DGK Veröffentlichungen der DGK

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften

Reihe C | Dissertationen | Heft Nr. 941

Rozhin Moftizadeh

Advanced Particle Filtering for Vehicle Navigation based on Collaborative Information

München 2024

Bayerische Akademie der Wissenschaften

ISSN 0065-5325 ISBN 978-3-7696-5353-3

Diese Arbeit ist gleichzeitig veröffentlicht in:
Wissenschaftliche Arbeiten der Fachrichtung Geodäsie und Geoinformatik der Leibniz Universität Hannover
ISSN 0174-1454, Nr. 397, Hannover 2024

DGK Veröffentlichungen der DGK

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften

Reihe C Dissertationen Heft Nr. 941

Advanced Particle Filtering for Vehicle Navigation based on Collaborative Information

Von der Fakultät für Bauingenieurwesen und Geodäsie
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des Grades
Doktor-Ingenieurin (Dr.-Ing.)
genehmigte Dissertation

von

Rozhin Moftizadeh, M. Sc.

München 2024

Bayerische Akademie der Wissenschaften

ISSN 0065-5325 ISBN 978-3-7696-5353-3

Diese Arbeit ist gleichzeitig veröffentlicht in:
Wissenschaftliche Arbeiten der Fachrichtung Geodäsie und Geoinformatik der Leibniz Universität Hannover
ISSN 0174-1454, Nr. 397, Hannover 2024

Adresse der DGK:

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften (DGK)

Alfons-Goppel-Straße 11 ● D – 80 539 München

Telefon +49 - 331 - 6264 1685 ● E-Mail post@dgk.badw.de

http://www.dgk.badw.de

Prüfungskommission:

Vorsitzender: Prof. Dr.-Ing. habil. Christian Heipke

Referent: PD Dr.-Ing. Hamza Alkhatib

Korreferenten: Prof. Dr.-Ing. Ingo Neumann
apl. Prof. Dr. techn. Franz Rottensteiner
Prof. Allison Kealy (Royal Melbourne Institute of Technology)

Tag der mündlichen Prüfung: 06.11.2023

ISSN 0065-5325 ISBN 978-3-7696-5353-3

Erklärung

Hiermit erkläre ich, dass ich die Regeln der geltenden Promotionsordnung kenne und eingehalten habe. Ich bin mit einer Prüfung nach den Bestimmungen der Promotionsordnung einverstanden.

Ich habe die Dissertation selbst verfasst und keine Textabschnitte von Dritten oder eigenen Prüfungsarbeiten ohne Kennzeichnung übernommen. Alle von mir benutzten Hilfsmittel und Quellen habe ich in meiner Arbeit angegeben.

Dritten habe ich weder unmittelbar noch mittelbar geldwerte Leistungen für Vermittlungstätigkeiten oder für die inhaltliche Ausarbeitung der Dissertation zukommen lassen.

Ich habe die Dissertation noch nicht als Prüfungsarbeit für eine staatliche oder andere wissenschaftliche Prüfung eingereicht.

Rozhin Moftizadeh

24. August 2023

Abstract

In recent years, Multi Sensor Systems (MSSs) have become increasingly important in various engineering fields due to their capability in data acquisition. Georeferencing of such MSSs in a global coordinate system is essential for an effective data analysis. While the Global Navigation Satellite System (GNSS) and Inertial Measurement Unit (IMU) are generally well-suited for georeferencing in rural areas, their reliability and accuracy diminish in urban environments due to issues such as the multipath effect. Moreover, using the IMU data leads to drifted results over time, which makes the use of this sensor less accurate across all environments. To address these challenges, filtering frameworks are commonly used in applications such as localization. These frameworks are favored because they allow for the recursive estimation of states, leading to a more efficient usage of memory.

To effectively address the challenges of MSS georeferencing in urban areas, the current thesis harnesses environmental information. Specifically, geometric data from existing urban infrastructure serve as additional inputs to enhance the localization process. The work emphasizes two key requirements. Firstly, there must be a mathematically defined information source representing the urban environment. Secondly, additional detailed environmental data are needed to effectively relate the MSS to this information source. For this purpose, the MSS is equipped with advanced sensors such as LiDARs and cameras to gather a high volume of observational data. Given the variety of these sensors, it becomes vital to develop a versatile filtering framework. By incorporating uncertainty information for the sensors and the observed environment, both explicit and implicit observation models must be dealt with. Furthermore, due to the inherent and unavoidable uncertainties, the framework is designed to operate without reliance on preliminary assumptions.

This dissertation aims to enhance the accuracy of MSS georeferencing in urban areas by leveraging environmental information. A key contribution of this work is the development of a particle filtering framework capable of handling both explicit and implicit observation models under uncertainty. Such a feature accommodates a range of sensors, including LiDAR and cameras. To further improve computational efficiency, the framework is integrated with the Kalman filter.

The developed framework is used to georeference an MSS in both in a real-world application and a simulated case study. The minimum and maximum accuracy of the obtained georeferenced positions are confirmed to be in decimeter and centimeter-level, respectively. Additionally, the methodology is applied on a parameter estimation example to realize its performance in applications other than georeferencing. The final results validate that the developed framework meets its stated objectives, providing a promising solution not only for MSS georeferencing in urban environments; but also, for other applications involving a large number of observations.

Keywords: Multi Sensor Systems, Georeferencing, Urban Localization, State Estimation, Particle Filtering, Kalman Filtering, Observation Models, Sensor Fusion, Uncertainty

Zusammenfassung

In den letzten Jahren haben Multi-Sensor Systeme (MSS) aufgrund ihrer Fähigkeit zur Datenerfassung in verschiedenen technischen Bereichen zunehmend an Bedeutung gewonnen. Die Georeferenzierung von solchen MSS in einem globalen Koordinatensystem ist für eine effiziente Datenanalyse unerlässlich. Während das Globale Navigationssatellitensystem (GNSS) und die Inertialmesseinheit (IMU) im Allgemeinen für die Georeferenzierung in ländlichen Gebieten sehr gut geeignet sind, nehmen ihre Zuverlässigkeit und Genauigkeit in städtischen Umgebungen aufgrund von Problemen wie dem Mehrwegeffekt ab. Darüber hinaus führt die Verwendung von IMU-Daten zu zeitlich driftenden Ergebnissen, was den Einsatz dieses Sensors in allen Umgebungen weniger genau macht. Um diesen Herausforderungen zu begegnen, werden in Anwendungen wie der Lokalisierung häufig Filterverfahren eingesetzt. Diese Ansätze werden bevorzugt, weil sie eine rekursive Schätzung von Zuständen ermöglichen, was zu einer effizienteren Nutzung der Rechenressourcen führt.
Um die Herausforderungen der MSS-Georeferenzierung in urbanen Gebieten effektiv zu bewältigen, werden in dieser Arbeit Umgebungsinformationen genutzt. Insbesondere dienen geometrische Daten aus der bestehenden städtischen Infrastruktur als zusätzlicher Input, um den Lokalisierungsprozess zu verbessern. Die Arbeit stellt zwei zentrale Anforderungen in den Vordergrund. Erstens muss ein mathematisch definiertes Datenformat zur Verfügung stehen, das die städtische Umgebung repräsentiert. Zweitens werden zusätzliche detaillierte Umgebungsinformationen benötigt, um das MSS effektiv mit dieser Informationsquelle zu verknüpfen. Zu diesem Zweck ist das MSS mit modernen Sensoren wie LiDARs und Kameras ausgestattet, um eine große Menge an Beobachtungsdaten zu generieren. Angesichts der Vielfalt dieser Sensoren ist die Entwicklung eines vielseitigen und flexiblen Filteransatzes unerlässlich. Durch die Einbeziehung von Unsicherheitsinformationen für die Sensoren und die beobachtete Umgebung müssen sowohl explizite als auch implizite Beobachtungsmodelle verarbeitet werden. Aufgrund der inhärenten und unvermeidlichen Unsicherheiten ist der Ansatz außerdem so konzipiert, dass er ohne Vorannahmen auskommt.
Diese Dissertation zielt darauf ab, die Genauigkeit der MSS-Georeferenzierung in städtischen Gebieten durch die Nutzung von Umgebungsinformationen zu verbessern. Ein wichtiger Beitrag dieser Arbeit ist die Entwicklung eines allgemeingültigen Partikelfilters, der sowohl explizite als auch implizite Beobachtungsmodelle unter Unsicherheit verarbeiten kann. Diese Ansatz ermöglicht die Unterstützung einer Reihe von Sensoren, einschließlich LiDAR und Kameras. Um die Recheneffizienz weiter zu verbessern, wird der Partikelfilter mit einem Kalmanfilter integriert.
Der entwickelte Ansatz wird für die Georeferenzierung eines MSS sowohl in einer realen Anwendung als auch in einer simulierten Fallstudie verwendet. Die minimale und maximale erreichte Genauigkeit der erhaltenen georeferenzierten Positionen liegt im Dezimeter- bzw. Zentimeterbereich. Zusätzlich wird die Methodik auf ein Beispiel zur Parameterschätzung angewendet, um die Leistungsfähigkeit in anderen Anwendungen als der Georeferenzierung zu demonstrieren. Die abschließenden Ergebnisse bestätigen, dass die entwickelten Verfahren die gesetzten Ziele erreichen und eine vielversprechende Lösung nicht nur für die Georeferenzierung von MSS in urbanen Umgebungen, sondern auch für andere Anwendungen mit einer großen Anzahl von Beobachtungen darstellen.

Schlagwörter: Multi-Sensor Systeme, Georeferenzierung, Urbane Lokalisierung, Zustandsschätzung, Partikelfilterung, Kalman-Filterung, Beobachtungsmodelle, Sensorfusion, Unsicherheit

Contents

List of Figures

List of Tables

List of Algorithms

Acronyms

6-DoF six degrees of freedom
APF auxiliary variable particle filter
AUV autonomous underwater vehicle
CI confidence interval
DBN dynamic Bayes network
DTM digital terrain model
ECEF earth-centered, earth-fixed
EKF extended Kalman filter
EnKF ensemble Kalman filter
ETRS89 European terrestrial reference system 1989
GNSS global navigation satellite system
GRV Gaussian random variable
ICP iterative closest point
IEKF iterated extended Kalman filter
IMU inertial measurement unit
IQR interquartile range
KF Kalman filter
KLD Kullback-Leibler divergence
LKF linear Kalman filter
LiDAR light detection and ranging
LoD-2 level of detail 2
LoD-3 level of detail 3
LUCOOP Leibniz University Cooperative Perception and Urban Navigation Dataset
MC Monte-Carlo
MSS multi sensor system
MSSs multi sensor systems
MAE mean absolute error
NN nearest neighbour
OSM OpenStreetMap
PDF probability density function
PF particle filter
PFI particle filter with implicit observation models
RBPF Rao-Blackwellized particle filter
R-EKPFI robust extended Kalman particle filter with implicit observation models
RMSE root mean square error
R-PFI robust particle filter with implicit observation models
SIR sequential importance resampling
STD standard deviation
TLS terrestrial laser scanner
TW-ToF two way time of flight
UAV unmanned aerial vehicle
UKF unscented Kalman filter
UTM universal transverse mercator
UWB ultra wide band
V1 vehicle 1

V2 vehicle 2
V3 vehicle 3
VCM variance covariance matrix
VLP-16 Velodyne VLP-16
i.i.d. independent and identically distributed

1 Introduction

1.1 Motivation

One of the outcomes of the rapid technological improvement in the recent decades is the increased demand for automation. Among the many categories in which automation is required, autonomous aerial and terrestrial vehicles have attracted a considerable amount of attention. An example of an autonomous aerial vehicle is a drone, while a car serves as an example in the terrestrial category. As the name suggests, these vehicles should function without steering input from humans. They can assist in various fields such as transporting passengers and acquiring data for various engineering tasks, which in turn facilitate the daily lives of humans.
An essential aspect in autonomous vehicles is their *localization* that must be carefully managed to ensure proper functionality. Localization, also known as *georeferencing*, involves determining the vehicle's pose relative to a super-ordinate coordinate system. The pose comprises six parameters, including the 3D position and orientation of the vehicle in a global coordinate system. These six parameters are often referred to as the six degrees of freedom (6-DoF). The definition of the global coordinate system depends on the application; for an autonomous car or drone, it could be the European terrestrial reference system 1989 (ETRS89), whereas for an indoor autonomous robot within an indoor environment, it might be a user-defined coordinate system.
Autonomous vehicles are usually equipped with more than one sensor due to limitations that arise from the sensor side, the surrounding environment or both. Consequently, the usage of single sensors in complex applications might not be feasible or it may lead to inefficiency issues. For example, inertial measurement units (IMUs) are prone to drift over time, making their data unreliable for extended periods (Grewal et al., 2001). Another example is the incapability of the cameras in capturing the environment in darkness. Consequently, using this sensor is limited to those cases in which light conditions are adequate. Additionally, light detection and ranging (LiDAR) sensors are affected by rainy weather conditions, which in the context of autonomous localization in urban environments can become critical. Sometimes the, sensor limitations arise from the geometrical condition of the surrounding environment rather than their intrinsic characteristics. The current thesis is motivated by these limitations, which in the context of autonomous localization in the urban environment can become critical. In such a case, high-rise buildings directly affect global navigation satellite system (GNSS) signals, resulting in unavailable, or unreliable sensor data. As a result, the vehicle can either not be localized or it is georeferenced with a high uncertainty, which can lead to dangerous situations. Consequently, to overcome the shortcomings of individual sensors in autonomous vehicles and achieve robust measurement, using multiple sensors is usually preferred. In this context, the resulting system is referred to as a multi sensor system (MSS). As a general term, MSS is a platform where various sensors are installed, which in case of autonomous vehicle should be referred to as kinematic MSS. To avoid limitations to only typical autonomous vehicles such as cars or drones, the term kinematic MSS is used in the subsequent parts of the thesis.
By using each sensor, a connection to the surrounding environment is established. Consequently, the vehicle can benefit from additional information available in the area. As an example, detected features in derived images by cameras that carry spatial information can be mentioned. Considering these features in combination with, e.g. GNSS and IMU data, can help in a more reliable georeferencing of kinematic MSSs in urban areas compared to using only GNSS and IMU data. Moreover, if other MSSs are present in the vicinity, connecting to them can aid the localization process. An example is a kinematic MSS with good visibility of GNSS signals that can assist

in localizing another kinematic MSS that is subject to signal blockage. Therefore, any source of knowledge that can assist the georeferencing problem is considered as *collaborative* information, which has been the focus of numerous recent investigations (e.g. Salameh et al. (2013), Hartzer and Saripalli (2021) and Malik et al. (2021)).
Depending on the sensor type, additional information source, and the requirements of the resulting georeferencing solutions, proposed methodologies differ. For instance, in the case of batch processing and considering the camera images, frameworks based on the bundle adjustment principle can be considered. As the name suggests, the georeferencing problem is solved in an offline mode. When computation time is critical, which is usually the case in autonomous vehicle localization, the interval-based methodologies can be employed. In these approaches, the camera images can be used for the purpose of georeferencing in real-time. However, the probabilistic aspects of the derived solutions, due to defined interval-based bounding boxes, are either largely neglected or only partially considered. Given existing uncertainties arising from the sensors, environment and mathematical frameworks, this work posits that the probabilistic aspect of the georeferencing solutions are critical and beneficial to consider. On the other hand, with increasing demand in automation, the development of real-time or near real-time approaches for georeferencing kinematic MSSs is of significant importance. Recursive state estimators within probabilistic approaches can effectively address these challenges. By recursively estimating the georeferencing parameters of a kinematic MSS over time, not only are the probabilistic aspects of the solutions preserved, but computation time is also significantly decreased. Moreover, among perception units, LiDAR sensors are preferred over cameras in this work due to the high memory and processing demand of camera images. The main challenge with low-cost LiDAR sensors is their lower accuracy, which compared to high-accuracy ones introduces additional challenges into the georeferencing process. The reason for focusing on low-cost LiDAR sensors is to make the resulting framework cost-effective. Having such a feature opens up the potential for its use in autonomous vehicle localization. The primary challenge of using low-cost LiDAR sensors is their reduced accuracy compared to their more expensive counterparts, introducing additional complexities into the georeferencing process. As previously mentioned, perceiving the environment enables the extraction of useful information that assist in georeferencing the MSSs. In the case of low-accurate sensor, gathering this additional information comes with a degree of uncertainty, which directly impacts the quality of the resulting solutions. Therefore, the main objective of the current thesis is to address the georeferencing challenges of kinematic MSSs in urban environments by incorporating low-cost LiDAR sensors within a recursive state estimation framework.

1.2 Recursive State Estimators for Georeferencing of Kinematic Multi Sensor Systems

In general, geoereferencing of kinematic MSSs encompasses a wide range of applications, occurring in both outdoor and indoor environments. Due to the unique characteristics of each of these environments, the required sensors and appropriate methodologies differ. As articulated in the previous section, the objective of the current thesis is to address the challenges of georeferencing kinematic MSSs in urban areas. Given that the focus of this work is on outdoor environments, the subsequent sections will describe research pertinent to this type of georeferencing.
In a broad categorization, based on the works of Schuhmacher and Böhm (2005) and Paffenholz (2012), the outdoor georeferencing of MSSs is classified as follows:

- “Sensor-driven Georefencing”
- “Target-driven Georeferencing”
- “Data-driven Georeferencing”

In “sensor-driven georeferencing”, MSSs can be localized by directly using the sensor data. In this case, as also explained in previous section, if the earth-centered, earth-fixed (ECEF) is the target frame in which the MSS pose is sought, the GNSS data can directly be used for this purpose. Examples of research related to sensor-driven georeferencing are the works of Paffenholz (2012) and Zair et al. (2015). The former work focuses on the georeferencing of 3D point clouds by fusing scanner and GNSS data. In the latter, the localization of MSSs in urban areas using GNSS data is investigated. In case of having an arbitrary super-ordinate coordinate system, a high-precision sensor, e.g. a laser tracker or a total station becomes essential to set up the global coordinate system. Subsequently, the MSS is localized relative to this system (Dennig et al. (2017) and Hartmann et al. (2018)). Moreover, regardless of the target coordinate system, the IMU sensors can always be directly used to derive the 3D orientation of the MSS. An example of research in this area is the work of Talaya et al. (2004), which uses IMU data to derive the direct orientation of terrestrial laser scanners. Nonetheless, in such a case, the unavoidable errors and biases resulting from the drifting effect over time should be well accounted for.
On occasion, pre-defined targets within a specific global coordinate system may exist. In such a case, if these targets are used to localize the MSS, the resulting georeferencing falls into the second category referred to as the “target-driven georeferencing”. The referenced targets that are used for this purpose can be of any type. For example, in a research by Elkhrachy and Niemeier (2006), 3D geometries are used for the matter of georeferencing, whereas Abmayr et al. (2008) have used already referenced flat markers for this purpose.
Instead of targets, sometimes it can happen that some specific datasets are available that are already defined with respect to a global coordinate system. In such a case, referred to as the “data-driven georeferencing”, these data can be used to georeference the MSS in the same super-ordinate coordinate system. For example, Soloviev et al. (2007) have used already referenced 3D point clouds for the matter of georeferencing. In some other investigations such as those by Li-Chee-Ming and Armenakis (2013), Dehbi et al. (2019), Unger (2020), Fernandez (2020) and Vogel (2020), it has been shown how digital terrain or city models could also be used for this purpose. In this context, Unger (2020) has used cameras in a recursive bundle adjustment to georeference an unmanned aerial vehicle (UAV) by using the level of detail 2 (LoD-2) city models. Similarly, Vogel (2020) has used LoD-2 city models for the matter of MSS georeferencing, but within a filtering framework. Furthermore, Fernandez (2020) has considered the LoD-2 city models within a collaborative navigation framework for georeferencing multiple MSSs.
In the recent decades, many investigations have been dedicated to data fusion frameworks within the MSS field. Due to the diversity of such applications, categorizing data fusion algorithms uniformly proves challenging. Comprehensive overviews can be found in the works of Castanedo (2013), Rosique et al. (2019) and Fayyad et al. (2020). According to a broad classification provided by Fayyad et al. (2020), the sensor fusion techniques can be divided into classical algorithms and deep-learning-based algorithms. The latter class of algorithms are out of the scope of the current work. Noteworthy among the classical algorithms are statistical methods, probabilistic methods, and interval analysis theory.
In practice, to address the challenges of MSS georeferencing using classical algorithms, filtering methods have consistently emerged as a suitable framework. Such frameworks are broad in type and vary depending on the application requirements, assumptions and the available information. For instance, when all the distributions in play are Gaussian and linear models accurately represent a MSS within a specific environment, the linear Kalman filter (LKF) emerges as an optimal recursive estimator (Simon, 2010). However, real-world applications often present challenges. The system’s inherent complexities and nonlinear behavior may violate the linearity assumption, while global uncertainties can disrupt the Gaussian assumption. Consequently, the LKF may not always serve as an optimal estimator in such scenarios. In such a case, other frameworks are needed, which based on a categorization by Alkhatib (2015) can be classified into two types. These two categories along with a selection of the investigations related to them are given in the following sub-sections.

1.2.1 Gaussian Approximate Methods

As mentioned above, in the case of linear models, the LKF could be used as an optimal estimator under the assumption that all the distributions involved are Gaussian. However, fulfilling these requirements in real world is not feasible. As explained by Denham and Pines (1966) and Gelb et al. (1974), in the case of nonlinear models, linearization could be done by using Taylor series expansion. By doing so, and by keeping the Gaussianity assumption, another realization of KF is derived, which is called the extended Kalman filter (EKF) (Simon, 2006). In a work by Tailanián et al. (2014), data from GNSS and IMU sensors are combined within the EKF framework to localize an UAV. In another work by Melendez-Pastor et al. (2017), the EKF is used to fuse data from four-wheel speed sensors and low-cost GNSS in real time for vehicle localization in urban areas where having signal outages is unavoidable.
In the case of having highly nonlinear models, the use of the first or higher order Taylor series expansion can lead to divergent estimates (Doucet et al., 2001). As explained by Bell and Cathey (1993), in such a case, one of the solutions is to iteratively linearize around the last updated state to overcome the linearization errors. Doing so, the resulting framework is called the IEKF. For the matter of MSS georeferencing, Vogel et al. (2018) have proposed an IEKF framework in which both the explicit and implicit observation models could be handled. An explicit observation model is a mathematical relation in which the observations can be described by the unknown parameters. In other words, it is possible to separate the observations from the unknowns in such observation models. On the contrary, in implicit cases, the observations cannot be estimated using the unknown parameters. To be more concise, these two are not mathematically separable. Implicit observation models, are typical when it comes to real applications where multiple types of sensors are involved and various information from the environment is to be considered. As studied by Schön et al. (2018), such a combination of multiple sensor and map data has several advantages. One of these advantages is the increased localization accuracy. Furthermore, the integrity and availability of georeferencing solutions over time can be ensured. These are aspects that are actively investigated in the i.c.sens Research Training Group. Among these researches, the aforementioned work of Vogel et al. (2018) is of great importance when it comes to real-world applications. This framework allows the use of multiple sensors as well as additional information that may be available from the environment, regardless of the type of observation models. The proposed IEKF methodology is further applied by Bureick et al. (2019), Vogel et al. (2019) and Vogel (2020) in the context of MSS georeferencing, which further validates its functionality.
Within the scope of the KF framework, methodologies like the unscented Kalman filter (UKF) and ensemble Kalman filter (EnKF) offer alternatives to handle highly nonlinear measurement equations and implicit observation models without resorting to the linearization process. The main idea of these methodologies is to propagate Gaussian random variables (GRVs) through the system dynamics to approximate the state distributions. In the case of the UKF, a deterministic sampling approach is used for this purpose (Julier and Uhlmann, 2004); while the EnKF uses a MC framework (Evensen, 2003). A number of investigations are dedicated to the application of these filters in different fields. In an investigation by Zhan and Wan (2007), a so-called “Iterated Unscented Kalman Filter” framework for the purpose of passive target tracking is introduced. Applying the developed framework on a simulated case and a real scenario has proved its robustness and more reliable estimations compared to the EKF and UKF. Moreover, Xian et al. (2016) have introduced a “Square Root Unscented Kalman Filter”, which can be used for the purpose of localization by using stereo cameras and inertial sensors. The main idea is to update the root of the state covariance directly in the filter. It is proved that doing so avoids the decomposition of the state covariance, which leads to the stability of the algorithm. Additionally, a paper by Allotta et al. (2016) addresses an UKF-based methodology, which is developed to be used for the matter of autonomous underwater vehicle (AUV) localization where no GNSS signal is available.
Furthermore, Herlambang et al. (2019) adapted the EnKF framework for the navigation of UAVs, incorporating nonlinear models of 6-DoF to minimize position errors. Tests on two simulated cases

confirmed a high accuracy rate of 99%. Additionally, a study by Apriliani et al. (2017) investigated the trajectory estimation of an AUV maneuvering in 6-DoF using the EnKF. Simulations results demonstrated the superior performance of the EnKF compared to another KF-based methodology, referred to as the "Fuzzy Kalman Filter". In Čurn et al. (2013), an EnKF-based approach was developed for cooperative localization. In the investigated scenario, all vehicles were equipped with GNSS and odometry. The main aim was to assist localizing vehicles lacking GNSS coverage. For that, proximity sensors such as radar were used to establish connections with other vehicles and benefit from their observations. In this context, information received from other vehicles was also influenced by their states, requiring accounting for correlations between state estimates and observation errors.
Due the limitations of the KF-based frameworks, their use becomes problematic in applications with high uncertainties. In such scenarios, alternative methodologies free from these preliminary assumptions are needed to more accurately represent reality.

