixes.
The two groups of pseudorange error-samples, $\{\zeta_{i,k}\}_{i=1}^{M}$ and $\{\bar{\zeta}_{i,k}\}_{i=1}^{M}$, are evaluated for all the epochs involved in the simulated trajectory and their temporal trends are shown in Figure 4.2. The choice of Pseudo Random Noise (PRN) sequence 32 is totally arbitrary and it is justified by the purpose of selecting a satellite which is almost always visible, and hence tracked, throughout the whole trajectory course. Moreover, Figure 4.3 shows the same pseudorange-errors' trends when taken in absolute value.

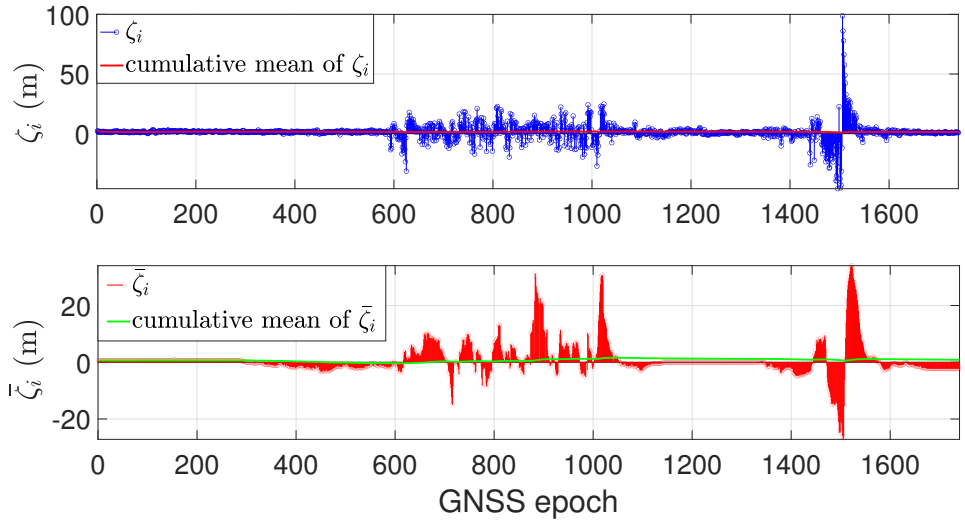


Figure 4.2: Comparison between measured pseudorange error sequence (blue-curve) and the true range-error sequence (red-curve). The results are based on raw ranging observables from GPS PRN 32.

From a qualitative perspective, a similar behaviour can be assessed between the two error sequences; this is particularly visible both in the range between epochs

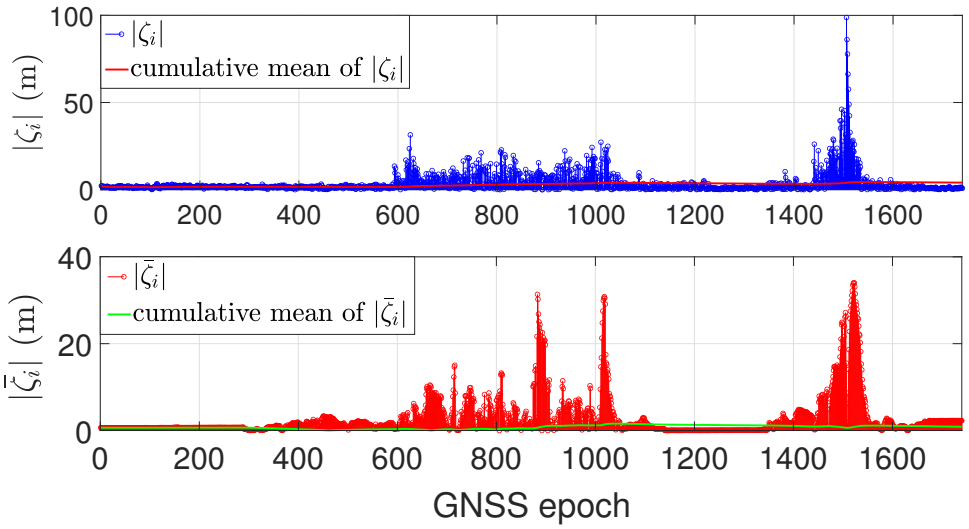


Figure 4.3: Comparison between measured pseudorange error sequence (blue-curve) and the true range-error sequence (red-curve), both considered in absolute value. The results are based on raw ranging observables from GPS PRN 32.

600 and 1000, where the time-series evidence an abrupt increment of the ranging error, and towards the end of the trajectory, after epoch 1300.

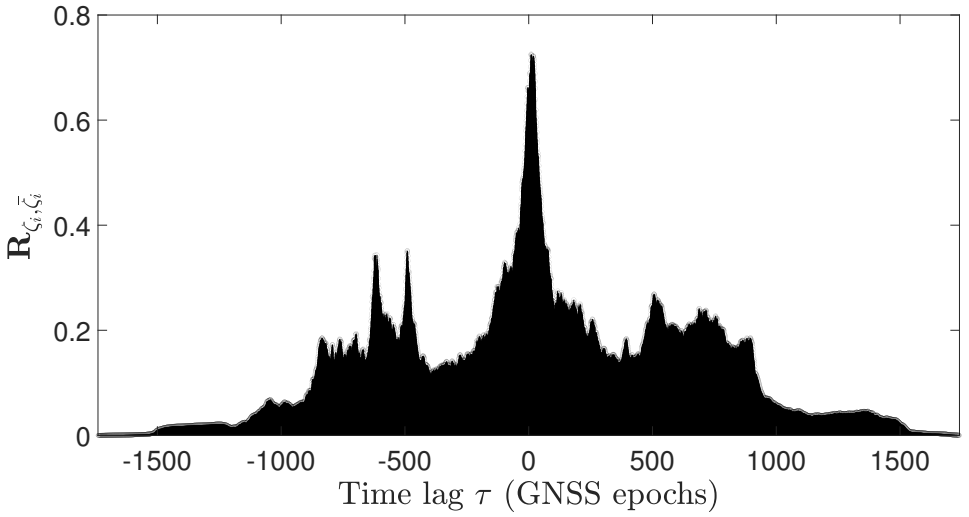


Figure 4.4: Cross-correlation between measured pseudorange error sequence and true range-error sequence obtained from ground-truth. The results are based on raw ranging observables from GPS PRN 32.

To quantitatively investigate on their degree of resemblance, the measured and true ranging-error sequences have been cross-correlated (numerically) and the outcome is shown in Figure 4.4. From analyzing the experimental values, it turns out that the main peak of the normalized correlation exceeds 0.7, which is a good indicator level for medium-to-high correlation.

The presented results set as important benchmarks useful to suggest some strategy allowing to effectively mitigate multipath, which is the major candidate phenomenon contributing to the increase in the ranging-error variance. In particular, some interesting remarks can be disclosed. First of all, the similarity in the temporal evolution between the two error sequences is a good evidence of the stability of inertial measurements. In fact, if the drifting effects of the IMU were considerably strong, then the ground-truth based pseudorange error sequence would result being uncorrelated from the other sequence since the inertial drift behaves as an accumulating bias-term over time. Actually, observing Figure 4.2, an average low inertial drift is indeed present and its effect is visible from the noise-like oscillation pattern having nearly null mean value. Overall, in the framework of the implemented TC navigation system, although relying on a low-cost MEMS-based mass-market grade IMU, the INS proves to behave stably and can be successfully used as reference for the computation of the measurement errors.
As a second point, by looking again at Figure 4.2, there are some discrete-time intervals where the swing of the error increases and the trend becomes more peaked. These effects are very likely provoked by multipath. In fact, as discussed at the beginning of the present section, multipath induces a systematic error (a bias) in the measured pseudorange. In other words, if there were no multipath and the only errors, after bias compensation, were caused by the IMU-drift plus some residual stochastic noise, then the amplitude of the pseudorange error would be small and the average null (unless in case of sensor bumping or excessive strains, it is reasonable to assume that inertial drift cannot induce considerable biases thanks to multiple robustness strategies operated on the IMU, such as sensor calibration, temperature compensation etc.). However, the effect of these spikes is exactly that of increasing the average amplitude of the error and, besides, their occurrence is absolutely irregular. To better catch the meaning of the last sentences, it may be helpful to examine Figure 4.5, where it is plotted the absolute difference between the two involved pseudorange error sequences. In those sectors of the trajectory which suffer from stronger multipath effects, the difference between the two pseudorange errors becomes higher and peaked with an increase both in the magnitude of the error and in its cumulative average.

In light of the above, the joint combination of all these comments suggests that a sensible approach to mitigate the effects induced by multipath would be attenuating (and even cutting) the peaks in the measured pseudorange error sequence, whenever

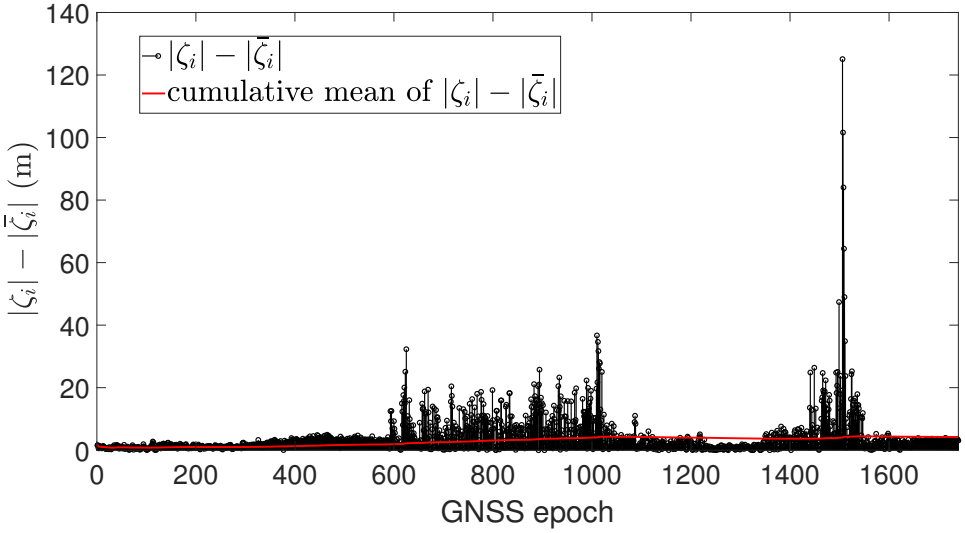


Figure 4.5: Residual sequence between GNSS-based pseudorange error and reference range error obtained using the ground-truth. The results are based on raw ranging observables from GPS PRN 32.

they occur, without affecting the low-frequency zero-mean noise-like pattern which is due to a blending of inertial bias-drift, inertial data noise plus some unknown residual contributions. To tell it differently, a profitable approach against multipath involves, on one hand, the containment and reduction of the amplitude of the pseudorange error in those sectors of the trajectory where multipath-related bias injection is stronger with several spikes induced in the measured error sequence and, on the other hand, the preservation of the low frequency oscillating pattern.

With this in mind, in the remainder of this section it is proposed a self-contained, low-complexity pre-processing strategy able to operate on-line mitigation of undesired multipath-bias injections without inhibiting the structure of the hybridization routine. To keep some mathematical and lexical formalisms, hereinafter ζ_i is going to indicate the measured (observed) pseudorange error-sequence based on raw GNSS ranges from satellite i, while $\tilde{\zeta}_i$ will refer to the smoothed error-pattern after pre-processing is applied. Specifically, $\zeta_{i,k}$ and $\tilde{\zeta}_{i,k}$ will indicate the observed and smoothed error-samples at k-th epoch, respectively.
The originally conceived idea to achieve multipath reduction involved a parametric strategy based on first-order derivative which, by its nature, brings information about the measured error-variation between consecutive GNSS-epochs. Specifically, such method was meant to trigger instantaneous jumps of the observed error-sequence ζ_i, for each tracked satellite i, possibly induced by multipath; their

mitigation, then, could be operated in an on-line fashion, that is by just relying on the difference between the newly observed error-sample $\zeta_{i,k}$, at current epoch k, and on the last measured error-sample $\zeta_{i,k-1}$, at previous epoch $k-1$. Despite cutting high-valued spikes, the smoothed sequence $\tilde{\zeta}_i$ turned out to be largely flattened in the low-error segments, thus being violated the low-frequency noise-like component characterizing the original sequence ζ_i.
Based on the former qualitative assessment, it has been elaborated an alternative approach proving to better fit with the aforementioned pre-requisites on the expected features of the smoothed error-trend. The idea is to implement a real-time multi-level scaling of the measured pseudorange-error conditional to some parameter σ_{th} tunable in real-time. In fact, the latter quantity, once properly initialized at system bootstrap and adjusted at every GNSS-epoch, allows to define multiple levels (i.e. ranges) of pseudorange error magnitudes, symmetric for positive and negative values, each of which is proportionally assigned a scaling factor. Then, at each integration epoch k, depending on the range where $\zeta_{i,k}$ is observed to fall, the scaling term associated to that level is applied to get the smoothed error-sample $\tilde{\zeta}_{i,k}$. In this way, the original measured error-pattern is not distorted and, besides, the adoption of an incremental scaling over multiple error-stripes guarantees both the abortion of spikes and the preservation of the low-frequency behaviour. Clearly, σ_{th}, as a tunable factor, establishes the impact of such pre-processing operated on the observed error-sequence. In fact, if a small value of σ_{th} is selected, then, for the same value of $\zeta_{i,k}$, the scaling is proportionally stronger (reducing σ_{th} has the effect of shifting the error-ranges towards smaller magnitudes) and peaks are severely cut-off; contrarily, if a higher value of σ_{th} is adopted, the error-scaling is softer and a weaker attenuation of the spikes is achieved. Furthermore, for those epochs where the measured error-sample is acceptably small, that is when multipath effect is evanescent and doesn't cause an increase in the pseudorange error variance, the pre-processing routine is suspended.
To offer a deeper insight on such mitigation technique, a pseudocode is supplied in Algorithm 4. Moreover, Figure 4.6 shows an example of the effect of this low-complexity pre-processing algorithm in smoothing the measured pseudorange error-sequence. Evidently, in those GNSS-epochs where multipath-related bias injection is stronger the error-peaks are attenuated but not completely zeroed, while in those segments characterized by small error-magnitude the measured and smoothed error-trends are perfectly matching because no compensation is applied (case for $p = 1$ in Algorithm 4).

The envisioned methodology bears multiple advantages. First of all, it allows to operate a real-time pre-processing of the pseudorange error ζ_i as it is measured, without involving any latency or requiring some extra buffering to keep memory of measured range-errors in previous epochs. Secondly, by operating a simple proportional scaling of the instantaneously observed error-sample $\zeta_{i,k}$, the added

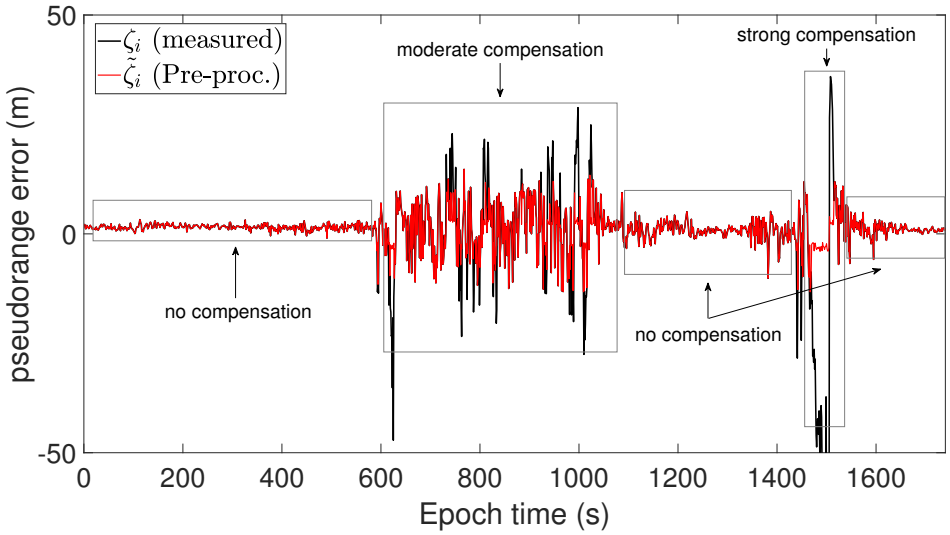


Figure 4.6: Comparison between measured and smoothed pseudorange error trends. The results are based on raw ranging observables from GPS PRN 32.

amount of computational burden inside the filtering algorithm is rather negligible. Furthermore, the developed multipath mitigation scheme is totally self-contained. Indeed, it directly operates on the measured pseudorange error at each epoch, and it does not need to rely on some aiding measurement neither of the same quantity nor of another quantity (for instance, Hatch filters require carrier-phase based pseudorange measurements or Doppler-error measurements). Last, but not least, the proposed solution is flexible because it applies transversally, as an independent module, to all the different Bayesian integrated architectures discussed in Chapter 3, without inhibiting the normal flow of the hybridization routine.

Algorithm 4 Pseudorange pre-processing (indirect formulation) for INS/GNSS TC-integrated navigation system - pseudocode

Assumptions:

- M tracked satellites at epoch k

Require:

- Knowledge of GNSS code-based pseudorange measurements $\{\rho_{i,k}^{GNSS}\}_{i=1}^{M}$ at epoch k.
- Knowledge of INS predictions to user-to-satellite ranges $\{\rho_{i,k}^{INS}\}_{i=1}^{M}$ at epoch k.
- Knowledge of the cumulative mean of pseudorange error-samples $\{\mu_{i,k-1}\}_{i=1}^{M}$ at past epoch $k-1$.
- Value selection for tunable parameter σ_{th}.

for $k = 1 : \inf$ **do**

  Compute (measured) pseudorange error-samples $\{\zeta_{i,k}\}_{i=1}^{M}$ at epoch k:

$$\zeta_{i,k} = \rho_{i,k}^{GNSS} - \rho_{i,k}^{INS}$$

  for $i = 1 : M$ **do**

    Define $f\left(\zeta_{i,k}, \mu_{i,k-1}, \sigma_{th}, p\right) = |\zeta_{i,k} - \mu_{i,k-1}| \leq 2^{p} \cdot \sigma_{th}$

    if $f\left(\zeta_{i,k}, \mu_{i,k-1}, \sigma_{th}, p = 5\right)$ **then**

$$\tilde{\zeta}_{i,k} = \frac{\zeta_{i,k}}{2^{p-1}}$$

    else if $f\left(\zeta_{i,k}, \mu_{i,k-1}, \sigma_{th}, p = 4\right)$ **then**

$$\tilde{\zeta}_{i,k} = \frac{\zeta_{i,k}}{2^{p-1}}$$

    else if $f\left(\zeta_{i,k}, \mu_{i,k-1}, \sigma_{th}, p = 3\right)$ **then**

$$\tilde{\zeta}_{i,k} = \frac{\zeta_{i,k}}{2^{p-1}}$$

11: **else if** $f\left(\zeta_{i,k}, \mu_{i,k-1}, \sigma_{th}, p = 2\right)$ **then**

12:
$$\tilde{\zeta}_{i,k} = \frac{\zeta_{i,k}}{2^{p-1}}$$

13: **else if** $f\left(\zeta_{i,k}, \mu_{i,k-1}, \sigma_{th}, p = 1\right)$ **then**

14:
$$\tilde{\zeta}_{i,k} = \frac{\zeta_{i,k}}{2^{p-1}}$$

15: **end if**

16: **end for**

17: Update cumulative mean for satellite i at epoch k:

$$\mu_{i,k} = \mu_{i,k-1} + \frac{\tilde{\zeta}_{i,k}}{k}$$

18: **end for**

Chapter 5

Performance assessment in real urban scenario

Chapter 3 has offered an exhaustive description and characterization of the major Bayesian architectures which can be adopted for fusing inertial measurements and Global Navigation Satellite System (GNSS) ranging measurements within the framework of an integrated Inertial Navigation System (INS)/GNSS navigation system in a Tightly Coupled (TC) configuration. In particular, starting from the state-of-art solution, represented by the Extended Kalman Filter (EKF) (Section 3.2), an incremental path, in terms of algorithmic complexity, has been followed; at first, by keeping the focus on Kalman-based methods, the Unscented Kalman Filter (UKF) (Section 3.3) has been discussed, and special attention has been devoted to the Unscented Transform (UT), a technique to generate deterministic sigma-points (Section 3.3.1). Afterwards, on the basis of the Sequential Monte-Carlo (SMC) integration concept and its approximation in terms of Sequential Importance Sampling (SIS), the Particle Filter (PF) has been covered (Section 3.4) with a highlight on both strengths and potential weaknesses. Eventually, the truly innovative and more advanced Unscented Particle Filter (UPF) scheme has been proposed, which cleverly combines the UT-technique, to improve the shaping of the proposal distribution for particles' generation (Section 3.5), with the PF flexibility in handling measurement-noise statistics and reflecting them on particles' weighting. Last, but not least, within the UPF-routine, a novel state-splitting approach has been investigated, which envisions a multiple weighting of each particle (a particle is, inherently, a possible realization of the state-vector) to estimate separately different portions of the state-space (Section 3.5.3).
Chapter 4, then, puts forward a set of innovative methodologies ultimately oriented to an enhancement of the filtering estimation performance, and develops along two main paths. On one hand, under the assumption of independent zero-bias normally

distributed GNSS ranging measurements, some adaptive models for the stochastic characterization of the variance on pseudorange and Doppler-shift observables are dealt with; the latter are meant to improve the internal responsiveness of the filter against changes in the navigated physical environment (Sections 4.1.3 and 4.1.4). On the other hand, by targeting a mitigation of multipath effects on the estimation of the travelled trajectory, a real-time, self-contained and low-complexity pre-processing routine, acting directly on input noisy pseudorange measurements, is treated. As such, the reader is also offered a qualitative illustration of the experimental intuitions that led to the former approach (Section 4.2).
In light of the above, the present chapter is primarily devoted to illustrate and analyze the performance of the aforementioned Bayesian architectures, when employed as fusion filters within an integrated INS/GNSS system. For testing purposes, the navigation module is operated on a high-accuracy and high-precision positioning and tracking problem, in the framework of vehicular land-based navigation. In particular, among the available performance indicators, the estimation accuracy is mostly considered; it is a measure of statistical bias (systematic error) and quantifies the degree of closeness of the integrated positioning solution to a ground-reference.

5.1 Methodology and underlying assumptions

The ultimate scope of the forthcoming assessment is to verify whether the expected accuracy gain, theoretically entailed by the investigated advanced Bayesian approaches (UKF and UPF, in ascending order of complexity), truly justifies the increase of algorithmic computational burden w.r.t. the original EKF scheme. In other words, keeping a global view on the available hybridization filters, the actual objective of the following analysis is a weighting between the cost-complexity impact of the integration algorithm and the accuracy performance in the estimate of the target trajectory, seeking for the identification of the most convenient integrated solution. Moreover, throughout the whole experimental study, Kalman strategies are maintained as ultimate benchmarks since, generally speaking, they represent a very good trade-off between performance and complexity. Based on these premises, the delineation of the experimental set-up, the identification of a methodology to carry out the assessment and the isolation of a set of underlying assumptions which are maintained in the course of the analysis are strictly mandatory.

To achieve a scientifically relevant validation, a field measurement campaign is conducted based on a car ride in a urban area nearby Polytechnic University of Turin (Turin, Italy). Two categories of observables are involved in this test:

- *High-rate inertial specific force and angular rate measurements* generated

as output from a low-cost MEMS-technology strap-down Inertial Measurement Unit (IMU), i.e. TDK Invensense MPU-9250, including two triads of accelerometers and gyroscopes.

- *Low-rate noisy pseudorange and Doppler measurements* which are output from a low-cost GNSS receiver, i.e. NVS NV08C-CSM.

The foregoing hardware was mounted on the vehicle platform and the collected measurements are synchronously blended in the real-time fully-software simulator of a TC-integrated INS/GNSS navigation system presented in Section 2.2.4. Running the multiple Bayesian routines covered in Chapter 3, the resulting experimental trajectory (it counts 1740 GNSS epochs) is investigated and compared against a highly accurate ground-truth. The latter has been retrieved from a multi-frequency, multi-constellation GNSS-receiver belonging to the Novatel OEM7 family (user manual available at [50]), which combines a tactical-grade IMU with Real-Time Kinematic (RTK) technique in order to supply real-time localization at sub-centimetre level accuracy. The ground reference for the tested car ride trajectory is supplied in Figure 5.1.
In the following results, some trajectory segments (sectors) are zoomed in, for which the original EKF-based fusion architecture manifests exceptional behaviours and that are commonly critical for GNSS-based Positioning, Navigation and Timing (PNT):

- Sectors A and B identify deep urban environments with poor satellites' visibility, unfavourable geometry and potentially strong multipath and shadowing effects.