1.2.2 Sequential Monte-Carlo Methods

The KF-based methodologies are applicable only if the requirements of this framework are fulfilled. In real-world scenarios, meeting these requirements —- specifically the Gaussianity assumption for the involved distributions and a reliable filter initialization —- may prove challenging. Besides, having highly nonlinear models along with the statistical information of the observation and process noise values being unknown leads to unreliable state estimations based on the KF approaches. In such cases, in order to have a realistic and reliable pose estimation, other strategies than KF should be applied among which the PF framework could be mentioned. Due to its simple principle, which can well deal with the global uncertainties, this filtering technique has attracted a significant amount of attention in the recent decades. In the current work, the global uncertainties refer to the highly uncertain initial values, unknown statistical information regarding the process and observation noises and any uncertainty that can arise from either the sensor side or the additional information of the surrounding environment. The PF approach uses the concept of MC estimating the states based on the Bayesian framework (Doucet et al. (2001) and Ristic et al. (2003)). The main idea of this filter is to derive the posterior distribution of the states by using randomly generated samples that are generally referred to as particles. Therefore, it is a non-parametric filter that unlike the Gaussian methodologies does not rely on a strict assumption regarding the involved posterior densities (Thrun et al., 2006). Due to having such a characteristic, it can well represent complex multi-modal state densities. Table 1.1 provides an overview of the KF-based frameworks in which the main characteristics of each framework is shown. As it can be seen, all of the KF-based frameworks have two common characteristics. Firstly, they are unable to encounter free-form posterior distributions, since in these frameworks having normally distributed states is the main assumption. Secondly, all of the KF-based approaches require a reliable initialization. Otherwise, they are prone to divergence over time. These two characteristics limit the usage of KF-based frameworks in application such as autonomous vehicle localization wherein having global uncertainties is unavoidable. It can be seen that such deficiencies are overcome in the PF framework, which are the main benefits of this filter. Numerous research endeavors are dedicated to the application of the PF in different fields. For localization purposes, Marchetti et al. (2006) compared two PF-based algorithms, namely the sequential importance resampling (SIR) and the auxiliary variable particle filter (APF), for a robot equipped with a camera in a soccer scenario.
In another research, González et al. (2009) have investigated the localization of a mobile robot by using three range measurement sensors, namely the ultra wide band (UWB) units. In this work, a PF framework is suggested in which the errors due to various effects such as multipath are well compensated for.

Table 1.1: *Overview of the KF-based frameworks and the PF.*

Filter name	Handling linear equations	Handling nonlinear equations	Having linearization error	Assuming Gaussian posterior	Handling free-form posterior	Requiring reliable initialization
LKF	✔	✘	✘	✔	✘	✔
EKF	✔	✔	✔	✔	✘	✔
IEKF	✔	✔	(✔)[a]	✔	✘	✔
UKF	✔	✔	✘	✔	✘	✔
EnKF	✔	✔	✘	✔	✘	✔
PF	✔	✔	✘	✘	✔	✘

[a]In the case of IEKF, the linearization error is not completely resolved. However, it is reduced.

Furthermore, Peker et al. (2011) have proposed a PF-based framework that incorporates the map topology to assign weights to the particles. The mentioned methodology, which is applied to 2D localization of a vehicle by using the GNSS and odometer data, shows the positive effect of using a road map for proper selection of the samples within the PF framework.
Additionally, Suhr et al. (2016) have developed a PF-based algorithm for localizing MSSs by fusing data derived from low-cost sensors on board. The main idea of the proposed framework is to engage low-volume digital maps within the likelihood estimation step. In such maps, it is sufficient to have road markings that are defined by a minimum number of points. The developed methodology is used for 2D localization of a vehicle that is equipped with a low-cost GNSS, low-cost IMU, a built-in wheel speed sensor and a single front camera. Results of the analysis show not only the reliability and efficiency of the proposed framework, but its capability to localize the vehicle even when it passes through tunnels, long urban canyons and under an elevated railroad.
In Kim et al. (2017), a novel fusion algorithm based on the PF framework is suggested in which the idea from the entropy information theory is integrated. The main idea is to involve each measurement model in the information gain, which in turn leads to a better performance of the filter. The suggested framework benefits from both vertical and road intensity information by using an efficient grid map strategy. The resulting approach is used to localize a MSS equipped with a 3D LiDAR, an IMU and a wheel odometry in an urban environment.
Moreover, Rormero et al. (2018) have proposed a so-called "Map-Aware Particle Filter" in which the information from a prior occupancy grid is used to constrain the vehicle poses. In this framework, a reweighing of each particle is performed, which is based on the validity of its current and previous positions with respect to the given map. It is shown that by using this methodology, not only the problem of dead reckoning can be well dealt with, but also by using a 2D LiDAR localization method, the errors that are caused by outdated and less accurate maps can be well compensated for. The proposed method is used to localize a robot in 2D space, which is equipped with wheel encoders for odometry, GNSS and an array of 2D LiDAR sensors.
In some recent works such as the one by Patoliya et al. (2022), a Rao-Blackwellized particle filter (RBPF)-based on a so-called Gmapping algorithm is designed for 2D localization of robots that are equipped with GNSS and LiDAR sensors. The developed framework is shown to be capable of generating 2D maps of unknown environments while providing a trajectory error of less than 0.1 cm. In this algorithm, the computational complexity resulting from a large number of particles is solved by using Gaussian approximation in the particle distribution. However, the proposed method suffers from the requirement of having planar surfaces only, which limits its application on surfaces with stairs.
In another research, Elhousni et al. (2022) proposes a deterministic cross-modal algorithm based on LiDAR and OpenStreetMap (OSM) data for 2D vehicle localization that does not require any learning components or labelled features. The main idea of this work is to generate simulated point cloud images as well as geometric constraints from OSM data, which could be combined with the

derived LiDAR data within the PF framework.
A major issue in PF is the choice of the proposal density from which the random samples should be generated. The more similar this density to the posterior density is, the less samples are required to be generated from it. However, since the posterior density is unknown, the selection of the proposal density is usually challenging. A poor choice of this density leads to a significantly high computation time. The reason lies in the rejection of most of the particles due to having significantly small importance weights. Consequently, a large number of particles are needed in this case to overcome the filter divergence. In order to help choosing a proper proposal density, some investigations suggest combining the KF framework with the PF. The main idea in such works is to select the proposal density by taking the sensor data into account. Of such investigations, the proposed "Unscented Particle Filter" by Van Der Merwe et al. (2000) can be mentioned in which the concept of UKF is applied to each particle in order improve the choice of the proposal density based on the available observations. The proposed methodology, which is applied on a signal processing example, is claimed to be suitable for engineering applications.
Additionally, Liang-Qun et al. (2005) have proposed a so-called "Iterated Extended Kalman Particle Filter" by means of which the proposal density is chosen based on applying the IEKF on each particle. In order to compare the performance of this filter with the one proposed by Van Der Merwe et al. (2000), the same numerical example given by Van Der Merwe et al. (2000) is used. The derived results show the superiority of the developed methodology by Liang-Qun et al. (2005) over Van Der Merwe et al. (2000). The reason for it is claimed to be the better performance of the IEKF framework compared to the UKF in highly nonlinear equations, which in turn leads to a higher precision of the resulting framework based on the IEKF (Liang-Qun et al., 2005). Furthermore, Alkhatib et al. (2012) have developed a so-called "Extended Kalman Particle Filter", which can be used for georeferencing a terrestrial laser scanner (TLS). In the same direction and by combining the IEKF and PF, Wu et al. (2013) have suggested a so-called "Modified Iterated Extended Kalman Particle Filter" framework that is used in satellite-to-satellite tracking. The simulation results of this work depict a higher tracking precision by the resulting filter compared to the EKF or by combining the EKF with PF.
Due to the capability of the PF framework in handling unavoidable uncertainties, its application in various fields is more preferred than the KF approaches. Since the MSS georeferencing is among the complicated applications, the current work claims that using the KF-based methodologies cannot always guarantee reliable solutions. Therefore, in this thesis the PF is selected as the basis for the developed georeferencing framework. Additionally, to overcome the challenge of high computation time, the EKF is chosen as the basis to reach a more reliable proposal density in the PF framework. Consequently, due to the less required number of particles, the computation time is significantly decreased.

1.3 Objectives

The primary objective of this thesis is to develop a PF-based framework specifically tailored for georeferencing kinematic MSSs in urban environments. As previously noted, the IMU data are subject to drifting and in inner city areas, due to, e.g. multipath effects, the GNSS data cannot be fully trusted. Consequently, a pure "sensor-driven georeferencing" approach is not feasible. To address this limitation, Bureick et al. (2019) employed a KF-based methodology that enables using the uncertain additional information of the surroundings – provided by the LoD-2 3D city models – to offset GNSS and IMU errors. In this context, LiDAR sensors were used to capture the surrounding environment. Figure 1.1 represents an urban area in which a drone and a car – both examples of kinematic MSSs – are to be localized. It shows that the environment can be captured via the scanners of the MSS, facilitating a link to the existing infrastructure. The green and red areas in this figure represent parts of the scanned data by means of the scanners on the drone and the car, respectively.

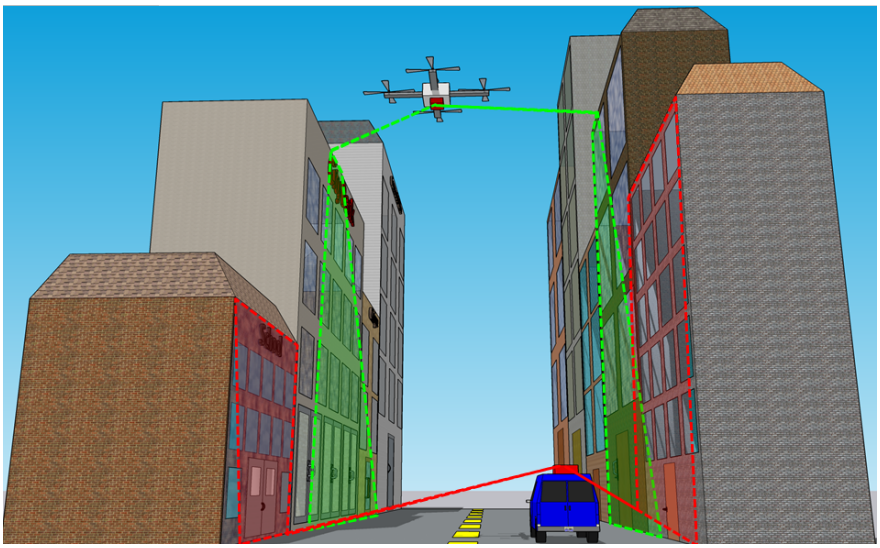

Figure 1.1: *Visualization of capturing the surrounding area of different MSSs by means of their scanners.*

To leverage the additional environmental information provided by LoD-2 3D city models, implicit observation models are required. However, the proposed PF frameworks in the research areas are limited to cases where only explicit measurement equations are used. Consequently, such methodologies are restricted to only those sensors whose data can be explicitly related to the states. Such a restriction avoids considering various sources of information, which in turn leads to a limited applicability of the resulting framework.
Therefore, the current work aims to develop a PF-based framework capable of handling implicit observation models for cases with a high number of observations. This focus stems from the overarching goal to improve the georeferencing of MSSs by exploiting the additional contextual information from the surroundings, which necessitates a large observational dataset for accurate environmental representation. Furthermore, it is required that the resulting framework can properly reflect the uncertainty of such information on the estimated states.
In addition, due to the large number of observations, it is possible to have misleading data. For a reliable state estimation, it is essential to detect and remove such observations before obtaining the states. Thus, the current work is devoted to robustifying the PF-based framework against existing anomalies.
Moreover, for accurate state estimation in highly nonlinear systems, a large particle set is generally required in PF-based frameworks. This increases computation time, which is undesirable in applications that require efficiency. Such applications include autonomous driving, which requires a real-time performance. To address this, the current work attempts to develop an efficient, yet robust, PF-based framework by utilizing a considerably smaller number of particles. The key innovation here is incorporating Kalman gain concepts from the KF-based approaches.
To assess the effectiveness of these frameworks, three case studies are investigated, two of which involve the use of the developed frameworks for georeferencing an MSS in both a simulated and a real-world application. The methodologies are adapted to accommodate various types of observation models, enabling the fusion of multiple sensors irrespective of their measurement equation type. The third case focuses on recursive parameter estimation for a mathematical model using the available observations, allowing for an examination of the framework's performance across diverse applications, which further investigates its generality.

1.4 Outline

The subsequent chapters of this thesis are organized as follows: Chapter 2 is dedicated to the fundamentals of recursive state estimation. In this chapter, the mathematical background of those methodologies that form the basis of this thesis are explained in detail. Understanding the content of this chapter helps to better follow the main novelties of the current work that are described in Chapter 3. In this chapter, three PF-based methodologies are introduced that can handle implicit observation models. First, a developed PF-based framework is explained that can handle implicit observation models in the presence of a large number of observations. Then, it is described how the developed framework can be improved to become resilient to outliers. In the last part, it is shown that by considering the concept of Kalman gain, the number of particles required can be drastically reduced, resulting in an efficient framework. The three developed methodologies are then applied to a recursive parameter estimation problem to demonstrate their practicability. In Chapter 4, the introduced approaches in Chapter 3 are used for kinematic MSSs in urban environments. In this chapter, the developed numerical approaches are further improved to allow the fusion of data from multiple sensors. The main aim for doing so is to achieve a framework that is independent of the type of observation models. In this chapter, two case studies are investigated in order to verify the functionality of the resulting approaches. In Chapter 5, the current work is concluded and some potential areas for future research are explained.

2 Fundamentals of Recursive State Estimation

MSS georeferencing is a challenging topic in the engineering field. On the one hand, it is prone to global uncertainties, and on the other hand, it often demands real-time performance. Therefore, the frameworks on which the newly developed georeferencing methodologies are based on should allow dealing with these aspects.
In the Bayesian framework, the uncertainty of the estimations is counted for by considering the unknown parameters as stochastic rather than deterministic (Fernández, 2011). In many problems such as dynamic state estimation, the complexity of a system is usually high enough that making a deterministic assumption for the unknown parameters may not be appropriate. Therefore, in these problems, even though other approaches are available, those methodologies that are based on the Bayesian formulation are usually preferred (Orlando et al. (2010) and Satish and Kashyap (1995)).
In addition to considering the unknown parameters as stochastic, the Bayesian framework has a recursive estimation feature. Therefore, in case of dynamic systems and at each time epoch, only the information from one time step earlier is taken into account. As a result, the computational complexity is significantly lower than the non-Bayesian strategies, which must consider all available information to estimate the unknown parameters at each epoch.
The current work is based on the Bayesian framework. Therefore, the main principles that form the basis of this dissertation are explained in the following sections. First, an overview of the Bayes filter is given in Section 2.1, which presents the recursive state estimation framework from a stochastic aspect. Then, in sections 2.2 and 2.3 two techniques to solve the Bayes filter, namely the KF and PF are explained.

2.1 Bayes Filter

The Bayes framework fundamentally relies on probabilistic principles. Hence, before explaining the Bayes filter, some basic concepts of probability are given in the following parts.
A pivotal notion in probability is the concept of *random variables*. According to Fristedt and Gray (2013), a random variable can be defined as the mathematical representation of a quantity that depends on random events. For example a random variable (x) can be seen as one value from a wider domain (X), so that $X = x$. Random variables can be categorized as either discrete or continuous. In the following parts, the mathematical notations are given only for the continuous space. However, the same definitions hold for the discrete case as well.
Unless otherwise stated, every random variable is assumed to have a *probability density function (PDF)*. A PDF is a mathematical function whose value at any given sample gives the relative likelihood that the value of a random variable would be equal to that sample (Clarke (1980) and Casella and Berger (2021)). The PDF of a random variable $X = x$ is shown as $p(X = x)$, which for simplicity is usually referred to as $p(x)$. The PDF of any random variable is non-negative ($p(x) \geq 0$) and its integral over the entire space always equals 1:

$$\int p(x)\ dx = 1. \tag{2.1}$$

In situations involving two random variables, for example x and y, their *joint distribution* is typically denoted by $p(x, y)$. If x and y are independent, the following relation holds for their joint distribution:

$$p(x, y) = p(x)\ p(y). \tag{2.2}$$

Another crucial concept in probability theory is *conditional probability*. If the value of a random variable x depends on the value of another random variable y, its conditional probability is expressed as:

$$p(x \mid y) = \frac{p(x, y)}{p(y)}. \tag{2.3}$$

Using the definition of conditional probability and the axioms of probability measures, the *theorem of total probability* is defined as follows:

$$p(x) = \int p(x \mid y)\ p(y)\ dy. \tag{2.4}$$

By means of the *Bayes rule*, the conditional probability $p(x \mid y)$ can be related to its inverse $p(y \mid x)$ as follows:

$$p(x \mid y) = \frac{p(y \mid x)\ p(x)}{p(y)}, \quad p(y) > 0. \tag{2.5}$$

In equation (2.5), $p(x \mid y)$ is referred to as the *posterior distribution* of the random variable x given a known random variable y. Moreover, $p(y \mid x)$ is referred to as the *likelihood* of the random variable y given the random variable x. Furthermore, $p(x)$ is called the *prior distribution* of x without incorporating any other random variables. In other words, $p(x)$ gives the prior knowledge of the PDF of x. Lastly, the factor $(p(y))^{-1}$ is a *normalizer* constant independent of x, ensuring that the resulting posterior distribution sums to 1.
According to the probabilistic rules, conditioning the Bayes rule on an arbitrary random variable such as $Z = z$ yields:

$$p(x \mid y, z) = \frac{p(y \mid x, z)\ p(x \mid z)}{p(y \mid z)}, \quad p(y \mid z) > 0. \tag{2.6}$$

Equation (2.6) forms the basis of the Bayes filter, which recursively estimates a set of unknown parameters based on available observations. To better understand the mathematical derivation of the Bayes filter, it is important to become familiar with some essential filter-related terminologies that are given in the following parts.
In filtering, the unknown parameters are time-dependent and typically referred to as *states*. The notation of a state vector at epoch k in time ($\boldsymbol{x}_k$) is:

$$\boldsymbol{x}_k = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_m \end{bmatrix}^T, \quad \boldsymbol{x} \in \mathbb{R}^m, \tag{2.7}$$

where x_m represents the m^{th} state. Moreover, the state vector in epoch k can be related to the state vector in epoch $k-1$ through a mathematical relation that is generally referred to as the *state transition function*. Such a function has a general form as follows, with $\boldsymbol{f}$ representing a vector of nonlinear functions that correspond to each state in $\boldsymbol{x}_k$:

$$\boldsymbol{x}_k = \boldsymbol{f}\left(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k\right) + \boldsymbol{w}_k, \tag{2.8}$$

where $\boldsymbol{w}$ is a Gaussian random vector with an expected value $\mathbf{0}$ and variance covariance matrix (VCM) $\boldsymbol{R}$, denoted as $\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R})$. By using such a random vector, which is also referred to as the *process noise*, the uncertainty of the state transition function can be modelled.
Similar to the state vector, an *observation* vector – also referred to as *measurements* – at epoch k in time ($\boldsymbol{l}_k$) can be written as:

$$\boldsymbol{l}_k = \begin{bmatrix} l_1 & l_2 & l_3 & \cdots & l_n \end{bmatrix}^T, \quad \boldsymbol{l} \in \mathbb{R}^n, \tag{2.9}$$

where l_n represents the n^{th} observation. Since the main concept is to use the observations to estimate the states, a mathematical model is needed to related these two set of variable to each other. Such a model is usually referred to as the *observation model* or *measurement equation*, which has a general form as follows:

$$\boldsymbol{l}_k = \boldsymbol{h}\left(\boldsymbol{x}_k\right) + \boldsymbol{\nu}_k, \tag{2.10}$$

where $\boldsymbol{h}$ represents a vector of nonlinear functions corresponding to the observations in $\boldsymbol{l}_k$. Additionally, $\boldsymbol{\nu}_k$ denotes the observation noise with an expected value of $\mathbf{0}$ and VCM $\boldsymbol{Q}$ such that $\boldsymbol{\nu} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q})$.
Occasionally, besides observations, some information might be available indicating how the state evolves in the environment (Thrun et al., 2006). Such information in filtering is usually referred to as the *control input*. For example, if the location of a moving object is of interest, its velocity can serve as the control data. A control input vector is denoted as:

$$\boldsymbol{u}_k = \begin{bmatrix} u_1 & u_2 & u_3 & \cdots & u_g \end{bmatrix}^T, \quad \boldsymbol{u} \in \mathbb{R}^g, \tag{2.11}$$

where u_g represents the g^{th} control input.
In the Bayes filter, states, observations and control input data are all treated as random variables. The primary goal is to derive the PDF of the states at each epoch k given all the observations and control input data up to that epoch. Considering the definitions given by equations (2.7) to (2.11) and using the Bayes rule given by equation (2.6), the core mathematical formulation of the Bayes filter emerges as:

$$\begin{aligned} \boldsymbol{p}\left(\boldsymbol{x}_k \mid \boldsymbol{l}_{1:k}, \boldsymbol{u}_{1:k}\right) &= \frac{\boldsymbol{p}\left(\boldsymbol{l}_k \mid \boldsymbol{x}_k, \boldsymbol{l}_{1:k-1}, \boldsymbol{u}_{1:k}\right)\ \boldsymbol{p}\left(\boldsymbol{x}_k \mid \boldsymbol{l}_{1:k-1}, \boldsymbol{u}_{1:k}\right)}{\boldsymbol{p}\left(\boldsymbol{l}_k \mid \boldsymbol{l}_{1:k-1}, \boldsymbol{u}_{1:k}\right)} \\ &= \boldsymbol{\eta}\ \boldsymbol{p}\left(\boldsymbol{l}_k \mid \boldsymbol{x}_k, \boldsymbol{l}_{1:k-1}, \boldsymbol{u}_{1:k}\right)\ \boldsymbol{p}\left(\boldsymbol{x}_k \mid \boldsymbol{l}_{1:k-1}, \boldsymbol{u}_{1:k}\right), \end{aligned} \tag{2.12}$$

where $\boldsymbol{p}\left(\boldsymbol{x}_k \mid \boldsymbol{l}_{1:k}, \boldsymbol{u}_{1:k}\right)$ represents the posterior distribution of the current states. Moreover, $\boldsymbol{p}\left(\boldsymbol{l}_k \mid \boldsymbol{x}_k, \boldsymbol{l}_{1:k-1}, \boldsymbol{u}_{1:k}\right)$ is the likelihood, which in the filtering context is also referred to as the *measurement probability*. Furthermore, $\boldsymbol{p}\left(\boldsymbol{x}_k \mid \boldsymbol{l}_{1:k-1}, \boldsymbol{u}_{1:k}\right)$ represents the prior distribution of the states. Finally, $\boldsymbol{\eta}$ is the normalization factor, which governs that $\boldsymbol{p}\left(\boldsymbol{x}_k \mid \boldsymbol{l}_{1:k}, \boldsymbol{u}_{1:k}\right)$ integrates to 1. Due to the need to consider all the available information up the current epoch, solving equation (2.12) is intractable.
The basic assumption in Bayes filter is that the temporal processes exhibit a characteristic known as the *Markov chain*. According to this principle, information from just the previous time-step

is adequate for estimating the current states, which governs the concept of *complete states*. As explained by Thrun et al. (2006), "a state $\boldsymbol{x}_k$ will be called *complete* if it is the best predictor of the future". The concept of complete state holds only theoretically. In real-world scenarios, listing all the existing variables that can have an influence on the states at each epoch in time is either impossible or it leads to a significant complexity. Therefore, usually a subset of all the state variables is used, which in turn leads to having an *incomplete state*. Even though the mathematical derivation of Bayes filter is based on the concept of complete state, it is proven that violations to such an assumption does not significantly affect this theorem (Thrun et al., 2006).
To better understand the Bayes filter, the *dynamic Bayes network (DBN)* is shown in Figure 2.1. By means of this model, the probabilistic evolution of states and observations over time can be visualized. It can be seen that by means of the states in the previous epoch ($\boldsymbol{x}_{k-1}$) and the current control data ($\boldsymbol{u}_k$), the current states ($\boldsymbol{x}_k$) can be derived. As a general term, this step in which the current states are estimated without using the observations is referred to as the *prediction step*. Such an estimation can then be modified by taking the current observations ($\boldsymbol{l}_k$) into account, which is referred to as the *update step*.

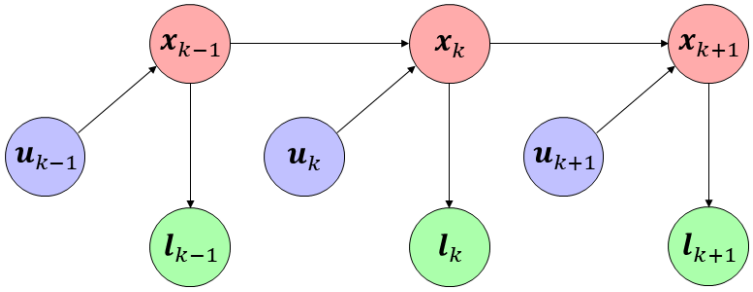

Figure 2.1: *Dynamic Bayes network or Hidden Markov model (adapted from Thrun et al. (2006)).*

Considering the given Bayes network in Figure 2.1 and exploiting the Markov chain and complete state assumptions, the likelihood notation in equation (2.12) is simplified as follows:

$$\boldsymbol{p}\left(\boldsymbol{l}_k \mid \boldsymbol{x}_k, \boldsymbol{l}_{1:k-1}, \boldsymbol{u}_{1:k}\right) = \boldsymbol{p}\left(\boldsymbol{l}_k \mid \boldsymbol{x}_k\right). \tag{2.13}$$

As it could be seen, any information related to the past ($\boldsymbol{l}_{1:k-1}$ and $\boldsymbol{u}_{1:k-1}$) as well as the control data in the current epoch ($\boldsymbol{u}_k$) is embedded in $\boldsymbol{x}_k$. The main reason for doing so is the assumption that at this stage the control input is already considered, when the likelihood is used to update the states.
Moreover, the complete state assumption leads to the following equation for the prior distribution:

$$\begin{aligned}\boldsymbol{p}\left(\boldsymbol{x}_k \mid \boldsymbol{l}_{1:k-1}, \boldsymbol{u}_{1:k}\right) &= \int \boldsymbol{p}\left(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}, \boldsymbol{l}_{1:k-1}, \boldsymbol{u}_{1:k}\right)\ \boldsymbol{p}\left(\boldsymbol{x}_{k-1} \mid \boldsymbol{l}_{1:k-1}, \boldsymbol{u}_{1:k-1}\right)\ \boldsymbol{dx}_{k-1} \\ &= \int \boldsymbol{p}\left(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}, \boldsymbol{u}_k\right)\ \boldsymbol{p}\left(\boldsymbol{x}_{k-1} \mid \boldsymbol{l}_{1:k-1}, \boldsymbol{u}_{1:k-1}\right)\ \boldsymbol{dx}_{k-1},\end{aligned} \tag{2.14}$$

where $\boldsymbol{p}\left(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}, \boldsymbol{u}_k\right)$ is the *transition probability* of the states. Moreover, $\boldsymbol{p}\left(\boldsymbol{x}_{k-1} \mid \boldsymbol{l}_{1:k-1}, \boldsymbol{u}_{1:k-1}\right)$ is the posterior distribution of the states in the previous epoch. In simple words, solving the integral in equation (2.14) results in propagating the posterior state distributions from the previous epoch to the current epoch without incorporating any recent observations. Note the presence of the $\boldsymbol{u}_k$ and $\boldsymbol{u}_{1:k-1}$ term on the right side of the equation (2.14). The reason – as also shown by the Bayes network in Figure 2.1 – is due to considering the control inputs at each epoch in which the states are to be estimated.
Finally, having the Markov chain principle leads to the following notation for $\boldsymbol{\eta}$:

$$\begin{aligned}\boldsymbol{\eta} &= \frac{1}{\boldsymbol{p}\left(\boldsymbol{l}_k \mid \boldsymbol{l}_{1:k-1}, \boldsymbol{u}_{1:k}\right)} \\ &= \frac{1}{\int \boldsymbol{p}\left(\boldsymbol{l}_k \mid \boldsymbol{x}_k, \boldsymbol{u}_{1:k}\right)\ \boldsymbol{p}\left(\boldsymbol{x}_k \mid \boldsymbol{l}_{1:k-1}, \boldsymbol{u}_{1:k}\right)\ d\boldsymbol{x}_k},\end{aligned} \tag{2.15}$$

where the concluding $\int \boldsymbol{p}\left(\boldsymbol{l}_k \mid \boldsymbol{x}_k, \boldsymbol{u}_{1:k}\right)\ \boldsymbol{p}\left(\boldsymbol{x}_k \mid \boldsymbol{l}_{1:k-1}, \boldsymbol{u}_{1:k}\right)\ d\boldsymbol{x}_k$ term for $\boldsymbol{p}\left(\boldsymbol{l}_k \mid \boldsymbol{l}_{1:k-1}, \boldsymbol{u}_{1:k}\right)$ is based on the *Chapman-Kolmogorov* principle. This principle, which holds for Markovian processes, enables transiting from one state to the other via all the possible intermediates (Metzler, 2000). Having equations (2.13) and (2.14), the general formulae of Bayes filter given by equation (2.12) can be written as follows:

$$\underbrace{\boldsymbol{p}\left(\boldsymbol{x}_k \mid \boldsymbol{l}_{1:k}, \boldsymbol{u}_{1:k}\right)}_{\boldsymbol{bel}(\boldsymbol{x}_k)} = \boldsymbol{\eta}\ \boldsymbol{p}\left(\boldsymbol{l}_k \mid \boldsymbol{x}_k\right) \underbrace{\int \boldsymbol{p}\left(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}, \boldsymbol{u}_k\right)\ \overbrace{\boldsymbol{p}\left(\boldsymbol{x}_{k-1} \mid \boldsymbol{l}_{1:k-1}, \boldsymbol{u}_{1:k-1}\right)}^{\boldsymbol{bel}\left(\boldsymbol{x}_{k-1}\right)}\ d\boldsymbol{x}_{k-1}}_{\overline{\boldsymbol{bel}}(\boldsymbol{x}_k)}. \tag{2.16}$$

In Algorithm 1, the pseudo-code of the Bayes filter for each epoch k in time is given. As given in line 1, the output is a set of posterior distributions – shown by $\boldsymbol{bel}\left(\boldsymbol{x}_k\right)$ – each of which corresponding to a specific state in the state vector $\boldsymbol{x}_k$. For that, the posterior distributions of the previous epoch ($\boldsymbol{bel}\left(\boldsymbol{x}_{k-1}\right)$), the current observation vector ($\boldsymbol{l}_k$) and the current control vector ($\boldsymbol{u}_k$) should be taken as input. According to lines 2 to 5, for each state in the state vector $\boldsymbol{x}_k$ the posterior distribution should first be predicted (line 3). After the prediction step, by considering the current observations ($\boldsymbol{l}_k$), the predicted distribution should be updated (line 4). Performing these two steps for all of the states results in a collection of posterior distributions, which is returned as output (line 6).

Algorithm 1: Pseudo-code of the Bayes filter adapted from Thrun et al. (2006).

1 $\boldsymbol{bel}\left(\boldsymbol{x}_k\right) = \textbf{Bayes\_filter}\Big(\boldsymbol{bel}\left(\boldsymbol{x}_{k-1}\right),\ \boldsymbol{l}_k,\ \boldsymbol{u}_k\Big)$

2 **for** *all* $\boldsymbol{x}_k$ **do**

Prediction step

3 $\overline{bel}\left(x_k\right) = \int p\left(x_k \mid x_{k-1}, u_k\right)\ bel\left(x_{k-1}\right)$ `equation (2.14)`

Update step

4 $bel\left(x_k\right) = \eta\ p\left(\boldsymbol{l}_k \mid x_k\right)\ \overline{bel}\left(x_k\right)$ `equation (2.16)`

5 **end**

6 return $\boldsymbol{bel}\left(\boldsymbol{x}_k\right)$

According to equation (2.16), to derive the posterior distribution of the states in Bayes filter, the probability distribution of the prior knowledge about the initial states ($\boldsymbol{p}\left(\boldsymbol{x}_{k-1} \mid \boldsymbol{l}_{1:k-1}, \boldsymbol{u}_{1:k-1}\right)$), the state transition probabilities ($\boldsymbol{p}\left(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}, \boldsymbol{u}_k\right)$) and the measurement probabilities ($\boldsymbol{p}\left(\boldsymbol{l}_k \mid \boldsymbol{x}_k\right)$) are required. In general, except for some highly specialized cases, the posterior distribution can only be approximated (Thrun et al., 2006). In other words, deriving an analytical solution to equation (2.16) in most practical applications is not possible. Therefore, depending on the assumptions and how the involved distributions are defined, there are different ways to implement the Bayes filter. In sections 2.2 and 2.3, two of the well-known strategies for this purpose, namely the KF and PF, are explained in detail.

2.2 Kalman Filter

A well-known technique to numerically solve the Bayes filter under certain assumptions is the KF. The basis of this framework is that the posterior distributions are Gaussian. Furthermore, it requires having *linear* functions in the prediction and update steps, which forms the LKF explained in Section 2.2.1. In case of having *nonlinear* functions, the Gaussian distribution assumption will be violated. In this case, different strategies should be taken that ensure having Gaussian posterior distributions. Of those techniques, the EKF, IEKF, UKF and EnKF can be mentioned. Due to the relevance of the EKF and IEKF to the current work, they are explained in Section 2.2.2.

2.2.1 Linear Kalman Filter

The LKF is the simplest form of the KF. It is an optimal estimator if all the engaged functions are linear and the PDFs of the process and measurement noise obey a Gaussian distribution. As a general term, Gaussian distributions over vectors are referred to as *multivariate* (Thrun et al., 2006). For a state vector $\boldsymbol{x}$, the multivariate Gaussian distribution has a notation as follows:

$$\boldsymbol{p}\left(\boldsymbol{x}\right) = \mathbf{det}\Big(2\pi\boldsymbol{\Sigma}\Big)^{-\frac{1}{2}}\,\mathbf{exp}\Bigg(-\frac{1}{2}\Big(\boldsymbol{x}-\boldsymbol{\mu}\Big)^{T}\cdot\,\boldsymbol{\Sigma}^{-1}\cdot\,\Big(\boldsymbol{x}-\boldsymbol{\mu}\Big)\Bigg), \tag{2.17}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean and VCM, respectively. Having these two moments, a multivariate Gaussian distribution can be derived in closed-form. Therefore, in the KF framework, the problem of deriving posterior distributions is reduced to estimating $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$.
According to Algorithm 1 line 3, the first step at each epoch k is to derive the predicted posterior $\overline{\boldsymbol{bel}}\left(\boldsymbol{x}_k\right)$. In LKF, to ensure having a multivariate Gaussian distribution for $\overline{\boldsymbol{bel}}\left(\boldsymbol{x}_k\right)$, the state transition probability $\boldsymbol{p}\left(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}, \boldsymbol{u}_k\right)$ needs to be Gaussian. For that purpose, a function is needed that is linear in its arguments with additive Gaussian noise. Such a function can be obtained by linearizing the state transition function as given by equation (2.8), which yields the following:

$$\boldsymbol{x}_k = \boldsymbol{F}_k \cdot \boldsymbol{x}_{k-1} + \boldsymbol{G}_k \cdot \boldsymbol{u}_k + \boldsymbol{w}_k, \tag{2.18}$$

where $\boldsymbol{F}$ is a matrix of size $m \times m$. Moreover, $\boldsymbol{G}$ is a matrix of size $m \times g$ with g being the size of the control vector.
Equation (2.18) and $\boldsymbol{R}$ can be taken as the first and second moments of the state transition probability $\boldsymbol{p}\left(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}, \boldsymbol{u}_k\right)$, respectively. Therefore, by substituting it in equation (2.17), the mathematical representation of the state transition probability can be derived as follows:

$$\begin{aligned}\boldsymbol{p}\left(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}, \boldsymbol{u}_k\right) = \mathbf{det}\Big(2\pi\boldsymbol{R}_k\Big)^{-\frac{1}{2}}\\ \mathbf{exp}\Bigg(-\frac{1}{2}\Big(\boldsymbol{x}_k - \boldsymbol{F}_k\cdot\boldsymbol{x}_{k-1} - \boldsymbol{G}_k\cdot\boldsymbol{u}_k\Big)^{T}\cdot\,\boldsymbol{R}_k^{-1}\cdot\,\Big(\boldsymbol{x}_k - \boldsymbol{F}_k\cdot\boldsymbol{x}_{k-1} - \boldsymbol{G}_k\cdot\boldsymbol{u}_k\Big)\Bigg).\end{aligned} \tag{2.19}$$

Using equation (2.14) and by substituting equation (2.19), the predicted belief $\overline{\boldsymbol{bel}}\left(\boldsymbol{x}_k\right)$ is derived as follows:

$$\begin{aligned}\overline{\boldsymbol{bel}}\left(\boldsymbol{x}_k\right) &= \int \underbrace{\boldsymbol{p}\left(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}, \boldsymbol{u}_k\right)}_{\sim \mathcal{N}\left(\boldsymbol{x}_k;\, \boldsymbol{F}_k \cdot \boldsymbol{x}_{k-1} + \boldsymbol{G}_k \cdot \boldsymbol{u}_k,\, \boldsymbol{R}_k\right)} \underbrace{\boldsymbol{bel}\left(\boldsymbol{x}_{k-1}\right)}_{\sim \mathcal{N}\left(\boldsymbol{x}_{k-1};\, \boldsymbol{\mu}_{k-1},\, \boldsymbol{\Sigma}_{k-1}\right)} \boldsymbol{dx}_{k-1} \\ &= \boldsymbol{\eta} \int \mathbf{exp}\Big(-\boldsymbol{L}_k\Big)\, \boldsymbol{dx}_{k-1},\end{aligned} \tag{2.20}$$

where $\boldsymbol{L}_k$ is:

$$\begin{aligned}\boldsymbol{L}_k = \frac{1}{2}\Big(\boldsymbol{x}_k - \boldsymbol{F}_k \cdot \boldsymbol{x}_{k-1} - \boldsymbol{G}_k \cdot \boldsymbol{u}_k\Big)^T \cdot\ \boldsymbol{R}_k^{-1} \cdot\ \Big(\boldsymbol{x}_k - \boldsymbol{F}_k \cdot \boldsymbol{x}_{k-1} - \boldsymbol{G}_k \cdot \boldsymbol{u}_k\Big) \\ + \frac{1}{2}\Big(\boldsymbol{x}_{k-1} - \boldsymbol{\mu}_{k-1}\Big)^T \cdot\ \boldsymbol{\Sigma}_{k-1}^{-1} \cdot\ \Big(\boldsymbol{x}_{k-1} - \boldsymbol{\mu}_{k-1}\Big).\end{aligned} \tag{2.21}$$

In equation (2.20), $\mathcal{N}\left(\boldsymbol{x}_k;\ \boldsymbol{F}_k \cdot \boldsymbol{x}_{k-1} + \boldsymbol{G}_k \cdot \boldsymbol{u}_k,\ \boldsymbol{R}_k\right)$ represents a Gaussian distribution for $\boldsymbol{x}_k$ with an expected value of $\boldsymbol{F}_k \cdot \boldsymbol{x}_{k-1} + \boldsymbol{G}_k \cdot \boldsymbol{u}_k$ and a VCM of $\boldsymbol{R}_k$. Such a mathematical interpretation holds for the similar notations in the current thesis.
Note that the belief distribution $\boldsymbol{bel}\left(\boldsymbol{x}_{k-1}\right)$ is a multivariate Gaussian distribution with mean $\boldsymbol{\mu}_{k-1}$ and VCM $\boldsymbol{\Sigma}_{k-1}$. The integral in equation (2.20) can be solved by defining a function that depends on $\boldsymbol{x}_{k-1}$. To do so, the first and second derivatives of the function $\boldsymbol{L}_k$ with respect to $\boldsymbol{x}_{k-1}$ can be used. After carrying out all the mathematical steps (given e.g. by Thrun et al. (2006)), it can be seen that $\overline{\boldsymbol{bel}}\left(\boldsymbol{x}_k\right)$ follows a multivariate Gaussian distribution with a mean vector $\overline{\boldsymbol{\mu}}_k$ and VCM $\overline{\boldsymbol{\Sigma}}_k$ as follows:

$$\overline{\boldsymbol{\mu}}_k = \boldsymbol{F}_k \cdot \boldsymbol{\mu}_{k-1} + \boldsymbol{G}_k \cdot \boldsymbol{u}_k. \tag{2.22}$$

$$\overline{\boldsymbol{\Sigma}}_k = \boldsymbol{F}_k \cdot \boldsymbol{\Sigma}_{k-1} \cdot \boldsymbol{F}_k^T + \boldsymbol{R}_k. \tag{2.23}$$

In the second step, as outlined in Algorithm 1 line 4, the obtained $\overline{\boldsymbol{bel}}\left(\boldsymbol{x}_k\right)$ needs to be updated by using the measurement probability $\boldsymbol{p}\left(\boldsymbol{l}_k \mid \boldsymbol{x}_k\right)$. In the LKF, in order to have a multivariate Gaussian distribution for $\boldsymbol{bel}\left(\boldsymbol{x}_k\right)$, it has to be ensured that $\boldsymbol{p}\left(\boldsymbol{l}_k \mid \boldsymbol{x}_k\right)$ is a multivariate Gaussian distribution. This necessitates a function that is linear in its arguments with additive Gaussian noise. Such a function can be derived by linearizing the observation model given by equation (2.10), which yields the following:

$$\boldsymbol{l}_k = \boldsymbol{H}_k \cdot \boldsymbol{x}_k + \boldsymbol{\nu}_k. \tag{2.24}$$

In equation (2.24), $\boldsymbol{H}$ is a matrix of size $n \times m$, where n represents the size of the observation vector. This equation effectively serves as the mean of the multivariate Gaussian distribution $\boldsymbol{p}\left(\boldsymbol{l}_k \mid \boldsymbol{x}_k\right)$. Hence substituting it in equation (2.17) yields:

$$\boldsymbol{p}\left(\boldsymbol{l}_k \mid \boldsymbol{x}_k\right) = \mathbf{det}\Big(2\pi \boldsymbol{Q}_k\Big)^{-\frac{1}{2}} \mathbf{exp}\left(-\frac{1}{2}\Big(\boldsymbol{l}_k - \boldsymbol{H}_k \cdot \boldsymbol{x}_k\Big)^T \cdot\ \boldsymbol{Q}_k^{-1} \cdot\ \Big(\boldsymbol{l}_k - \boldsymbol{H}_k \cdot \boldsymbol{x}_k\Big)\right). \tag{2.25}$$

Using equation (2.16) and by substituting equation (2.25), the posterior belief $\boldsymbol{bel}\left(\boldsymbol{x}_k\right)$ is derived as follows:

$$\begin{aligned} \boldsymbol{bel}\left(\boldsymbol{x}_k\right) &= \boldsymbol{\eta} \quad \underbrace{\boldsymbol{p}\left(\boldsymbol{l}_k \mid \boldsymbol{x}_k\right)}_{\sim \mathcal{N}(\boldsymbol{l}_k;\, \boldsymbol{H}_k \cdot \boldsymbol{x}_k,\, \boldsymbol{Q}_k)} \quad \underbrace{\overline{\boldsymbol{bel}}\left(\boldsymbol{x}_k\right)}_{\sim \mathcal{N}\left(\boldsymbol{x}_k;\, \overline{\boldsymbol{\mu}}_k,\, \overline{\boldsymbol{\Sigma}}_k\right)} \\ &= \boldsymbol{\eta}\ \mathbf{exp}\Big(- \boldsymbol{J}_k\Big), \end{aligned} \tag{2.26}$$

where $\boldsymbol{J}_k$ is:

$$\boldsymbol{J}_k = \frac{1}{2}\Big(\boldsymbol{l}_k - \boldsymbol{H}_k \cdot \boldsymbol{x}_k\Big)^T \cdot\ \boldsymbol{Q}_k^{-1} \cdot\ \Big(\boldsymbol{l}_k - \boldsymbol{H}_k \cdot \boldsymbol{x}_k\Big) + \frac{1}{2}\Big(\boldsymbol{x}_k - \overline{\boldsymbol{\mu}}_k\Big)^T \cdot\ \overline{\boldsymbol{\Sigma}}_k^{-1} \cdot\ \Big(\boldsymbol{x}_k - \overline{\boldsymbol{\mu}}_k\Big). \tag{2.27}$$

Equation (2.27) is quadratic with respect to $\boldsymbol{x}_k$. Thus, $\boldsymbol{bel}\left(\boldsymbol{x}_k\right)$ is characterized as a multivariate Gaussian. To determine its parameters, both the first and second derivatives of $\boldsymbol{J}_k$ should be employed. Upon completing the necessary mathematical operations (given e.g. by Thrun et al. (2006)), the mean vector $\boldsymbol{\mu}_k$ and VCM denoted by $\boldsymbol{\Sigma}_k$, can be expressed as follows:

$$\boldsymbol{\mu}_k = \overline{\boldsymbol{\mu}}_k + \boldsymbol{K}_k \cdot \Big(\boldsymbol{l}_k - \boldsymbol{H}_k \cdot \overline{\boldsymbol{\mu}}_k\Big). \tag{2.28}$$

$$\boldsymbol{\Sigma}_k = \Big(\boldsymbol{I} - \boldsymbol{K}_k \cdot \boldsymbol{H}_k\Big) \cdot \overline{\boldsymbol{\Sigma}}_k. \tag{2.29}$$

In equations (2.28) and (2.29), $\boldsymbol{K}_k$ is called the *Kalman gain*, which is calculated as follows:

$$\boldsymbol{K}_k = \overline{\boldsymbol{\Sigma}}_k \cdot \boldsymbol{H}_k^T \cdot \Big(\boldsymbol{H}_k \cdot \overline{\boldsymbol{\Sigma}}_k \cdot \boldsymbol{H}_k^T + \boldsymbol{Q}_k\Big)^{-1}. \tag{2.30}$$

Algorithm 2 summarizes the aforementioned derivations discussed earlier. As given in line 1, the input variables include the mean and VCM of the multivariate Gaussian distribution from the previous epoch, represented by ($\boldsymbol{\mu}_{k-1}$ and $\boldsymbol{\Sigma}_{k-1}$) respectively, along with the observation vector and control vector of the current epoch, denoted as ($\boldsymbol{l}_k$ and $\boldsymbol{u}_k$). It is worth noting that lines 2 to 3 in this algorithm correspond to line 3 in Algorithm 1. Moreover, lines 4 to 6 of Algorithm 2 correspond to line 4 in Algorithm 1.

Algorithm 2: Pseudo-code of the linear Kalman filter adapted from Thrun et al. (2006).

1 $\left[\boldsymbol{\mu}_k,\ \boldsymbol{\Sigma}_k\right] = \textbf{linear\_Kalman\_filter}\left(\boldsymbol{\mu}_{k-1},\ \boldsymbol{\Sigma}_{k-1},\ \boldsymbol{l}_k,\ \boldsymbol{u}_k\right)$

Prediction step

2 $\overline{\boldsymbol{\mu}}_k = \boldsymbol{F}_k \cdot \boldsymbol{\mu}_{k-1} + \boldsymbol{G}_k \cdot \boldsymbol{u}_k$ `equation (2.22)`

3 $\overline{\boldsymbol{\Sigma}}_k = \boldsymbol{F}_k \cdot \boldsymbol{\Sigma}_{k-1} \cdot \boldsymbol{F}_k^T + \boldsymbol{R}_k$ `equation (2.23)`

Update step

4 $\boldsymbol{K}_k = \overline{\boldsymbol{\Sigma}}_k \cdot \boldsymbol{H}_k^T \cdot \left(\boldsymbol{H}_k \cdot \overline{\boldsymbol{\Sigma}}_k \cdot \boldsymbol{H}_k^T + \boldsymbol{Q}_k\right)^{-1}$ `equation (2.30)`

5 $\boldsymbol{\mu}_k = \overline{\boldsymbol{\mu}}_k + \boldsymbol{K}_k \cdot \left(\boldsymbol{l}_k - \boldsymbol{H}_k \cdot \overline{\boldsymbol{\mu}}_k\right)$ `equation (2.28)`

6 $\boldsymbol{\Sigma}_k = \left(\boldsymbol{I} - \boldsymbol{K}_k \cdot \boldsymbol{H}_k\right) \cdot \overline{\boldsymbol{\Sigma}}_k$ `equation (2.29)`

7 return $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$

2.2.2 (Iterated) Extended Kalman Filter

The LKF is an optimal estimator if both the state transition function and observation model are linear. In such a case, the strict assumption of having a Gaussian distribution for all the involved variables should be met. However, in real-world applications, the existing functions are usually nonlinear. Therefore, instead of having linear equations such as (2.18) and (2.24), the state transition function and observation model might have nonlinear forms as given by equations (2.8) and (2.10).