- Sector C corresponds to a mild urban context with dense foliage but better visibility conditions.

- Sector D determines a cut-off slice of the vehicle path where the average accuracy performance of any Bayesian hybridization routine tends to dramatically worsen, likely due to the presence of highly reflective and diffractive surfaces.

To have a deeper understanding of these challenging areas, Figure 5.2 captures an environmental snapshot for each indicated sector. Just for comment, in previous chapters it has been always addressed the generic concept of GNSS-receiver and its related measurements, without forcing any assumption on a specific constellation. Without loss of generality, the dataset for the current assessment involves Global Positioning System (GPS) code-ranging data only.
Reached this point, it is due remarking that, besides the intrinsic high cardinality of the integrated system state-space (Section 2.2.1), only the spatial components related to the moving-target horizontal position are object of investigation. In this

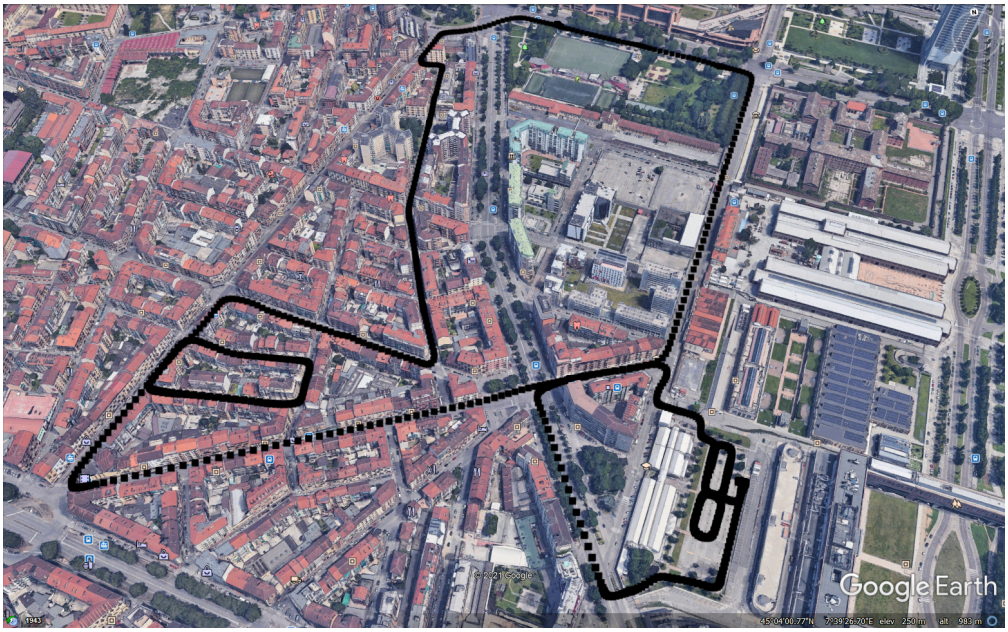


Figure 5.1: Ground-truth for the tested vehicular trajectory in the assessment (Google Earth image).

sense, the vertical position component or, more rigorously, the body geodetic height (referring to a geodetic coordinate system with an ellipsoidal approximation of the Earth surface based on WGS84-model), is not examined since, in the context of GNSS-based positioning, it is typically penalized by the geometry of the multilateration problem to be solved.
Concerning the assessment criterion, the Root-Mean-Square Error (RMSE) metric is selected. Precisely, by considering the horizontal position in Universal Transverse Mercator (UTM) coordinates (Easting/Northing), such measure expresses the standard deviation of the error between the estimated trajectory and the ground-truth (i.e. ground reference) over consecutive integration epochs. Defining $\left(E_{TC}^{k}, N_{TC}^{k}\right)$ and $\left(E_{GT}^{k}, N_{GT}^{k}\right)$ the horizontal UTM components of the k-th position fix (at k-th GNSS-epoch) associated to the TC-integrated unit and the ground-truth, respectively, the corresponding RMSE can be computed as:

$$RMSE_{EN}^{k} = \sqrt{\left(E_{TC}^{k} - E_{GT}^{k}\right)^{2} + \left(N_{TC}^{k} - N_{GT}^{k}\right)^{2}} \tag{5.1}$$

Furthermore, it is fundamental to highlight a couple of baseline assumptions on the integrated system state-space model (Section 2.2.1), which are maintained and shared across the presented experimental results:

(a) Sector A. **(b)** Sector B.

(c) Sector C. **(d)** Sector D.

Figure 5.2: Challenging navigation environments along the experimental trajectory involved in the real dataset employed for the assessment (Google Earth images).

- Linearized model for GNSS-observables (Equation 2.7).
- Zero-mean Gaussian modelling for the statistical distribution of the errors affecting noisy GNSS pseudorange and Doppler measurements.

As far as the observation model is concerned, the linearized approximation is employed in the update-step for all the different Bayesian estimators, even for those (UKF and UPF) which could directly handle the non-linear pseudorange (1.8) and Doppler (1.26) equations. This choice has been already motivated in Section 3.3.2, where it was shown, via a Cumulative Density Function (CDF) plot on the horizontal RMSE of the UKF estimated trajectory (Figure 3.2), a nearly perfect matching between the two curves obtained when using both the linearized and the non-linear models. For the specified stochastic GNSS-observables' error characterization, it is used in all the examined navigation filters. In particular, in Kalman-based approaches, it impacts on the construction of the observation-noise covariance matrix $\mathbf{R}_k$ at each integration time; on the contrary, for the UPF, it reflects on the weighting of the particles' set.

Although the Gaussianity assumption is a standard trademark for Kalman filters, for the UPF and, in particular, for the cascaded PF-stage, it may end-up creating some sub-optimality in the algorithm. In fact, one of the key-features of PFs is the possibility of handling heterogeneous models for processing the input measurements.

The remainder of this chapter is organized as follows. Section 5.2 assesses the accuracy performance of the various baseline Bayesian estimators when implemented with a semi-static stochastic weighting scheme for ranging measurements noise. Later on, Section 5.3 steers the focus on adaptive filtering strategies and attempts to validate the behaviours of residual based and redundant measurement based observation noise covariance estimation algorithms. Eventually, Section 5.4 concentrates on the self-contained pseudorange pre-processing module and its impact on the quality and robustness of the position estimates under challenging signal environments. Besides, all the aforementioned sections are developed according to a common structure, which first analyses Kalman paradigms and UPF-based schemes separately and, then, transversally combine the two Bayesian families to highlight mutual performance differences.

5.2 Legacy Bayesian architectures

This Section considers the legacy versions of the various Bayesian algorithms discussed in Chapter 3 and characterizes their performances, in terms of horizontal accuracy of the estimated vehicular trajectory, following a threefold approach. First, Kalman-only filtering architectures are accounted for, by verifying the response of the UKF paradigm over the original EKF approach. Then, the attention is oriented towards the proposed UPF solution and its estimation performance is verified as a function of the number of particles, according to an increasingly high degree of computational complexity. Eventually, owning a transversal perspective, all the available fusion schemes are jointly handled and compared, in order to portray the gains (or the losses) of each approach over the others. Besides, dealing with the stochastic characterization of the errors affecting input noisy GNSS code-ranging observables, the Carrier-to-Noise Ratio based measurement weighting strategy (i.e. semi-static approach) of Section 4.1.2 is selected for each Bayesian integration routine.

5.2.1 Kalman-based solutions: EKF and UKF

The first experimental result of Figure 5.3 compares the estimated vehicle trajectories, by both the EKF-based (Section 3.2.2) and the UKF-based (Section 3.3.2) INS/GNSS integrated navigation systems, with respect to the sub-centimetre accurate ground truth.

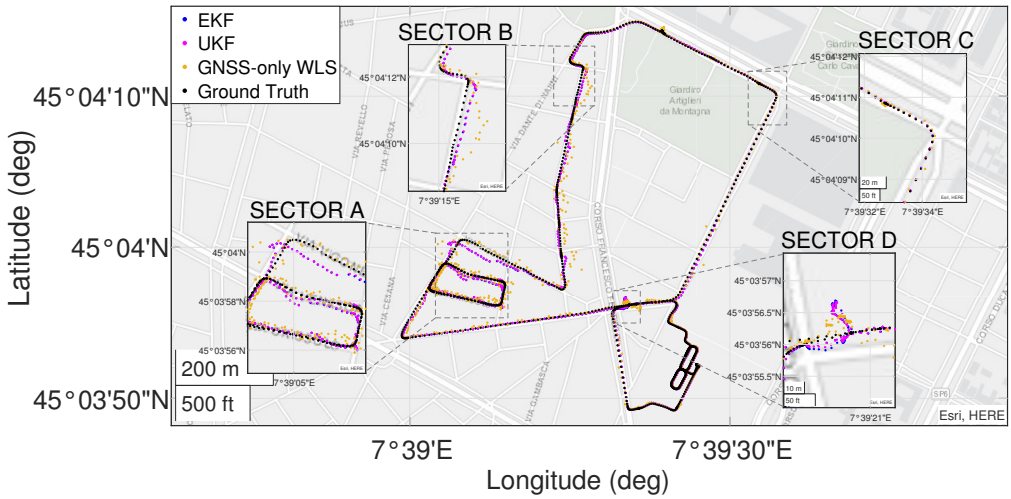


Figure 5.3: 2-D experimental trajectory in latitude/longitude (LLH) spherical coordinates. Comparison between EKF and UKF estimates w.r.t. ground-truth.

Figure 5.3 also shows the experimental trajectory output by a standalone GNSS-receiver (a GPS solution is considered in this specific framework) running, at each integration epoch, a Weighted Least-Squares (WLS) routine to retrieve a Position, Velocity, Timing (PVT) estimate. It is important to point out that, while the TC navigation unit, thanks to the integrated INS, can guarantee a long-term stable positioning solution with high update-rate (INS bridges GNSS outages and dead-reckons position, velocity and attitude states between consecutive GNSS epochs), a low-rate standalone WLS solution exists only in case a minimum of four satellites are tracked and the geometric matrix $\mathbf{H}$ is full-rank (Section 1.3.2). Observing the different trajectories, it is apparent the superior accuracy performance of the integrated solution w.r.t. a GNSS-only WLS estimate, where the former is tighter to the ground truth and behaves in a smoother fashion over time. In particular, comparing the EKF and UKF estimates, it is not possible to appreciate consistent differences at this qualitative level. Furthermore, focusing on the zoomed sectors A and B, it is evident the injection of a bias in the integrated solution, and this is likely caused by multipath. Last, but not least, in sector D, the accuracy performance for both filters degenerates.

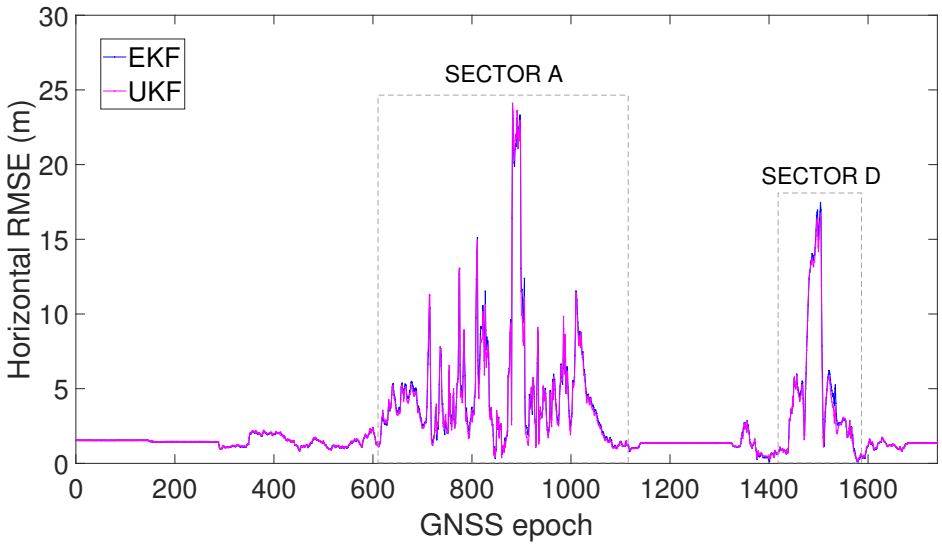


Figure 5.4: RMSE on the horizontal position (E/N) in UTM-coordinates. Comparison between EKF and UKF over the estimated trajectory fixes.

The temporal behaviour, over consecutive integration epochs, of the horizontal position RMSE is shown in Figure 5.4 for both Kalman architectures. In particular, the critical trajectory sectors A (Figure 5.2a) and D (Figure 5.2d) are identified and the average error increments are blatant. Such biasing effects are induced by the presence of tall buildings (and other environmental obstacles) potentially

obstructing LOS-propagation and enhancing scattering phenomena (i.e. multipath). Observing the global error-trend, it can be evidenced the presence of a steady bias floor amounting to about 1.5 m, which is caused by some alignment issues in the tactical-grade IMU used for the ground-truth.
As it was already foreseen from the graphical representation of the estimated trajectories, there is a tiny difference, in terms of accuracy performance, between the EKF and UKF. In fact, the RMSE curves overlap to a large extent, even though the UKF seems to gain, on average, some very small margin w.r.t. the EKF.
The latter observation can be better appreciated by considering the empirical CDFs reported in Figure 5.5.
Despite the maximum horizontal error for the EKF (23.7 m) is slightly smaller than that of the UKF (24.1 m), the latter architecture always enhances the accuracy performance by reducing the error of few centimetres at almost all percentiles. In particular, the maximum improvement is measured at the intermediate 75-th percentile with a value of 3.49 % (9.9×10^{-2} m). It means that, on average over the whole trajectory, 75 % of times the UKF estimate, in terms of horizontal position component, is affected by an error which is 9.9×10^{-2} m smaller than that of the EKF, when both misalignments are measured against a common truth. This is an encouraging result if contextualized within a perspective of balancing between performance and complexity, but the absolute performance gain is certainly minimal. Anyway, moving from the EKF to the UKF, the algorithmic complexity is just slightly increased.

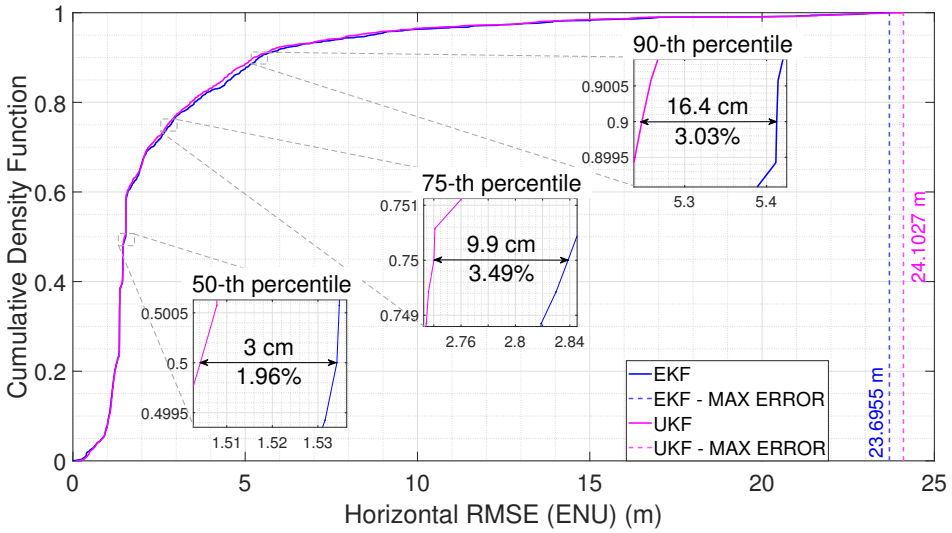


Figure 5.5: Empirical CDF (ECDF) of the RMSE on the horizontal position (E/N) in UTM-coordinates. Comparison between legacy EKF and UKF architectures.

Another aspect which should be pointed out, concerns the accuracy gain as a function of the error magnitude. For small average horizontal errors, in a bounded range up to 3 m, the accuracy improvement, in units of centimetres, is relatively limited, but the corresponding percentage improvement, especially at 75-th percentile, is the largest measured. On the contrary, moving towards higher percentiles and, in particular, observing what happens at the 90-th level, the centimetre gain on the horizontal error is nearly twice as much as it is at 75-th percentile, but the percentage improvement is smaller.
Eventually, although not invalidating the comparison, it cannot be neglected the existence of the aforementioned small-scale bias floor which impacts on the absolute accuracy performance measured for each Bayesian implementation. In fact, all the involved error CDFs would be expected to shift to the left such that to roughly bound the horizontal RMSE below 2 m.

5.2.2 Unscented Particle Filter (UPF) and state-splitting method

In this Section, the attention shifts on the proposed UPF scheme, which is isolated from the previous Kalman-based approaches and its performance is analyzed by varying the number of particles in the cascaded PF-stage. Specifically, two baseline UPF architectures are accounted for, which mainly differ on how particles, together with their relative weights, contribute to the a-posteriori state-estimation:

- Legacy algorithm (Section 3.5.1), where each weighted particle, representing a possible realization of the system state, contributes totally to the a-posteriori state vector and state-covariance estimates. Hereinafter, it will be addressed as *plain* UPF version.

- Modified routine (Section 3.5.3), where each particle is assigned multiple weights and each weight is used to estimate only a portion of the state-vector. Hereinafter, it will be addressed as *state-splitting* UPF version.

A comparison of the estimated trajectories, considering both aforementioned architectures, when varying the number of randomly generated particles, is supplied in Figure 5.6.
First of all, by looking at sector A, it is apparent that the trajectory output by the state-splitting architecture suffers from a larger error w.r.t. the plain version, when both are referenced to the ground-truth. In other words, it seems that, in presence of multipath, the splitting method still grants a smooth trajectory estimate but it is more prone to absorb an unpredictable bias, potentially affecting pseudorange measurements, which detaches the estimated vehicle position fixes from the ground-reference. This effect, in particular, gets even stronger as the

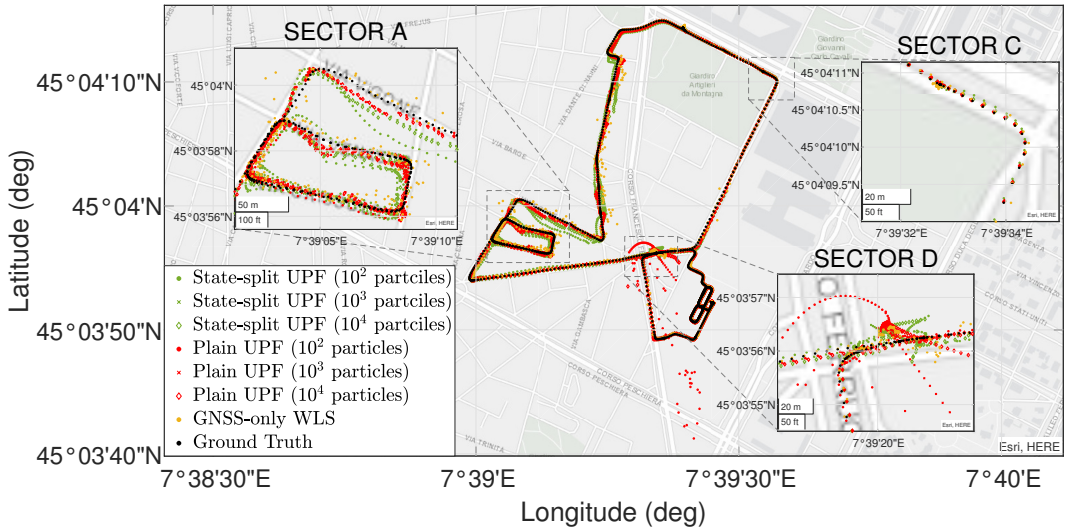


Figure 5.6: 2-D experimental trajectory in latitude/longitude (LLH) spherical coordinates. Comparison, over the integration epochs, between plain and state-splitting UPF architectures at 10^2, 10^3 and 10^4 particles.

number of particles decreases. Furthermore, such bias, once onset, is dragged over a non-negligible set of consecutive GNSS epochs and the filter manifests weak reactivity in recovering from it.
Instead, moving the attention on sector D, the state-splitting UPF is able to keep the error bounded, while the plain routine starts diverging. Hence, given these experimental evidences, a hypothetical suggestion is that, in harsh navigation conditions with poor satellite visibility and strong measurement noise, the Bayesian UPF-strategy with state-splitting offers higher robustness to the integrated navigation system and higher stability to the positioning solution, but it is more sensitive to multipath-related bias injections and slower in recovering from them.
Contrarily, in the mild urban context of sector C, both UPF paradigms exhibit a good accuracy performance, even for a small number of particles, and it is hard, by just observing the trajectories, to identify which implementation outperforms the others. Anyway, the choice of highlighting such sector stems from the will to observe what happens to the horizontal position estimate when the vehicle inertial dynamics are evolving much faster. As a matter of fact, the INS proves to be stable (as discussed in Chapter 4) and there is no critical enhancement of the IMU-drift. The former comments can be further supported by investigating on Figure 5.7, which displays the horizontal position RMSE and highlights the critical trajectory segments where peaky error behaviours pop up. Interestingly, between epochs 1400 and 1600, the horizontal error measured for the plain architecture explodes to unacceptably large values above 100 m, irrespective of the number of employed

particles (the case at 10^2 particles is not shown due to filter divergence). The state-splitting variant, instead, succeeds in keeping the maximum error bounded and the filter is stable even for a limited amount of particles.

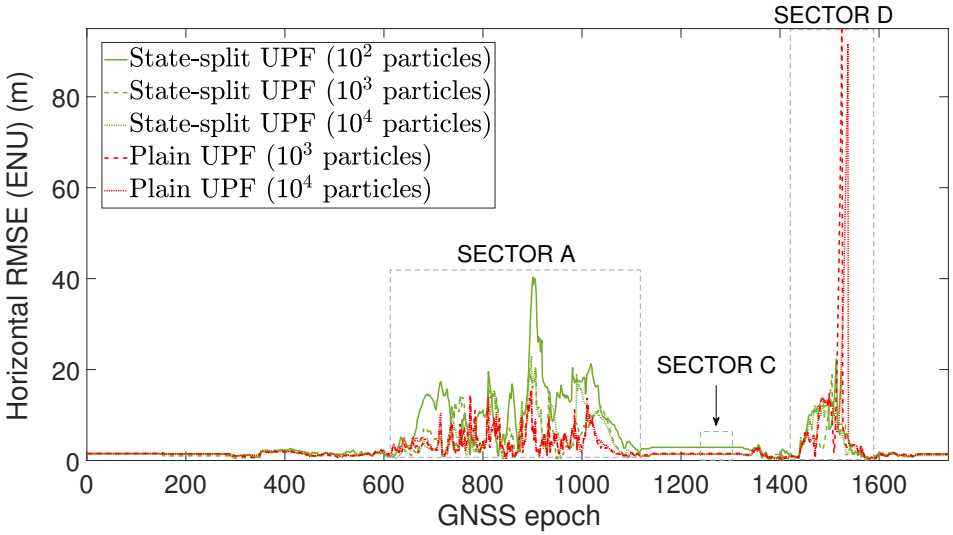


Figure 5.7: RMSE on the horizontal position (E/N) in UTM-coordinates. Comparison between plain and state-splitting UPF architectures at 10^2, 10^3 and 10^4 particles.