Employing nonlinear functions often results in non-Gaussian distributions. Figure 2.2 depicts such an effect for a one-dimensional random variable x, which is assumed to have a Gaussian distribution $p(x)$, depicted in blue. From Figure 2.2a, it is evident that when transforming x via a linear function such as $a \cdot x + b$ leads to a random variable y, which retains a Gaussian distribution $p(y)$, shown in green. However, Figure 2.2b employs a nonlinear function ($g(x)$). This nonlinear transformation causes the resulting distribution y to deviate from Gaussian. By utilizing the MC method to generate a large set of samples and compute their mean and covariance, we obtain the MC estimates. The derived Gaussian representation of $p(y)$ is then visualized as a black curve in the upper left part of Figure 2.2b. In case of more information regarding the MC method or the process of obtaining these estimates, related sources such as the work by Kroese et al. (2013) can be used. It should be noted that the MC methods are used to derive the true posterior distributions that can have any other shape than Gaussian. However, the main purpose here is to realize how well the EKF can obtain the true distribution. For that purpose, the MC estimates are used to represent the true distribution as Gaussian. The obtained result of the EKF is then compared with this distribution to evaluate its performance. The main idea of EKF is to approximate such a Gaussian distribution by applying Taylor series expansion to $g(x)$. Doing so, as depicted in the upper right part of Figure 2.2c, a linear function tangent to $g(x)$ at the mean of $p(x)$ can be derived (red line), which in turn leads to a linear transformation of x. The dashed red line in the upper left part of Figure 2.2c shows the output of EKF, which is the mean of a Gaussian distribution that is shown by a red curve.

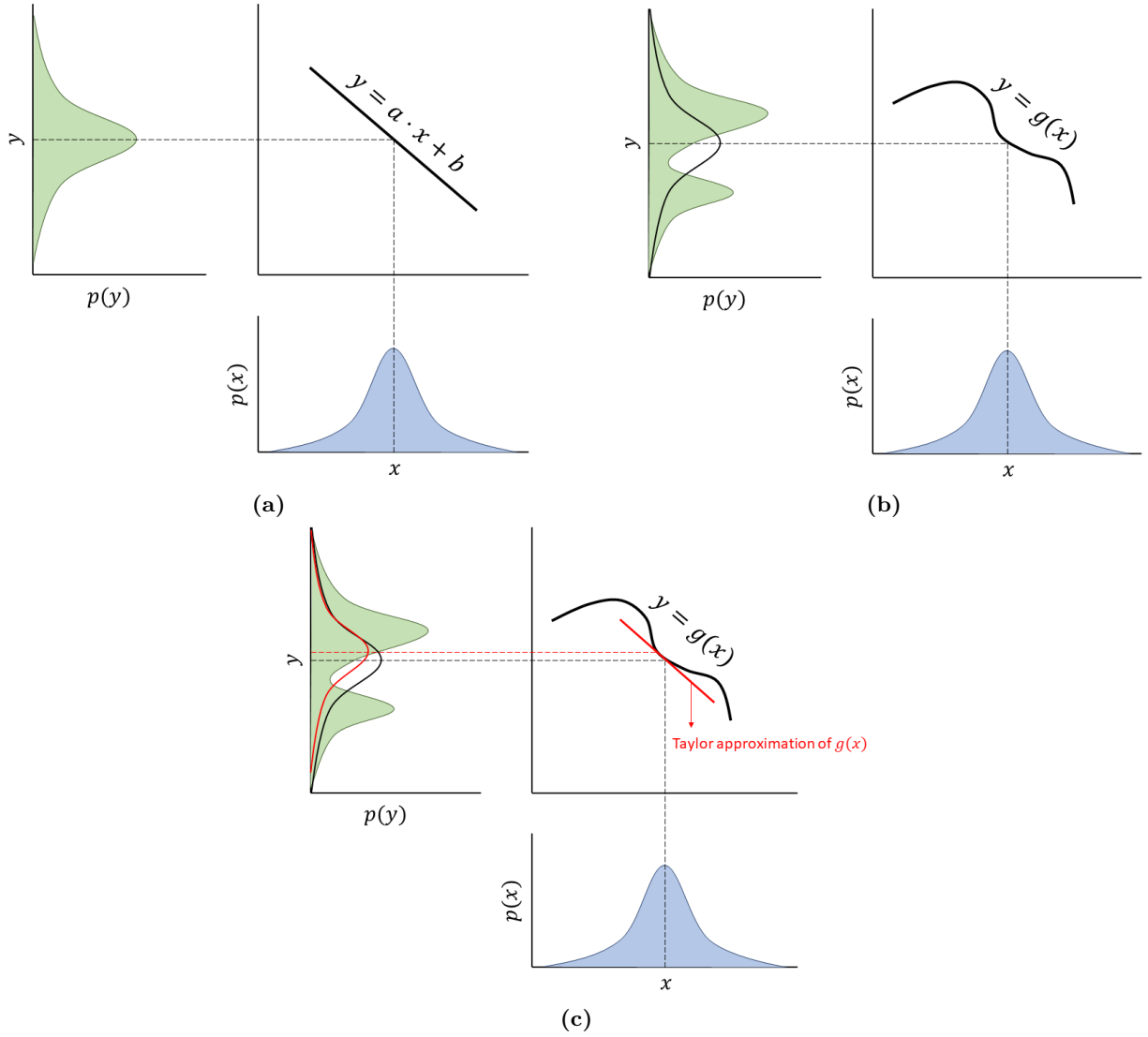

Figure 2.2: *The effect of: (a) linear, (b) nonlinear, and (c) linearized transformation of a one-dimensional normally distributed random variable x on the resulting random variable y (adapted from Thrun et al. (2006)).*

Using the first degree of Taylor series expansion, the nonlinear state transition function $\boldsymbol{f}\left(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k\right)$ at the most recent estimate $(\boldsymbol{\mu}_{k-1})$ can be approximated by a line as follows:

$$\begin{aligned} \boldsymbol{f}\left(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k\right) &\approx \boldsymbol{f}\left(\boldsymbol{\mu}_{k-1}, \boldsymbol{u}_k\right) + \underbrace{\boldsymbol{f}'\left(\boldsymbol{\mu}_{k-1}, \boldsymbol{u}_k\right)}_{:=\boldsymbol{F}_{x,k}} \cdot \Big(\boldsymbol{x}_{k-1} - \boldsymbol{\mu}_{k-1}\Big) \\ &= \boldsymbol{f}\left(\boldsymbol{\mu}_{k-1}, \boldsymbol{u}_k\right) + \boldsymbol{F}_{x,k} \cdot \Big(\boldsymbol{x}_{k-1} - \boldsymbol{\mu}_{k-1}\Big), \end{aligned} \tag{2.31}$$

where $\boldsymbol{F}_{x,k}$ is a matrix of size $m \times m$ called *Jacobian*, which is calculated as follows:

$$\boldsymbol{F}_{x,k} := \frac{\partial \boldsymbol{f}\left(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k\right)}{\partial \boldsymbol{x}_{k-1}}. \tag{2.32}$$

By substituting equation (2.31) in (2.17), the mathematical representation of the state transition

probability is derived as follows:

$$\boldsymbol{p}\left(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}, \boldsymbol{u}_k\right) \approx \mathbf{det}\left(2\pi \boldsymbol{R}_k\right)^{-\frac{1}{2}} \mathbf{exp}\left(-\frac{1}{2}\left[\boldsymbol{x}_k - \boldsymbol{f}\left(\boldsymbol{\mu}_{k-1}, \boldsymbol{u}_k\right) - \boldsymbol{F}_{x,k} \cdot \left(\boldsymbol{x}_{k-1} - \boldsymbol{\mu}_{k-1}\right)\right]^T \cdot \boldsymbol{R}_k^{-1} \cdot \left[\boldsymbol{x}_k - \boldsymbol{f}\left(\boldsymbol{\mu}_{k-1}, \boldsymbol{u}_k\right) - \boldsymbol{F}_{x,k} \cdot \left(\boldsymbol{x}_{k-1} - \boldsymbol{\mu}_{k-1}\right)\right]\right). \quad (2.33)$$

With a similar strategy explained in Section 2.2.1, by substituting equation (2.33) in (2.14), the predicted $\overline{\boldsymbol{bel}}\left(\boldsymbol{x}_k\right)$ proves to be a multivariate Gaussian distribution with the following mean and VCM:

$$\overline{\boldsymbol{\mu}}_k = \boldsymbol{f}(\boldsymbol{\mu}_{k-1}, \boldsymbol{u}_k). \quad (2.34)$$

$$\overline{\boldsymbol{\Sigma}}_k = \boldsymbol{F}_{x,k} \cdot \boldsymbol{\Sigma}_{k-1} \cdot \boldsymbol{F}_{x,k}^T + \boldsymbol{R}_k. \quad (2.35)$$

Applying the first degree of Taylor series expansion on the nonlinear observation model $\boldsymbol{h}\left(\boldsymbol{x}_k\right)$ at the most recent estimate $(\overline{\boldsymbol{\mu}}_k)$ results in the following approximation:

$$\begin{aligned} \boldsymbol{h}\left(\boldsymbol{x}_k\right) &\approx \boldsymbol{h}\left(\overline{\boldsymbol{\mu}}_k\right) + \underbrace{\boldsymbol{h}'\left(\overline{\boldsymbol{\mu}}_k\right)}_{\coloneqq \boldsymbol{H}_{x,k}} \cdot \left(\boldsymbol{x}_k - \overline{\boldsymbol{\mu}}_k\right) \\ &= \boldsymbol{h}\left(\overline{\boldsymbol{\mu}}_k\right) + \boldsymbol{H}_{x,k} \cdot \left(\boldsymbol{x}_k - \overline{\boldsymbol{\mu}}_k\right), \end{aligned} \quad (2.36)$$

where $\boldsymbol{H}_{x,k}$ is a matrix of size $n \times m$, which is calculated as follows:

$$\boldsymbol{H}_{x,k} \coloneqq \frac{\partial \boldsymbol{h}\left(\boldsymbol{x}_k\right)}{\partial \boldsymbol{x}_k}. \quad (2.37)$$

By substituting equation (2.36) in (2.17), the mathematical representation of the measurement probability is derived as follows:

$$\boldsymbol{p}\left(\boldsymbol{l}_k \mid \boldsymbol{x}_k\right) = \mathbf{det}\left(2\pi \boldsymbol{Q}_k\right)^{-\frac{1}{2}} \mathbf{exp}\left(-\frac{1}{2}\left[\boldsymbol{l}_k - \boldsymbol{h}\left(\overline{\boldsymbol{\mu}}_k\right) - \boldsymbol{H}_{x,k} \cdot \left(\boldsymbol{x}_k - \overline{\boldsymbol{\mu}}_k\right)\right]^T \cdot \boldsymbol{Q}_k^{-1} \cdot \left[\boldsymbol{l}_k - \boldsymbol{h}\left(\overline{\boldsymbol{\mu}}_k\right) - \boldsymbol{H}_{x,k} \cdot \left(\boldsymbol{x}_k - \overline{\boldsymbol{\mu}}_k\right)\right]\right). \quad (2.38)$$

Similar to the strategy explained in Section 2.2.1, substituting equation (2.38) in (2.16) leads to a

normally distributed posterior belief $\boldsymbol{bel}\left(\boldsymbol{x}_k\right)$ with the following first and second moments:

$$\boldsymbol{\mu}_k = \overline{\boldsymbol{\mu}}_k + \boldsymbol{K}_k \cdot \Big(\boldsymbol{l}_k - \boldsymbol{h}\left(\overline{\boldsymbol{\mu}}_k\right) \Big). \tag{2.39}$$

$$\boldsymbol{\Sigma}_k = \Big(\boldsymbol{I} - \boldsymbol{K}_k \cdot \boldsymbol{H}_{x,k} \Big) \cdot \overline{\boldsymbol{\Sigma}}_k. \tag{2.40}$$

The Kalman gain ($\boldsymbol{K}_k$) is calculated as follows:

$$\boldsymbol{K}_k = \overline{\boldsymbol{\Sigma}}_k \cdot \boldsymbol{H}_{x,k}^T \cdot \Big(\boldsymbol{H}_{x,k} \cdot \overline{\boldsymbol{\Sigma}}_k \cdot \boldsymbol{H}_{x,k}^T + \boldsymbol{Q}_k \Big)^{-1}. \tag{2.41}$$

In Algorithm 3, pseudo-code of the EKF is given, which summarizes the above-mentioned derivations. Similar to Algorithm 2, lines 2 to 4 and lines 6 to 8 correspond to lines 3 and 4 of Algorithm 1.

Algorithm 3: Pseudo-code of the extended Kalman filter adapted from Thrun et al. (2006).

1 $\Big[\boldsymbol{\mu}_k,\ \boldsymbol{\Sigma}_k\Big] = \textbf{extended\_Kalman\_filter}\Big(\boldsymbol{\mu}_{k-1},\ \boldsymbol{\Sigma}_{k-1},\ \boldsymbol{l}_k,\ \boldsymbol{u}_k\Big)$

Prediction step

2 $\overline{\boldsymbol{\mu}}_k = \boldsymbol{f}\left(\boldsymbol{\mu}_{k-1}, \boldsymbol{u}_k\right)$ `equation (2.34)`

3 $\boldsymbol{F}_{x,k} = \dfrac{\partial \boldsymbol{f}\left(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k\right)}{\partial \boldsymbol{x}_{k-1}}$ `equation (2.32)`

4 $\overline{\boldsymbol{\Sigma}}_k = \boldsymbol{F}_{x,k} \cdot \boldsymbol{\Sigma}_{k-1} \cdot \boldsymbol{F}_{x,k}^T + \boldsymbol{R}_k$ `equation (2.35)`

Update step

5 $\boldsymbol{H}_{x,k} = \dfrac{\partial \boldsymbol{h}\left(\boldsymbol{x}_k\right)}{\partial \boldsymbol{x}_k}$ `equation (2.37)`

6 $\boldsymbol{K}_k = \overline{\boldsymbol{\Sigma}}_k \cdot \boldsymbol{H}_{x,k}^T \cdot \Big(\boldsymbol{H}_{x,k} \cdot \overline{\boldsymbol{\Sigma}}_k \cdot \boldsymbol{H}_{x,k}^T + \boldsymbol{Q}_k \Big)^{-1}$ `equation (2.41)`

7 $\boldsymbol{\mu}_k = \overline{\boldsymbol{\mu}}_k + \boldsymbol{K}_k \cdot \Big(\boldsymbol{l}_k - \boldsymbol{h}\left(\overline{\boldsymbol{\mu}}_k\right) \Big)$ `equation (2.39)`

8 $\boldsymbol{\Sigma}_k = \Big(\boldsymbol{I} - \boldsymbol{K}_k \cdot \boldsymbol{H}_{x,k} \Big) \cdot \overline{\boldsymbol{\Sigma}}_k$ `equation (2.40)`

9 return $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$

In the EKF, approximating the state transition function and observation model using Taylor series expansions leads to a *linearization error*. Such an error is an indicator of the dissimilarity between the estimated Gaussian distribution by means of the EKF and the derived Gaussian from highly accurate MC estimates. As explained by Thrun et al. (2006), both the degree of uncertainty and the degree of local nonlinearity of the involved functions influence this linearization error. Such a linearization error leads to a biased mean and a scaled uncertainty in the derived Gaussian distribution. These effects can be better explained by referring to Figure 2.2c. The deviation of the dashed red line from the dashed black line represents the linearization error. If $p\left(x\right)$ is more uncertain, such a deviation gets larger than in a case in which $p\left(x\right)$ has a higher certainty. The same

also holds for a case in which the degree of nonlinearity of $g(x)$ is higher, which leads to a larger linearization error than a case with lower degree of nonlinearity. To mitigate the linearization errors inherent in the EKF, one can apply the Taylor series expansion successively to the state considered most probable at the linearization time. The modified EKF approach is termed the IEKF, which its corresponding pseudo-code is given by Algorithm 4. In this algorithm, C indicates the total number of iterations that the EKF should be applied.

Algorithm 4: Pseudo-code of the iterated extended Kalman filter adapted from Thrun et al. (2006).

$\left[\boldsymbol{\mu}_k,\ \boldsymbol{\Sigma}_k\right] = \textbf{iterated\_extended\_Kalman\_filter}\left(\boldsymbol{\mu}_{k-1},\ \boldsymbol{\Sigma}_{k-1},\ \boldsymbol{l}_k,\ \boldsymbol{u}_k\right)$

Prediction step

$\overline{\boldsymbol{\mu}}_k = \boldsymbol{f}\left(\boldsymbol{\mu}_{k-1}, \boldsymbol{u}_k\right)$ `equation (2.34)`

$\boldsymbol{F}_{x,k} = \dfrac{\partial \boldsymbol{f}\left(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k\right)}{\partial \boldsymbol{x}_{k-1}}$ `equation (2.32)`

$\overline{\boldsymbol{\Sigma}}_k = \boldsymbol{F}_{x,k} \cdot \boldsymbol{\Sigma}_{k-1} \cdot \boldsymbol{F}_{x,k}^T + \boldsymbol{R}_k$ `equation (2.35)`

Update step

$\boldsymbol{\mu}_{k,0} = \overline{\boldsymbol{\mu}}_k$

$\boldsymbol{\Sigma}_{k,0} = \overline{\boldsymbol{\Sigma}}_k$

for $c = 0 : C-1$ **do**

  $\boldsymbol{K}_{k,c} = \boldsymbol{\Sigma}_{k,c} \cdot \boldsymbol{H}_{x,k,c}^T \cdot \left(\boldsymbol{H}_{x,k,c} \cdot \boldsymbol{\Sigma}_{k,c} \cdot \boldsymbol{H}_{x,k,c}^T + \boldsymbol{Q}_k\right)^{-1}$ `equation (2.41)`

  $\boldsymbol{\mu}_{k,c+1} = \boldsymbol{\mu}_{k,c} + \boldsymbol{K}_{k,c} \cdot \left(\boldsymbol{l}_k - \boldsymbol{h}\left(\boldsymbol{\mu}_{k,c}\right)\right)$ `equation (2.39)`

  $\boldsymbol{\Sigma}_{k,c+1} = \left(\boldsymbol{I} - \boldsymbol{K}_{k,c} \cdot \boldsymbol{H}_{x,k,c}\right) \cdot \boldsymbol{\Sigma}_{k,c}$ `equation (2.40)`

end

$\boldsymbol{\mu}_k = \boldsymbol{\mu}_{k,C-1}$

$\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}_{k,C-1}$

return $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$

2.2.3 Versatile (Iterated) Extended Kalman Filter

The principle of EKF as well as its iterated version (IEKF) explained in Section 2.2.2 holds for cases with explicit observation models. As mentioned in Chapter 1, in an explicit observation model, the equation directly relates the state of the system to the observed measurements. Such an observation model has a form as given by equation (2.10). In contrast, in an implicit observation model, instead of expressing the measurements as a direct function of the state, the implicit form defines a relationship between the state and the measurements as follows:

$$\boldsymbol{h}\left(\boldsymbol{l}_k + \boldsymbol{\nu}_k, \boldsymbol{x}_k\right) = \boldsymbol{0}, \tag{2.42}$$

where $\boldsymbol{h}$ represents nonlinear observation models.

The choice of whether to use an explicit or implicit measurement equation often depends on the specifics of the application and the nature of the measurements. It should be noted that every explicit observation model can indeed be rephrased in the form of an implicit model. The explicit representation is direct and often more intuitive, especially when the relationship between the state and observations is straightforward. However, the implicit model becomes particularly useful in scenarios where the relation between the state and observations is more complex or it is not easily expressible in an explicit form. In such cases, the implicit model serves as a more flexible framework, allowing to define a mathematical relation between the state and observations without explicitly solving for one variable in terms of the others.
In Dang (2007), an IEKF approach is introduced that can handle observation models of implicit type. While the Bayesian framework remains fundamentally applicable, the main challenge lies in the specific implementations. Deriving solutions based on certain methodologies, such as those used in the LKF, becomes challenging due to the implicit nature of the relationship. Specifically, defining the likelihood ($\boldsymbol{p}(\boldsymbol{l}_k \mid \boldsymbol{x}_k)$) becomes intricate. Consequently, the standard approaches to optimization, like the one represented in equation (2.27), are not directly applicable. In such scenarios, with the constraints imposed by the implicit relation, alternative methods like the use of Lagrangian multipliers become necessary to define the objective function. To achieve this, equation (2.42) has to be first linearized. According to Dang (2007), linearization of an implicit observation model by using the first degree of Taylor series expansion results in the following equation:

$$\begin{aligned}\boldsymbol{h}\left(\boldsymbol{l}_k + \boldsymbol{\nu}_k, \boldsymbol{x}_k\right) &\approx \boldsymbol{h}\left(\check{\boldsymbol{l}}_k, \check{\boldsymbol{x}}_k\right) + \boldsymbol{H}_{x,k} \cdot \left(\boldsymbol{x}_k - \check{\boldsymbol{x}}_k\right) + \boldsymbol{H}_{l,k} \cdot \left(\boldsymbol{l}_k^+ - \check{\boldsymbol{l}}_k\right) \\ &= \boldsymbol{H}_{x,k} \cdot \boldsymbol{x}_k + \boldsymbol{H}_{l,k} \cdot \boldsymbol{l}_k^+ + \underbrace{\boldsymbol{h}\left(\check{\boldsymbol{l}}_k, \check{\boldsymbol{x}}_k\right) - \boldsymbol{H}_{x,k} \cdot \check{\boldsymbol{x}}_k - \boldsymbol{H}_{l,k} \cdot \check{\boldsymbol{l}}_k}_{\boldsymbol{\delta}_k} \\ &= \boldsymbol{H}_{x,k} \cdot \boldsymbol{x}_k + \boldsymbol{H}_{l,k} \cdot \boldsymbol{l}_k^+ + \boldsymbol{\delta}_k \stackrel{!}{=} \boldsymbol{0}.\end{aligned} \tag{2.43}$$

where $\check{\boldsymbol{x}}_k$ and $\check{\boldsymbol{l}}_k$ are those state and observation vectors in which the observation model $\boldsymbol{h}$ is to be linearized. Also, $\boldsymbol{x}_k$ and $\boldsymbol{l}_k^+$ indicate the obtained state vector and the modified observation vector at epoch k, respectively. Moreover, $\boldsymbol{H}_{x,k}$ is a matrix of size $n \times m$, which is calculated by taking the partial derivative of the observation model $\boldsymbol{h}$ with respect to the states $\boldsymbol{x}_k$ as follows:

$$\boldsymbol{H}_{x,k} \coloneqq \frac{\partial \boldsymbol{h}\left(\boldsymbol{l}_k + \boldsymbol{\nu}_k, \boldsymbol{x}_k\right)}{\partial \boldsymbol{x}_k}. \tag{2.44}$$

It should be noted that in principle, equation (2.44) is similar to (2.37). The only intention to rewrite it is to show that the derivatives should be obtained with respect to the implicit observation model $\boldsymbol{h}\left(\boldsymbol{l}_k + \boldsymbol{\nu}_k, \boldsymbol{x}_k\right)$ rather than the explicit case $\boldsymbol{h}\left(\boldsymbol{x}_k\right)$.
Furthermore, $\boldsymbol{H}_{l,k}$ is a matrix of size $n \times u$, which is calculated by taking the partial derivative of the observation model $\boldsymbol{h}$ with respect to the observations $\boldsymbol{l}_k$ as follows:

$$\boldsymbol{H}_{l,k} \coloneqq \frac{\partial \boldsymbol{h}\left(\boldsymbol{l}_k + \boldsymbol{\nu}_k, \boldsymbol{x}_k\right)}{\partial \boldsymbol{l}_k}. \tag{2.45}$$

After linearizing the implicit observation model, the objective function can be defined using the Lagrangian multipliers as follows:

$$\begin{aligned}\boldsymbol{\zeta}_{IEKF-GHM} = &\begin{bmatrix} \boldsymbol{l}_k^+ - \boldsymbol{l}_k \\ \boldsymbol{x}_k - \overline{\boldsymbol{\mu}}_k \end{bmatrix}^T \begin{bmatrix} \boldsymbol{Q}_k & \boldsymbol{0} \\ \boldsymbol{0} & \overline{\boldsymbol{\Sigma}}_k \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{l}_k^+ - \boldsymbol{l}_k \\ \boldsymbol{x}_k - \overline{\boldsymbol{\mu}}_k \end{bmatrix} \\ &- 2 \cdot \boldsymbol{\lambda}_k^T \cdot \left(\boldsymbol{H}_{x,k} \cdot \boldsymbol{x}_k + \boldsymbol{H}_{l,k} \cdot \boldsymbol{l}_k^+ + \boldsymbol{\delta}_k \right),\end{aligned} \tag{2.46}$$

where $\boldsymbol{\lambda}$ is the Lagrangian multiplier. By minimizing the given objective function, the optimal states ($\boldsymbol{x}_k$) and modified observations $\left(\boldsymbol{l}_k^+\right)$ can be derived.
Similar to the explained IEKF in Section 2.2.2, iterations are employed to reduce the linearization error associated with implicit observation models. Consequently, $\check{\boldsymbol{x}}_k$ and $\check{\boldsymbol{l}}_k$ in equation (2.43) are iteratively replaced by the estimated state and observation vectors until a predefined abort criterion is reached. For the initial iteration, $\check{\boldsymbol{x}}_k$ and $\check{\boldsymbol{l}}_k$ are set as follows:

$$\begin{aligned}\check{\boldsymbol{x}}_{k,0} &= \overline{\boldsymbol{\mu}}_k \\ \check{\boldsymbol{l}}_{k,0} &= \boldsymbol{l}_k.\end{aligned} \tag{2.47}$$

In each iteration (c) and based on minimizing equation (2.46), the optimal state is calculated as follows:

$$\check{\boldsymbol{x}}_{k,c+1} = \overline{\boldsymbol{\mu}}_k - \boldsymbol{K}_{k,c} \cdot \left(\boldsymbol{h}\left(\check{\boldsymbol{l}}_{k,c}, \check{\boldsymbol{x}}_{k,c}\right) + \boldsymbol{H}_{l,k,c} \cdot \left(\boldsymbol{l}_k - \check{\boldsymbol{l}}_{k,c}\right) + \boldsymbol{H}_{x,k,c} \cdot \left(\overline{\boldsymbol{\mu}}_k - \check{\boldsymbol{x}}_{k,c}\right) \right). \tag{2.48}$$

where $\boldsymbol{K}_{k,c}$ is the Kalman gain that is derived as follows:

$$\boldsymbol{K}_{k,c} = \overline{\boldsymbol{\Sigma}}_k \cdot \boldsymbol{H}_{x,k,c}^T \cdot \left(\boldsymbol{O}_{k,c} + \boldsymbol{S}_{k,c}\right)^{-1}. \tag{2.49}$$

To obtain the optimal observations, the following relations hold:

$$\boldsymbol{O}_{k,c} = \boldsymbol{H}_{x,k,c} \cdot \overline{\boldsymbol{\Sigma}}_k \cdot \boldsymbol{H}_{x,k,c}^T, \tag{2.50}$$

$$\boldsymbol{S}_{k,c} = \boldsymbol{H}_{l,k,c} \cdot \boldsymbol{Q}_k \cdot \boldsymbol{H}_{l,k,c}^T, \tag{2.51}$$

$$\boldsymbol{G}_{k,c} = \boldsymbol{Q}_k \cdot \boldsymbol{H}_{l,k,c}^T \cdot \left(\boldsymbol{O}_{k,c} + \boldsymbol{S}_{k,c}\right)^{-1}, \tag{2.52}$$

$$\check{\boldsymbol{l}}_{k,c+1} = \boldsymbol{l}_k - \boldsymbol{G}_{k,c} \cdot \left(\boldsymbol{h}\left(\check{\boldsymbol{l}}_{k,c}, \check{\boldsymbol{x}}_{k,c}\right) + \boldsymbol{H}_{l,k,c} \cdot \left(\boldsymbol{l}_k - \check{\boldsymbol{l}}_{k,c}\right) + \boldsymbol{H}_{x,k,c} \cdot \left(\overline{\boldsymbol{\mu}}_k - \check{\boldsymbol{x}}_{k,c}\right) \right). \tag{2.53}$$

In the above equations, $\boldsymbol{H}_{x,k,c}$ and $\boldsymbol{H}_{l,k,c}$ in each iteration are calculated by using $\check{\boldsymbol{x}}_{k,c}$ and $\check{\boldsymbol{l}}_{k,c}$ in equations (2.44) and (2.45), respectively. After reaching the predefined abortion criterion, the filtered states and modified observations will be:

$$\begin{aligned}\boldsymbol{\mu}_k &= \check{\boldsymbol{x}}_{k,C-1} \\ \boldsymbol{l}_k^+ &= \check{\boldsymbol{l}}_{k,C-1},\end{aligned} \tag{2.54}$$

where $C-1$ is the last iteration in which the abortion criterion is reached. Such a criterion is user-defined and depends on the application. However, it is usually set to be a maximum number of iterations or a threshold for the difference between the consecutive filtered states as well as the consecutive estimated observations. Another possibility is to set the criterion to a threshold that results in fulfilling the implicit observation model.
Based on Dang (2007), after deriving the filtered state vector, its corresponding the uncertainty ($\boldsymbol{\Sigma}_k$) can be calculated as follows:

$$\boldsymbol{D}_k = \boldsymbol{I} - \boldsymbol{K}_{k,C-1} \cdot \boldsymbol{H}_{x,k,C-1}, \tag{2.55}$$

$$\boldsymbol{\Sigma}_k = \boldsymbol{D}_k \cdot \overline{\boldsymbol{\Sigma}}_k \cdot \boldsymbol{D}_k^T + \boldsymbol{K}_{k,C-1} \cdot \boldsymbol{S}_{k,C-1} \cdot \boldsymbol{K}_{k,C-1}^T. \tag{2.56}$$

In the suggested framework by Dang (2007), even though the observations are modified, their uncertainty information ($\boldsymbol{Q}$) remains unchanged. However, based on the work by Vogel (2020), by applying the principle of uncertainty propagation to equation (2.53) in the last iteration (C), the modified VCM of the observations can be derived as follows:

$$\hat{\boldsymbol{Q}}_k = \boldsymbol{Q}_k + \boldsymbol{G}_{k,C-1} \cdot \boldsymbol{S}_{k,C-1} \cdot \boldsymbol{G}_{k,C-1}^T - \boldsymbol{U}_k \cdot \overline{\boldsymbol{\Sigma}}_k \cdot \boldsymbol{U}_k^T. \tag{2.57}$$

To derive $\boldsymbol{U}_k$ in equation (2.57), the following equation holds:

$$\boldsymbol{U}_k = \boldsymbol{G}_{k,C-1} \cdot \boldsymbol{H}_{x,k,C-1}. \tag{2.58}$$

Moreover, the framework proposed by Dang (2007) is designed cases where only implicit observation models are present. However, Vogel (2020) demonstrates that any explicit observation model can be transformed into an implicit one. Considering such a transformation, the suggested methodology of Dang (2007) becomes applicable to scenarios in which both implicit and explicit observation models exist. Such a framework is referred to as *versatile* IEKF by Vogel (2020).
To adapt an explicit observation model into an implicit form, as described by Vogel (2020), equation (2.10) should be reformulated as follows:

$$\boldsymbol{l}_k - \boldsymbol{\nu}_k - \boldsymbol{h}\left(\boldsymbol{x}_k\right) = \boldsymbol{0}. \tag{2.59}$$

Linearizing the above equation by the first degree Taylor series expansion and rearranging the elements leads to:

$$-\boldsymbol{H}_{x,k} \cdot \boldsymbol{x}_k + \underbrace{\boldsymbol{I}}_{\boldsymbol{H}_{l,k}} \cdot \boldsymbol{l}_k - \underbrace{\boldsymbol{\nu}_k - \boldsymbol{h}\left(\check{\boldsymbol{x}}_k\right) - \boldsymbol{H}_{x,k} \cdot \check{\boldsymbol{x}}_k}_{\boldsymbol{\delta}_k} = \boldsymbol{0}. \tag{2.60}$$

When comparing equation (2.60) with (2.43), it becomes evident that the general formulation of an implicit observation model can be applied to an explicit case.
The versatile IEKF is summarized in Algorithm 5. In this algorithm, lines 2 and 3 represent the prediction step, which is similar to that of e.g. Algorithm 4. Lines 4 to 13 represent the update step of this filter in which iterations are applied to reduce the linearization errors. In each iteration $(c+1)$, the derived states and modified observations from the last iteration (c) are used as the points in which the linearization is applied. Upon completion of the final iteration $(C-1)$, the

uncertainty of the filtered states ($\boldsymbol{\Sigma}_k$) and the modified observations $\left(\hat{\boldsymbol{Q}}_k\right)$ are calculated according to lines 16 and 19, respectively. It should be noted that the iterations in line 6 are started from 0 until $C-1$ to properly count for the initialization step.

Algorithm 5: Pseudo-code of the versatile iterated extended Kalman filter adapted from Vogel (2020).

1 $\left[\boldsymbol{\mu}_k,\ \boldsymbol{\Sigma}_k,\ \boldsymbol{l}_k^+,\ \hat{\boldsymbol{Q}}_k\right] =$
$\textbf{versatile\_iterated\_extended\_Kalman\_filter}\left(\boldsymbol{\mu}_{k-1},\ \boldsymbol{\Sigma}_{k-1},\ \boldsymbol{l}_k,\ \boldsymbol{u}_k\right)$

Prediction step

2 $\overline{\boldsymbol{\mu}}_k = \boldsymbol{g}\left(\boldsymbol{\mu}_{k-1}, \boldsymbol{u}_k\right)$ `equation (2.34)`

3 $\overline{\boldsymbol{\Sigma}}_k = \boldsymbol{F}_{x,k} \cdot \boldsymbol{\Sigma}_{k-1} \cdot \boldsymbol{F}_{x,k}^T + \boldsymbol{R}_k$ `equation (2.35)`

Update step

4 $\check{\boldsymbol{x}}_{k,0} = \overline{\boldsymbol{\mu}}_k$

5 $\check{\boldsymbol{l}}_{k,0} = \boldsymbol{l}_k$

6 **for** $c = 0 : C-1$ **do**

7 $\quad \boldsymbol{O}_{k,c} = \boldsymbol{H}_{x,k,c} \cdot \overline{\boldsymbol{\Sigma}}_k \cdot \boldsymbol{H}_{x,k,c}^T$ `equation (2.50)`

8 $\quad \boldsymbol{S}_{k,c} = \boldsymbol{H}_{l,k,c} \cdot \boldsymbol{Q}_k \cdot \boldsymbol{H}_{l,k,c}^T$ `equation (2.51)`

9 $\quad \boldsymbol{K}_{k,c} = \overline{\boldsymbol{\Sigma}}_k \cdot \boldsymbol{H}_{x,k,c}^T \cdot \left(\boldsymbol{O}_{k,c} + \boldsymbol{S}_{k,c}\right)^{-1}$ `equation (2.49)`

10 $\quad \check{\boldsymbol{x}}_{k,c+1} = \overline{\boldsymbol{\mu}}_k - \boldsymbol{K}_{k,c} \cdot \left(\boldsymbol{h}\left(\check{\boldsymbol{l}}_{k,c}, \check{\boldsymbol{x}}_{k,c}\right) + \boldsymbol{H}_{l,k,c} \cdot \left(\boldsymbol{l}_k - \check{\boldsymbol{l}}_{k,c}\right) + \boldsymbol{H}_{x,k,c} \cdot \left(\overline{\boldsymbol{\mu}}_k - \check{\boldsymbol{x}}_{k,c}\right)\right)$ `equation (2.48)`

11 $\quad \boldsymbol{G}_{k,c} = \boldsymbol{Q}_k \cdot \boldsymbol{H}_{l,k,c}^T \cdot \left(\boldsymbol{O}_{k,c} + \boldsymbol{S}_{k,c}\right)^{-1}$ `equation (2.52)`

12 $\quad \check{\boldsymbol{l}}_{k,c+1} = \boldsymbol{l}_k - \boldsymbol{G}_{k,c} \cdot \left(\boldsymbol{h}\left(\check{\boldsymbol{l}}_{k,c}, \check{\boldsymbol{x}}_{k,c}\right) + \boldsymbol{H}_{l,k,c} \cdot \left(\boldsymbol{l}_k - \check{\boldsymbol{l}}_{k,c}\right) + \boldsymbol{H}_{x,k,c} \cdot \left(\overline{\boldsymbol{\mu}}_k - \check{\boldsymbol{x}}_{k,c}\right)\right)$ `equation (2.53)`

13 **end**

14 $\boldsymbol{\mu}_k = \check{\boldsymbol{x}}_{k,C-1}$

15 $\boldsymbol{D}_k = \boldsymbol{I} - \boldsymbol{K}_{k,C-1} \cdot \boldsymbol{H}_{x,k,C-1}$

16 $\boldsymbol{\Sigma}_k = \boldsymbol{D}_k \cdot \overline{\boldsymbol{\Sigma}}_k \cdot \boldsymbol{D}_k^T + \boldsymbol{K}_{k,C-1} \cdot \boldsymbol{S}_{k,C-1} \cdot \boldsymbol{K}_{k,C-1}^T$ `equation (2.56)`

17 $\boldsymbol{l}_k^+ = \check{\boldsymbol{l}}_{k,C-1}$

18 $\boldsymbol{U}_k = \boldsymbol{G}_{k,C-1} \cdot \boldsymbol{H}_{x,k,C-1}$ `equation (2.58)`

19 $\hat{\boldsymbol{Q}}_k = \boldsymbol{Q}_k + \boldsymbol{G}_{k,C-1} \cdot \boldsymbol{S}_{k,C-1} \cdot \boldsymbol{G}_{k,C-1}^T - \boldsymbol{U}_k \cdot \overline{\boldsymbol{\Sigma}}_k \cdot \boldsymbol{U}_k^T$ `equation (2.57)`

20 return $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ and $\boldsymbol{l}_k^+$ and $\hat{\boldsymbol{Q}}_k$

2.3 Particle Filter

In Section 2.2, one of the strategies to solve the Bayes filter, namely the KF was explained. As described, the main idea of KF is to represent beliefs by means of multivariate Gaussian distribu-

tions. However, in practice, assuming uni-modal beliefs may not always capture the complexities of reality. Moreover, the true beliefs might be too complicated that deriving their analytical form is either not possible or requires high computational efforts. In such cases, the MC methods are considered as suitable approaches for solving the Bayes filter and hence concluding the related distributions.
In general, the MC methods are means to deal with experimental mathematics by using random samples. According to a categorization by Hammersley (2013), the mathematical problems are either theoretical or experimental. The theoretical problems are those that can be directly solved by using available mathematical resources. On the contrary, in the experimental mathematics, such resources are not available and hence the problems can be solved by doing experiments on a set of random samples. In case of the MC methods, the main idea is to derive the posterior distributions by means of such randomly generated samples. One of the well-known MC methods that is based on the concept of *sequential importance sampling* is the PF (Djuric et al., 2003). The PF framework is among the so-called *non-parametric filters* that can be used to approximate the posteriors by using a finite number of values that each correspond to a specific region in state space (Thrun et al., 2006). This filter has attracted a significant amount of attention in the recent decades due to its simple implementation and intuitive principle.
In the PF, the beliefs are represented by a set of samples, referred to as *particles*. These particles t are randomly generated from the pertinent distributions. Unlike other filter, in the PF, no assumption about the PDF of the beliefs is made and hence it can represent a broad space of distributions. In Figure 2.3, the sample representation of the distribution of an arbitrary one-dimensional random variable x is shown. The blue area with a blue outline represents the true distribution of x (referred to as $Tr\,(x)$) and the blue bars are random samples that are generated from this distribution. As it can be seen, many samples are located in the region of the peak and few samples elsewhere. Therefore, if the density function of the true distribution was not available, the regions of high and low probabilities could have been inferred from the random samples.

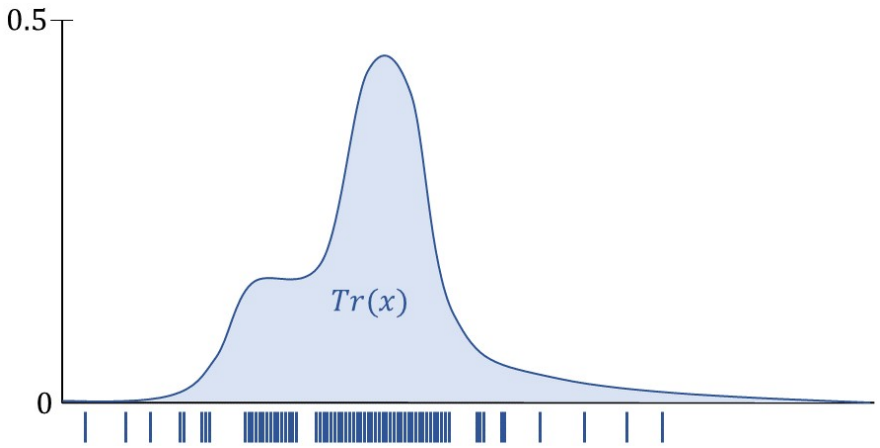

Figure 2.3: *Sample representation of the true distribution of a one-dimensional random variable x.*

In practice, the true belief is typically unknown, making it impossible to generate random samples from it. The primary concept of PF is to approximate the true belief by a set of weighted samples that are randomly drawn from a known distribution. Within the context of PF, this known distribution is termed the *proposal distribution*, while the true belief is called the *target distribution*. In the following parts, a proposal density function is denoted by Pr and Tr is used to represent a target density function.
In Figure 2.4, the given example in Figure 2.3 is used to describe the main principle of PF. In this example, it is assumed that sampling from $Tr\,(x)$ is not possible. Instead, we sample from $Pr\,(x)$ ensuring that the property $Tr\,(x) > 0 \;\rightarrow\; Pr\,(x) > 0$ holds. The red bars visually represent the sample representation of $Pr\,(x)$. The main idea of PF is to give certain weights to these samples

in a way that the resulting weighted samples represent the unknown $Tr\,(x)$, as shown by Figure 2.5. The magenta bars are indeed the generated samples from $Pr\,(x)$, which are weighted in a way that best represent the target distribution $Tr\,(x)$. Equation (2.61) encapsulates this process mathematically. In this equation, E_{Tr} and E_{Pr} denote the expectations for $Tr\,(x)$ and $Pr\,(x)$, respectively. Moreover, $\int_{Tr}$ shows the integral for a case in which $Tr\,(x)$ is known. A similar conviction applies to $\int_{Pr}$, but for the case in which $Pr\,(x)$ is known. As it can be interpreted from this equation, the expectation of $Tr\,(x)$ is equal to the expectation of $\omega\,(x) \cdot Pr\,(x)$.

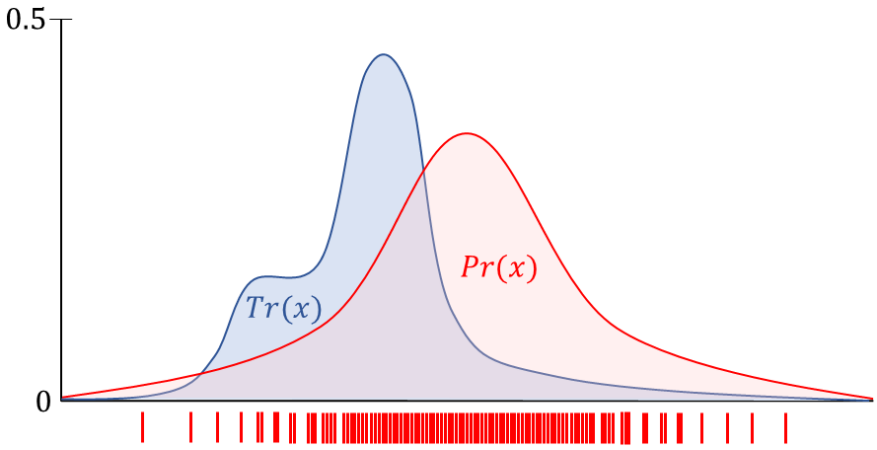

Figure 2.4: *Sample representation of the proposal distribution of a one-dimensional random variable x.*

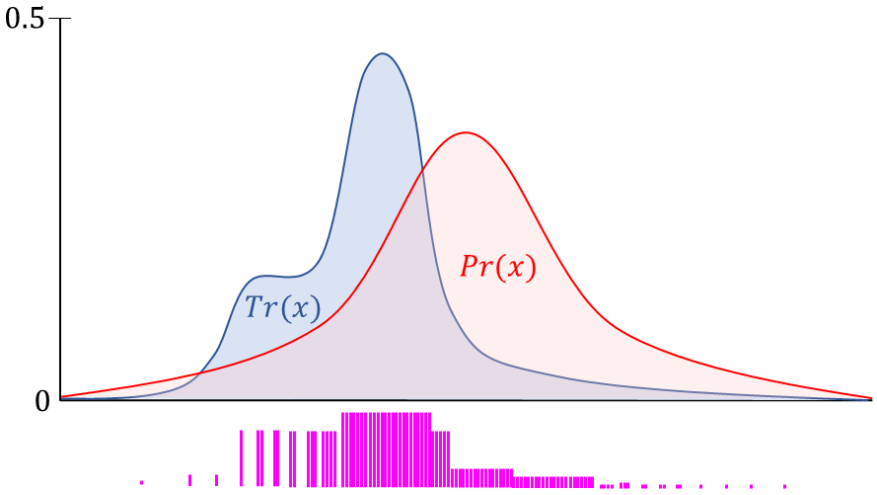

Figure 2.5: *Sample representation of the true distribution of a one-dimensional random variable x based on the drawn samples from the proposal distribution.*

$$\begin{aligned} E_{Tr} &= \int_{Tr} Tr\,(x)\;\; dx \\ &= \int_{Pr} \underbrace{\frac{Tr\,(x)}{Pr\,(x)}}_{:=\omega(x)} Pr\,(x)\;\; dx \\ &= E_{Pr}\Big[\omega\,(x)\Big]. \end{aligned} \tag{2.61}$$

In theory, calculating the expectation of a function requires generating an infinite number of sam-

ple from it. However, In practice, this is unfeasible. Thus, the space is considered as discrete, represented by a finite number of samples. By doing so, the expectation of a known function such as $Pr\left(x\right)$ can be computed as:

$$E_{Pr} = \int_{Pr} Pr\left(x\right)\ dx \quad \rightarrow \quad \frac{1}{S}\sum_{s=1}^{S} x^{[s]}, \tag{2.62}$$

where S denotes the total number of the samples generated from $Pr\left(x\right)$ and $x^{[s]}$ represents the s^{th} sample. With a similar justification, equation (2.61) can be reformulated as:

$$E_{Tr} = \int_{Pr} Tr\left(x\right)\ dx \quad \rightarrow \quad \left[\sum_{s=1}^{S} \omega^{[s]}\right]^{-1} \sum_{s=1}^{S} \omega^{[s]} x^{[s]}, \tag{2.63}$$

where $\omega^{[s]}$ represents the weight assigned to $x^{[s]}$ that according to equation (2.61) is determined as:

$$\omega^{[s]} = \frac{Tr\Big(x^{[s]}\Big)}{Pr\Big(x^{[s]}\Big)}. \tag{2.64}$$

In PF, the mathematical concepts given by equations (2.61) to (2.64) are applied to Bayes filter. The aim is to derive the target belief using samples generated randomly from a proposal distribution. Note that in the subsequent sections, instead of a one-dimensional random variable, the formulations are given for a multidimensional case.
To derive the primary formulae of Bayes filter, as given in equation (2.16), based on a sample representation, each particle should be considered as a set of state sequences:

$$\boldsymbol{x}_{0:k}^{[s]} = \boldsymbol{x}_{0}^{[s]}, \boldsymbol{x}_{1}^{[s]}, \ldots, \boldsymbol{x}_{k}^{[s]}. \tag{2.65}$$

In equation (2.65), $\boldsymbol{x}_{0:k}^{[s]}$ represents a matrix of size $u \times k$, which encompasses the s^{th} sample of the state vector $\boldsymbol{x}$ from the start to epoch k. By doing so, we can represent the target belief, taking into account the state sequences of all the samples, as:

$$\boldsymbol{bel}\left(\boldsymbol{x}_{0:k}\right) = \boldsymbol{p}\left(\boldsymbol{x}_{0:k} \mid \boldsymbol{l}_{1:k}, \boldsymbol{u}_{1:k}\right). \tag{2.66}$$

Applying such a sample representation to the Bayes filter, equation (2.16) can be rephrased accordingly:

$$\underbrace{\boldsymbol{p}\left(\boldsymbol{x}_{0:k} \mid \boldsymbol{l}_{1:k}, \boldsymbol{u}_{1:k}\right)}_{\boldsymbol{bel}(\boldsymbol{x}_{0:k})} = \boldsymbol{\eta}\ \boldsymbol{p}\left(\boldsymbol{l}_{k} \mid \boldsymbol{x}_{k}\right)\ \underbrace{\boldsymbol{p}\left(\boldsymbol{x}_{k} \mid \boldsymbol{x}_{k-1}, \boldsymbol{u}_{k}\right)\ \overbrace{\boldsymbol{p}\left(\boldsymbol{x}_{0:k-1} \mid \boldsymbol{l}_{1:k-1}, \boldsymbol{u}_{1:k-1}\right)}^{\boldsymbol{bel}\left(\boldsymbol{x}_{0:k-1}\right)}}_{\overline{\boldsymbol{bel}}(\boldsymbol{x}_{0:k})}. \tag{2.67}$$

Comparing equation (2.67) with equation (2.16), it is evident that the integral symbol in equation (2.67) has been omitted. The reason for this is in equation (2.67), the posterior is obtained by considering all the states from start to the current epoch. On the contrary, in equation (2.16), only the most recent state is taken into account for this purpose. Therefore, to account for the effect of the past states on the current posterior, considering an integral over the state space is essential.

Typically, in PF, the term $\overline{\boldsymbol{bel}}\,(\boldsymbol{x}_{0:k})$ in equation (2.67) is used as the proposal distribution. Therefore, according to equation (2.64), the particle weights can be derived as follows:

$$\begin{aligned}
\boldsymbol{\omega}_k^{[1:S]} &= \frac{\textit{target belief}}{\textit{proposal belief}} \\
&= \frac{\boldsymbol{bel}\,(\boldsymbol{x}_{0:k})}{\overline{\boldsymbol{bel}}\,(\boldsymbol{x}_{0:k})} \\
&= \frac{\boldsymbol{\eta}\ \boldsymbol{p}\,(\boldsymbol{l}_k \mid \boldsymbol{x}_k)\ \ \boldsymbol{p}\,(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}, \boldsymbol{u}_k)\ \ \boldsymbol{p}\,(\boldsymbol{x}_{0:k-1} \mid \boldsymbol{l}_{1:k-1}, \boldsymbol{u}_{1:k-1})}{\boldsymbol{p}\,(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}, \boldsymbol{u}_k)\ \ \boldsymbol{p}\,(\boldsymbol{x}_{0:k-1} \mid \boldsymbol{l}_{1:k-1}, \boldsymbol{u}_{1:k-1})} \\
&= \boldsymbol{\eta}\ \boldsymbol{p}\,(\boldsymbol{l}_k \mid \boldsymbol{x}_k)\,.
\end{aligned} \tag{2.68}$$

Equation (2.68) implies that the weight of each particle in epoch k has a direct relation to its likelihood in the same epoch. In other words, a particle that has a higher likelihood is more probable to be correct and hence it gets a higher weight, also referred to as *importance weight*. Note that due to having the same terms in numerator and denominator, the impact of state sequence for each particle is neglected. Therefore, in order to derive the weight of each particle, only its value in the current epoch is required.