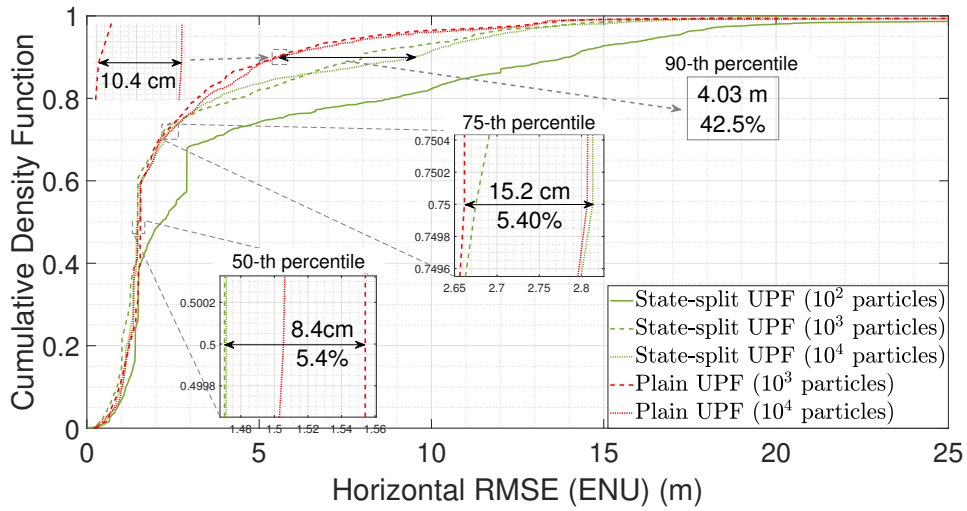
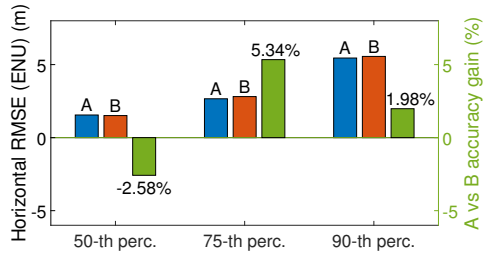
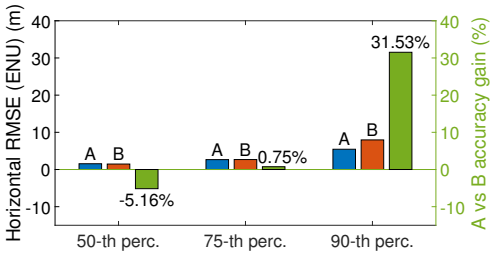


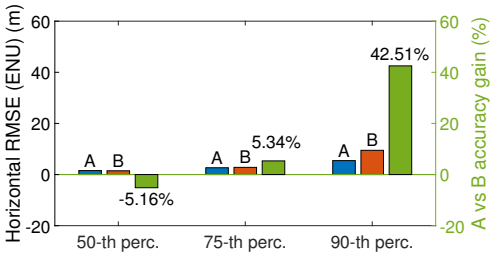
Figure 5.8: Empirical CDF (ECDF) of the RMSE on the horizontal position (E/N) in UTM-coordinates. Comparison between plain and state-splitting UPF architectures at 10^2, 10^3 and 10^4 particles.



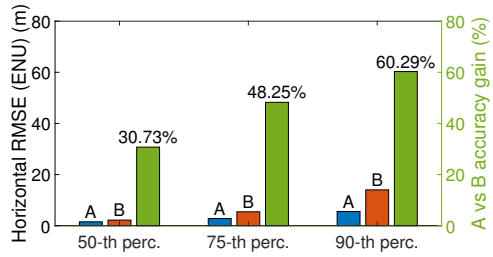
(a) Plain UPF 10^3 particles (A) & Plain UPF 10^4 particles (B).



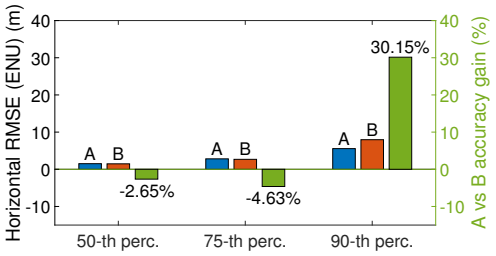
(b) Plain UPF 10^3 particles (A) & state-splitting UPF 10^3 particles (B).



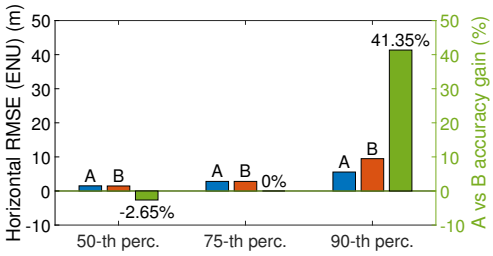
(c) Plain UPF 10^3 particles (A) & state-splitting UPF 10^4 particles (B).



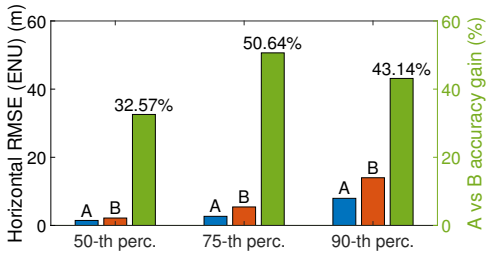
(d) Plain UPF 10^4 particles (A) & state-splitting UPF 10^2 particles (B).



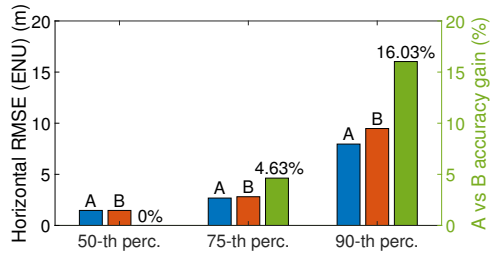
(e) Plain UPF 10^4 particles (A) & state-splitting UPF 10^3 particles (B).



(f) Plain UPF 10^4 particles (A) & state-splitting UPF 10^4 particles (B).



(g) State-splitting UPF 10^3 particles (A) & state-splitting UPF 10^2 particles (B).



(h) State-splitting UPF 10^3 particles (A) & state-splitting UPF 10^4 particles (B).

Figure 5.9: One-to-one comparison among different couples of baseline UPF architectures (plain and state-splitting) in terms of percentage accuracy gain/loss considering 50-th,75-th and 90-th percentiles.

Table 5.1: Horizontal (E/N) position RMSE measured, for both plain and state-splitting UPF routines, at 50-th, 75-th and 90-th percentiles.

UPF architecture	50-th percentile Error (m)	75-th percentile Error (m)	90-th percentile Error (m)
Plain (10^3 particles)	1.55	2.66	5.45
Plain (10^4 particles)	1.51	2.81	5.56
State-splitting (10^2 particles)	2.18	5.43	14
State-splitting (10^3 particles)	1.47	2.68	7.96
State-splitting (10^4 particles)	1.47	2.81	9.48

The empirical CDFs for the horizontal position RMSE, associated to the analyzed UPF approaches as a function of the number of particles, are shown in Figure 5.8. Accordingly, Table 5.1 summarizes the measured horizontal errors at the percentiles of interest (50-th, 75-th and 90-th). Although, for lower percentiles, the estimated errors are comparable between plain and state-splitting algorithms, at higher percentiles the measured horizontal error for the latter UPF solution is nearly twice as much the error measured for plain filter version. In fact, this is not at all a surprising experimental outcome since it simply reflects the tendency of the splitting-based scheme to easily absorb some bias (likely induced by multipath) which, above all, is slowly neutralized.
In order to draw up some additional comments to assess the performances of the multiple UPF strategies investigated within this section, it might be helpful to resort to Figure 5.9, where the involved Bayesian schemes are treated in pairs and, for each couple, focusing on the 50-th, 75-th and 90-th percentiles, it is highlighted the percentage accuracy gain (or loss) of one filter over the other. Again, only the accuracy performance on the horizontal position component is accounted for.
Concerning the plain UPF algorithmic solution applied to an integrated INS/GNSS navigation system, it is not apparently advantageous to increase the number of particles in the aim of gaining some better accuracy. Indeed, looking at Figure 5.9a, when moving to higher percentiles the UPF with 10^3 particles grants a better horizontal accuracy w.r.t. the case at 10^4 particles. The latter evidence is satisfactory because the lower the number of particles, the smaller the amount of handled computational burden. On the other hand, focusing on the UPF variant with state-splitting through Figures 5.9g and 5.9h, the implemented solution with 10^3 particles outperforms, by no means, the low-complexity stage at 10^2 particles (5.9g). On top of that, from Figure 5.9h, it comes out again that an increase in the number of particles is not rewarded by an accuracy enhancement. Hence, for both filtering variants, the mid-complexity solution at 10^3 particles sets as the best

compromise between performance and complexity.
In the end, seeking for a cross-comparison between the two UPF architectural proposals, Figures 5.9b and 5.9f deliver a one-to-one comparison at a fixed number of samples (10^3 and 10^4). Both plots share an intrinsic analogy: the plain UPF performance is worse than the state-splitting one at 50-th percentile (and even for lower percentiles, if considering Figure 5.8), but a complete turnaround is observed for higher percentiles, especially at 90-th one. The interpretation of this result is fully aligned to previous observations; the low-percentile loss is plausibly induced by the large horizontal error which is measured in sector D (Figure 5.6), which is anyway a short-term performance degradation (low percentile). On the other side, averaging over the full trajectory, the split-solution is affected by a larger bias (Figure 5.8) and this directly mirrors the larger error-magnitude for high percentiles. A different reading of the foregoing outcomes can be achieved adopting the RMSE magnitude as viewpoint. As such, the state-splitting paradigm grants a finer estimation quality in the high-accuracy CDF segment where horizontal errors below 2 m are involved; on the contrary, for larger misalignment ranges, the plain architecture yields better performance.
To conclude, weighting the variety of results discussed in this section, the plain UPF architecture at 10^3 particles ends up providing the most advantageous trade-off between estimation accuracy and complexity when considering the RMSE trend globally. Instead, in case the interest is limited to the high accuracy region, the state-splitting implementation turns out to be rather effective.

5.2.3 UPF vs Kalman schemes

Previous Sections 5.2.1 and 5.2.2 have been devoted to the assessment and characterization of the two disjoint families of Bayesian architectures, namely PF-based strategies (UPF) and Kalman-based (EKF and UKF) methods, respectively. The present section, then, proposes a synthesis, achieved through experimental validation, between the multiple and heterogeneous filtering schemes in the view of picking out, if possible, the most promising and best balanced architectural solution for an estimator to be operated as navigation filter in a TC INS/GNSS system.
As a starting point, some performance hints can be caught from examining Figure 5.10, where the estimated horizontal trajectories of the moving vehicles are shown. Looking at Sector A, it seems that the standard UPF scheme, at both 10^3 and 10^4 particles, is tighter to the truth w.r.t. the other filters, and this is partially confirmed in Sector B, where the Kalman approaches estimate a trajectory which is roughly as accurate as that of the UPF. Clearly, the UPF at 10^2 particles does not have enough samples to catch a system state-estimate which, considering horizontal position components only, can get close to the reference (i.e. true state-components). The latter problem culminates into the filter divergence in sector D, and this is

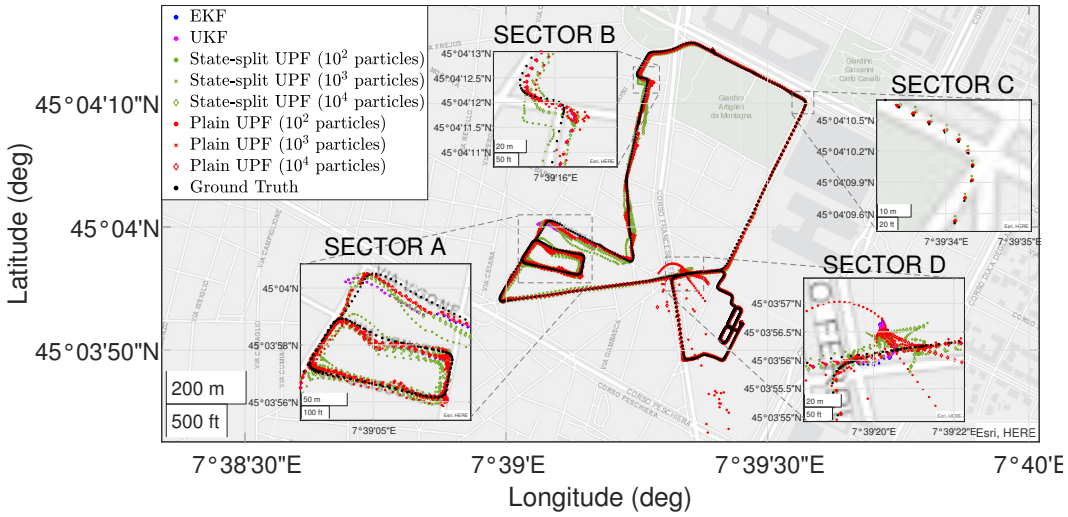


Figure 5.10: 2-D experimental trajectory in latitude/longitude (LLH) spherical coordinates. Comparison between EKF, UKF and UPF (plain and state-splitting architectures at 10^2, 10^3 and 10^4 particles).

the reason why it will not be analysed further in the remainder of this section. Anyway, qualitatively speaking, there is not a filtering solution which is strikingly outperforming the others in harsh trajectory segments (i.e. A,B and D). Certainly, the state-splitting UPF, both at 10^3 and 10^4 particles, is suffering more from multipath interference with an estimated trajectory which is shifted. Eventually, in parallel to the discussion of Section 5.2.2, the state-splitting version at 10^2 particles, despite being more robust and providing a smoother trajectory estimate than the equivalent plain solution, is characterized by a poorer accuracy performance. Therefore, it is neglected hereinafter. To pursue a deeper characterization of the considered architectures, it would be interesting to inspect Figure 5.11, which shows their horizontal RMSE CDFs. At 50-th percentile, the best accuracy is achieved by the state-splitting UPF with a measured horizontal error of 1.47 m (refer to Table 5.2), which implies an accuracy gain of 3.4×10^{-2} m w.r.t. the UKF and an even larger gain of 6.4×10^{-2} m w.r.t. the EKF. The latter result is promising, especially when noticing that the maximum error is bounded to less than 2 m almost 60 % of time. Moving to the higher 75-th percentile, it is possible to observe an inversion of the initial trend which brings ahead the plain UPF implementation at 10^3 particles, whose accuracy performance starts prevailing over the other filtering schemes. Simultaneously, the curves associated to the state-splitting solutions (10^3 and 10^4 particles) bend down the right and, from here on, the achieved performances become worse than those provided by Kalman estimators. Then, at 90-th percentile, the UKF strategy outperforms, limiting the horizontal position error to

5.25 m (Table 5.2). Based on this last experimental evidence, it could be asserted that the UKF identifies the best behaving architecture with a reasonably limited complexity, since 90 % of times it grants the estimated position to have a maximum error of 5 m w.r.t. the ground-truth. However, prior to come to any conclusion, it is advisable to take a look at Figure 5.12, where the different architectures are compared at couples to emphasize the accuracy improvement or deterioration of one integrated system over the other.

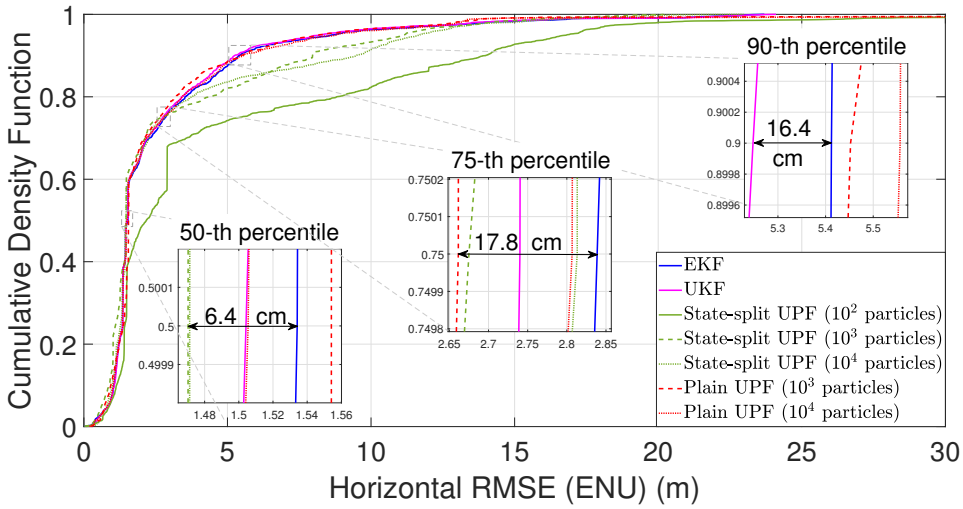


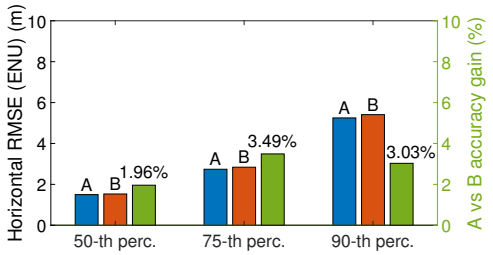
Figure 5.11: Empirical CDF (ECDF) of the RMSE on the horizontal position (E/N) in UTM-coordinates. Comparison between EKF, UKF and UPF (plain and state-splitting architectures at 10^2, 10^3 and 10^4 particles).

Figure 5.12a definitely validates what has been discussed in Section 5.2.1, with an overall improved accuracy performance of the UKF over the EKF, at the cost of a moderate complexity enhancement (the implemented UKF, in fact, relies on $2n$ sigma points only, where $n = 17$ is the integrated system state-space dimension). Focusing on the standard UPF algorithm employing 10^3 particles (Figures 5.12b and 5.12c), it can be highlighted, at the 75-th percentile, an horizontal accuracy gain over the Kalman architectures. In particular, compared to the EKF, the improvement amounts to 6.27 %. Moreover, for higher percentiles and, specifically, at the 90-th one, the same UPF scheme looses some accuracy w.r.t. Kalman filters, but moderately. The latter fact suggests that, very likely, the UPF is not able to gain at all percentiles (i.e. in absolute terms) since its performance is somehow penalized by the permanent adoption of a zero-mean Gaussian density model for the noise affecting input GNSS-measurements. Accordingly, for those trajectory sectors where the external environment is expected to degrade considerably the

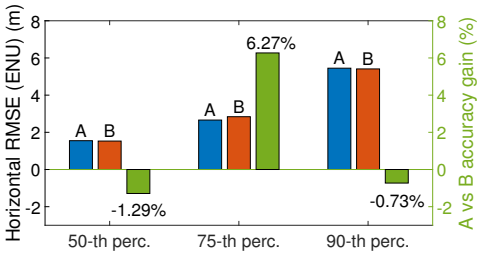
observables quality, the Gaussian model is probably not the best stochastic fit to approximate the true unknown noise distribution and it would be necessary to resort to alternative analytic models. Unfortunately, the non-stationary and non-ergodic measurement-noise process evolution complicates the elaboration of unambiguous statistical characterizations; hence, a Gaussian approximation is still rather effective. Keeping the same UPF architecture and increasing by a factor 10 the number of particles (Figures 5.12f and 5.12g), it improves the accuracy performance at lower percentiles (50-th percentile) but, for high percentiles, the measured loss is higher than that for the 10^3 particles solution. Consequently, the filtering scheme at 10^3 particles identifies the best trade-off between estimation accuracy and computational complexity. Moving the attention on the state-splitting UPF, Figures 5.12d and 5.12f compare the implementation at 10^3 particles against both versions of Kalman filters. Evidently, at both 50-th and 75-th percentiles, it is measured an accuracy gain for the UPF over both the EKF and the UKF. However, for higher percentiles, the UPF accuracy behaviour worsens (refer to Figure 5.11 for a visual understanding) and, at 90-th percentile, the estimated performance loss is above 30 %.
Coming to a conclusion, the UKF demonstrates to slightly improve, at almost all percentiles, the horizontal accuracy over the original EKF, with a contained complexity enhancement. The increase of computational burden, w.r.t. Kalman solutions, brought about by the UPF architecture does not seem to be justified by a remarkable gain over the other fusion algorithms. Potentially, if valuable measurement models other than Gaussian were accessible, the observed UPF improvement would be much higher.

Table 5.2: Horizontal (E/N) position RMSE measured for EKF, UKF and UPF (both plain and state-splitting versions), at 50-th, 75-th and 90-th percentiles.

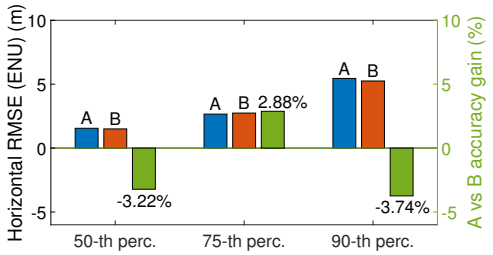
Bayesian architecture	50-th percentile Error (m)	75-th percentile Error (m)	90-th percentile Error (m)
EKF	1.534	2.839	5.412
UKF	1.504	2.740	5.248
Plain UPF (10^3 particles)	1.554	2.661	5.452
Plain UPF (10^4 particles)	1.505	2.807	5.556
State-splitting UPF (10^2 particles)	2.175	5.426	14
State-splitting UPF (10^3 particles)	1.470	2.675	7.964
State-splitting UPF (10^4 particles)	1.471	2.813	9.483



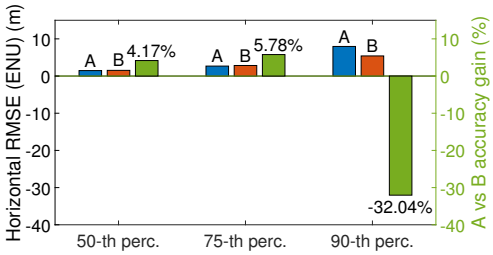
(a) UKF (A) & EKF (B).



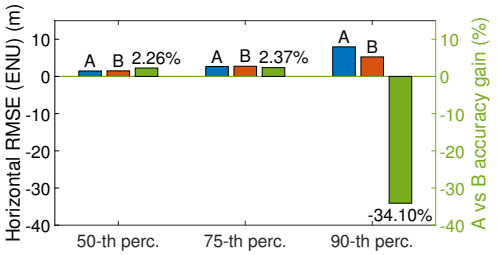
(b) Plain UPF 10^3 particles (A) & EKF (B).



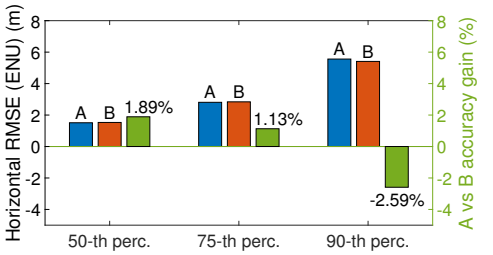
(c) Plain UPF 10^3 particles (A) & UKF (B).



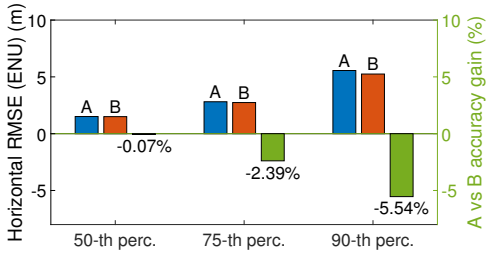
(d) State-splitting UPF 10^3 particles (A) & EKF (B).



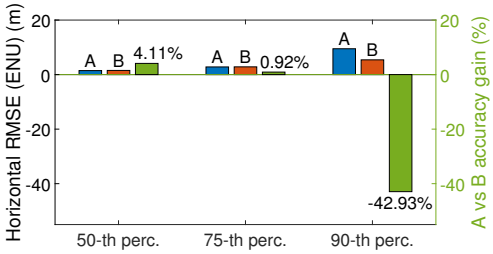
(e) State-splitting UPF 10^3 particles (A) & UKF (B).



(f) Plain UPF 10^4 particles (A) & EKF (B).



(g) Plain UPF 10^4 particles (A) & UKF (B).



(h) State-splitting UPF 10^4 particles (A) & EKF (B).

Figure 5.12: Comparison at couples between EKF, UKF and baseline UPF (plain and state-spitting) architectures in terms of percentage accuracy gain/loss considering 50-th,75-th and 90-th percentiles.

5.3 Bayesian architectures with adaptive noise variance models

Section 5.2 characterized the accuracy performance, in terms of vehicle horizontal position components (Easting/Northing), of the legacy (or baseline) Bayesian estimation strategies covered in Chapter 3, by testing them with real INS and GPS data. As a matter of fact, a zero-mean Gaussian model was adopted for the stochastic description of noise affecting input GPS measurements, thus letting their variance as the only tunable statistical parameter inside the integration filter (one degree of freedom). Given so, the semi-static (i.e. with weak adaptivity pattern) Carrier-to-Noise-density ratio (C/N_0) based variance weighting strategy was accounted for throughout the assessment.
In light of the above, this Section deals with the adaptive variance modelling paradigms discussed in Chapter 4 and, specifically, the Residual-based Adaptive Estimation (RAE) scheme (Section 4.1.3) and the Redundant Measurement Noise Covariance Estimation (RMNCE) method (Section 4.1.4) are experimentally tested and validated. The performance assessment develops across three stages of analysis. First, the estimated trajectory by an adaptive UKF is compared against that of an adaptive EKF, to verify the response of the former approach over the latter. Moreover, both architectures are compared against the legacy implementations to ultimately assess whether adaptive solutions really bring about some advantage at the cost of higher complexity. Secondly, the Adaptive Unscented Particle Filter (AUPF) paradigm is targeted and, by fixing the number of particles to 10^3, the estimation performance of RAE and RMNCE implementations are verified in detail and compared to the baseline filtering model. In the end, the different adaptive fusion algorithms are transversally handled to highlight the strengths and weaknesses of each approach over the others.

5.3.1 Adaptive Kalman filters

Considering both legacy and adaptive (RAE and RMNCE) implementations of EKF and UKF Bayesian architectures, Figure 5.13 shows the estimated vehicle trajectories in the horizontal spatial component (the altitude behaviour is not considered in this assessment) using geodetic latitude and longitude coordinates. The same figure also reports the estimated vehicle path by a standalone GNSS receiver running a WLS solution at each epoch.
Looking at Sector A, it turns out that all solutions have a very similar trend but the baseline routines get closer to the ground-truth (reference trajectory).
Furthermore, from a purely qualitative perspective, fixing the measurement-noise variance model, there is a tiny mismatch between UKF and EKF estimates with differences of few centimetres in the horizontal accuracy performance. The latter

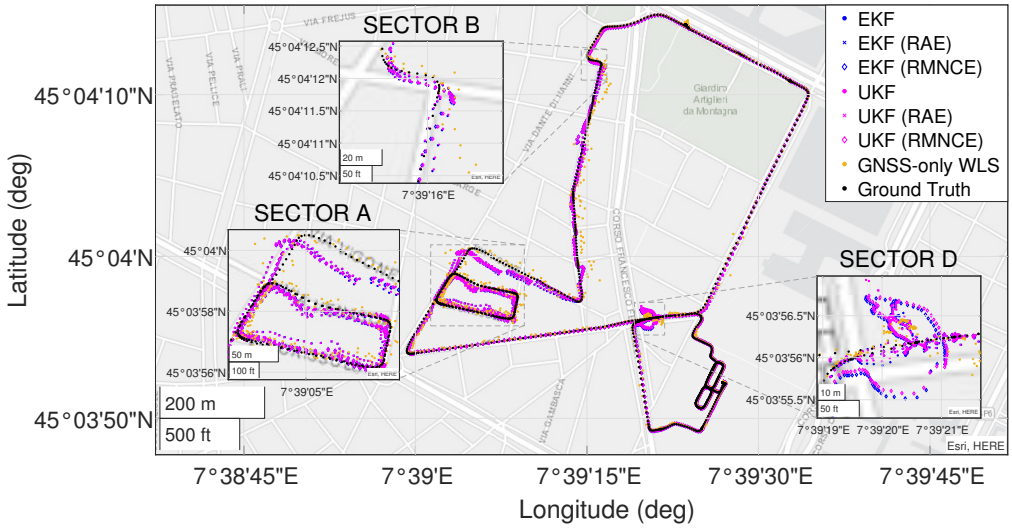


Figure 5.13: 2-D experimental trajectory in latitude/longitude (LLH) spherical coordinates. Comparison between legacy (C/N_0 based) and adaptive (RAE and RMNCE) EKF and UKF algorithms.

statement further confirms the observations discussed in Section 5.2.1. Contrarily, fixing the same Bayesian architecture and accounting for adaptive noise-variance estimation methods only, it seems the RMNCE scheme suffers less (better accuracy) from multipath w.r.t. the RAE approach. To investigate on the possible reasons, it is useful to review the key features of the adaptive algorithms presented in Sections 4.1.3 and 4.1.4. In fact, the RAE method estimates the measurement noise covariance matrix $\mathbf{R_k}$ as the sum of the residual-sequence covariance matrix $\mathbf{C}_{\mathbf{r}_k}$ (Equation 4.8) and a term which depends on the a-posteriori state-covariance update $\mathbf{P}_{k|k}$ (Equation 4.10). As for $\mathbf{C}_{\mathbf{r}_k}$, it is built from measurement residuals (Equation 4.9) and, if few raw observables were affected by a strong bias induced by multipath, this would inevitably reflect on the estimate of their residual covariance. More importantly, considering the dependence on $\mathbf{P}_{k|k}$, if the a-posteriori state estimate $\hat{\mathbf{x}}_{k|k}$ were biased, this effect would mirror on the state-covariance $\mathbf{P}_{k|k}$ itself and, in turns, would badly affect the estimate of $\mathbf{R_k}$. Thus, to summarize, the noise covariance estimated by RAE has a double point of failure which can be smashed by multipath. On the contrary, the computation of $\mathbf{R_k}$ by the RMNCE strategy involves First-Order Self-Differences (FOSDs) (4.14), which correspond to an approximated, discrete, first-order derivative. Therefore, considering a sufficiently large window of collected pseudorange and Doppler measurements, the derivative over the window implements a basic low-pass filtering which can help in attenuating the impact of multipath and of its induced bias.
Moving the attention on Sector B, the accuracy performance of adaptive methods

improves considerably and gets tighter to the true trajectory. Therefore, although loosing some performance when navigating in critical environments with heavy scattering phenomena, adaptive implementations appear to enhance the estimation accuracy over baseline Kalman algorithms when travelling more favourable scenarios.

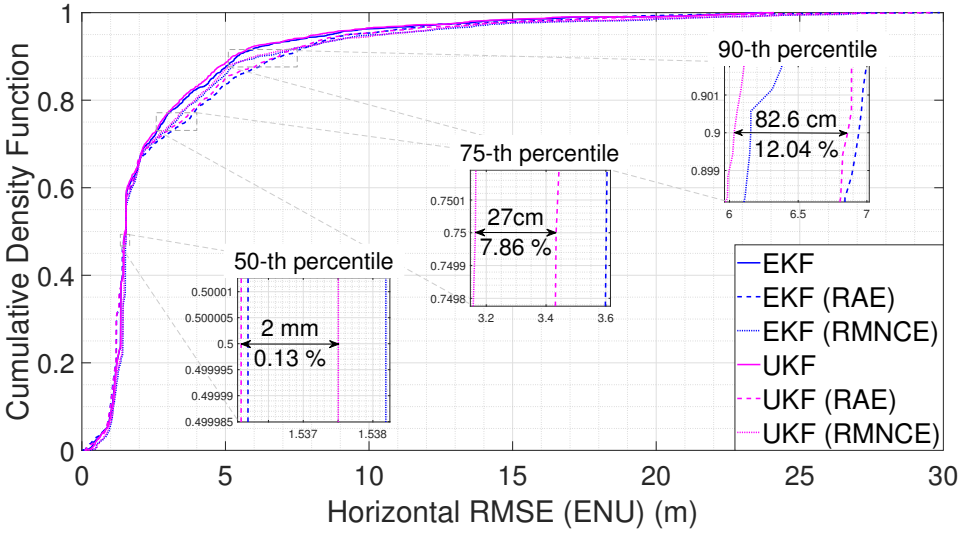


Figure 5.14: Empirical CDF (ECDF) of the RMSE on the horizontal position (E/N) in UTM-coordinates. Comparison between EKF and UKF employing both C/N_0 based (legacy implementation) and adaptive (RAE and RMNCE) methods for noise-covariance estimation.

Although the foregoing qualitative assessment allows to catch a general idea about the performance trend and the behaviour of the investigated adaptive Kalman strategies, it is certainly not enough to accomplish an exhaustive assessment and a scientific validation. Therefore, it would be recommended to investigate on Figure 5.14, showing up the empirical CDFs of the RMSE on the horizontal position components. Starting from low percentiles and, specifically, targeting to the 50-th percentile, the accuracy performance of the legacy routines is slightly better than that achieved with either of the adaptive schemes (refer to Table 5.3), especially when considering the baseline UKF implementation. Isolating the adaptive strategies (zoom in Figure 5.14), the filtering variant with RAE brings a ridiculous improvement of few millimetres over the corresponding RMNCE method, hence being considerable equivalent to the latter. Furthermore, scanning Table 5.3, the legacy EKF roughly converges to the corresponding RAE-based adaptive version, with a mutual horizontal error variation of about 2×10^{-3} m. Consequently, for

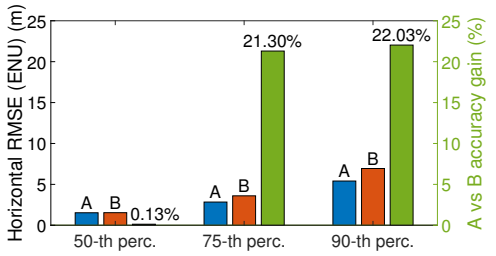
small-valued horizontal RMSE, say within 1.6 m, the best trajectory estimates are achieved by the baseline UKF routine with a moderate centimetre improvement ($\simeq 30 \times 10^{-2}$ m) w.r.t. to the remainder Kalman variants. Anyway, it is the due point here to stress that, for tiny horizontal errors, small gains of few centimetres could be potentially relevant for some high-accuracy positioning applications.
Moving towards higher percentiles, the multiple CDF lines start bending towards higher errors and branch out, with the legacy architectures granting a superior performance over adaptive methods by owning an average horizontal RMSE gain of about 50×10^{-2} m. In particular, at the 75-th percentile, the adaptive RMNCE and RAE implementations of the UKF measure considerable gaps, compared to the non-adaptive filter version, of 42.4×10^{-2} m and 69.4×10^{-2} m, respectively (Table 5.3). In addition, focusing on adaptive approaches only, the RMNCE method proves to be more accurate, with measured horizontal errors having smaller magnitude on average. For instance, at the 75-th percentile, the UKF with RMNCE gives an error which is 27×10^{-2} m smaller w.r.t. the one with RAE. Even greater, at the 90-th percentile the error-gap between RMNCE and RAE rises up to 82.6×10^{-2} m. The latter results demonstrate that, for higher errors very likely caused by multipath, the RMNCE paradigm behaves more resiliently and is capable of better mitigating the impact of such harsh phenomenon.

Digging further in the analysis of the involved Kalman filtering strategies, it is convenient to look at Figure 5.15, where the aforementioned Bayesian estimators are isolated in pairs and mutually compared in terms of percentage accuracy. First of all, observing all sub-plots at the 50-th percentile level, it is plain to see that there is not an architecture which is considerably improving, or worsening, its horizontal error performance over the other schemes. Actually, accounting for Figures 5.15b and 5.15d, the legacy UKF scheme racks up a small improvement of about 2 % w.r.t. its corresponding adaptive versions and sets, for small errors, as the filtering solution supplying the trajectory fixes closest to those of the ground-truth.
On the contrary, for higher percentiles, the framework becomes more complex. Restricting the sight on adaptive implementations, Figures 5.15e and 5.15f evidently emphasize an accuracy gain of the RMNCE method over the RAE which increases with the error. This is perfectly aligned with the former comments; the higher the error magnitude, the stronger the probability for the same error to be caused by multipath and, in presence of the latter unwanted effect, the RMNCE algorithm demonstrates higher robustness and flexibility. Given so, it is easier to understand the percentage trends of Figures 5.15a and 5.15b contrasted to those of Figures 5.15c and 5.15d. In fact, at higher percentiles, baseline Kalman routines, which overall manifest superior performance, have a percentage accuracy gain which is stronger over the RAE scheme (above 20 %) than over the RMNCE (below 13 %). As a matter of fact, this behaviour is verified at the 75-th percentile and it is even

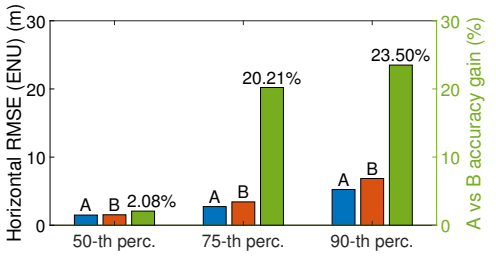
more stressed at the 90-th one.
Another interesting validation is caught from the lineup between Figures 5.15c and 5.15d; fixing the same adaptive method (i.e. RMNCE), the percentage accuracy improvement, at all percentiles (that is, both for smaller and larger errors), is a bit higher for the UKF routine.
Eventually, considering adaptive implementations only and fixing one method (RAE or RMNCE), the UKF proves to perform better than the EKF, but the percentage improvements are minor.

Table 5.3: Horizontal (E/N) position RMSE measured for EKF and UKF employing both C/N_0 based (legacy implementation) and adaptive (RAE and RMNCE) methods for noise-covariance estimation, at 50-th, 75-th and 90-th percentiles.

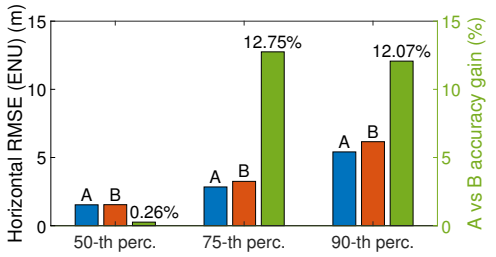
Kalman architecture	50-th percentile Error (m)	75-th percentile Error (m)	90-th percentile Error (m)
Legacy EKF	1.534	2.839	5.412
Legacy UKF	1.504	2.740	5.248
Adaptive EKF (RAE)	1.536	3.601	6.941
Adaptive UKF (RAE)	1.536	3.434	6.860
Adaptive EKF (RMNCE)	1.538	3.248	6.155
Adaptive UKF (RMNCE)	1.538	3.164	6.034



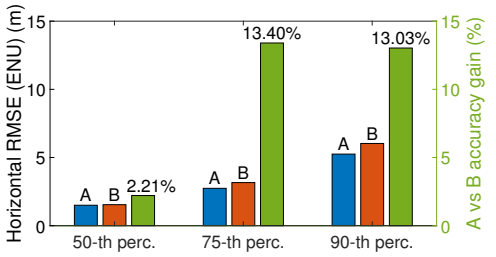
(a) Legacy EKF (A) & Adaptive EKF (RAE) (B).



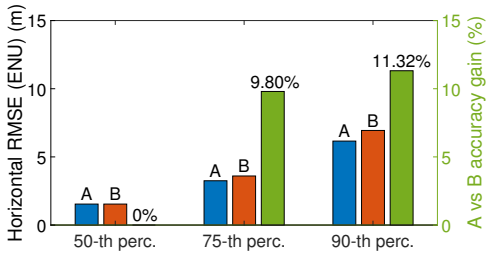
(b) Legacy UKF (A) & Adaptive UKF (RAE) (B).



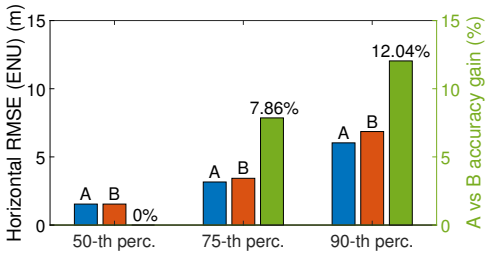
(c) Legacy EKF (A) & Adaptive EKF (RMNCE) (B).



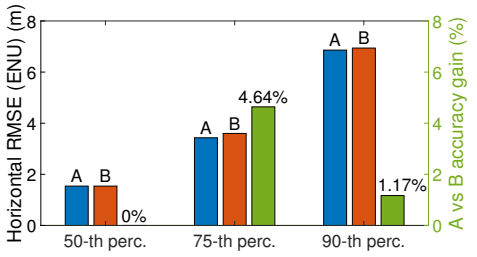
(d) Legacy UKF (A) & Adaptive UKF (RMNCE) (B).



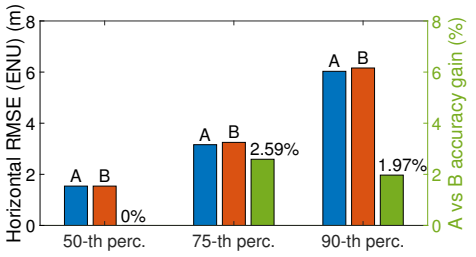
(e) Adaptive EKF (RMNCE) (A) & Adaptive EKF (RAE) (B).



(f) Adaptive UKF (RMNCE) (A) & Adaptive UKF (RAE) (B).



(g) Adaptive UKF (RAE) (A) & Adaptive EKF (RAE) (B).



(h) Adaptive UKF (RMNCE) (A) & Adaptive EKF (RMNCE) (B).

Figure 5.15: Comparison at couples between EKF and UKF (baseline and adaptive architectures) in terms of percentage accuracy gain/loss considering 50-th, 75-th and 90-th percentiles.

5.3.2 Adaptive Unscented Particle Filter (AUPF)

In this section the focus is moved towards the AUPF architecture, which is assessed when using both RAE (Section 4.1.3) and RMNCE (Section 4.1.4) strategies and its performance is compared against the legacy UPF algorithm. Similarly to Section 5.2.2, both plain and state-splitting implementations are considered for the underlying PF-stage. Moreover, throughout the analysis, the number of particles is kept fixed to 10^3, which was found to set a worthy trade-off between estimation accuracy and algorithmic complexity.

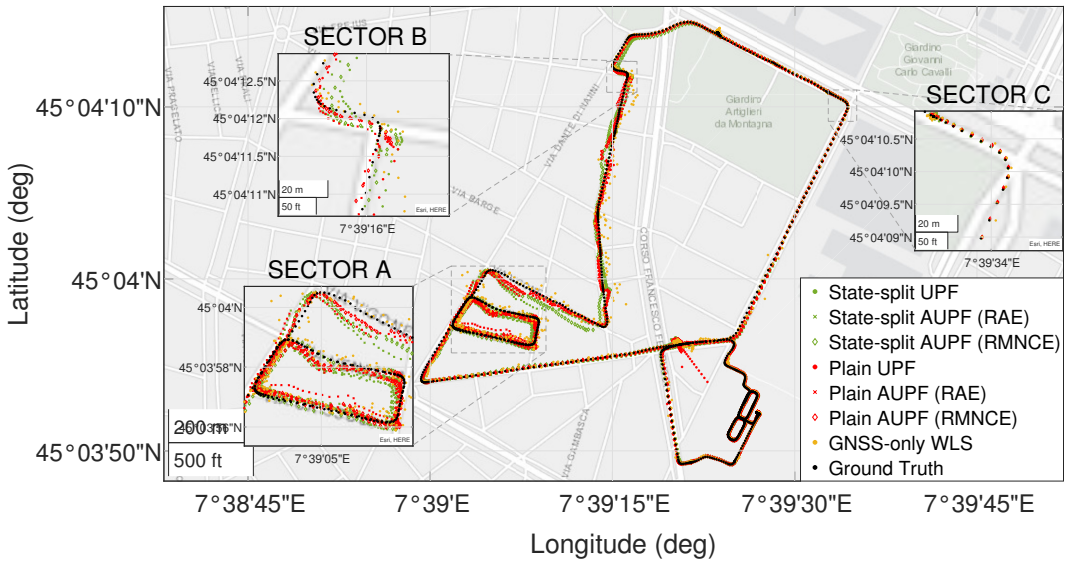


Figure 5.16: 2-D experimental trajectory in latitude/longitude (LLH) spherical coordinates. Comparison between plain and state-splitting UPF and AUPF (RAE and RMNCE based) architectures at 10^3 particles.

Starting from a first high-level comparative characterization, the estimated trajectories of the tracked vehicular target are provided in Figure 5.16.
Looking at Sector A, it is apparent that, on average, plain UPF and AUPF implementations supply fixes that, at each integration epoch, are tighter to the reference positions. It turns out that, in harsh navigation scenarios with heavy signal deterioration, the state-splitting configuration, irrespective of the potential adaptability in the measurement-noise covariance estimation, is particularly penalized w.r.t. the plain architecture. In fact, it tends to easily absorb a larger bias which is slowly neutralized over the set of consecutive epochs. The reason for such behaviour is inherently linked to the operating principle of the splitting method (Section 3.5.3), which optimizes particle weights towards a restricted set of state-components and limits the group of measurements determining them. More

explicitly, accounting for the body 3-D position states only, the particle weights exploited to estimate the position are exclusively conceived from pseudorange measurements and, as a matter of fact, multipath deteriorates more the quality of these observables than that of Doppler ones. Consequently, when navigating in highly reflective environments, particle weights of the state-splitting architecture cannot take advantage from a mixing of pseudorange and Doppler measurements (as it happens for plain implementations) which, hypothetically, might relax the impact of the considered impairment. Furthermore, similarly to Section 5.3.1, for the same AUPF architecture (plain or state-splitting), the RMNCE variance model, thanks to its approximate derivative operation in terms of FOSDs, better manages to relax multipath bias on ranging measurements, thus allowing a more accurate positioning solution w.r.t. the RAE scheme.
Shifting the attention on Sector B, the plain AUPF implementation still seems to offer better estimation accuracy over the state-splitting variant, and the RAE method tends to reduce the gap with the alternative RMNCE technique. The similarity of the trajectory trends between sectors A and B suggests two aspects. First, it is possible to extract a further assessment of the good IMU sensors quality which, even in conditions of higher vehicle dynamics, do not drift markedly. Secondly, the slight improvement of the RAE trajectory indicates a softening of multipath influence.
Eventually, in the mild-urban scenario of Sector C the trajectories are nearly matching each other and it is not possible to understand which architecture is gaining better performance. In other words, Sector C involves horizontal errors below 2 m (i.e. high accuracy) and an alternative methodology is necessary to refine the assessment.

In light of the former comments, it is useful to investigate on Figure 5.17 which illustrates, for the PF-based architectures treated so far, the CDF curves of the RMSE on the horizontal position in UTM coordinates.
Starting from the small error region (within 2 m), it is apparent that state-splitting implementations, both for the legacy UPF and the AUPF, outperform plain architectures, with an average accuracy gain around 10×10^{-2} m. As repeatedly remarked in former sections, it does not count so much the magnitude of such gain (which, by the way, is rather small) but the range of average horizontal errors where the improvement is measured. Indeed, 10×10^{-2} m of difference upon errors below 2 m, that is high-accuracy applications, can be regarded as a quite satisfactory enhancement. Moreover, for both plain and state-splitting AUPF architectures, comparing the adaptive observation variance models, it turns out that RAE overcomes the RMNCE performance. In particular, observing the 50-th percentile zoom in Figure 5.17, the RAE scheme improves the horizontal error of about 13 % w.r.t. the RMNCE paradigm.
The experimental evidences highlighted so far advantage the RAE approach when

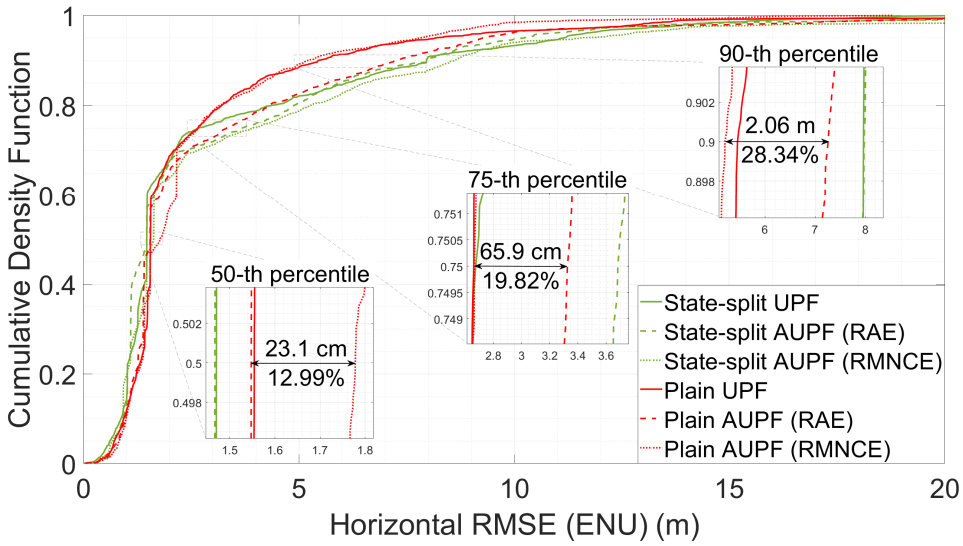


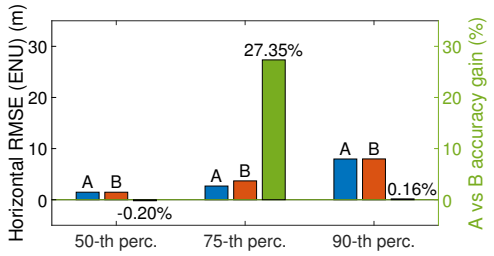
Figure 5.17: Empirical CDF (ECDF) of the RMSE on the horizontal position (E/N) in UTM-coordinates. Comparison between standard and state-splitting UPF architectures, implemented both with semi-static and adaptive noise covariance estimation, at 10^3 particles.

small errors are considered, that is when the input GNSS-based ranging measurements are affected by low noise and an almost null bias. In such scenario, the use of residuals (defined as the difference between the input measurements and the a-posteriori estimated measurements) and, in particular, the use of their covariance $\mathbf{C}_{\mathbf{r}_k}$ to evaluate the variances of input observables, is favourable in finely tracking the evolution of noise statistics.
Moving to higher percentiles, which means to consider a higher magnitude for the horizontal error and the impact of harsher impairments (e.g. multipath), the accuracy performance of state-splitting UPF and AUPF implementations definitely deteriorates, as well as it becomes blatant the trend inversion among the CDF curves associated to the adaptive schemes. As regards the latter aspect, reading at Table 5.4, the superiority of the RMNCE-based AUPF is clear and it gets further stressed for increasingly high percentiles. For example, considering the zoom of Figure 5.17 at 75-th percentile level, the plain AUPF architecture with RMNCE gains almost 20 % of accuracy over the corresponding RAE-based implementation; what's more, such improvement is even bigger at the 90-th percentile (i.e. average horizontal errors above 6 m) where it nearly touches 30 %. Once again, the foregoing results fix the strength and robustness of RMNCE in mitigating tough error sources and its resilience in keeping the maximum horizontal error bounded.