After deriving the importance weights, there are two ways to continue the state estimation by PF. One way is to *resample* the particles based on their importance weights. Doing so, the particles that have a lower importance weight are replaced by those that have a higher weight. As mentioned by Thrun et al. (2006), the resampling step can be interpreted as the probabilistic implementation of the Darwinian idea of *survival of the fittest*. In other words, those particles that have a higher importance weight can better represent the posterior density and hence they are chosen more frequently.

Alternatively, after calculating the importance weights, state estimation can proceed without resampling the particles. This method ensures that regions with low posterior probability, which still contain particles, are also considered. Therefore, more particles are needed in order to have a reliable estimate of the posterior distribution. The required number of particles depends on the complexity of the target belief. In case of having a multi-modal belief, using few particles leads to a poor estimation; whereas taking more particles into consideration can correctly capture the modes of the distribution. Consequently, the accuracy of the resulting solutions depends on the number of the particles. Such an effect becomes less significant in case of having simple posterior distributions.

In practice, it usually proves useful to include the resampling step in PF. Doing so, the computational resources concentrate on regions of high posterior probability, leading to a more efficient algorithm (Thrun et al., 2006).

In Algorithm 6, the pseudo-code of the PF is given. The set of particles in epoch k is notated as follows:

$$\boldsymbol{\chi}_k \coloneqq \boldsymbol{x}_k^{[1]}, \boldsymbol{x}_k^{[2]}, \boldsymbol{x}_k^{[3]}, ..., \boldsymbol{x}_k^{[S]}. \tag{2.69}$$

As shown in line 1, the set of particles in the previous epoch $\left(\boldsymbol{\chi}_{\boldsymbol{k-1}}\right)$, the observations and control vectors of the current epoch ($\boldsymbol{l}_k$ and $\boldsymbol{u}_k$) serve as the input variables. The output of the algorithm is the set of particles in the current epoch $\left(\boldsymbol{\chi}_{\boldsymbol{k}}\right)$ that represent the target distribution as well as the estimated state vector ($\boldsymbol{\mu}_k$) and its VCM ($\boldsymbol{\Sigma}_k$).

The algorithm initiates by allocating an empty set for the particles, as indicated in line 2. To populate this set, the first step is to generate the particles from a proposal distribution (line 4). Afterwards, based on the estimated likelihoods, each particle is assigned an importance weight

(line 5). The initialized empty set in line 2 is eventually filled with the particles along with their importance weights according to line 6. If no resampling step is carried out, the resulting $\overline{\boldsymbol{\chi}}_k$ will be the output of the algorithm $\left(\boldsymbol{\chi}_k\right)$. However, as mentioned, resampling is typically employed to focus the particles on the high-probability regions of the target belief. The general workflow of this step is given by lines 8 to 11. The main idea is to replicate the particles with higher importance weights. Therefore, after the resampling step, the total number of particles still remains to be S. However, the distribution of these particles is closer to the target belief compared to the particle set before the resampling step. Multiple resampling approaches are discussed in the literature, with Li et al. (2015) offering an overview of resampling methods in PF. One of such methods is the *residual resampling*, which is also used in the current work. Therefore it is explained in the following section.

Algorithm 6: Pseudo-code of the particle filter adapted from Thrun et al. (2006).

1 $\left[\boldsymbol{\chi}_k,\ \boldsymbol{\mu}_k,\ \boldsymbol{\Sigma}_k\right] = \textbf{particle\_filter}\left(\boldsymbol{\chi}_{k-1},\ \boldsymbol{l}_k\ \boldsymbol{u}_k\right)$
2 $\overline{\boldsymbol{\chi}}_k = \boldsymbol{\chi}_k = \emptyset$
3 **for** $s = 1 : S$ **do**
  Prediction step
4 sample $\boldsymbol{x}_k^{[s]} \sim \boldsymbol{p}\left(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}^{[s]}, \boldsymbol{u}_k\right)$ `according to` $\overline{\boldsymbol{bel}}\left(\boldsymbol{x}_{0:k}\right)$ `in equation (2.67)`
  Update step
5 $\omega_k^{[s]} = \boldsymbol{p}\left(\boldsymbol{l}_k \mid \boldsymbol{x}_k^{[s]}\right)$ `equation (2.68)`
6 add $\boldsymbol{x}_k^{[s]}$ to $\overline{\boldsymbol{\chi}}_k$
7 **end**
  (Re-) Sampling step
8 **for** $s = 1 : S$ **do**
9 draw s with probability $\propto \omega_k^{[s]}$
10 add $\boldsymbol{x}_k^{[s]}$ to $\boldsymbol{\chi}_k$
11 **end**
  State estimation step
12 $\boldsymbol{\mu}_k = \frac{1}{S}\sum \boldsymbol{\chi}_k$
13 $\boldsymbol{\Sigma}_k = \frac{1}{S-1}\sum \left(\boldsymbol{\chi}_k - \boldsymbol{\mu}_k\right) \cdot \left(\boldsymbol{\chi}_k - \boldsymbol{\mu}_k\right)^T$
14 return $\boldsymbol{\chi}_k$, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$

Residual Resampling

As explained by Beadle and Djuric (1997) and Liu and Chen (1998), the residual (remainder) resampling consists of two steps. In the first step, those particles with a weight greater than $\frac{1}{S}$ are replicated deterministically. In the second step, a random replication of these particles based on the residual of their weights should be carried out. According to this approach, the particle $\boldsymbol{x}_k^{[s]}$ is replicated with a minimum of $N_k^{[s]}$ and with a maximum of $N_k^{[s]} + R_k^{[s]}$ times. $N_k^{[s]}$ and $R_k^{[s]}$ are the number of replications derived from the first and second steps, respectively. To calculate $N_k^{[s]}$, the following applies:

$$N_k^{[s]} = \left\lfloor S \times \omega_k^{[s]} \right\rfloor, \tag{2.70}$$

where $\lfloor\ \rfloor$ represents the maximum integer value of the $S \times \omega_k^{[s]}$ multiplication. The calculated $N_k^{[s]}$ from equation (2.70) can be used to calculate the residual of the weight $\left(\hat{\omega}_k^{[s]}\right)$ as follows:

$$\hat{\omega}_k^{[s]} = \omega_k^{[s]} - \frac{N_k^{[s]}}{S}. \tag{2.71}$$

In the second step, the particles are replicated based on the calculated residuals from equation (2.71). For that, random resampling methods such as *multinomial* and *stratified* approaches can be used in which selection of the particles is proportional to their remaining weight (Li et al., 2015). The total number of replications in epoch k resulting from the second stage should be:

$$\begin{aligned} R_k &= S - N_k \\ N_k &= \sum_{s=1}^{S} N_k^{[s]}. \end{aligned} \tag{2.72}$$

Algorithm 7 gives the pseudo-code of the residual resampling. The algorithm starts by deriving the number of the deterministic replications along with the remainder of the weights for each particle (lines 2 to 5). Following that, based on the derived number of replications (N_k), the deterministic replication step is carried out (lines 6 to 13). Thereafter, the derived weight residuals should be first normalized (lines 14 to 16) and then cumulatively summed up (lines 17 to 20). Finally, based on the accumulated sum of the weight residuals and by using a random resampling technique (here the stratified approach according to Li et al. (2015)), the second replication step is carried out (lines 21 to 31). In line 24, U represents the uniform distribution.

Algorithm 7: Pseudo-code of the residual resampling adapted from Li et al. (2015).

$\left[\underbrace{\left\{\boldsymbol{x}_k^{[s]}\right\}_{s=1}^{S}}_{\boldsymbol{\chi}_k}\right] = \textbf{residual\_resampling}\left(\underbrace{\left\{\boldsymbol{x}_k^{[s]}\right\}_{s=1}^{S}}_{\overline{\boldsymbol{\chi}}_k}, \left\{\omega_k^{[s]}\right\}_{s=1}^{S}\right)$

for $s = 1 : S$ **do**
  $N_k^{[s]} = \left\lfloor S \times \omega_k^{[s]} \right\rfloor$ `equation (2.70)`
  $\hat{\omega}_k^{[s]} = \omega_k^{[s]} - \frac{N_k^{[s]}}{S}$ `equation (2.71)`
end
$n = 0$
for $s = 1 : S$ **do**
  for $i = 1 : N_k^{[s]}$ **do**
    $n = n + 1$
    $\boldsymbol{x}_k^{[n]} = \boldsymbol{x}_k^{[s]}$
  end
end
$N_k = n$
for $s = 1 : S$ **do**
  $\hat{\omega}_k^{[s]} = \hat{\omega}_k^{[s]} \times \frac{S}{S - N_k}$
end
$Q_k^{[1]} = \hat{\omega}_k^{[1]}$
for $s = 2 : S$ **do**
  $Q_k^{[s]} = Q_k^{[s-1]} + \hat{\omega}_k^{[s]}$
end
$n = 0$
$s = 1$
while $n \leq S - N_k$ **do**
  $u_0 \sim U\left(0, \frac{1}{S - N_k}\right]$
  $u = u_0 + \frac{n}{S - N_k}$
  while $Q_k^{[s]} < u$ **do**
    $s = s + 1$
  end
  $n = n + 1$
  $\boldsymbol{x}_k^{[N_k+n]} = \boldsymbol{x}_k^{[s]}$
end
return $\left\{\boldsymbol{x}_k^{[s]}\right\}_{s=1}^{S}$

3 Advanced Particle Filtering

As explained in Section 2.3, one of the frameworks to implement the Bayes filter is the PF. Due to the capability of this framework in handling highly nonlinear and non-Gaussian systems, it has become popular in various communities such as signal processing (Gordon et al., 1993) and robotics (Thrun et al., 2006). Nonetheless, the general formulation of the PF is based on explicit observation models. Consequently, its usage is constrained to those cases in which the observations can be exclusively expressed by using the states. For example, the GNSS data can be used to obtain the 3D position of an autonomous vehicle by using an explicit observation model. However, as mentioned in the previous chapters, using such data in urban environments is not always possible or beneficial. In such a case, using e.g. LiDARs to benefit from the existing geometrical information of the environment can compensate for the issue of the GNSS data. However, depending on the uncertainty of such data and the type of the geometrical information, an implicit relation between the states and the observations may be possible in which the observations cannot be exclusively obtained by using the states. This will lead to the violation of the basic formulation of PF in which the likelihood is estimated by comparing the sensor data with the estimated observations using the explicit observation models. To our knowledge and until the time of writing this dissertation, there is no PF framework that can overcome such an issue. Therefore, the main aim of this work is to develop a PF methodology that can handle implicit observation models as well.
In Section 3.1, it is explained how the general framework of PF can be adapted to cases in which an implicit relation between the sensor data and the states exists. The newly developed framework in this case is referred to as the PFI, which involves having a large number of observations. The reason for such a requirement is the main motive of the current thesis in which the georeferencing problem in urban areas is to be solved by considering additional information of the environment. In order to benefit from such information, the environment should be captured intensively. As a result, a large number of sensor data are obtained that should be used in the desired state estimator to obtain the georeferencing solutions. Apart from this, in cases with a high-dimensional state vector, a large number of observations are usually required to estimate the states reliably. Therefore, a framework that can handle a large number of observations is also essential for applications other than georeferencing.
Moreover, Section 3.2 is dedicated to the second developed framework, which is referred to as the R-PFI. The main idea of this new methodology is to account for potential challenges of the PFI approach that can arise due to existing outliers, which can be the case when the number of observations is large. In such a situation, the estimated likelihood of each particle can be strongly influenced leading to diverged solutions over time. For example, in case of using the PFI framework for georeferencing an autonomous vehicle equipped with a LiDAR in an urban area, the existing outliers can lead to a wrong estimation of the 6-DoF. Therefore, the current work asserts the importance of detecting the outliers prior to the likelihood estimation step. In the R-PFI framework, a strategy is introduced that can well overcome the existence of such misleading data. By using the proposed framework, the outliers can be detected and removed in the early stages of the filtering. Consequently, the negative impact of such data on the estimated likelihood can be eliminated, which in turn leads to reliable estimates.
The basis of both the PFI and R-PFI frameworks is a large amount of sensor data. Consequently, a large number of particles is required to reliably estimate the states. It should be noted that in the PF-based frameworks in general, a high number of particles is required to reduce or overcome uncertainties including an unreliable initialization. However, when the number of observations is high, even more particles are required to avoid instabilities or divergence of the filter due to the

combination of the likelihoods from different data points. This leads to an increased computational complexity, which is also the case for the PFI and R-PFI frameworks. Having a high computation time is not desirable in real-time applications such as autonomous vehicles. Therefore, Section 3.3 explains how the computational complexity of the R-PFI can be reduced. The result is not only a robust but also an efficient state estimator, which is called R-EKPFI. The main idea of this newly developed framework is to combine it with the versatile IEKF in order to use significantly fewer number of particles.
The following derivations and methodologies are based on the assumption that the measurements are independent and identically distributed (i.i.d.). This implies that each measurement does not influence the others, and they all come from the same family of distributions.

3.1 Particle Filter with Implicit Observation Models

In the general formulation of PF – as explained in Section 2.3 – the main idea is to estimate the states based on the importance weight of the particles. As shown by equation (2.68), to estimate the importance weights, the likelihoods should be derived. In case of explicit observation models and for each particle $\left(\boldsymbol{x}_k^{[s]}\right)$, the likelihoods are derived by comparing the estimated observations $\left(\hat{\boldsymbol{l}}_k^{[s]}\right)$ with the sensor data $(\boldsymbol{l}_k + \boldsymbol{\nu}_k)$. For that, equation (2.59) can be rewritten for one particle as follows:

$$\boldsymbol{l}_k - \boldsymbol{\nu}_k - \boldsymbol{h}\left(\boldsymbol{x}_k^{[s]}\right) = \boldsymbol{0}, \tag{3.1}$$

where $\boldsymbol{h}\left(\boldsymbol{x}_k^{[s]}\right)$ represents the estimated observations by using particle $\boldsymbol{x}_k^{[s]}$ in the explicit observation model given by equation (2.10) as follows:

$$\hat{\boldsymbol{l}}_k^{[s]} = \boldsymbol{h}\left(\boldsymbol{x}_k^{[s]}\right). \tag{3.2}$$

Using equation (3.2) in (3.1) yields the following measure, which is used to estimate the likelihoods in the general framework of PF:

$$\hat{\boldsymbol{v}}_k^{[s]} = \boldsymbol{l}_k - \boldsymbol{\nu}_k - \hat{\boldsymbol{l}}_k^{[s]}. \tag{3.3}$$

In case of having an implicit observation model with a general formulation given by equation (2.42), estimating the observations $\left(\hat{\boldsymbol{l}}_k^{[s]}\right)$ by using the states is not possible. Consequently, having an equation similar to (3.2) is not feasible, which in turn leaves the question of how the importance weight for each particle $\left(\omega_k^{[s]}\right)$ should be derived.
In the developed framework of PFI by the current work, a similar principle to equation (3.3) is applied to derive a measure for the likelihood estimation (Moftizadeh et al., 2021). Rewriting the implicit observation model given by equation (2.42) for particle $\left(\boldsymbol{x}_k^{[s]}\right)$ yields:

$$\boldsymbol{h}\left(\boldsymbol{l}_k + \boldsymbol{\nu}_k, \boldsymbol{x}_k^{[s]}\right) = \boldsymbol{0}. \tag{3.4}$$

It is claimed that the left side of equation (3.4) can be treated similar to the left side of equation

(3.1) in order to determine how well the observation model is satisfied. Consequently, the following measure can be considered as an equivalent equation to (3.3):

$$\hat{\boldsymbol{r}}_k^{[s]} = \boldsymbol{h}\left(\boldsymbol{l}_k + \boldsymbol{\nu}_k, \boldsymbol{x}_k^{[s]}\right), \tag{3.5}$$

where $\hat{\boldsymbol{r}}_k^{[s]}$ is referred to as the *implicit residual*. The reason for choosing this term is on the one hand the underlying implicit observation model. On the other hand, the uncertainties stemming from e.g. the observations lead to the resulting $\hat{\boldsymbol{r}}_k^{[s]}$ to have non-zero values.
It should be noted that in case of having an explicit observation model, $\hat{\boldsymbol{v}}_k^{[s]}$ follows the same distribution as $\boldsymbol{\nu}_k$. The reason is due to the possibility of estimating the observations by using equation (3.2). Those particles that lead to estimated observations $\left(\hat{\boldsymbol{l}}_k\right)$ close to the real ones $(\boldsymbol{l}_k)$ result in residuals $\left(\hat{\boldsymbol{v}}_k^{[s]}\right)$ that are within the uncertainty domain $(\boldsymbol{\nu}_k)$ of the sensor data. However, such a claim does not necessarily hold for the distribution of $\hat{\boldsymbol{r}}_k^{[s]}$. The reason is the uncertainty of equation (3.4), which stems from not only the observations, but also the information that implicitly relates the sensor data to the states. The aspect of uncertain additional information and its consideration in the case of implicit observation models is discussed further in Section 3.2.
Using the suggested concept of implicit residual and based on the mathematical formulations of Bayes filter, the likelihood is calculated as follows:

$$\begin{aligned} \omega_k^{[s]} &= p\left(\boldsymbol{l}_k \mid \boldsymbol{x}_k^{[s]}\right) \\ &= \prod_{i=1}^{n} p\left(\hat{r}_{k,i}^{[s]}\right) \\ &\rightarrow \sum_{i=1}^{n} ln\left(p\left(\hat{r}_{k,i}^{[s]}\right)\right). \end{aligned} \tag{3.6}$$

In equation (3.6), the log likelihood is employed to ensure numerical stability. Multiplying many probabilities, especially small ones, can cause numerical underflow. By taking the logarithm, we convert the product of probabilities into a sum, mitigating potential numerical issues. Furthermore, the logarithm can simplify the optimization process by making the objective function more concave.
In Algorithm 8, the pseudo-code of the PFI framework is given. As it can be seen, the general workflow of this methodology is similar to that given by Algorithm 6. The main difference is in line 5 in which the implicit residual is calculated.
In case of having outliers, the explained PFI framework is subject to divergence over time. In this dissertation, an outlier is defined as an observation that results in an implicit residual value that is inconsistent with the implicit residual values derived from the other observations. Such misleading observations lead to the early removal of the potentially good particles by using equation (3.6). Having such outliers leads to the implicit residual vector $\left(\hat{\boldsymbol{r}}_k^{[s]}\right)$ to have some values that lie in a higher magnitude compared to the other implicit residuals. Considering such values in equation (3.6) leads to having small importance weights for all of the particles. As a result, distinguishing between the correct and incorrect samples becomes critical, which in turn can lead to the removal of the potentially good particles over time. Since the PF framework has a random nature, such a divergence is also random leading to an unstable filter, which is not favored in real-word applications. Apart from the potential diverging effect of the outliers, having them leads to degraded posterior distributions. The reason, as explained earlier, is the removal of potentially good particles, resulting in a set of resampled particles that do not represent the correct states in each epoch.

Given the mentioned issue, there is a pressing need to develop an approach that effectively manages estimating the importance weight of the particles by removing the outliers. The current work suggests to use the IQR methodology to detect the outliers based on their resulting implicit residuals. The resulting methodology, which is referred to as the R-PFI and ensures a robust state estimation, is explained in detail in the following section.

Algorithm 8: Pseudo-code of the PFI.

$\left[\boldsymbol{\chi}_k,\ \boldsymbol{\mu}_k,\ \boldsymbol{\Sigma}_k\right] = \mathbf{PFI}\left(\boldsymbol{\chi}_{k-1},\ \boldsymbol{l}_k,\ \boldsymbol{u}_k\right)$

$\overline{\boldsymbol{\chi}}_k = \boldsymbol{\chi}_k = \emptyset$

for $s = 1 : S$ **do**

Prediction step

sample $\boldsymbol{x}_k^{[s]} \sim \boldsymbol{p}\left(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}^{[s]}, \boldsymbol{u}_k\right)$ `according to` $\overline{\boldsymbol{bel}}\left(\boldsymbol{x}_{0:k}\right)$ `in equation (2.67)`

Update step

$\hat{\boldsymbol{r}}_k^{[s]} = \boldsymbol{h}\left(\boldsymbol{l}_k + \boldsymbol{\nu}_k, \boldsymbol{x}_k^{[s]}\right)$ `equation (3.5)`

$\omega_k^{[s]} = \sum_{i=1}^{n} ln\left(p\left(\hat{r}_{k,i}^{[s]}\right)\right)$ `equation (3.6)`

add $\boldsymbol{x}_k^{[s]}$ to $\overline{\boldsymbol{\chi}}_k$

end

(Re-) Sampling step

for $s = 1 : S$ **do**

draw s with probability $\propto \omega_k^{[s]}$

add $\boldsymbol{x}_k^{[s]}$ to $\boldsymbol{\chi}_k$

end

State estimation step

$\boldsymbol{\mu}_k = \frac{1}{S} \sum \boldsymbol{\chi}_k$

$\boldsymbol{\Sigma}_k = \frac{1}{S-1} \sum \left(\boldsymbol{\chi}_k - \boldsymbol{\mu}_k\right) \cdot \left(\boldsymbol{\chi}_k - \boldsymbol{\mu}_k\right)^T$

return $\boldsymbol{\chi}_k$, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$

3.2 Robust Particle Filter with Implicit Observation Models

As explained in Section 3.1, using equation (3.6) for estimating the importance weights can lead to the divergence of the filter due to the existing outliers. Such misleading observations result in a high variation between the components of the implicit residual vector. As a result, using equation (3.6) leads to small importance weight for all of the particles. Consequently, the filter can fail to correctly distinguish between the potentially good and bad particles. Therefore, it is possible that a number of the potentially good samples are removed in the early epochs, which in turn leads to the divergence of the filter. In order to overcome this issue, the current work suggests detecting

such misleading observations by applying the IQR method on the absolute of the implicit residual vector $\left|\hat{\boldsymbol{r}}_k^{[s]}\right|$. Doing so, the outliers can be detected and removed before estimating the importance weight of the particles.
The basis of the IQR method is that the data have a symmetric distribution. Therefore, using it to detect outliers implies that the implicit residuals have a symmetric distribution. However, as explained in Section 3.1, in addition to the uncertainty of the observations, the uncertainty of the information that implicitly relates them to the states affects the underlying distribution of the $\hat{\boldsymbol{r}}_k^{[s]}$. Therefore, in practical applications, a symmetric distribution for the implicit residuals cannot be guaranteed. However, in the current work, the propagation of the uncertainty of the additional information in the implicit observation models is neglected. The reason is that in practical applications, the uncertainty values of such information are either unavailable or inaccurate. Therefore, to avoid complications, it is proposed that the implicit residuals follow the same family of distributions as the observations. The later parts of this section explain how the uncertainty of additional information is compensated in the R-PFI framework.
In real-world applications, observations are usually expected to have Gaussian distributions. Therefore, the assumption of a Gaussian distribution for the implicit residuals is claimed to be safe in the current work. As a result, the symmetric distribution requirement of the IQR method is well satisfied. In the case of non-symmetric distributions for the implicit residuals, alternative outlier detection methods to the IQR method should be used.
In addition, to make the filter more robust to the large number of observations, the current work proposes to reduce the dimensionality of $\hat{\boldsymbol{r}}_k^{[s]}$ after detecting the outliers and before estimating the importance weight of the particles. It is claimed that using equation (3.6) after detecting the outliers overcomes the problem of having small importance weights due to such misleading observations. However, due to the large number of data, using the joint PDF can still lead to small importance weights. This effect becomes more noticeable as the number of observations increases. As a result, a clear distinction between the potentially good and bad particles becomes difficult, and the filter may exhibit instability. To reduce the dimensionality of $\hat{\boldsymbol{r}}_k^{[s]}$, the current work proposes to use the mean of its absolute values. In this way, the quality of each particle is evaluated based on a representative of the implicit residual vector instead of using all of its components by using the joint PDF. Considering the mean instead of the median is intended to reflect the effect of each observation on the implicit residual vector. Therefore, it is recommended to avoid using the median instead of the mean. Doing so can falsify the filter and cause it to diverge. Moreover, it should be emphasized that the detection of outliers before this step is essential. Otherwise, their negative influence will directly affect the calculated mean, which in turn will lead to incorrect selection of particles.
In the following sections, the IQR method is explained first. Then, the details of the developed R-PFI method are elaborated.