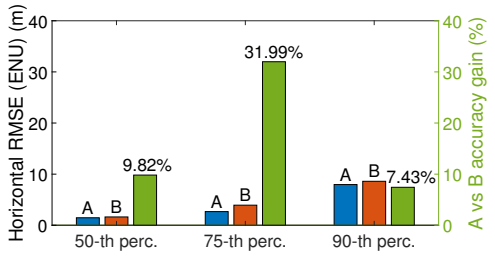
A further assessment of the involved Bayesian methodologies is provided in Figure 5.18, where the UPF and AUPF routines are compared in pairs to emphasize the mutual percentage accuracy improvements. First of all, Figures 5.18a and 5.18b indicate that, for state-splitting architectures, the use of adaptive paradigms for the measurement noise covariance estimation is actually not advantageous. In fact, such result should be particularly read for low percentiles where, in the high accuracy CDF region, it is well established the finer performance of state-splitting implementations over plain ones. On the contrary, accounting for Figures 5.18c and 5.18d, the frame for plain filtering versions seems more garbled. In particular, considering the plain AUPF with RAE scheme, it gains a little for small horizontal errors while its performance is definitely worse at high percentiles. Potentially, by extending the width of the residuals buffer for the construction of the covariance matrix $\mathbf{R}_k$, it might be possible to increase the percentage improvement in the high-accuracy region. For the RMNCE-based AUPF solution, instead, its employment is suggested only for high percentiles or, equivalently, for large horizontal errors. To tell it differently, the use of such adaptive approach could make sense when the tracked moving target is travelling a challenging environment. Anyway, in the 90-th percentile region, where the horizontal error magnitude is above 6 m (Table 5.4), the increase of computational burden induced by the adoption of RMNCE is certainly not upheld by such a moderate accuracy gain of 4.6 %.
After that, Figure 5.18f highlights that, for larger horizontal errors, the adaptive RMNCE approach is superior than RAE and the accuracy performance divergence between the two methods augments proportionally to the considered percentile (i.e. proportionally to an increase of the RMSE value). Eventually, it turns out that the percentage improvement, w.r.t. the state-splitting version, brought about by the RMNCE paradigm implemented on a plain AUPF architecture is considerably large (up to 40 %, looking at Figure 5.15h).

Table 5.4: Horizontal (E/N) position RMSE measured, for both UPF and AUPF (plain and state-splitting) architectures, at 50-th, 75-th and 90-th percentiles. A fixed number of 10^3 particles is considered.

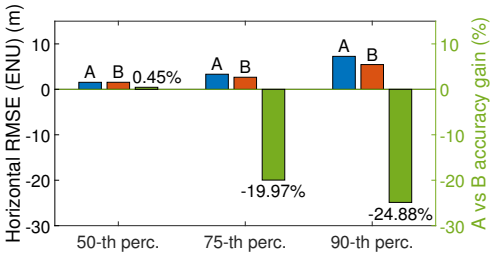
UPF architecture	50-th percentile Error (m)	75-th percentile Error (m)	90-th percentile Error (m)
State-splitting UPF	1.470	2.675	7.964
State-splitting AUPF (RAE)	1.467	3.682	7.977
State-splitting AUPF (RMNCE)	1.630	3.933	8.603
Plain UPF	1.554	2.661	5.452
Plain AUPF (RAE)	1.547	3.325	7.258
Plain AUPF (RMNCE)	1.778	2.666	5.201



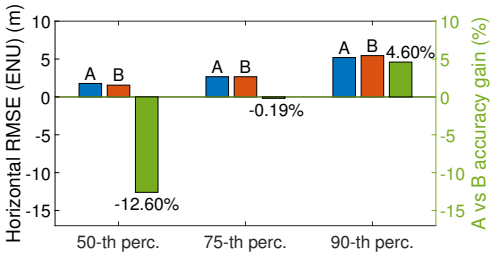
(a) State-splitting UPF (A) & State-splitting AUPF (RAE) (B).



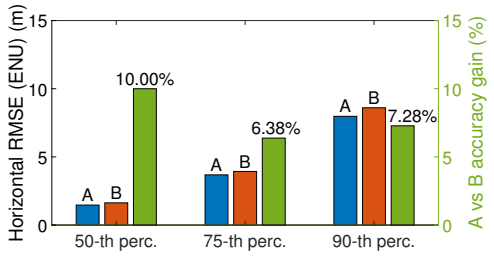
(b) State-splitting UPF (A) & State-splitting AUPF (RMNCE) (B).



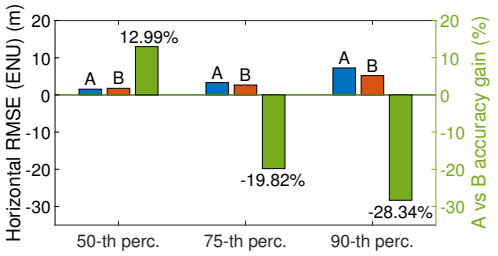
(c) Plain AUPF (RAE) (A) & Plain UPF (B).



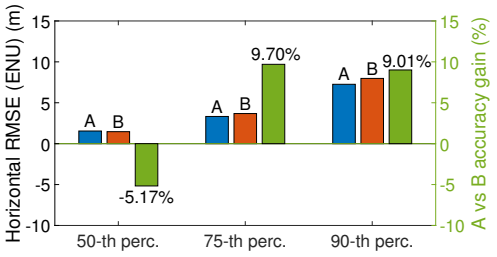
(d) Plain AUPF (RMNCE) (A) & Plain UPF (B).



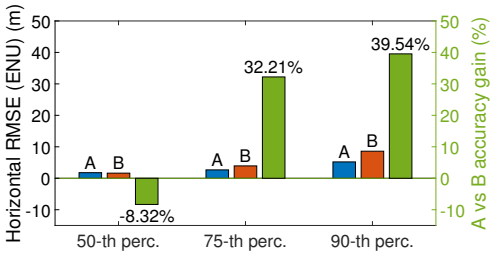
(e) State-splitting AUPF (RAE) (A) & State-splitting AUPF (RMNCE) (B).



(f) Plain AUPF (RAE) (A) & Plain AUPF (RMNCE) (B).



(g) Plain AUPF (RAE) (A) & State-splitting AUPF (RAE) (B).



(h) Plain AUPF (RMNCE) (A) & State-splitting AUPF (RMNCE) (B).

Figure 5.18: One-to-one comparison, in terms of percentage gain/loss at multiple percentiles, among different couples of plain and state-splitting UPF and AUPF architectures. A fixed number of 10^3 particles is considered.

5.3.3 Adaptive Kalman Filters vs AUPF

After characterizing separately the accuracy performance of both legacy and adaptive Kalman filters and Unscented Particle filters, this section combines them transversally to ultimately identify the most convenient statistical integration scheme. In the framework of Kalman solutions, both EKF and the more advanced UKF are considered, while, in the UPF context, the same algorithmic strategies assessed in previous Section 5.3.2 are treated in the following (the number of particles is still kept equal to 10^3). Before entering the analysis, it is right to premise that, contrarily to previous sections, the experimental vehicle trajectories are not shown here, mainly because of the large number of compared hybridization routines. The first adopted comparative methodology relies on Figure 5.19, which

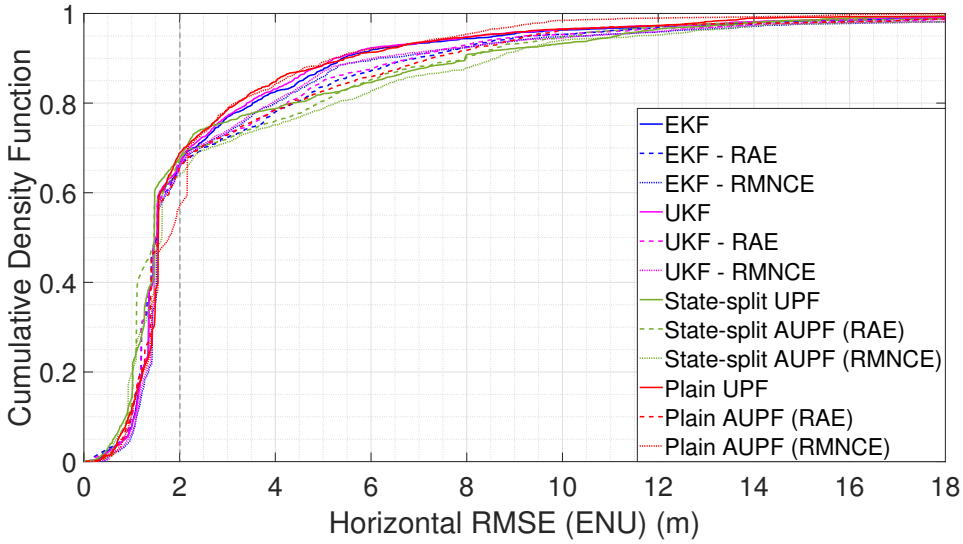


Figure 5.19: Empirical CDF (ECDF) of the RMSE on the horizontal position (E/N) in UTM-coordinates. Comparison between legacy and adaptive EKF, UKF and UPF architectures (plain and state-splitting implementations).

incorporates, at a glance, all the multiple CDF curves on the horizontal RMSE whose behaviour is explored. For sake of clarity, it has been chosen to divide it into two segments, by isolating the low error region (below 2 m) from the remainder part. Starting form the high-accuracy slice, the reader is invited to refer to Figure 5.20, which provides a zoom on Figure 5.19. As largely expected on the basis of the experimental outcomes observed in former analysis, state-splitting PF-based architectures prevail over the other filtering solutions. This behaviour is true since very small errors around 50×10^{-2} m but it becomes definitely stressed above 1 m. Interestingly, above 50×10^{-2} m, the state-splitting UPF architecture manifests a

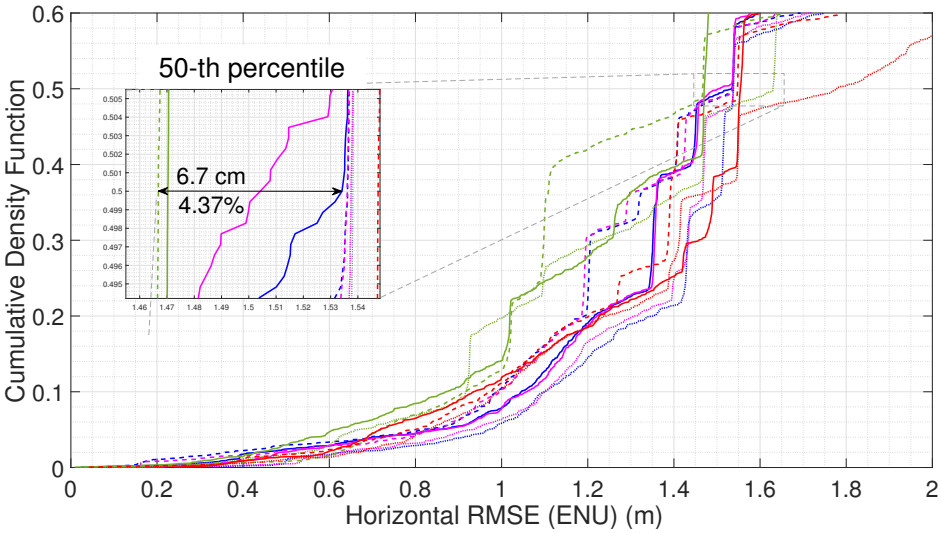


Figure 5.20: Zoom of Figure 5.19 for horizontal RMSE values within 2 m.

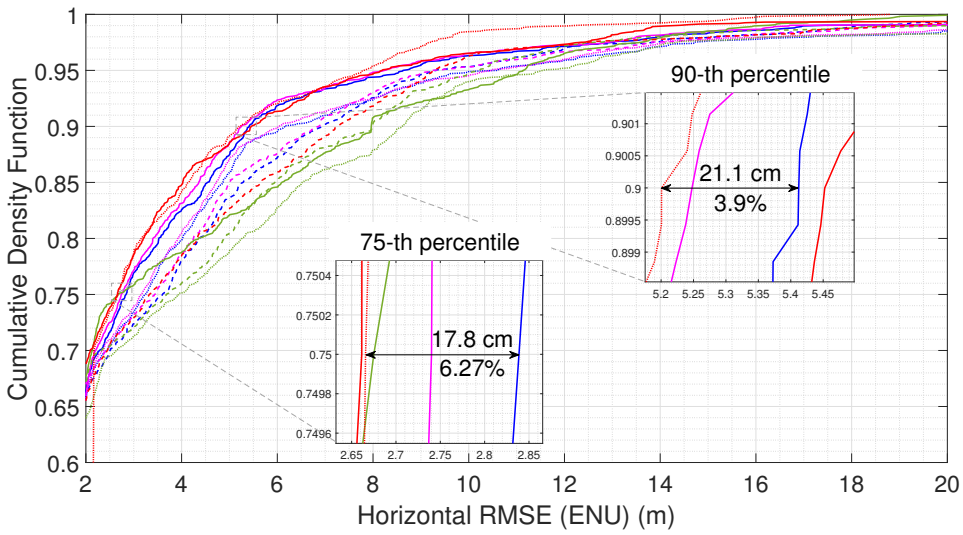


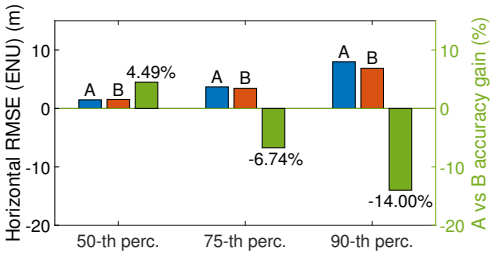
Figure 5.21: Zoom of Figure 5.19 for horizontal RMSE above 2 m.

greater performance, but around the metre, the AUPF with RAE scheme comes out and enhances the accuracy over the other filters up to 25×10^{-2} m. Anyway, this contrasting course reabsorbs towards 1.5 m and, at the 50-th percentile, state-splitting UPF and RAE-based AUPF perform equivalently. Moreover, at the same percentile, the state-splitting UPF improves the horizontal accuracy by a 4.37 % factor w.r.t. the legacy EKF, and this margin gets narrower when referring to the

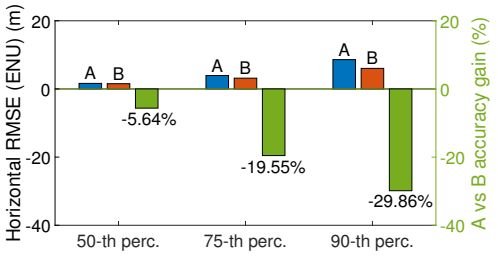
UKF.
The former evidences, once more, establish the prominent performance in the high-accuracy region of the state-splitting method, which takes advantage from the finer weighting scheme applied to the set of randomly generated particles (Section 3.5.3). It is worth highlighting that, all the involved outcomes, are determining a scientifically-relevant, preliminary validation, since they result from testing the integration algorithms on real data, which have been collected from a field test and are not synthetically created.
Still observing the low-error segment, regardless of the selected Bayesian stage, it is plain to see that the average accuracy of the RMNCE paradigm is inferior than RAE. In fact, when measurement noise is weak, the use of residuals, in place of FOSDs, brings the fusion filter with more relevant pieces of information which are helpful to faithfully infer observables' variances.

Shifting the attention on larger RMSE error values (Figure 5.21), two aspects emerge. On one hand, the accuracy performance of state-splitting UPF and AUPF implementations degrades significantly, with an average loss of more than 1.5 m over corresponding plain architectures. For example, at the 85-th percentile, the error committed by the state-splitting AUPF with RMNCE is 2.4 m bigger than that committed by the corresponding plain filter version (36.3 % loss). At further high percentiles, the loss gets even bigger touching the 40 % threshold towards 90-th percentile level. On the other hand, for horizontal errors above 4 m, the plain RMNCE-based AUPF architecture slightly improves its accuracy w.r.t. lower percentiles. In particular, at the 75-th level, it is measured a modest 6.27 % improvement over the legacy EKF, which is however nearly halved at higher error levels.
Overall, in the low-precision region, the outperforming filters are the UPF and the RMNCE-based AUPF, both taken with plain implementations. Compared each other, they can be regarded as equivalent, being the mutual error differences almost irrelevant. However, in terms of complexity, the gap between these architectures is consistent, hence not being advisable to augment the Bayesian structure applying an adaptive strategy.

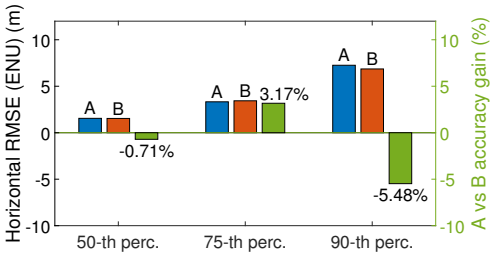
An alternative viewpoint, more strictly oriented to assess RMNCE and RAE noise covariance estimation strategies, is detailed in Figure 5.22, where few couples of adaptive Bayesian schemes are compared based on the percentage horizontal accuracy metric. In particular, in the framework of Kalman methods, it is considered only the UKF since, for almost all percentiles, it tends to slightly improve the performance over the original EKF.



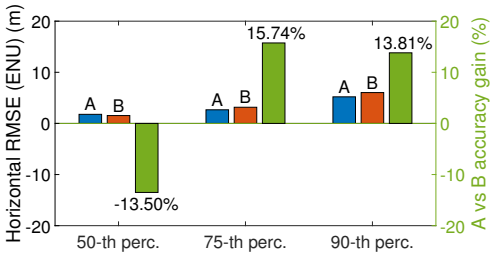
(a) State-splitting AUPF (RAE) (A) & Adaptive UKF (RAE) (B).



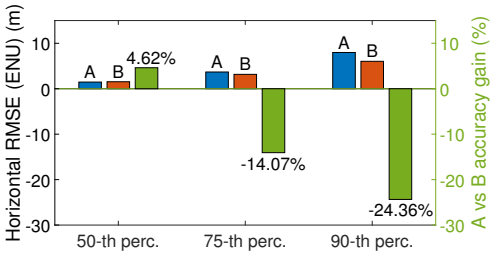
(b) State-splitting AUPF (RMNCE) (A) & Adaptive UKF (RMNCE) (B).



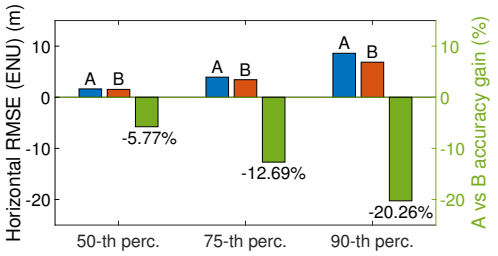
(c) Plain AUPF (RAE) (A) & Adaptive UKF (RAE) (B).



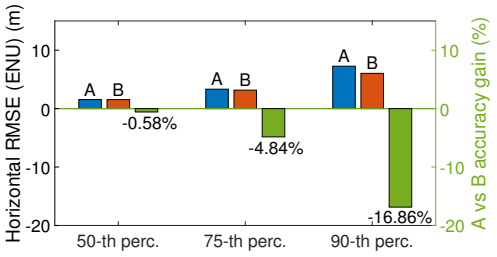
(d) Plain AUPF (RMNCE) (A) & Adaptive UKF (RMNCE) (B).



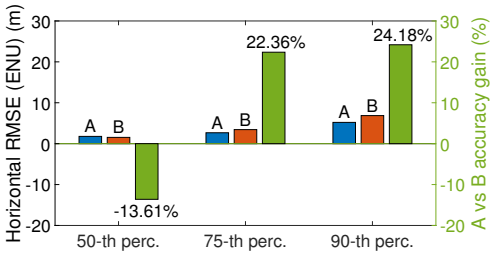
(e) State-splitting AUPF (RAE) (A) & Adaptive UKF (RMNCE) (B).



(f) State-splitting AUPF (RMNCE) (A) & Adaptive UKF (RAE) (B).



(g) Plain AUPF (RAE) (A) & Adaptive UKF (RMNCE) (B).



(h) Plain AUPF (RMNCE) (A) & Adaptive UKF (RAE) (B).

Figure 5.22: One-to-one comparison among different couples of adaptive (RAE and RMNCE) UKF and AUPF (10^3 particles) architectures, in terms of percentage accuracy gain (or loss) at different percentiles.

Given so, Figures 5.22a and 5.22b offer interesting comparisons between state-splitting AUPFs and adaptive UKFs. Evidently, the results are globally unfavourable for the former filtering solutions with consistent accuracy losses (especially for the RMNCE-based AUPF at high percentiles) w.r.t. the adaptive Kalman estimator. However, a moderate improvement of 4.49 % is measured for the AUPF with RAE strategy at 50-th percentile level (Figure 5.22a). These results underline, just from a different perspective, the good accuracy behaviour granted by the state-splitting methodology for high-accuracy estimation, as well as they emphasize its average performance degradation for larger RMSE.
Another meaningful trend pops up in Figure 5.22d, where a plain AUPF filtering scheme with RMNCE is compared against an adaptive UKF employing the same covariance estimation strategy. Interestingly, for higher percentiles, the percentage accuracy gain is respectable and, considering the measured horizontal errors specified in Table 5.5, it translates into 49.8×10^{-2} m (75-th percentile) and 83.3×10^{-2} m (90-th percentile) of difference in terms of distance of the estimated trajectory fixes from the ground-truth. Such outcome explains, with an alternative indicator, the solid performance of the RMNCE under harsh navigation conditions with stronger multipath effects.
A further hint on this aspect can be extracted from Figure 5.22e, where the RAE-based state-splitting AUPF gains about 7×10^{-2} m of estimation accuracy over the UKF with RMNCE (Table 5.5). At higher percentiles, then, the performance turnaround is blatant. Anyway, the last comments, together with the percentage levels specified in Figure 5.22g, are relatively weak and less significant than others, since both the state-splitting and the plain AUPF variants with RAE demonstrate, on average, to supply position estimates which are quite erroneous if compared to reference fixes.

Final remarks

In this Section it has been offered an extensive assessment of the accuracy performance of adaptive strategies for measurement noise covariance, both in the framework of Kalman architectures and in the UPF context. Referring to the former class of filters, it is definitely not advisable the employment neither of the RAE scheme nor of the RMNCE method within the UKF and EKF estimation structures (refer also to Table 5.3), unless additional signal processing stages are introduced (e.g. the pseudorange pre-processing stage analyzed in the following section). For the UPF paradigm, instead, the review is trickier. For small percentiles, in the high accuracy region of the error CDF, the RAE-based state-splitting AUPF, taking well behaving filtering solutions as terms of comparison, gains 5.17 % of accuracy w.r.t. the equivalent plain implementation (Figure 5.18g) and a smaller 2.46 % w.r.t. the legacy UKF fusion algorithm. In fact, these are rather small improvements that,

however, might become significant when accounting for horizontal errors below 1.5 m. Nevertheless, the selection of such adaptive PF-based architecture must be weighted by the complexity enhancement induced by the RAE-routine, which varies proportionally to the residuals' buffer size. Moving to higher RMSE values, it has been commented already that the RMNCE-based AUPF improves the accuracy by few centimetres over the UPF, when both are taken in their legacy versions (i.e. plain implementations). Consequently, the increase of complexity brought about by the adaptability in the covariance estimate is not adequately rewarded by such a limited performance enhancement, especially if scaled to the considered horizontal error levels.

Table 5.5: Horizontal (E/N) position RMSE measured for adaptive EKF, UKF and AUPF (plain and state-splitting) architectures at 50-th, 75-th and 90-th percentiles. A fixed number of 10^3 particles is considered.

Bayesian architecture	50-th percentile Error (m)	75-th percentile Error (m)	90-th percentile Error (m)
Adaptive EKF (RAE)	1.536	3.601	6.941
Adaptive EKF (RMNCE)	1.538	3.248	6.155
Adaptive UKF (RAE)	1.536	3.434	6.860
Adaptive UKF (RMNCE)	1.538	3.164	6.034
State-splitting AUPF (RAE)	1.467	3.682	7.977
State-splitting AUPF (RMNCE)	1.630	3.933	8.603
Plain AUPF (RAE)	1.547	3.325	7.258
Plain AUPF (RMNCE)	1.778	2.666	5.201

5.4 Legacy and Adaptive Bayesian filters with pre-processing

At the beginning of this chapter, in Section 5.2, the legacy Bayesian architectures described in Chapter 3 have been assessed in their accuracy performance, when implemented with a C/N_0 based approach involving measurement-noise covariance $\mathbf{R}_k$ estimation. In particular, for both Monte-Carlo and Kalman integration strategies, the analysis has been conducted by testing the INS/GNSS TC system on a dataset involving real inertial and GPS observables. Later, in Section 5.3, the former fusion methodologies have been characterized when augmented with adaptive noise-variance modelling paradigms, such as the RAE and RMNCE algorithms presented in Sections 4.1.3 and 4.1.4, respectively. In the remainder of this chapter, then, the aforementioned integrated filtering solutions are further expanded by including the self-contained pseudorange pre-processing module proposed in Section 4.2. For the assessment, it is preserved the same methodology of previous sections, involving experimental rating based on the CDFs of the horizontal position RMSE. Furthermore, analogously to the foregoing analysis, results are presented as a three-stage path. First, the attention is conveyed to Kalman filters only; then, the focus is transferred onto the UPF estimation pattern; eventually, the relevantly performing filtering stages, identified through the two former steps, are mutually compared, seeking for the identification of the most advantageous architecture to be applied to INS/GNSS sensor fusion.

5.4.1 Pseudorange pre-processing in Kalman filters

In the framework of Kalman estimators, both legacy and adaptive implementations are accounted for, in the view of assessing the potential benefits, in terms of estimation accuracy and filtering robustness, induced by pre-processing pseudorange data prior to integrate them within the state-space observation model. To avoid increasing too much the number of compared Bayesian routines, adaptive architectures without measurement pre-processing (Section 5.3.1) are neglected hereinafter, as they were found poorly behaving w.r.t. legacy EKF and UKF algorithms.

Figure 5.23 shows up the empirical CDF curves for the RMSE on the horizontal position components (E/N) in UTM coordinates.
Starting from the high-precision segment (Figure 5.24), where the involved horizontal errors stand below 2 m, the accuracy performance of the RAE-based adaptive Kalman implementations with pre-processing module stands out over the other filtering variants. This behaviour starts becoming particularly evident towards the metre, and it is highly marked in the range from 1 m to 1.4 m. For example, at

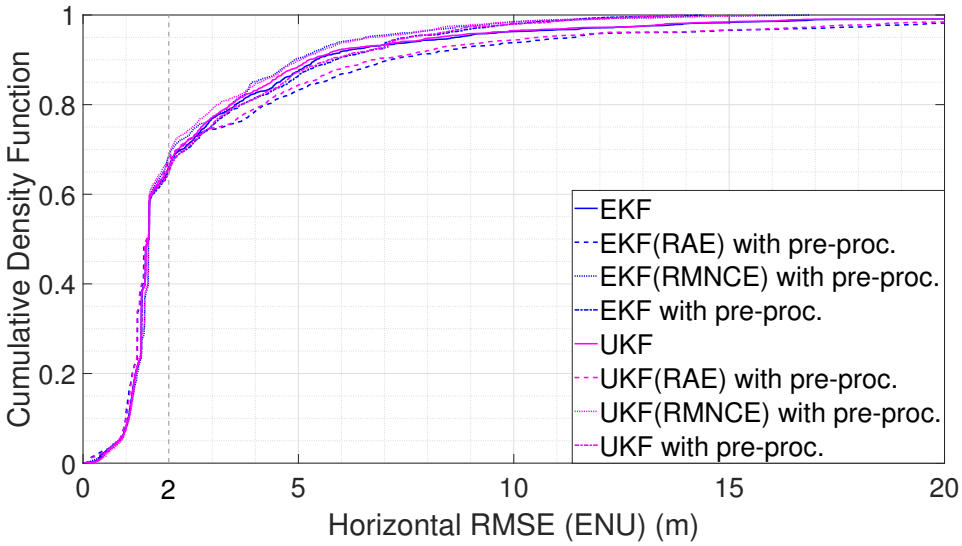


Figure 5.23: Empirical CDF (ECDF) of the RMSE on the horizontal position (E/N) in UTM-coordinates. Comparison between legacy and adaptive EKF and UKF implementations with pseudorange pre-processing module.

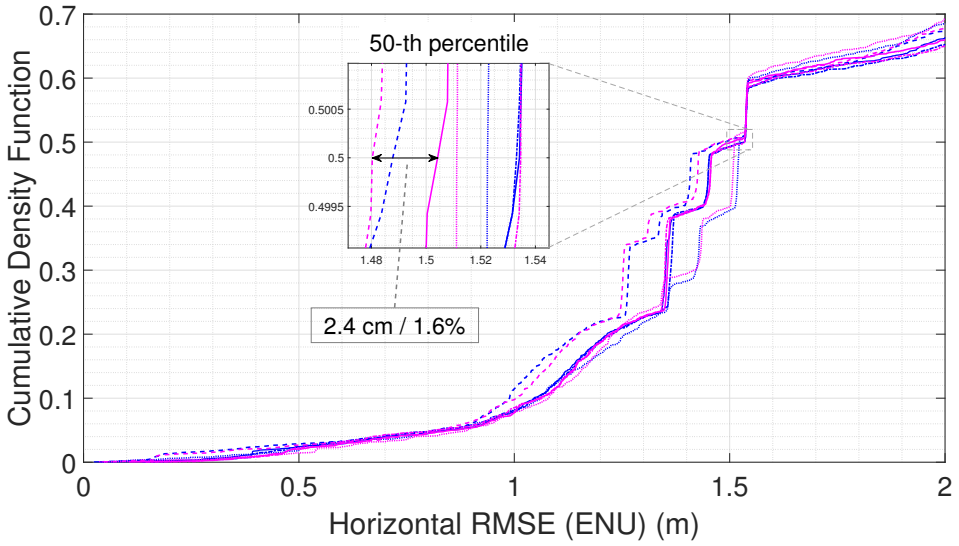


Figure 5.24: Zoom of Figure 5.23 for horizontal RMSE values below 2 m.

the 30-th percentile level, the measured relative distance between the estimated position fixes from the adaptive RAE-based UKF with pre-filtering and the legacy UKF amounts to 10.2×10^{-2} m, which corresponds to a 7.5 % accuracy gain of the former solution over the latter. Then, increasing the horizontal error towards 1.5 m,

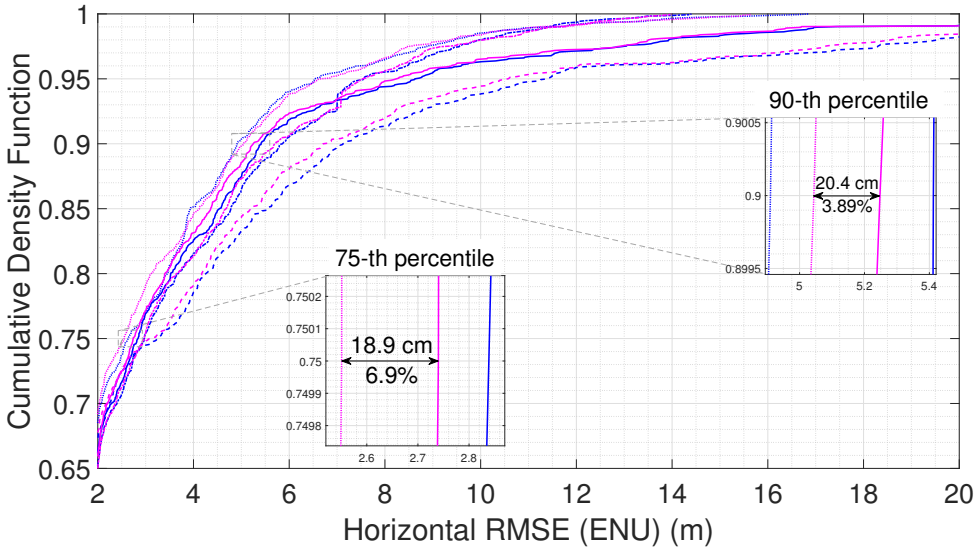


Figure 5.25: Zoom of Figure 5.23 for horizontal RMSE values above 2 m.

the gap narrows and, at the 50-th percentile level (zoom in Figure 5.24), it equals 2.4×10^{-2} m, which corresponds to a minimal accuracy improvement of 1.6 %. In parallel, still considering small errors, the average accuracy profile of the adaptive-RMNCE Kalman variant with pre-processing is the worst. However, from 1.6 m on (about 60-th percentile), the CDF trend overturns. Overall, for high-accuracy, the RAE paradigm jointly combined with pseudorange pre-processing seems to improve the performance over the legacy EKF and UKF architectures, but a single-dataset assessment prevents from stating whether such adaptive implementation is definitely advisable.

Moving to higher horizontal errors (Figure 5.25), perfectly in agreement with former Section 5.3.1, the adaptive RAE scheme suffers performance deterioration induced by multipath. On the other hand, the mixing between the RMNCE paradigm exploiting FOSDs, which calculate an approximate first-order derivative on a buffer of noisy pseudorange measurements, and the measurement pre-processing algorithm strengthen the estimation capability of the integrated navigation system. In fact, the accuracy behaviour of the resulting adaptive filtering architecture is better than that of legacy Kalman routines and, even for high percentiles, the maximum horizontal error is bounded to acceptable values. More in depth, Figure 5.25 highlights the mutual differences in terms of RMSE on the horizontal position fixes. At the 75-th percentile level, which corresponds to horizontal misalignments in a range between 2 m and 3 m, the RMNCE-based UKF with pre-processing commits an error, w.r.t. the ground-truth, which is 18.9×10^{-2} m smaller than the

error measured for the legacy architecture, hence achieving a moderate accuracy gain of 6.9 %. Moreover, at the higher 90-th percentile (error range between 5 m and 6 m), the error difference is slightly increased (20.4×10^{-2} m) but the percentage improvement is smaller (3.89 %).
Another interesting aspect, which anyway considers the entire error-CDF at all percentiles, concerns the legacy EKF and UKF implementations when augmented with pre-processing. In fact, no relevant performance variations are triggered, measuring both negligible percentage accuracy improvements in some error ranges and minor accuracy losses (few centimetres) in other CDF segments. In light of the above, it seems that, at least for Kalman filters, pseudorange pre-processing impacts more incisively on an estimation performance enhancement when operated jointly with adaptive noise-covariance estimation strategies. Anyway, the choice of the most feasible architecture, which balances at best accuracy and complexity, is nearly unaffected by the pre-processing scheme, since it infinitesimally impacts on the sensor fusion algorithm computational complexity.

Table 5.6 specifies the measured horizontal RMSE for the aforementioned Kalman strategies. It is interesting to compare it against Table 5.3, which summarizes the same quantities for the adaptive filtering schemes without pre-processing. The accuracy improvement determined by the introduction of such strategy is rather apparent. For instance, focusing on the adaptive RAE scheme applied to the UKF, the insertion of pre-processing causes a reduction of the horizontal error by 5.6×10^{-2} m (3.6 % accuracy gain) at the 50-th percentile (low RMSE region). As another example, considering the RMNCE-based adaptive EKF at the 90-th percentile, the horizontal error contraction made possible by pre-filtering amounts to 1.25 m, which translates in a non-negligible percentage enhancement of 20.23 %. More generally, according to a rule-of-thumb evaluation, pre-processing within the adaptive RMNCE Kalman fusion structure mitigates the accuracy loss of about 60×10^{-2} m at the 75-th percentile and of about 1 m at the 90-th level, but the effect is irrelevant on the low-error CDF slice (high-accuracy). On the contrary, the application of pre-processing within the adaptive RAE Kalman structure attenuates by about 5×10^{-2} m the error in the high-accuracy region of the CDF, but it turns out to be quite ineffective at higher percentiles.

5.4.2 UPF and pseudorange pre-processing

Having characterized the impact of pseudorange pre-processing in the Kalman framework, this section analyzes plain and state-splitting AUPF implementations (10^3 particles), which are compared each other to investigate the potential improvements deriving from the application of the pre-processing stage.

Table 5.6: Horizontal (E/N) position RMSE measured for legacy and adaptive Kalman architectures with pseudorange pre-processing at 50-th, 75-th and 90-th percentiles.

Kalman architecture	50-th percentile Error (m)	75-th percentile Error (m)	90-th percentile Error (m)
Legacy EKF	1.534	2.839	5.412
Adaptive EKF (RAE) with pre-proc.	1.488	3.240	7.123
Adaptive EKF (RMNCE) with pre-proc.	1.522	2.634	4.910
Legacy EKF with pre-proc.	1.533	2.967	5.858
Legacy UKF	1.504	2.740	5.248
Adaptive UKF (RAE) with pre-proc.	1.480	3.084	6.710
Adaptive UKF (RMNCE) with pre-proc.	1.511	2.551	5.044
Legacy UKF with pre-proc.	1.535	2.970	5.706

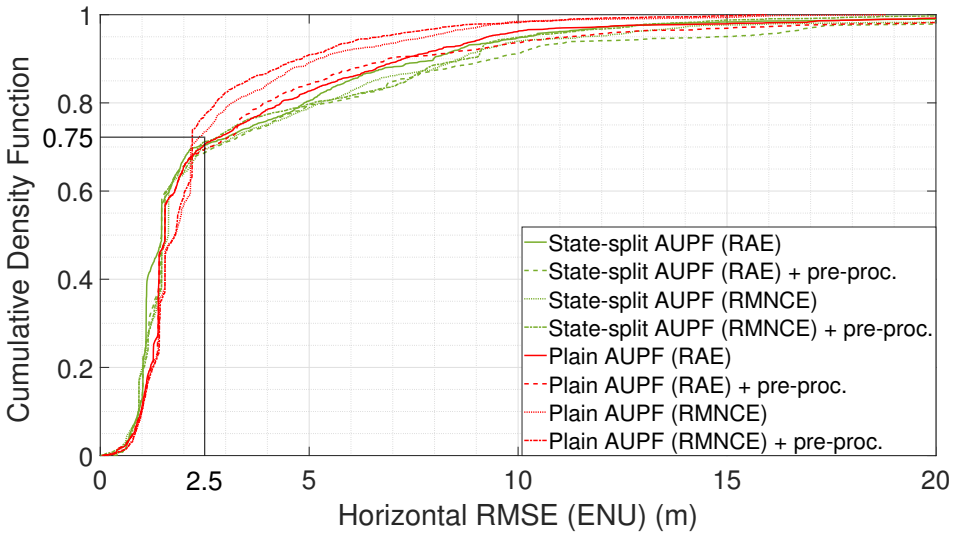


Figure 5.26: Empirical CDF (ECDF) of the RMSE on the horizontal position (E/N) in UTM-coordinates. Comparison between AUPF implementations with and without pre-processing module.

The starting point of the assessment targets Figure 5.26, which presents the CDF lines of the RMSE on the horizontal position components associated to the tracked vehicular target.

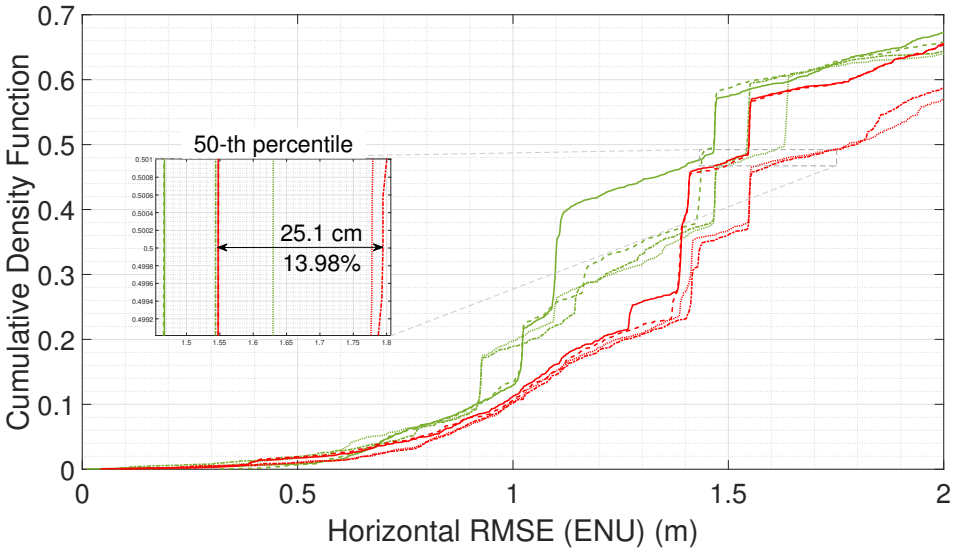


Figure 5.27: Zoom of Figure 5.26 for horizontal RMSE values below 2 m.

Looking first at Figure 5.27, where the low horizontal error region is zoomed in, the average accuracy performance of the state-splitting AUPF architecture is superior than the plain implementation. Interestingly, the best behaving scheme is represented by the RAE-based AUPF variant without pseudorange pre-processing stage. For such fusion strategy, the percentage accuracy gap when using or not pre-filtering is rather wide, especially for errors included between 1 m and 1.5 m. For example, at the 40-th percentile level, the state-splitting AUPF with RAE gains 21.2 % accuracy over the same architecture just augmented with pseudorange pre-processing. The latter percentage improvement translates into 30×10^{-2} m of difference in terms of average distance of the estimated fixes from the ground-truth positions. However, for slightly increasing horizontal RMSE, the separation between CDF curves reduces, as well as it decreases the accuracy penalty of plain AUPF routines versus state-splitting ones. In particular, as highlighted in the zoom at the 50-th percentile level, the measured accuracy loss affecting the plain AUPF with RMNCE and pre-processing, w.r.t. the equivalent state-splitting version, equals 13.98 %. Anyway, such error trend is more or less maintained up to 2 m, where a remarkable performance inversion between plain and state-splitting AUPF paradigms takes place, as it will be discussed shortly. Observing further Figure 5.27, another relevant aspect can be grasped, dealing with the impact of pseudorange pre-processing in the high-accuracy region. Plainly, the CDFs associated to the same Bayesian structure, which only differ for the presence of the data pre-filtering module, are tightly evolving and match to a large extent, especially for plain implementations, hence signalling that the former methodology is not crashing.

The reason is that, by its nature, pre-processing targets to relax the spiky behaviour affecting noisy code-ranging measurements, whose major responsible is multipath. In other words, pre-processing aims at enhancing the accuracy and robustness of the integrated solution when the fusion algorithm is involved with highly deteriorated signal measurements due to a challenging physical environment. Obviously, such operating condition is not mirrored by the high-precision CDF segment, but rather it is embodied in the low-accuracy portion, with horizontal errors well greater than 2 m.

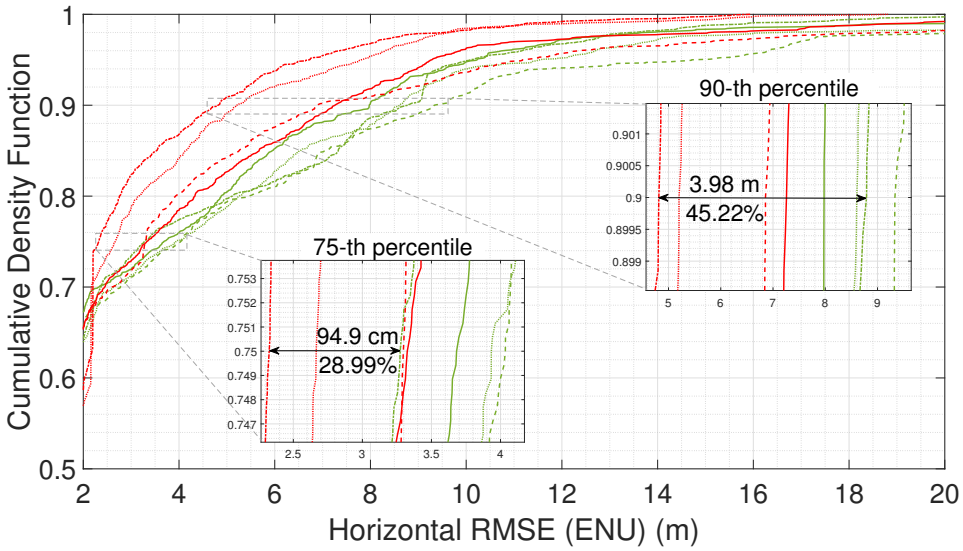


Figure 5.28: Zoom of Figure 5.26 for horizontal RMSE values above 2 m.

Figure 5.28 shows up the remainder portion of the involved CDF curves for values of the horizontal RMSE between 2 m and 20 m. As anticipated in previous paragraph, pseudorange pre-processing is primarily meant to enhance the integrated system performance under critical signal environments, where multipath bias injections are strong and provoke a degradation of the filter estimation accuracy. Blatantly, the pre-filtering scheme positively impacts in the medium-to-high error region, by determining an accuracy enhancement over all the other architectures. In particular, the effect is marked on plain AUPF implementations, while state-splitting ones, in line with previous results, suffer considerable performance loss when multipath gets heavier. Concerning the plain AUPF with RMNCE, its outperformance over the other UPF and AUPF architectures was already outlined in Figure 5.21, but it was also emphasized its infeasibility due to a non-advantageous cost-complexity trade-off. Here, the expansion of the former structure, by introducing a low-complexity pre-processing, increases the accuracy gain further. The latter statement can be

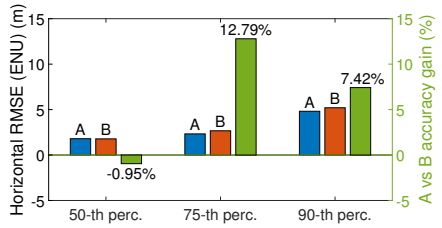
better understood by referring to Figure 5.29a, which compares the plain RMNCE-based AUPF with pseudorange pre-processing against the analogous version without pre-processing. At the 75-th percentile, the additional extra-gain granted by pre-filtering amounts to 12.79 %, which corresponds to an average horizontal error which is reduced by 34.1×10^{-2} m. Then, at the higher 90-th percentile, the average error cut is slightly higher (38.9×10^{-2} m) but the percentage enhancement smaller (7.42 %).
Another interesting comparison is caught through the zooms of Figure 5.28, where the plain AUPF architecture with RMNCE and pre-processing is compared against the alternative state-splitting solution. In such case, the accuracy gap is much wider. At the 75-th level, the former fusion scheme achieves an average improvement in the estimation of horizontal position fixes by 94.9×10^{-2} m (28.99 %) while, at the 90-th percentile, the accuracy boost is huge (3.98 m, corresponding to 45.22 %).