3.2.1 Interquartile Range Method

In the IQR method, the data is not necessarily assumed to be normally distributed, and an *outlier* is often defined as a data point that lies outside 1.5 times the IQR above the third quartile or below the first quartile (Barbato et al., 2011).
Assuming we have n data points, to detect the outliers using this method, first the *IQR* should be calculated. The IQR is a range in which 50% of the data lies. To determine this range, the data ($\boldsymbol{y}$) should first be sorted in ascending order. Afterwards, the *lower quartile* (Q_1) and the *upper quartile* (Q_3) can be derived:

$$\begin{aligned} Q_1 &= y_{(n+1)\times\frac{1}{4}} \\ Q_3 &= y_{(n+1)\times\frac{3}{4}}, \end{aligned} \tag{3.7}$$

where the subscripts $(n+1)\times\frac{1}{4}$ and $(n+1)\times\frac{3}{4}$ correspond to the 25% and 75% percentile of the data, respectively. Having the lower and upper quartiles, the IQR is calculated as follows:

$$IQR = Q_3 - Q_1. \tag{3.8}$$

After deriving the interquartile range, any data point with a value lower than $Q_1 - 1.5 \times IQR$ or higher than $Q_3 + 1.5 \times IQR$ is identified as outlier. The definition for outliers is based on the range proposed by Tukey et al. (1977), often referred to as the *Tukey's fences*:

$$[\ Q_1 - k \times IQR,\ Q_3 + k \times IQR\ ], \tag{3.9}$$

for some non-negative constant k. With the assumption of Gaussian distribution for the data points and the definition of outliers as points that lie outside of three standard deviation, the constant k is determined to be 1.5. This is because in a Gaussian distribution, the lower and upper quartile correspond to -0.675σ and $+0.675\sigma$. Using these values in equation (3.9) in combination with (3σ) as the definition for outliers, the constant k is derived to be approximately 1.7. For practicality and based on common standards in the literature, k is typically chosen to be 1.5 (Tukey et al., 1977). The explained concept of the IQR for detecting the outliers can also be visualized by *box plots*. Box plots are tools that display the statistical information of data sets (Frigge et al. (1989) and Potter et al. (2006)). They are also referred to as the *5-number summary approach*. The reason is that in box plots, a set of data points can be visualized by using the minimum and maximum range values, the lower and upper quartiles, and the median. Figure 3.1 depicts a box plot based on the definitions of the IQR method for detecting possible outliers. As it can be seen, the box that is extended from the lower to upper quartile depicts the IQR. Also, the *whiskers* are the vertical lines that are extended from the two ends of the box to the Tukey's fences. As it is shown, the outliers are then those data points that are beyond the Tukey's fences.

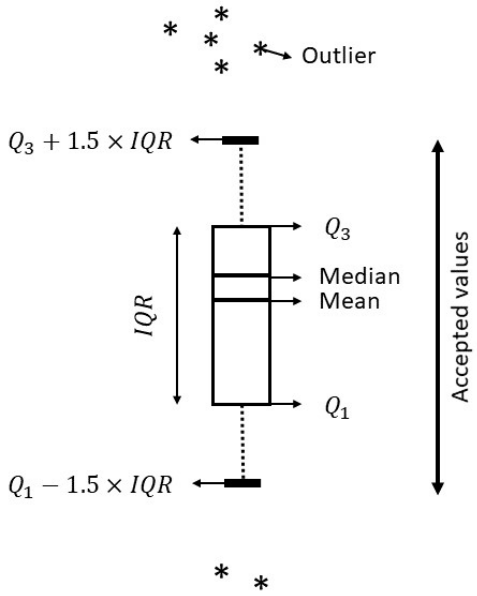

Figure 3.1: *Visualization of the IQR method by means of a box plot.*

As mentioned earlier, the main basis of the R-PFI is to apply the IQR method to the absolute of the implicit residual vector $\left(\left|\hat{\boldsymbol{r}}_k^{[s]}\right|\right)$. After removing the outliers, the importance weight is calculated based on the mean of the remaining absolute values of the implicit residual vector as follows:

$$\tilde{r}_k^{[s]} = \frac{1}{q}\sum_{j=1}^{q}\left|\hat{r}_{k,j}^{[s]}\right| \;, \quad \forall \left|\hat{r}_{k,j}^{[s]}\right| \in \left|\hat{\boldsymbol{r}}_k^{[s]}\right| : \quad \left|\hat{r}_{k,j}^{[s]}\right| \leq Q_1 - 1.5\times IQR \;\mid\; \left|\hat{r}_{k,j}^{[s]}\right| \geq Q_3 + 1.5\times IQR, \tag{3.10}$$

where q indicates the size of the implicit residual vector after removing the outliers ($q \leq n$). In other words, a smaller mean value indicates that the particle is closer to the true state compared to a larger mean value that is resulted by using another particle. Therefore, instead of calculating the joint PDF to derive the importance weight (see equation (3.6)), only the PDF of the mean of the absolute values should be considered:

$$\begin{aligned} \omega_k^{[s]} &= \boldsymbol{p}\left(\boldsymbol{l}_k \mid \boldsymbol{x}_k^{[s]}\right) \\ &= p\left(\tilde{r}_k^{[s]}\right). \end{aligned} \tag{3.11}$$

To obtain the $p\left(\tilde{r}_k^{[s]}\right)$ in equation (3.11), the appropriate distribution for the implicit residuals should be used. As mentioned earlier, the current work assumes a Gaussian distribution for the implicit residuals. In such a case, $p\left(\tilde{r}_k^{[s]}\right)$ can be calculated by considering $\mathcal{N}\left(\tilde{r}_k^{[s]};\, 0,\, \sigma_{\tilde{r}}\right)$. The latter mathematical expression indicates that the obtained $\tilde{r}_k^{[s]}$ belongs to a Gaussian distribution with an expected value of 0 and a standard deviation (STD) of $\sigma_{\tilde{r}}$. The current work claims that $\sigma_{\tilde{r}}$ is application dependent, which should be specified accordingly. In an optimal case where the only source of uncertainty is from the observations, their uncertainty can be propagated to the estimated mean ($\tilde{r}$) to obtain $\sigma_{\tilde{r}}$. However, in real-world applications, there are various sources of uncertainty that affect the estimated mean, and thus determining $\sigma_{\tilde{r}}$ by simply propagating the error of the observations does not lead to reliable results. According to the investigations of this thesis, doing

so can lead to instability of the filter. Therefore, it is essential to also consider the uncertainty of the information that implicitly relates the observations to the states for determining $\sigma_{\tilde{r}}$. If such an uncertainty is readily available and can be formulated mathematically, it is recommended to propagate it along with the uncertainty of the observations to obtain a realistic estimate for $\sigma_{\tilde{r}}$. However, as mentioned earlier, such uncertainty information may not be available or reliable. In such situations, a penalty for the uncertainty of the additional information should be reflected in $\sigma_{\tilde{r}}$. Therefore, the current work claims that $\sigma_{\tilde{r}}$ is a design term that should be specified depending on the application.
In Algorithm 9, the pseudo-code of the proposed R-PFI methodology is given. As it can be seen, the general structure of this algorithm is similar to that given by Algorithm 8. The main novelty of the proposed methodology is in lines 5 to 10 in which calculating the importance weight of each particle based on the implicit observation model is shown.
In the R-PFI method, similar to the PFI method, a large number of particles is required to obtain a reliable estimate of the states. This is due to the unknown posterior distribution from which the particles should be sampled. Consequently, a proposal distribution is chosen for this purpose, which may differ significantly from the posterior distribution. As a result, a large number of samples is required to compensate for the poor choice of the proposal distribution and to obtain reliable estimates. Due to the large number of particles, the computational time of these filters is significantly high, which is a disadvantage for real-world applications such as autonomous driving, where computational time plays a key role.
In the next section, the concept of the versatile IEKF explained in Chapter 2 is used in the R-PFI to overcome the problem of inappropriate proposal distributions. As a result, a significantly smaller number of particles is required, which in turn leads to the efficiency of the resulting framework, called the R-EKPFI.

3.3 Robust Extended Kalman Particle Filter with Implicit Observation Models

The proposed R-PFI framework – explained in Section 3.2 – has two features. First, it can overcome the negative impact of the outliers on the importance weight of the particles by using the IQR method. Second, it avoids having small importance weights due to a large number of observations by reducing the dimensionality of the implicit residual vector $\left(\hat{\boldsymbol{r}}_k^{[s]}\right)$. As a result, calculating the joint PDF to obtain the importance weights is avoided. These two features provide a robust state estimator in cases with a large number of observations. However, as mentioned in Section 3.2, due to the unknown posterior distribution of the states, a large number of particles is required in this framework to guarantee a reliable state estimation. As a result, the R-PFI framework has a high computational cost, which is not desirable in those real-world applications that call for real-time performance.
To overcome the high computational time of R-PFI, the versatile IEKF methodology – explained in Section 2.2.3 of Chapter 2 is included to modify the proposal distribution. In this way, the proposal distribution is designed to ensure that the sampled particles more effectively capture the posterior distribution compared to the R-PFI algorithm. This is achieved by adjusting each particle using the Kalman gain, which guides them towards regions of higher posterior probability. This results in a substantial reduction in the number of particles required, which in turn leads to a considerable decrease in computation time. The reason for using the versatile IEKF is the existence of implicit observation models, which are not addressed in the general formulation of the IEKF.
The framework of R-EKPFI is inspired by the concept of *constraints* in the PF framework, which are additional prior information such as physical laws or geometrical restrictions that improve state estimation. In this work, the observations and their corresponding observation model act as constraints on each particle after the prediction step. Usually, due to the poor choice of the proposal

Algorithm 9: Pseudo-code of the R-PFI.

$\left[\boldsymbol{\chi}_k,\ \boldsymbol{\mu}_k,\ \boldsymbol{\Sigma}_k\right] = \textbf{R-PFI}\left(\boldsymbol{\chi}_{k-1},\ \boldsymbol{l}_k,\ \boldsymbol{u}_k\right)$

$\overline{\boldsymbol{\chi}}_k = \boldsymbol{\chi}_k = \emptyset$

for $s = 1 : S$ **do**

Prediction step

sample $\boldsymbol{x}_k^{[s]} \sim \boldsymbol{p}\left(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}^{[s]}, \boldsymbol{u}_k\right)$ `according to` $\overline{\boldsymbol{bel}}\left(\boldsymbol{x}_{0:k}\right)$ `in equation (2.67)`

Update step

$\hat{\boldsymbol{r}}_k^{[s]} = \boldsymbol{h}\left(\boldsymbol{l}_k + \boldsymbol{\nu}_k, \boldsymbol{x}_k^{[s]}\right)$ `equation (3.5)`

$Q_1 = y_{(n+1)\times\frac{1}{4}}$ `equation (3.7)`

$Q_3 = y_{(n+1)\times\frac{3}{4}}$ `equation (3.7)`

$IQR = Q_3 - Q_1$ `equation (3.8)`

Outlier detection

$\forall \left[\left|\hat{r}_{k,j}^{[s]}\right|\right]_{j=1}^{j=q} \in \left|\hat{\boldsymbol{r}}_k^{[s]}\right| : \quad \left|\hat{r}_{k,j}^{[s]}\right| \leq Q_1 - 1.5 \times IQR \ \mid \ \left|\hat{r}_{k,j}^{[s]}\right| \geq Q_3 + 1.5 \times IQR\ , \quad q \leq n$

$\tilde{r}_k^{[s]} = \frac{1}{q}\sum_{j=1}^{j=q}\left|\hat{r}_{k,j}^{[s]}\right|$ `equation (3.10)`

$\omega_k^{[s]} = p\left(\tilde{r}_k^{[s]}\right)$ `equation (3.11)`

add $\boldsymbol{x}_k^{[s]}$ to $\overline{\boldsymbol{\chi}}_k$

end

(Re-) Sampling step

for $s = 1 : S$ **do**

draw s with probability $\propto \omega_k^{[s]}$

add $\boldsymbol{x}_k^{[s]}$ to $\boldsymbol{\chi}_k$

end

State estimation step

$\boldsymbol{\mu}_k = \frac{1}{S}\sum \boldsymbol{\chi}_k$

$\boldsymbol{\Sigma}_k = \frac{1}{S-1}\sum\left(\boldsymbol{\chi}_k - \boldsymbol{\mu}_k\right)\cdot\left(\boldsymbol{\chi}_k - \boldsymbol{\mu}_k\right)^T$

return $\boldsymbol{\chi}_k$ and $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$

distribution in the R-PFI approach, similar to that of the general PF, the particles do not satisfy the observation model. The strategy proposed in the current work uses the concept of Kalman gain to modify each particle based on the observations. To increase efficiency of the R-EKPFI, iterations within the versatile IEKF are avoided, which in turn means that each particle is modified only once. The current work claims that since multiple particles are involved, the linearization error is averaged across them, resulting in an overall reduced error. Furthermore, as explained in Section 2.2.3, the observations are modified by using the modified states. However, in the context of R-EKPFI, the modification of the observations is avoided. The reason is to avoid introducing additional uncertainty into the filter by modifying the observations with non-optimal particles. To further explain, by not performing iterations to modify each particle, it cannot be guaranteed that the modified particle is the best representative of the state. Consequently, after each particle is modified, if the observations are modified accordingly, there is a risk of degrading the original sensor data. As a result, subsequent particles will be modified using worsened observations, which in turn will lead to incorrect modification. Therefore, the performance of the filter is negatively affected, resulting in divergence or unreliable estimates. To be clearer, the current work claims that modifying each particle only once reduces the overall error of the linearization, but the modified particle is still not good enough to be used for the purpose of modifying available observations. Therefore, instead of using the equation (2.48) of the versatile IEKF, the following equation is used to modify each particle $\left(\boldsymbol{x}_k^{[s]}\right)$:

$$\check{\boldsymbol{x}}_k^{[s]} = \boldsymbol{x}_k^{[s]} - \boldsymbol{K}_k^{[s]} \cdot \boldsymbol{h}\left(\boldsymbol{l}_k + \boldsymbol{\nu}_k, \boldsymbol{x}_k^{[s]}\right), \tag{3.12}$$

where $\boldsymbol{x}_k^{[s]}$ and $\check{\boldsymbol{x}}_k^{[s]}$ show the particle before and after modification, respectively. To derive the Kalman gain based on $\boldsymbol{x}_k^{[s]}$, the following equation holds:

$$\begin{aligned}
\boldsymbol{O}_k^{[s]} &= \boldsymbol{H}_{x,k}^{[s]} \cdot \overline{\boldsymbol{\Sigma}}_{\boldsymbol{\chi}_k} \cdot \boldsymbol{H}_{x,k}^{[s]\,T} \\
\boldsymbol{S}_k^{[s]} &= \boldsymbol{H}_{l,k}^{[s]} \cdot \boldsymbol{Q}_k \cdot \boldsymbol{H}_{l,k}^{[s]T} \\
\boldsymbol{K}_k^{[s]} &= \overline{\boldsymbol{\Sigma}}_{\boldsymbol{\chi}_k} \cdot \boldsymbol{H}_{x,k}^{[s]\,T} \cdot \left(\boldsymbol{O}_k^{[s]} + \boldsymbol{S}_k^{[s]}\right)^{-1},
\end{aligned} \tag{3.13}$$

where $\boldsymbol{H}_{x,k}^{[s]}$ and $\boldsymbol{H}_{l,k}^{[s]}$ should be calculated by using $\boldsymbol{x}_k^{[s]}$ and $\boldsymbol{l}_k$ and based on equations (2.44) and (2.45), respectively. In addition, $\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\chi}_k}$ is the VCM of the states, which is derived by using the predicted particles. After modifying the particle $\boldsymbol{x}_k^{[s]}$, the predicted $\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\chi}_k}$ should also be modified as follows:

$$\begin{aligned}
\boldsymbol{L}_k^{[s]} &= \boldsymbol{I} - \boldsymbol{K}_k^{[s]} \cdot \boldsymbol{H}_{x,k}^{[s]} \\
\boldsymbol{\Sigma}_{\boldsymbol{\chi}_k} &= \boldsymbol{L}_k^{[s]} \cdot \overline{\boldsymbol{\Sigma}}_{\boldsymbol{\chi}_k} \cdot \boldsymbol{L}_k^{[s]T} + \boldsymbol{K}_k^{[s]} \cdot \boldsymbol{S}_k^{[s]} \cdot \boldsymbol{K}_k^{[s]T}.
\end{aligned} \tag{3.14}$$

The modified particle $\left(\check{\boldsymbol{x}}_k^{[s]}\right)$ is considered a sample from the posterior distribution. However, instead of directly using $\check{\boldsymbol{x}}_k^{[s]}$ in the subsequent steps of the filter, it is recommended to generate $\boldsymbol{x}_k^{[s]}$ from a normal distribution with an expected value and uncertainty corresponding to $\check{\boldsymbol{x}}_k^{[s]}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\chi}_k}$, respectively $\left(\mathcal{N}\left(\boldsymbol{x}_k^{[s]};\, \check{\boldsymbol{x}}_k^{[s]},\, \boldsymbol{\Sigma}_{\boldsymbol{\chi}_k}\right)\right)$. The reason for this is to preserve the randomness of the resulting R-EKPFI frame. Such a feature accounts for any negative effects due to outliers in the

observations. Although the existence of such misleading observations alters the particles incorrectly, it is claimed that the regeneration of the modified particle avoids systematic deterioration of the particles. Repeating this strategy for all particles yields a modified set of samples that more closely aligns with the posterior distribution. Therefore, as will be demonstrated in Section 3.4, state estimation can be performed using a significantly smaller number of particles.
Algorithm 10 gives the pseudo-code of the proposed R-EKPFI. The only difference to Algorithm 9 is from line 5 to line 11. In this segment, the proposed distribution (line 4 of both algorithms) is replaced by a more certain one obtained by using observations. After generating a new set of particles, the rest of the algorithm (starting from line 12) remains unchanged and is analogous to lines 5 through 18 in Algorithm 9.

Algorithm 10: Pseudo-code of the R-EKPFI.

$\left[\boldsymbol{\chi}_k,\ \boldsymbol{\mu}_k,\ \boldsymbol{\Sigma}_k\right] = \textbf{R-EKPFI}\left(\boldsymbol{\chi}_{k-1},\ \boldsymbol{l}_k,\ \boldsymbol{u}_k\right)$

$\overline{\boldsymbol{\chi}}_k = \boldsymbol{\chi}_k = \emptyset$

for $s = 1 : S$ **do**

Prediction step

sample $\boldsymbol{x}_k^{[s]} \sim \boldsymbol{p}\left(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}^{[s]}, \boldsymbol{u}_k\right)$ `according to` $\overline{\boldsymbol{bel}}\left(\boldsymbol{x}_{0:k}\right)$ `in equation (2.67)`

Modification step

$\boldsymbol{O}_k^{[s]} = \boldsymbol{H}_{x,k}^{[s]} \cdot \overline{\boldsymbol{\Sigma}}_{\boldsymbol{\chi}_k} \cdot \boldsymbol{H}_{x,k}^{[s]\,T}$

$\boldsymbol{S}_k^{[s]} = \boldsymbol{H}_{l,k}^{[s]} \cdot \boldsymbol{Q}_k \cdot \boldsymbol{H}_{l,k}^{[s]\,T}$

$\boldsymbol{K}_k^{[s]} = \overline{\boldsymbol{\Sigma}}_{\boldsymbol{\chi}_k} \cdot \boldsymbol{H}_{x,k}^{[s]\,T} \cdot \left(\boldsymbol{O}_k^{[s]} + \boldsymbol{S}_k^{[s]}\right)^{-1}$ `equation (3.13)`

$\check{\boldsymbol{x}}_k^{[s]} = \boldsymbol{x}_k^{[s]} - \boldsymbol{K}_k^{[s]} \cdot \boldsymbol{h}\left(\boldsymbol{l}_k + \boldsymbol{\nu}_k, \boldsymbol{x}_k^{[s]}\right)$ `equation (3.12)`

$\boldsymbol{L}_k^{[s]} = \boldsymbol{I} - \boldsymbol{K}_k^{[s]} \cdot \boldsymbol{H}_{x,k}^{[s]}$

$\boldsymbol{\Sigma}_{\boldsymbol{\chi}_k} = \boldsymbol{L}_k^{[s]} \cdot \overline{\boldsymbol{\Sigma}}_{\boldsymbol{\chi}_k} \cdot \boldsymbol{L}_k^{[s]\,T} + \boldsymbol{K}_k^{[s]} \cdot \boldsymbol{S}_k^{[s]} \cdot \boldsymbol{K}_k^{[s]\,T}$ `equation (3.14)`

sample $\boldsymbol{x}_k^{[s]} \sim \mathcal{N}\left(\boldsymbol{x}_k^{[s]};\ \check{\boldsymbol{x}}_k^{[s]},\ \boldsymbol{\Sigma}_{\boldsymbol{\chi}_k}\right)$

Update step

$\hat{\boldsymbol{r}}_k^{[s]} = \boldsymbol{h}\left(\boldsymbol{l}_k + \boldsymbol{\nu}_k, \boldsymbol{x}_k^{[s]}\right)$ `equation (3.5)`

$Q_1 = y_{(n+1)\times\frac{1}{4}}$ `equation (3.7)`

$Q_3 = y_{(n+1)\times\frac{3}{4}}$ `equation (3.7)`

$IQR = Q_3 - Q_1$ `equation (3.8)`

Outlier detection

$\forall \left[\left|\hat{r}_{k,j}^{[s]}\right|\right]_{j=1}^{j=q} \in \left|\hat{\boldsymbol{r}}_k^{[s]}\right| : \quad \left|\hat{r}_{k,j}^{[s]}\right| \leq Q_1 - 1.5 \times IQR \ \mid \ \left|\hat{r}_{k,j}^{[s]}\right| \geq Q_3 + 1.5 \times IQR\ , \quad q \leq n$

$\tilde{r}_k^{[s]} = \frac{1}{q} \sum_{j=1}^{j=q} \left|\hat{r}_{k,j}^{[s]}\right|$ `equation (3.10)`

$\omega_k^{[s]} = p\left(\tilde{r}_k^{[s]}\right)$ `equation (3.11)`

add $\boldsymbol{x}_k^{[s]}$ to $\overline{\boldsymbol{\chi}}_k$

end

(Re-) Sampling step

for $s = 1 : S$ **do**

draw s with probability $\propto \omega_k^{[s]}$

add $\boldsymbol{x}_k^{[s]}$ to $\boldsymbol{\chi}_k$

end

State estimation step

$\boldsymbol{\mu}_k = \frac{1}{S} \sum \boldsymbol{\chi}_k$

$\boldsymbol{\Sigma}_k = \frac{1}{S-1} \sum \left(\boldsymbol{\chi}_k - \boldsymbol{\mu}_k\right) \cdot \left(\boldsymbol{\chi}_k - \boldsymbol{\mu}_k\right)^T$

return $\boldsymbol{\chi}_k$ and $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$

3.4 Numerical Validation

In order to evaluate the performance of the introduced PF-based methodologies, namely the PFI, R-PFI and R-EKPFI, a numerical example is investigated in the following. In addition, the versatile IEKF method given by Algorithm 5 is also applied to the same example as a baseline against which to compare the other three methods. This example provides numerical support for the statements made in the previous sections about each of the newly developed methods, which shows their general applicability for more complicated applications such as MSS georeferencing. Therefore, their application, including the pros and cons of using them, is presented in practice, rather than remaining purely theoretical.

3.4.1 Problem Formulation

The main purpose of the investigated example is to estimate the parameters of a 3D plane by considering a total number of 10000 observations. However, to allow the use of filtering approaches, the parameter estimation is done recursively over 100 epochs using 100 observations that are randomly selected from these 10000 data. To be more concise, the total number of observations is divided over the 100 epochs so that there are no repeated observations over the epochs.
The state vector at each epoch k is as follows:

$$\boldsymbol{\mu}_k = \begin{bmatrix} n_{x,k} & n_{y,k} & n_{z,k} & d_k \end{bmatrix}^T , \tag{3.15}$$

where $n_{x,k}$, $n_{y,k}$ and $n_{z,k}$ are the 3D components of the normal vector $\boldsymbol{n}_k$, while d_k represents the distance of the plane to the origin. Since the parameters of the plane do not change over time, the state transition function is mainly affected by the process noise ($\boldsymbol{w}_k$) as follows:

$$\begin{aligned} \boldsymbol{\mu}_k &= \boldsymbol{F}_k \cdot \boldsymbol{\mu}_{k-1} + \boldsymbol{w}_k \ , \qquad \boldsymbol{w}_k \ \sim \ \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{R}_k\right)) \\ \boldsymbol{F}_k &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} . \end{aligned} \tag{3.16}$$

As explained in Section 2.1 of Chapter 2, adding the process noise is to overcome the uncertainties in the transition function. In principle, for this particular example, it can be claimed that such an uncertainty does not exist due to the parameters being constant over time. However, the current work claims that the inclusion of the process noise is essential. If the filters are not reliably initialized, not considering the process noise can lead to wrong predictions. If the unreliability of the predictions is high, considering the observations in the update step cannot compensate for it. As a result, the filter diverges over time. For this example, each component of $\boldsymbol{w}_k$ follows a Gaussian distribution with mean $\mu_w = 0$ and standard deviation $\sigma_w = 0.001$. It is important to note that the process noise values have a significant impact on the estimations. In general, the determination of appropriate values is an optimization problem that is application-dependent. However, for this example no optimization is performed in the sense of comparing the effect of different process noise values on the accuracy of the estimated states. Instead, the standard deviation σ_w is chosen in such a way that any unreliable predictions of the filters can be resolved by considering the observations in the update step. As a result, the considered value $\sigma_w = 0.001$ is not the most optimal value. It is a value at which the filters do not fail to estimate the states. The current work suggests that this value is sufficient to compare the performance of the developed PF-based methods with the versatile IEKF framework.