To achieve a deeper insight on the assessment, the reader is invited to take a look at Figure 5.18. In particular, it is interesting to jointly comment on Figures 5.29b, 5.29c and 5.29d. Picking the plain AUPF implementation with RMNCE observation variance modelling and code-ranging pre-processing, it always shows sub-optimal estimation performance for horizontal RMSE below 2 m. In particular, at the 50-th percentile, the estimated position fixes are, on average, shifted from the ground-truth 30×10^{-2} m more than those produced by alternative AUPF schemes implementing the pre-filtering strategy. Consequently, if the application target requires very-high accuracy and the physical environment is favourable, the former Bayesian architecture might not be recommended. However, if higher percentiles are accounted for (i.e. higher RMSE values), the accuracy gain for the same architecture is definitely not questionable and it even increases with the horizontal error. As an example, in Figure 5.29d, the 90-th percentile improvement of the RMNCE-based AUPF with pre-processing amounts to 48.4 %, which signifies an horizontal error contraction by 4.52 m. To be honest, the latter consistent error reduction results from the combination of two effects: the performance degradation of the state-splitting paradigm under multipath signal environments, on one hand, and the remarkable accuracy improvement and error bounding granted by the same adaptive filter when augmented with pre-processing.
In Figure 5.29h, the state-splitting AUPF with RAE and pseudorange pre-filtering is compared to the equivalent RMNCE-based AUPF variant. In the high-accuracy CDF region, the RAE paradigm tends to prevail, with an average accuracy improvement of 5.05 %, which is moderate but relevant for small RMSE magnitude. On the contrary, at higher percentiles, especially at the 75-th level, the trend reverses. Eventually, taking into account Figures 5.29f and 5.29g at high percentiles, the accuracy loss of the state-splitting AUPF (RMNCE) with pre-filtering w.r.t. a plain AUPF with RAE is not as heavily stressed as for the cases diagrammed

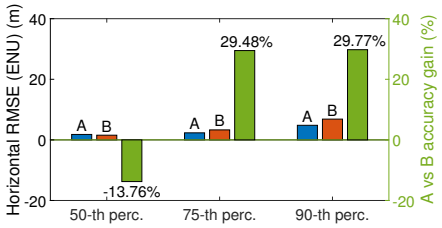
in Figures 5.29c and 5.29d. Two main reasons explain such difference: first of all, for higher horizontal errors above 2 m, the RAE scheme is weaker than RMNCE in faithfully characterizing the variance of noisy pseudorange measurements potentially impacted by multipath; secondly, the RMNCE technique joined to a state-splitting architecture contributes in mitigating the performance deterioration in the low-accuracy CDF segment.

Table 5.7: Horizontal (E/N) position RMSE measured for plain and state-splitting AUPF architectures (10^3 particles) with/without pseudorange pre-processing, at 50-th, 75-th and 90-th percentiles.

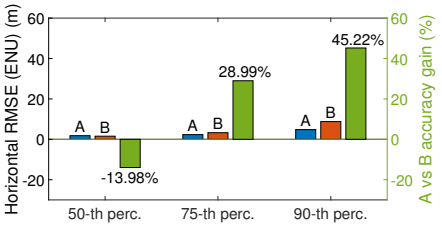
UPF architecture	50-th percentile Error (m)	75-th percentile Error (m)	90-th percentile Error (m)
State-splitting AUPF (RAE)	1.467	3.682	7.977
State-splitting AUPF (RAE) with pre-proc.	1.466	4.035	9.332
State-splitting AUPF (RMNCE)	1.630	3.933	8.603
State-splitting AUPF (RMNCE) with pre-proc.	1.544	3.274	8.790
Plain AUPF (RAE)	1.547	3.325	7.258
Plain AUPF (RAE) with pre-proc.	1.548	3.297	6.856
Plain AUPF (RMNCE)	1.778	2.666	5.201
Plain AUPF (RMNCE) with pre-proc.	1.795	2.325	4.815



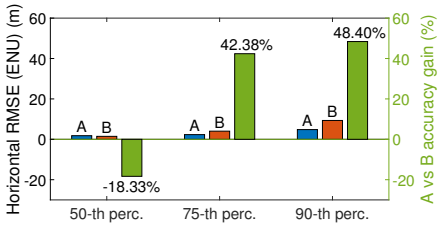
(a) Plain AUPF (RMNCE+pre-proc.) (A) & Plain AUPF (RMNCE) (B).



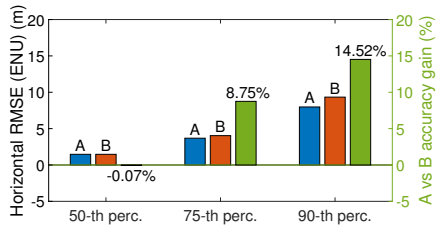
(b) Plain AUPF (RMNCE+pre-proc.) (A) & Plain AUPF (RAE+pre-proc.) (B).



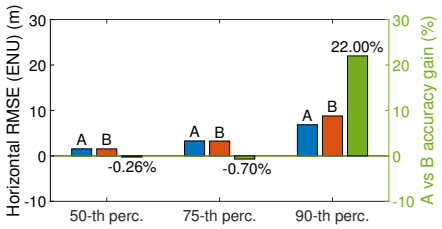
(c) Plain AUPF (RMNCE+pre-proc.) (A) & state-splitting AUPF (RMNCE+pre-proc.) (B).



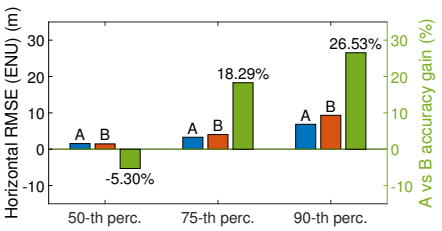
(d) Plain AUPF (RMNCE+pre-proc.) (A) & state-splitting AUPF (RAE+pre-proc.) (B).



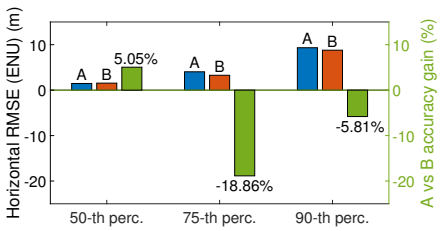
(e) State-splitting AUPF (RAE) (A) & state-splitting AUPF (RAE+pre-proc.) (B).



(f) Plain AUPF (RAE+pre-proc.) (A) & state-splitting AUPF (RMNCE+pre-proc.) (B).



(g) Plain AUPF (RAE+pre-proc.) (A) & state-splitting AUPF (RAE+pre-proc.) (B).



(h) State-splitting AUPF (RAE+pre-proc.) (A) & State-splitting AUPF (RMNCE+pre-proc.) (B).

Figure 5.29: One-to-one comparison among couples of plain and state-splitting AUPF architectures with/without pseudorange pre-processing.

5.4.3 Kalman filters vs AUPF with pre-processing

The final section of this chapter tries to collect and compare the most relevantly performing fusion architectures encountered so far. More specifically, the focus is oriented, for both Kalman-based and PF-based implementations, towards the filtering variants including adaptive RAE (Section 4.1.3) and RMNCE (Section 4.1.4) models for online measurement-noise covariance estimation and, possibly, code-ranging data pre-processing (Section 4.2). As a further remark, the selection of one specific fusion scheme is guided by the foregoing experimental evaluations which are strictly concerned with estimation accuracy on the horizontal position state.

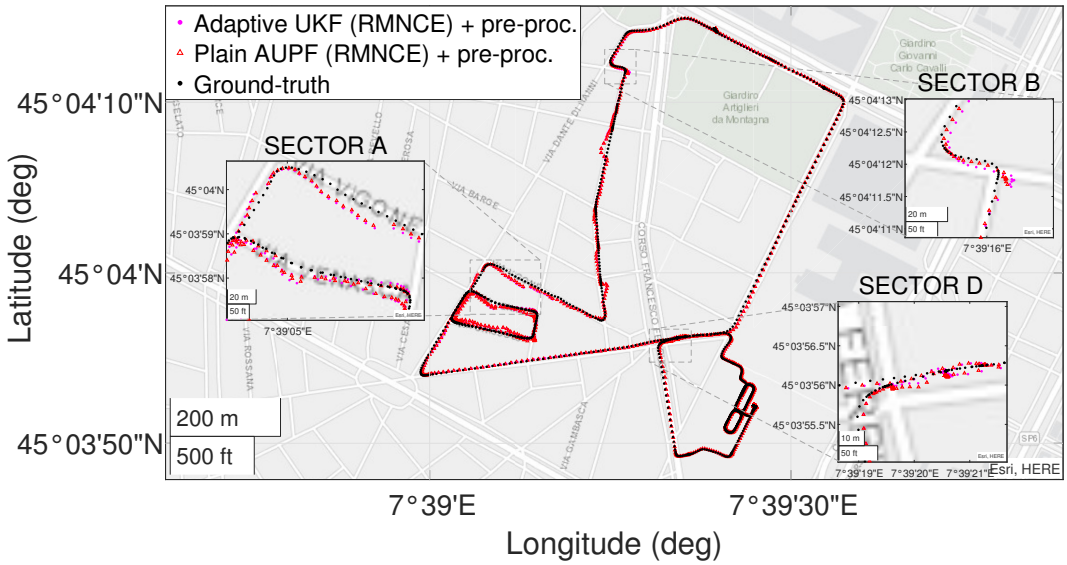


Figure 5.30: 2-D experimental trajectory in latitude/longitude (LLH) spherical coordinates. Comparison between RMNCE-based adaptive UKF and AUPF (plain architecture at 10^3 particles) with pseudorange pre-processing.

A first result, offering a purely qualitative assessment, is provided in Figure 5.30, which depicts the estimated vehicle trajectories by the UKF and the plain AUPF, both implemented with adaptive RMNCE strategy and augmented with the low-complexity pre-filtering module. As an additional aid, the same Figure incorporates some zooms on specific trajectory sectors that, in the course of the assessment, have been found to critically prove both the robustness of the filtering algorithm and the accuracy preservation in the solution to the localization problem.
In Sector A, both Bayesian fusion algorithms demonstrate robustness against multipath and, even in case some bias accumulates in the estimated trajectory fixes, it gets compensated with high responsiveness. Moving the attention on Sector

B, the AUPF estimated position fixes are tighter to the ground-truth than those supplied by the adaptive UKF. Finally, looking at Sector D, where most of the Bayesian schemes treated so far (especially legacy architectures) dramatically loose accuracy or even diverge, the considered adaptive filters with pre-processing turn out to preserve a rather high accuracy level in the horizontal position estimates, and neither a solution divergence nor an error bouncing are triggered.

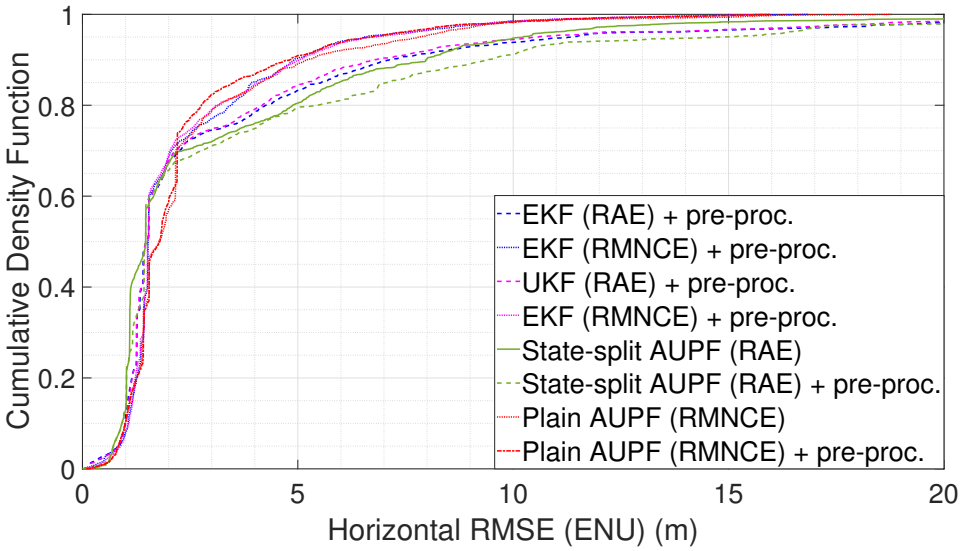


Figure 5.31: Empirical CDF (ECDF) of the RMSE on the horizontal position (E/N) in UTM-coordinates. Comparison between adaptive EKF, UKF and AUPF implementations (plain and state-splitting) with pseudorange pre-processing.

To acquire a clearer perspective on the average accuracy behaviour of the aforementioned Bayesian strategies and of few other high performance ones, it might be useful to account for the CDF trend of the RMSE on the horizontal (E/N) position components (Figure 5.31). Starting form the high-accuracy segment, Figure 5.32 considers average horizontal errors below 2 m only. Once again, analogously to the observations of Section 5.4.2, the state-splitting RAE-based AUPF architecture, when augmented with data pre-processing, is apparently outperforming the remainder fusion algorithms, involving both adaptive solutions in the Kalman family and alternative plain AUPF schemes. For example, looking at the proposed 50-th percentile zoom, the average error is reduced by 1.4×10^{-2} m w.r.t. an adaptive (RAE-based) UKF routine with pre-processing. The corresponding percentage accuracy gain is slightly below 1 %, which is not a minor improvement both because it is measured for magnitude values of the RMSE around 1.5 m and, more importantly, because the same percentage enhancement is much larger at slightly lower percentiles (in particular, a maximum gain is observed between 1.1 m and

1.2 m). As far as adaptive Kalman filters (with pre-processing) are concerned, their low-error trend is mid-way between plain and state-splitting AUPF paradigms. However, it is noticeable a finer behaviour of the estimator variant relying on RAE modelling rather than the one adopting RMNCE. Again, this is not surprising in light of former considerations, since the use of measurement residuals, when short misalignments are involved, is highly advantageous to precisely track noise statistics. Eventually, the average less accurate position estimates, at any percentile below the 50-th level, are those output by a plain AUPF with RMNCE, hence upholding the ineffectiveness of FOSDs and Second-Order Mutual Differences (SOMDs) in carefully approximating the variance parameter when Gaussian observation noise is little.

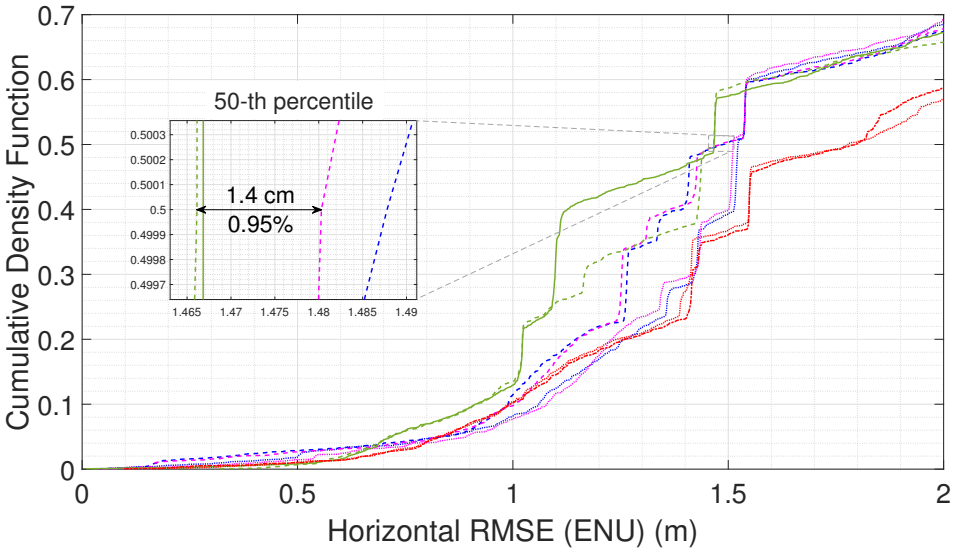


Figure 5.32: Zoom of Figure 5.31 for horizontal RMSE below 2 m.

By the other side, Figure 5.33 outlines the CDFs for horizontal RMSE magnitudes above 2 m. Apparently, the former high-accuracy trend is overturned, and the plain RMNCE-based AUPF architecture, augmented with pseudorange pre-processing, proves superior accuracy performance w.r.t. the other involved Bayesian architectures. Nevertheless, RMNCE-adaptive EKF and UKF strategies with pre-filtering are not outdone at all, and the performance interplay between such filtering methodologies is plain since average horizontal errors above 5 m. More scrupulously, in a RMSE range between 2 m and 5 m, the accuracy gap between the foregoing AUPF solution and the adaptive UKF routine is wider. For instance, as highlighted in the zoom at the 75-th percentile, the AUPF gains 8.86 % accuracy. In other words, as an average statement, 75 % of time the AUPF error is 22.6×10^{-2} m smaller than that of the adaptive UKF. Furthermore, this performance difference enhances

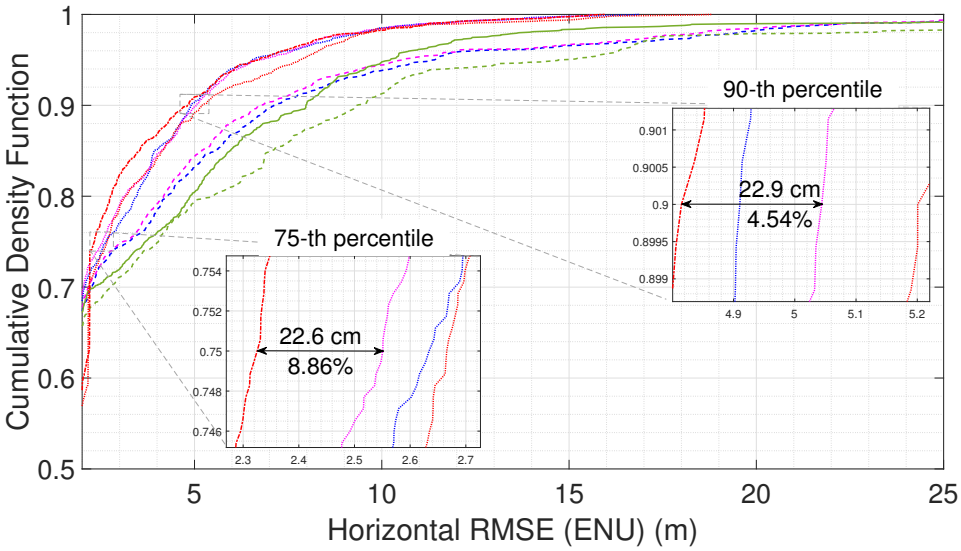


Figure 5.33: Zoom of Figure 5.31 for horizontal RMSE above 2 m.

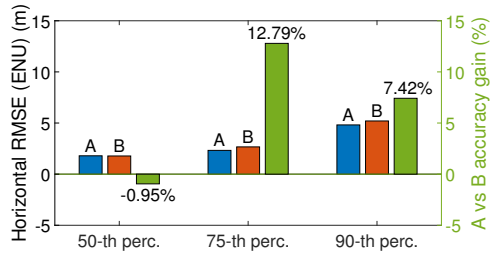
proportionally to the RMSE, and a peak is observed around 3 m. Then, from the 85-th percentile level on, the accuracy gap progressively reduces and, at the 90-th level (zoom of Figure 5.33), it is nearly halved. Beyond, for horizontal errors above 5 m, it can be stated that the considered CDFs match to a large extent.
The story is well different for state-splitting AUPF architectures which, suffering multipath, make hard in keeping the average horizontal misalignment bounded to small values. For instance, at the 90-th percentile, the measured RMSE is about 3 m larger than that committed by plain AUPF implementations. Moreover, as an additional term of comparison, the plain AUPF scheme produces a maximum average horizontal error which is slightly below 20 m, while the state-splitting strategy measures twice as much the value.

A thorough investigation on the aforementioned filtering paradigms is pursued with the analysis of Figure 5.34, which compares some of them in pairs at the percentiles of interest. Starting from the simultaneous observation of Figures 5.34a, 5.34b, 5.34c and 5.34d, it comes out a key result of the entire assessment path. In fact, in the medium-to-low accuracy CDF segment, that is for horizontal RMSE above 2 m, the plain AUPF architecture with RMNCE and pre-processing is unquestionably gaining accuracy over any other advanced Bayesian solution. Particularly, looking at Kalman implementations for the comparison (5.34b and 5.34c), the average improvement at the 75-th percentile is of about 31×10^{-2} m (11.73 %) w.r.t. the adaptive EKF, and of 22.6×10^{-2} m (8.86 %) over the adaptive UKF. Thus, at the 75-th level, the RMNCE-based UKF with data pre-filtering is

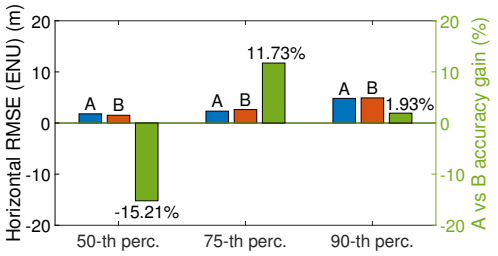
slightly less sub-optimal (smaller accuracy loss) than the equivalent EKF routine. Contrarily, at the 90-th percentile, the AUPF gains 9.5×10^{-2} m (1.93 %) over the EKF and 22.9×10^{-2} m (4.54 %) over the UKF, hence reversing the sub-optimality order. Overall, these results stress the importance of jointly combining, within the hybridization filter, the RMNCE paradigm with the pre-processing module, in the aim of better counteracting multipath in signal-degraded navigation scenarios.
Moving to Figures 5.34e and 5.34f, the ultimate attempt is showing the advantage of adopting the state-splitting paradigm to multiply weight error particles in PF-based architectures. It has been already clarified that, for horizontal errors above 2 m (percentiles higher than the 60-th level), such approach suffers a consistent accuracy degradation, with the corresponding CDF curves that markedly bend towards the right. Instead, it is important to highlight the state-splitting behaviour in high accuracy, especially when augmented with an adaptive RAE observation variance modelling stage. At the 50-th percentile, the improvement is more than moderate, with an average horizontal error reduction of 2.1×10^{-2} m w.r.t. the adaptive EKF and of 1.3×10^{-2} m w.r.t. the adaptive UKF. However, sliding down to errors below 1.4 m, the enhancement gets more marked (refer to Figure 5.32). Eventually, expanding the state-splitting AUPF with ranging measurement pre-processing, it doesn't seem to pay much. Indeed, comparing Figure 5.34g to Figure 5.34e and Figure 5.34h to Figure 5.34f, the percentage accuracy gain variation at the 50-th percentile is minimal (few millimetres of difference).

Table 5.8: Horizontal (E/N) position RMSE measured for adaptive EKF, UKF and AUPF architectures (plain and state-splitting) with pseudorange pre-processing, at 50-th, 75-th and 90-th percentiles.

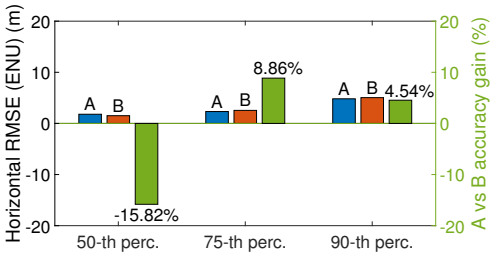
Bayesian architecture	50-th percentile Error (m)	75-th percentile Error (m)	90-th percentile Error (m)
Adaptive EKF (RMNCE) with pre-proc.	1.522	2.634	4.910
Adaptive UKF (RMNCE) with pre-proc.	1.511	2.551	5.044
Adaptive EKF (RAE) with pre-proc.	1.488	3.240	7.123
Adaptive UKF (RAE) with pre-proc.	1.480	3.084	6.710
Plain AUPF (RMNCE) with pre-proc.	1.795	2.325	4.815
Plain AUPF (RMNCE)	1.778	2.666	5.201
State-splitting AUPF (RAE) with pre-proc.	1.466	4.035	9.332
State-splitting AUPF (RAE)	1.467	3.682	7.977



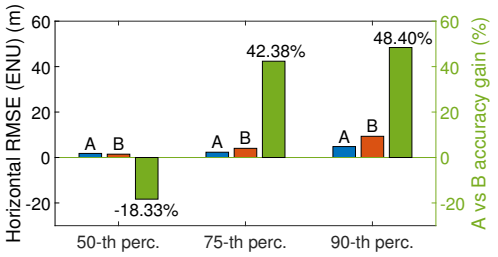
(a) Plain AUPF (RMNCE+pre-proc.) (A) & Plain AUPF (RMNCE) (B).



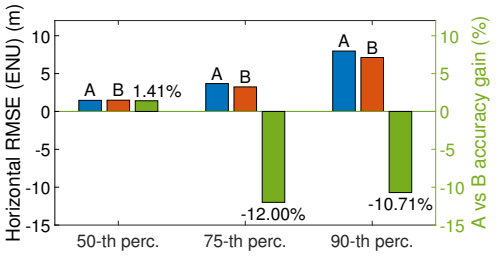
(b) Plain AUPF (RMNCE+pre-proc.) (A) & Adaptive EKF (RMNCE+pre-proc.) (B).



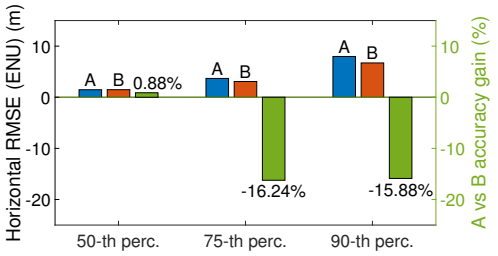
(c) Plain AUPF (RMNCE+pre-proc.) (A) & Adaptive UKF (RMNCE+pre-proc.) (B).



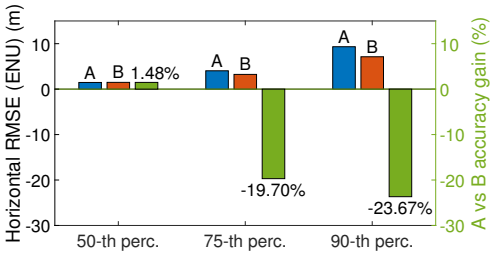
(d) Plain AUPF (RMNCE+pre-proc.) (A) & State-splitting AUPF (RAE+pre-proc.) (B).



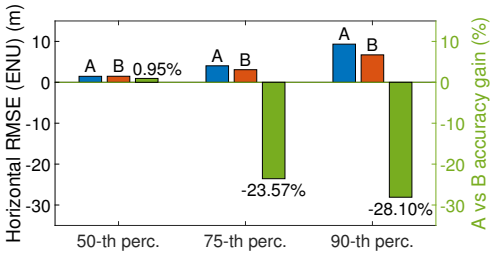
(e) State-splitting AUPF (RAE) (A) & Adaptive EKF (RAE+pre-proc.) (B).



(f) State-splitting AUPF (RAE) (A) & Adaptive UKF (RAE+pre-proc.) (B).



(g) State-splitting AUPF (RAE+pre-proc.) (A) & Adaptive EKF (RAE+pre-proc.) (B).



(h) State-splitting AUPF (RAE+pre-proc.) (A) & Adaptive UKF (RAE+pre-proc.) (B).

Figure 5.34: One-to-one comparison among couples of adaptive EKF, UKF and AUPF (plain and state-splitting) architectures with/without pseudorange pre-processing.

Conclusions

Global Navigation Satellite System (GNSS) is a ubiquitous radio-navigation technology providing reliable and long-term stable localization capabilities at affordable costs. In the framework of land-vehicle positioning and navigation, dense urban areas identify challenging environments, where the presence of multipath and the poor satellite visibility conditions detrimentally affect the quality of GNSS-standalone Positioning, Navigation and Timing (PNT) solutions. It follows that, for critical applications requiring high precision and accuracy, the absolute positioning paradigm offered by satellite-based multilateration becomes infeasible, hence paving the way to advanced multi-sensor navigation units. In such context, a strong research effort has been put on coupling GNSS with an Inertial Navigation System (INS), by leveraging the complementary natures of the two navigation sensors to enhance both quality and robustness in the localization solution.
The recent advances in stochastic sensor fusion methodologies based on Bayesian estimation have laid the means for the studies involved with this thesis. The most common sub-optimal Kalman filter implementation, in terms of the Extended Kalman Filter (EKF), sets as the state-of-art integration module for a INS/GNSS system. When properly configured in its parameters attaining to the discrete system state-space model, the EKF proves good effectiveness in the state-estimation process.

Given this starting point and pursuing an indirect (error-based) state-space formulation, it has been proposed the implementation of a more complex and advanced Unscented Particle Filter (UPF) architecture. It implements a synthesis between the Unscented Transform (UT) function, engaged in the importance distribution refinement for particles' draw, and the Sequential Importance Sampling (SIS) concept, which fixes the operational trademark of any Sequential Monte-Carlo (SMC) filter. Focusing on the underlying PF-stage within the UPF scheme and looking over canonical particle weighting, it has been put forward an innovative state-splitting criterion, which does not introduce extra computational burden and can be extended to any PF model. It envisions a multiple weighting of random particles such that each weight contributes to the estimation of just a portion of

the whole state-space. Although this original weighting paradigm has not been conceived within this thesis, the current research has offered the first assessment on a real dataset with sub-centimetre accurate ground-truth. From the outcomes of Chapter 5, state-splitting architectures have been found improving the estimation performance in the high-accuracy region, that is for positioning errors below 2 m.

One of the chief motivations fuelling the research on the UPF and, more in general, on PFs, is the huge filter flexibility in handling and accommodating multiple models, in terms of analytic distributions, for the stochastic description of noise affecting input GNSS-observables. This perspective carries a great potential over Kalman filters which, by their nature, are forced to select Gaussian models. Nevertheless, in a dynamic system which fast evolves in a dynamic environment, the non-stationary and non-ergodic measurement-noise process behaviour complicates the elaboration of alternative, unified statistical characterizations, and a further research effort is demanded on this topic. Accepting zero-mean Gaussian measurement densities, the optimization was then oriented along two main paths; on one hand, the exploration of adaptive paradigms for real-time measurement-noise covariance estimation, thus allowing for a prompt reflection of environmental changes on the filter statistical information; on the other hand, the design and implementation of a portable, low-complexity and self-contained signal processing module to mitigate multipath-related bias injections in the integrated PNT solution. As far as the first aspect is concerned, Residual-based Adaptive Estimation (RAE) and Redundant Measurement Noise Covariance Estimation (RMNCE) algorithms have been investigated. From the final assessment on the accuracy of the horizontal position estimates, two important hints have been grasped. First, the introduction of an adaptive RAE routine in the state-splitting UPF architecture grants a performance gain in the high accuracy. Secondly, the combination of the adaptive RMNCE strategy with the self-contained pre-filtering unit allows to effectively counteract multipath and to improve the estimation quality in the low-precision.

Putting an eye to the future, it is worth expanding the INS/GNSS unit to support multi-constellation signal processing and dual-frequencies, thus easing the deployment of sophisticated RAIM techniques to hold up the integrity and availability of the hybrid PNT solution. Another research pattern should foster the elaboration of convenient inertial models to support the design of a Tightly Coupled (TC)-scheme in direct configuration, where the fusion routine outputs total navigation states in place of correcting factors. Furthermore, a profitable architectural evolution might envision the integration of additional aiding sensor technologies (e.g. visual sensors, UWB solutions etc.). Eventually, the navigation filter observation model could be extended as well, to bear collaborative positioning paradigms or network assistance strategies.

Bibliography

[1] P. Misra and P. Enge. *Global Positioning System : Signals, Measurements, and Performance*. Revised Second Edition. Ganga-Jamuna Press, Dec. 2010. ISBN: 978-0-97095-442-8. URL: https://www.navtechgps.com/global-positioning-system-signals-measurements-and-performance-revised-second-edition-paperback/ (cit. on pp. 2, 4, 5, 8, 10, 11, 13–17, 93).

[2] ESA - Navipedia. *GPS Space Segment*. [online]. 2011. URL: https://gssc.esa.int/navipedia/index.php/GPS_Space_Segment (cit. on p. 4).

[3] GPS.gov. *Space Segment*. Official U.S. government information about the Global Positioning System (GPS). URL: https://www.gps.gov/systems/gps/space/ (cit. on p. 4).

[4] GPS.gov. *The Global Positioning System*. Official U.S. government information about the Global Positioning System (GPS). URL: https://www.gps.gov/systems/gps/ (cit. on p. 5).

[5] European Space Agency (ESA). *What is Galileo?* [online]. URL: http://www.esa.int/Applications/Navigation/Galileo/What_is_Galileo (cit. on p. 5).

[6] ESA - Navipedia. *Galileo Signal Plan*. [online]. 2011. URL: https://gssc.esa.int/navipedia/index.php/Galileo_Signal_Plan (cit. on p. 5).

[7] ESA - Navipedia. *GNSS signal*. [online]. 2011. URL: https://gssc.esa.int/navipedia/index.php/GNSS_signal (cit. on pp. 6, 7).

[8] ESA - Navipedia. *GNSS Receivers General Introduction*. [online]. 2011. URL: https://gssc.esa.int/navipedia/index.php/GNSS_Receivers_General_Introduction (cit. on p. 7).

[9] ESA - Navipedia. *Front End*. [online]. 2011. URL: https://gssc.esa.int/navipedia/index.php/Front_End (cit. on p. 9).

[10] C. Cristodaro. «Advanced Integration of GNSS and External Sensors for Autonomous Mobility Applications». Prof. F. Dovis,Supervisor. PhD thesis. Turin, Italy: ScuDo,Politecnico di Torino, May 2019. DOI: http://hdl.handle.net/11583/2734313 (cit. on pp. 10, 23, 24, 30–32).

[11] C. Boucher. «Geodetic Reference Frames: 40 Years of Technological Progress and of International Cooperation: 1970–2010». In: *Reference Frames for Applications in Geosciences*. Ed. by Z. Altamimi and X. Collilieux. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 1–4. ISBN: 978-3-642-32998-2. DOI: 10.1007/978-3-642-32998-2_1 (cit. on pp. 14, 24).

[12] E.D. Kaplan, J.L. Leva, D. Milbert, and M.S. Pavloff. «Chapter 2 - Fundamentals of Satellite Navigation». In: *Understanding GPS: Principles and Applications*. Ed. by E.D. Kaplan and C. Hegarty. Second Edition. 16 Sussex Street, London: Artech House, 2017, pp. 21–63. ISBN: 978-1-63081-058-0. URL: https://us.artechhouse.com/Understanding-GPSGNSS-Principles-and-Applications-Third-Edition-P1871.aspx (cit. on pp. 17, 18).

[13] W. Wrigley. «History of Inertial Navigation». In: *NAVIGATION* 24.1 (1977), pp. 1–6. DOI: https://doi.org/10.1002/j.2161-4296.1977.tb01262.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.2161-4296.1977.tb01262.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2161-4296.1977.tb01262.x (cit. on p. 22).

[14] A. B. Chatfield. «Chapter 1 - Introduction». In: *Fundamentals Of High Accuracy Inertial Navigation*. Reston, Virginia: American Institute of Aeronautics and Astronautics, 2014, pp. 1–8. ISBN: 978-1-56347-243-5. DOI: 10.2514/5.9781600866463.0001.0011. URL: https://doi.org/10.2514/5.9781600866463.0001.0011 (cit. on p. 22).

[15] A. Bose, K.N. Bhat, and T. Kurian. «Chapter 2 - Autonomous Strapdown Inertial Navigation». In: *Fundamentals Of Navigation and Inertial Sensors*. Rimjhim House, Delhi: PHI Learning Pvt, Ltd. - Eastern Economy Editions, Jan. 1997, pp. 19–73. ISBN: 978-81-203-4859-2. DOI: 10.1002/9781119547860.ch2. URL: https://doi.org/10.1002/9781119547860.ch2 (cit. on pp. 22, 23).

[16] J. Zhou. «Low-cost MEMS-INS/GPS integration using nonlinear filtering approaches». PhD thesis. Siegen, Germany: University of Siegen, Center for Sensorsystems (ZESS), Apr. 2013. URL: https://dspace.ub.uni-siegen.de/handle/ubsi/750 (cit. on pp. 22, 23, 28, 39, 47, 49, 50, 59, 61, 69–74, 79).

[17] M.S. Grewal, L.R. Weill, and A.P. Andrews. «Chapter 2 - Fundamentals of Satellite and Inertial Navigation». In: *Global Positioning Systems, Inertial Navigation, and Integration*. Hoboken, New Jersey, USA: John Wiley & Sons, Ltd, 2007, pp. 18–52. ISBN: 978-0-47-009972-8. DOI: 10.1002/9780470099728.ch2. URL: https://doi.org/10.1002/9780470099728.ch2 (cit. on p. 23).

[18] P.D. Groves. «Chapter 4 - Inertial Sensors». In: *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. 2nd edition. 16 Sussex Street, London: Artech House, 2008, pp. 97–120. ISBN: 978-1-58053-255-6. URL: https://us.artechhouse.com/Principles-of-GNSS-Inertial-and-Multisensor-Integrated-Navigation-Systems-Second-Edition-P2046.aspx (cit. on p. 23).

[19] CHRobotics LLC. *Understanding Euler Angles*. online resource. Nov. 2010. URL: http://www.chrobotics.com/library/understanding-euler-angles (cit. on p. 23).

[20] D. Titterton and J.L. Weston. «Chapter 3 - Basic principles of strapdown inertial navigation systems». In: *Strapdown Inertial Navigation Technology*. 2nd edition. Stevenage, United Kingdom: The Institution of Engineering and Technology, 2004, pp. 17–58. ISBN: 978-0-86341-358-2. DOI: 10.1049/PBRA017E. URL: https://doi.org/10.1049/PBRA017E (cit. on pp. 24–26).

[21] A. Noureldin, T.B. Karamat, and J. Georgy. «Chapter 8 - INS/GPS Integration». In: *Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration*. Berlin, Heidelberg: Springer, 2013, pp. 247–271. ISBN: 978-3-642-30465-1. DOI: 10.1007/978-3-642-30466-8. URL: https://doi.org/10.1007/978-3-642-30466-8 (cit. on pp. 28, 29, 39).

[22] R. Conley, R. Cosentino, E.D. Kaplan, C. Hegarty, J.L. Leva, M.U. de Haag, and K. Van Dyke. «Chapter 7 - Performance of Stand-Alone GPS». In: *Understanding GPS: Principles and Applications*. Ed. by E.D. Kaplan and C. Hegarty. Second Edition. 16 Sussex Street, London: Artech House, 2017, pp. 21–63. ISBN: 978-1-63081-058-0. URL: https://us.artechhouse.com/Understanding-GPSGNSS-Principles-and-Applications-Third-Edition-P1871.aspx (cit. on pp. 29, 92).

[23] M.G. Petovello. «Real-Time Integration of a Tactical-Grade IMU and GPS for High-Accuracy Positioning and Navigation». Prof. G. Lachapelle,Supervisor. PhD thesis. Calgary, Alberta: University of Calagary, Department of Geomatics Engineering, Apr. 2003. URL: http://www.geomatics.ucalgary.ca/links/GradTheses.html (cit. on pp. 29, 31, 32, 35–38, 43).

[24] C. Jekeli. *Inertial Navigation Systems with Geodetic Applications*. Berlin, Boston: De Gruyter, Oct. 2012. ISBN: 978-3-11-080023-4. DOI: 10.1515/9783110800234. URL: https://doi.org/10.1515/9783110800234 (cit. on pp. 30, 40, 41).

[25] M.G. Petovello, C. O'Driscol, and G. Lachapelle. «Weak Signal Carrier Tracking Using Extended Coherent Integration with an Ultra-Tight GNSS/IMU Receiver». In: *Proceedings of the European Navigation Conference*. Toulouse, France, 2008. URL: https://schulich.ucalgary.ca/labs/position-

location-and-navigation/files/position-location-and-navigation/petovello2008_conference_a.pdf (cit. on pp. 30, 43).

[26] M. Karaim. «Ultra-tight GPS/INS Integrated System for Land Vehicle Navigation in Challenging Environments». Prof. A. Noureldin, Supervisor. PhD thesis. Kingston, Ontario: Queen's University, Department of Electrical and Computer Engineering, May 2019 (cit. on p. 30).

[27] G. Falco, M. Pini, and G. Marucco. «Loose and Tight GNSS/INS Integrations: Comparisonof Performance Assessed in Real Urban Scenarios». In: *Sensors* 255.17 (Jan. 2017). DOI: 10.3390/s17020255. URL: https://doi.org/10.3390/s17020255 (cit. on pp. 31, 32, 39, 96).

[28] I. Skog and P. Handel. «In-Car Positioning and Navigation Technologies—A Survey». In: *IEEE Transactions on Intelligent Transportation Systems* 10.1 (2009), pp. 4–21. DOI: 10.1109/TITS.2008.2011712. URL: https://doi.org/10.1109/TITS.2008.2011712 (cit. on pp. 31, 32, 53).

[29] K. Li, B. Hu, L. Chang, and Y. Li. «Comparison of direct navigation mode and indirect navigation mode for integrated SINS/GPS». In: *Transactions of the Institute of Measurement and Control* 38.1 (2016), pp. 3–13. DOI: 10.1177/0142331214568236. eprint: https://doi.org/10.1177/0142331214568236. URL: https://doi.org/10.1177/0142331214568236 (cit. on p. 34).

[30] J. Zhou, S. Knedlik, and O. Loffeld. «INS/GPS Tightly-coupled Integration using Adaptive Unscented Particle Filter». In: *Journal of Navigation* 63.3 (2010), pp. 491–511. DOI: 10.1017/S0373463310000068. URL: https://doi.org/10.1017/S0373463310000068 (cit. on pp. 38, 77, 79–83, 90, 92, 96–100).

[31] R.E. Griffin and A.P. Sage. «Sensitivity analysis of discrete filtering and smoothing algorithms». In: *AIAA Journal* 7.10 (Oct. 1969), pp. 1890–1897. DOI: 10.2514/3.5477. eprint: https://doi.org/10.2514/3.5477. URL: https://doi.org/10.2514/3.5477 (cit. on p. 45).

[32] P. Closas Gomez. «Bayesian Signal Processing Techniques for GNSS Receivers: from multipath mitigation to positioning». Prof. A. Fernandez Rubio, Supervisor. PhD thesis. Barcelona, Spain: Universitat Politecnica de Catalunya, Department of Signal Theory and Communications, June 2009. URL: https://theses.eurasip.org/theses/310/bayesian-signal-processing-techniques-for-gnss/ (cit. on pp. 45–54, 59–62, 68–72, 75, 76).

[33] M.A. Caceres Duran, P. Closas, E. Falletti, C. Fernández-Prades, M. Nájar, and F. Sottile. «Chapter 6 - Signal Processing for Hybridization». In: *Satellite and Terrestrial Radio Positioning Technique*. Ed. by Davide Dardari, Emanuela Falletti, and Marco Luise. Oxford: Academic Press, 2012, pp. 317–382. ISBN: 978-0-12-382084-6. DOI: 10.1016/B978-0-12-382084-6.00006-4.

URL: https://doi.org/10.1016/B978-0-12-382084-6.00006-4 (cit. on pp. 47–49, 53, 69).

[34] P. Xie and M. G. Petovello. «Measuring GNSS Multipath Distributions in Urban Canyon Environments». In: *IEEE Transactions on Instrumentation and Measurement* 64.2 (2015), pp. 366–377. DOI: 10.1109/TIM.2014.2342452 (cit. on p. 58).

[35] S. Julier, J. Uhlmann, and H.F. Durrant-Whyte. «A new method for the nonlinear transformation of means and covariances in filters and estimators». In: *IEEE Transactions on Automatic Control* 45.3 (2000), pp. 477–482. DOI: 10.1109/9.847726. URL: https://doi.org/10.1109/9.847726 (cit. on pp. 59, 60).

[36] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. «A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking». In: *IEEE Transactions on Signal Processing* 50.2 (2002), pp. 174–188. DOI: 10.1109/78.978374. URL: https://doi.org/10.1109/78.978374 (cit. on pp. 68, 72, 74).

[37] P. Rebeschini and R. van Handel. «Can local particle filters beat the curse of dimensionality?» In: *The Annals of Applied Probability* 25.5 (2015), pp. 2809–2866. DOI: 10.1214/14-AAP1061. URL: https://doi.org/10.1214/14-AAP1061 (cit. on p. 69).

[38] D. Crisan and A. Doucet. «A survey of convergence results on particle filtering methods for practitioners». In: *IEEE Transactions on Signal Processing* 50.3 (2002), pp. 736–746. DOI: 10.1109/78.984773. URL: https://doi.org/10.1109/78.984773 (cit. on p. 69).

[39] A. Doucet, S. Godsill, and C. Andrieu. «On sequential Monte Carlo sampling methods for Bayesian filtering». In: *Statistics and Computing* 10 (3 July 2000). DOI: 10.1023/A:1008935410038. URL: https://doi.org/10.1023/A:1008935410038 (cit. on p. 72).

[40] M. A. Nicely and B. E. Wells. «Improved Parallel Resampling Methods for Particle Filtering». In: *IEEE Access* 7 (2019), pp. 47593–47604. DOI: 10.1109/ACCESS.2019.2910163. URL: https://doi.org/10.1109/ACCESS.2019.2910163 (cit. on p. 74).

[41] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. «Novel Approach to Nonlinear and Non-Gaussian Baysian State Estimation». In: *IEEE Proceedings F (Radar and Signal Processing)* 140 (2 Apr. 1993), 107–113(6). URL: https://digital-library.theiet.org/content/journals/10.1049/ip-f-2.1993.0015 (cit. on p. 77).

[42] A. Doucet, S. Godsill, and C. Andrieu. «On Sequential Simulation-Based Methods for Bayesian Filtering». In: *Statistics and Computing* 10 (3 July 2000). DOI: 10.1023/A:1008935410038. URL: https://doi.org/10.1023/A:1008935410038 (cit. on p. 77).

[43] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan. «The Unscented Particle Filter». In: *Advances in Neural Information Processing Systems*. Ed. by T. Leen, T. Dietterich, and V. Tresp. Vol. 13. Cambridge, Massachusetts, US: MIT Press, 2001, pp. 584–590. URL: https://proceedings.neurips.cc/paper/2000/file/f5c3dd7514bf620a1b85450d2ae374b1-Paper.pdf (cit. on p. 78).

[44] F. Sottile, H. Wymeersch, M. A. Caceres, and M. A. Spirito. «Hybrid GNSS-Terrestrial Cooperative Positioning Based on Particle Filter». In: *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*. 2011, pp. 1–5. DOI: 10.1109/GLOCOM.2011.6134002. URL: https://doi.org/10.1109/GLOCOM.2011.6134002 (cit. on p. 85).

[45] H. Kuusniemi. «Usel-level reliability and quality monitoring in satellite-based personal navigation». Prof. J. Takala and Prof. G. Lachapelle, Supervisors. PhD thesis. Tampere, Finland: Tampere University of Technology, 2005. URL: https://tutcris.tut.fi/portal/en/publications/usellevel-reliability-and-quality-monitoring-in-satellitebased-personal-navigation(b55448d2-0785-418a-9ccc-83aec2d47c31).html (cit. on pp. 94, 95).

[46] A.H. Mohamed and K.P. Schwarz. «Adaptive Kalman Filtering for INS/GPS». In: *Journal of Geodesy* 73.4 (1999), pp. 193–203. ISSN: 1432-1394. DOI: 10.1007/s001900050236. URL: https://doi.org/10.1007/s001900050236 (cit. on pp. 96–98).

[47] Z. Qifan, Z. Hai, L. You, and L. Zheng. «An Adaptive Low-Cost GNSS/MEMS-IMU Tightly-Coupled Integration System with Aiding Measurement in a GNSS Signal-Challenged Environment». In: *Sensors* 15.9 (2015), pp. 23953–23982. ISSN: 1424-8220. DOI: 10.3390/s150923953. URL: https://www.mdpi.com/1424-8220/15/9/23953 (cit. on pp. 101–103).

[48] L. Zheng, H. Zhang, Z. Qifan, and H. Che. «An Adaptive Low-Cost INS/GNSS Tightly-Coupled Integration Architecture Based on Redundant Measurement Noise Covariance Estimation». In: *Sensors* 17.9 (2017). ISSN: 1424-8220. DOI: 10.3390/s17092032. URL: https://www.mdpi.com/1424-8220/17/9/2032 (cit. on pp. 101–103).

[49] P.W. Ward, J.W. Betz, and C.J. Hegarty. «Chapter 6 - Interference, Multipath, and Scintillation». In: *Understanding GPS: Principles and Applications*. Ed. by E.D. Kaplan and C. Hegarty. Second Edition. 16 Sussex Street, London: Artech House, 2017, pp. 21–63. ISBN: 978-1-63081-058-0. URL: https://us.artechhouse.com/Understanding-GPSGNSS-Principles-and-Applications-Third-Edition-P1871.aspx (cit. on p. 104).

[50] *OEM7 Commands and Logs Reference Manual*. OM-20000169. Version v17. HEXAGON, NovAtel. Jan. 2021. URL: https://docs.novatel.com/oem7/Content/PDFs/OEM7_Commands_Logs_Manual.pdf (cit. on p. 117).