The observation model is based on the Hesse normal form, which can be written as follows (Bôcher, 1915):

$$(\boldsymbol{P}_k + \boldsymbol{v}_k) \cdot \boldsymbol{n}_k - d_k = \boldsymbol{0} \ , \qquad \boldsymbol{v}_k \ \sim \ \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_k) \, , \tag{3.17}$$

where $\boldsymbol{P}_k$ is a matrix of size $n \times 3$ containing a total of $n = 100$ observation data points with 3D coordinates $\begin{bmatrix} P_{x,i} & P_{y,i} & P_{z,i} \end{bmatrix}_{i=1}^{n}$. Moreover, $\boldsymbol{v}_k$ denotes the noise in the observations, with a VCM of $\boldsymbol{Q}_k$ with size 300×300. In this example, each component of $\boldsymbol{v}_k$ follows a Gaussian distribution with mean $\mu_v = 0$ and standard deviation $\sigma_v = 0.5$.
In addition, the initialization of the filters is as follows:

$$\boldsymbol{\mu}_0 = \tilde{\boldsymbol{\mu}} + \boldsymbol{a} \cdot \tilde{\boldsymbol{\mu}}, \tag{3.18}$$

where $\tilde{\boldsymbol{\mu}}$ represents the true values of parameters of the plane. Additionally, $\boldsymbol{a}$ is a vector of random errors, each generated from a Gaussian distribution with the standard deviation of $\sigma_a = 0.1$. The intention is to initialize the state vector using these randomly perturbed true states. Choosing $\sigma_a = 0.1$ means that the initialization differs 10% from the ground truth. In the current work, this is considered a poor initialization, which can further show the performance of each filter. Considering values greater than 0.1 for σ_a caused the versatile IEKF to fail. Therefore, considering $\sigma_a = 0.1$ ensures that all the filters work. However, the effect of such a poor initialization is reflected in the results.
Using the Hesse normal form of the observation model requires a unit normal vector. Consequently, it is crucial to ensure the magnitude of the estimated normal vector is $\|\boldsymbol{n}_k\| = 1$. Having such a constraint in the introduced PF-based methods requires additional modification of the particles based on the applied restriction. It should be noted that the taken strategy in this regard is limited to the current example and cannot serve as a global approach for other geometrical constraints. In this thesis, the topic of constraints as an additional exclusive restriction is out of scope. Considering such a limiting information in the current example is solely to show the potential of the introduced PF-based frameworks to handle constraints.
This constraint is applied in two steps in both the PFI and the R-PFI. First, the normal vector of the particles is normalized following the prediction step. This normalization occurs after line 4 of algorithms 8 and 9 and prior to the update step. A second normalization is conducted after estimating the state vector using resampled particles. For the R-EKPFI, in addition to these two steps, intermediate normalization is applied after modifying the particles during the KF step. This occurs after line 11, and before entering the update step.
In the versatile IEKF methodology, the unit normal vector requirement is addressed using the *projection method*. Since the constraint-related methodologies are not the focus of the current work, the reader is referred to the work of Simon and Chia (2002). The algorithm employed in the versatile IEKF, which includes projection method, can also be found in Vogel et al. (2018).
In addition, the number of particles used in case of the PFI and R-PFI is 1000, while in case of the R-EKPFI framework a number of 20 particles is used to estimate the state vector in each epoch.

3.4.2 Evaluation Measures

In order to evaluate the performance of the different filters, two evaluation measures, namely the RMSE and the KLD are investigated. In general, the RMSE is an estimator that is used to realize the divergence of the estimated states from the ground truth. Furthermore, by using the KLD measure, the deviations of the derived posterior distribution from the true one can be determined. If the true distribution of states is available, the use of the RMSE measure can be neglected. The reason is that the KLD measure gives a more complete insight into the performance of the filter by

evaluating the obtained posterior distributions. To be more precise, knowing the divergence of the resulting posterior distributions from the true ones implies the correctness of the estimated states. However, in this thesis, the true distribution of states is unknown. Furthermore, the performance of the PF-based methods is to be realized against the versatile IEKF. Consequently, the KLD measure is used to obtain the dissimilarity of the estimated posterior distributions by the PFI, R-PFI, and R-EKPFI to the derived Gaussian distributions from the versatile IEKF. To evaluate the performance of each filter, and since the true parameters of the planes are known, the divergence of the estimated states from the ground truth is derived using the RMSE measure.
In the following, the use of RMSE is first explained in Section 3.4.2.1. Then the general idea of the KLD measure is described in Section 3.4.2.2. These two sections are followed by the results of the analyses, which are given in Section 3.4.3.

3.4.2.1 Root Mean Square Error

The RMSE is a widely-recognized statistical measure used to evaluate the model performance (Chai and Draxler, 2014). Based on the general formulation of RMSE, and assuming that the true state vector ($\tilde{\boldsymbol{\mu}}$) is available, the following equation is applicable:

$$\boldsymbol{RMSE}_k = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(\boldsymbol{\mu}_k - \tilde{\boldsymbol{\mu}})^2}, \tag{3.19}$$

where $\boldsymbol{RMSE}_k$ is referred to as the *accumulative RMSE* in the current work. The reason behind is calculating the RMSE not only based on the current estimations ($\boldsymbol{\mu}_k$), but also considering all the estimations up to the current epoch. Therefore, k is the total number of the epochs until the current epoch ($i = k$). The main reason for accumulating the errors of the estimated states is to monitor the performance of the filters over time. This measure helps to determine whether the performance is improving, degrading, or constant. An improving performance results in a decreasing pattern of the accumulative RMSE, while a degrading performance leads to an increasing pattern of the accumulative RMSE.
Equation (3.19) is valid when the filters are applied just once. However, to ensure the stability of the methodologies, they are applied to different data sets. For this purpose, the principle of MC simulation is utilized (Mooney, 1997). The idea is to iteratively generate new observations by adding random noise to the true ones. Doing so, the methodologies are indeed applied on different data sets. Therefore, their functionality can better be evaluated compared to applying them only once. In the current example, a total of 50 MC runs are considered. The reason for considering 50 MC runs is the computational restriction. Also, it is claimed that using 50 runs yields a general insight into the performance of the filters. The given results in Section 3.4.3 are averages over these MC runs. In order to derive the average $RMSE$ at each epoch k over the MC runs $\left(\overline{\boldsymbol{RMSE}_k}\right)$, the following equation holds:

$$\overline{\boldsymbol{RMSE}_k} = \frac{1}{M}\sum_{j=1}^{M}\boldsymbol{RMSE}_{k,j}, \tag{3.20}$$

where M indicates the total number of the MC runs, which in this example is $M = 50$.

3.4.2.2 Kullback-Leibler Divergence

KLD is a well-known statistical distance that is used to compare an estimated distribution (Q) with a reference distribution (P) (Kullback and Leibler, 1951). In other words, it is an indicator of how similar $\left(\hat{D}_{KL}(P,Q) = 0\right)$ or dissimilar $\left(\hat{D}_{KL}(P,Q) > 0\right)$ two distributions are. The benefit of this

measure becomes evident in applications that result in multi-modal distributions. For example, in georeferencing of MSSs and due to different sources of uncertainty, having a uni-modal distribution cannot be ensured. In such a case, it is essential to confirm the correctness of the estimated posterior distribution, which can be obtained by using PF-based frameworks.
In a continuous measurable space (χ), as given by Bishop and Nasrabadi (2006), the following equation for the KLD measure $\left(\hat{D}_{KL}\left(P,Q\right)\right)$ holds:

$$\hat{D}_{KL}\left(P,Q\right) = \int_{\chi} P\left(dx\right) \cdot log\left(\frac{P\left(dx\right)}{Q\left(dx\right)}\right). \tag{3.21}$$

For discrete probability distributions, equation (3.21) is adapted as follows (MacKay, 2003):

$$\hat{D}_{KL}\left(P,Q\right) = \sum_{x\in\chi} P\left(x\right) \cdot log\left(\frac{P\left(x\right)}{Q\left(x\right)}\right). \tag{3.22}$$

To estimate the KLD based on equations (3.21) and (3.22), the same measurable space (χ) is required for P and Q. This means that the number of samples in the two distributions must be the same. Additionally, both P and Q distributions are required to be known. In the PF, however, one primary aim is to derive unknown posterior distributions, rendering the reference distribution P typically unknown.
One of the ideas is to use histograms, but this approach limits the use of equation (3.22) to low-dimensional (Chou et al., 2011) state vectors. However, in practical applications such as georeferencing of MSSs, the state vector contains at least six parameters to be estimated. Therefore, it must be ensured that the obtained posterior distributions for these six states are reliable. To do so, the use of equation (3.22) is not possible, as it does not allow for a joint measure of the correctness of the estimated distributions. Recognizing these constraints Chou et al. (2011) proposed the following equation to address both the issue of varying sample sizes and high dimensionality of state vectors:

$$\hat{D}_{KL}\left(P,Q\right) = \frac{d}{n}\sum_{i=1}^{n} log\frac{\alpha_{k_i}\left(i\right)}{\rho_{l_i}\left(i\right)} + \frac{1}{n}\sum_{i=1}^{n}\left[\psi\left(l_i\right) - \psi\left(k_i\right)\right] + log\frac{m}{n-1}, \tag{3.23}$$

where d is the dimension of the state vector, and n is the number of samples $\{\boldsymbol{X}_1, ..., \boldsymbol{X}_n\}$ derived from the reference distribution (P). Additionally, m is the number of samples $\{\boldsymbol{Y}_1, ..., \boldsymbol{Y}_m\}$ from distribution Q which is compared with P. $\alpha_{k_i}\left(i\right)$ denotes the Euclidean distance between $\boldsymbol{X}_i$ and its k_i nearest neighbours (NNs) in $\{\boldsymbol{Y}_j\}$, while $\rho_{l_i}\left(i\right)$ indicates the Euclidean distance between $\boldsymbol{X}_i$ and its l_i in $\{\boldsymbol{X}_j\}_{j\neq i}$. As Chou et al. (2011) explains, instead of the Euclidean distance, any other distance measures can be used. Lastly, ψ is the Digamma function, which is the logarithmic derivative of the Gamma function (Abramowitz and Stegun, 1968).
In the investigated example, the proposed KLD measure by Chou et al. (2011) is used. The primary reason for this choice is twofold: first, the state vector is highly dimensional ($u = 4$), and second, the goal is to compare the performance of the developed PF-based frameworks with the versatile IEKF. Therefore, the sample sizes are different.

3.4.3 Results

In the following, results of the estimated planes by the PFI, R-PFI, R-EKPFI and versatile IEKF are given. In the current example, results of the R-PFI and R-EKPFI show an average removal of outliers of approximately 2% in each epoch. It should be noted that no artificial outliers are created

for this test experiment. When talking about removing outliers, those observations are meant that do not fit the pattern of the majority of the data. To be more precise, those observations are meant that result in implicit residual values that are not in the order of magnitude of the other values resulting from the rest of the data points. The removal of such data points by the R-PFI and R-EKPFI suggests that they are capable of dealing effectively with such discrepancies should they occur in real-world applications.
The following figures depict the results for the estimated n_x and d. Corresponding plots to the n_y and n_z are provided in the Appendix (A.1). In the legend of the plots, the versatile IEKF is simply referred to as IEKF. Figure 3.2 depicts the average accumulative RMSE for the estimated n_x and d across the 50 MC runs. The reason for not specifying a unit for the plots related to d is that for this example, it is considered to have a unit of length. Moreover, equation (3.20) is employed to calculate the RMSE values in each epoch. As observed, the PFI and R-PFI yield poor estimations compared to the R-EKPFI and versatile IEKF algorithms. In the left plot of this figure, it is evident that the outlier removal in the R-PFI leads to improved estimations compared to the PFI, where no outliers are detected and removed. However, as seen in the right plot, the same cannot be said for the estimated d state. As explained in Section 3.4.1, equation (3.18) is used to initialize the state vector ($\boldsymbol{\mu}_0$). Across the MC runs, this mostly results in normalized vectors close to the true ones. However, the initialized distances of the planes can deviate substantially from the true distances. In addition, in the R-PFI algorithm fewer observations are considered for estimation due to outlier elimination. Nevertheless, similar to the R-PFI, 1000 particles are utilized in each epoch. Consequently, it is concluded that this setting cannot rectify the issues stemming from incorrect initial states. In such a case, increasing the number of the particles in the R-PFI helps to compensate for the incorrect initialization. A similar conclusion applies to the estimated states by the PFI algorithm. By increasing the number of particles, the adverse effects of outliers is expected to be mitigated, leading to improved estimations. The reason for such a claim – as explained in Section 3.2 – is the failure of the PFI to distinguish between the good and bad particles. Therefore, if the number of particles is not sufficient, such an incorrect selection of particles cannot be compensated, which leads to the divergence of the filter. However, in the case of a large number of particles, such an incorrect resampling can be corrected over time due to the variation that exists between the particles. Both initialization and the number of particles influences the reliability of the estimations by the PFI and R-PFI algorithm. These challenges are overcome by R-EKPFI algorithm. As seen in the same figure, the estimated n_x and d states by the R-EKPFI are in the same range of those derived by the versatile IEKF. Remarkably, the R-EKPFI accomplishes this using 98% fewer particles than the PFI and R-PFI, demonstrating the efficacy of particle correction by observations in this framework.

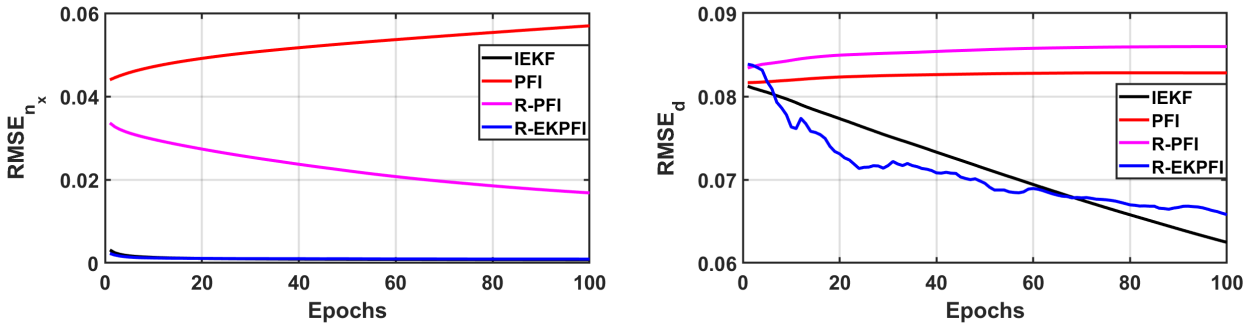

Figure 3.2: *Average accumulative RMSE of the estimated n_x (left plot) and d (right plot) over 50 MC runs.*

By applying MC runs, it is possible to not only carry out the estimations in each epoch, but also to compute the statistical values related to these estimations. Doing so allows for a more robust evaluation of the performance of the filters. The reason for this is that by using MC runs, different

sets of observations are considered, which in turn ensures that the performance of the filters is not coincidental. Figures 3.3 and 3.4 depict the 95% confidence interval (CI), minimum and maximum bounds (Min - Max bound), along with the mean and median values of the average accumulative RMSE values over the MC runs for each filter. In both figures, a wider 95% CI means less stability in the estimated RMSE values, revealing inconsistencies across MC runs. In such a case, due to the lower stability, the mean and median values differ from each other. Such a pattern can be seen in the statistical values of the PFI and R-PFI estimates. Both the estimated n_x and d values exhibit a wide CI, indicating unstable estimations with these algorithms. Conversely, the estimated n_x values by the versatile IEKF show a narrower CI, validating the higher stability of the estimations by this algorithm. As it can be seen in this case, the mean and median value do not deviate significantly. However, a wide CI can be seen for the estimated d values. This shows a high instability of the estimations, highlighting the impact of uncertain initial values on the versatile IEKF. For the R-EKPFI, it can be seen that the CI of the estimated n_x values is comparable to the versatile IEKF and significantly narrower than the CIs derived for the PFI and R-PFI frameworks. This underscores the advantage of adjusting the particles based on the available observations in the R-EKPFI algorithm. Additionally, this adjustment notably affects the estimated d values. Unlike the other filters, using the R-EKPFI results in a narrower CI and, thus, more stable estimations across the MC runs. To better compare the numerical results, the corresponding statistical values to the last epoch of figures 3.3 and 3.4 is given in tables 3.1 and 3.2, respectively. In these tables, Min and Max indicate the minimum and maximum bounds, respectively. Furthermore, ↑ CI (95%) and ↓ CI (95%) refer to the upper and lower bounds of the 95% CI.

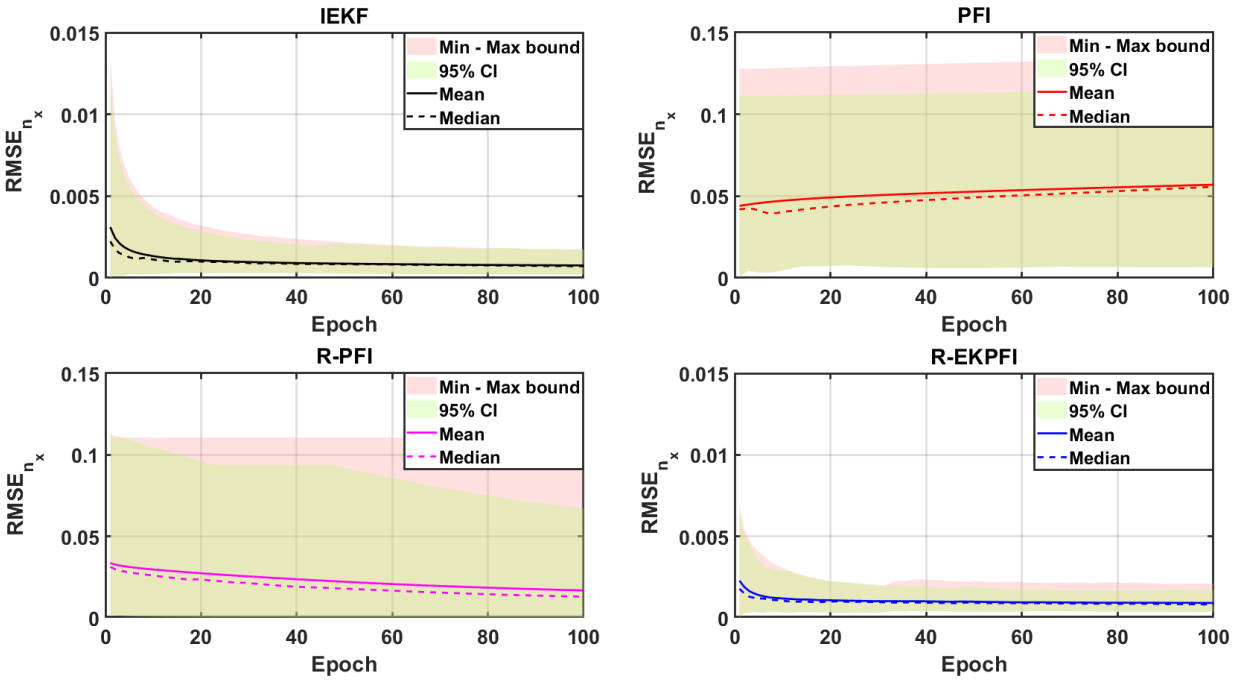

Figure 3.3: *Statistical values of the accumulative RMSE of the estimated n_x over 50 MC runs.*

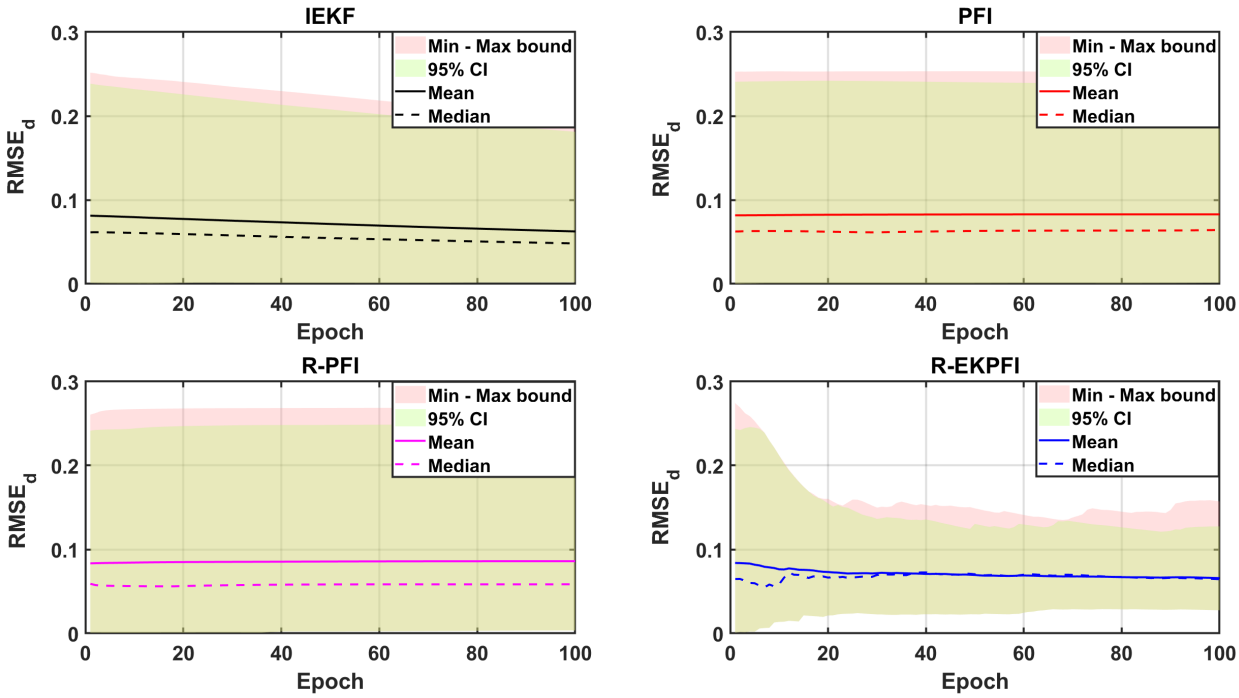

Figure 3.4: *Statistical values of the accumulative RMSE of the estimated d over 50 MC runs.*

Table 3.1: *Statistical values of the accumulative RMSE of the estimated n_x in the last epoch over the 50 MC runs by means of the different filters. Red and green colors indicate the minimum and maximum values in each column, respectively.*

	$RMSE_{n_x}$					
	Min	Max	Mean	Median	↑ CI (95%)	↓ CI (95%)
versatile IEKF	$1.91 \cdot 10^{-4}$	0.0018	$7.59 \cdot 10^{-4}$	$7.19 \cdot 10^{-4}$	0.0017	$1.91 \cdot 10^{-4}$
PFI	0.0065	0.1343	0.0570	0.0556	0.1152	0.0065
R-PFI	$3.48 \cdot 10^{-4}$	0.1104	0.0168	0.0130	0.0671	$3.48 \cdot 10^{-4}$
R-EKPFI	$3.80 \cdot 10^{-4}$	0.0021	$9.15 \cdot 10^{-4}$	$8.19 \cdot 10^{-4}$	0.0017	$3.80 \cdot 10^{-4}$

Table 3.2: *Statistical values of the accumulative RMSE of the estimated d in the last epoch over the 50 MC runs by means of the different filters. Red and green colors indicate the minimum and maximum values in each column, respectively.*

	$RMSE_d$					
	Min	Max	Mean	Median	↑ CI (95%)	↓ CI (95%)
versatile IEKF	0.0017	0.20	0.0625	0.0482	0.1812	0.0017
PFI	0.0014	0.2514	0.0828	0.0641	0.2387	0.0014
R-PFI	0.0035	0.2683	0.0860	0.0585	0.2498	0.0035
R-EKPFI	0.0275	0.1572	0.0658	0.0645	0.1275	0.0275

When the estimated states are compared to the ground truth, the RMSE values give a measure of the accuracy for each filter. In addition to accuracy, the precision of each filter can also be derived by investigating the calculated VCM of the estimated states. In the case of the versatile IEKF, a smaller STD indicates higher precision, meaning that the objective function is minimized

more successfully than when the STD is larger. In the PF-based methods, the precision indicates the scatter between the particles, which is an indicator of the agreement between the particles regarding the estimated state. The foundations of the KF-based and PF-based methods in terms of precision are different. However, the current work claims that in both frameworks, the precision can be interpreted as the amount of reliability that can be considered for the estimated states. The stability of the precision is investigated by examining the STD over the MC runs.
Figure 3.5 presents the precision of the estimated n_x and d states for each filter. It can be seen that in case of the estimated n_x, the highest and lowest precision belongs to the versatile IEKF and PFI, respectively. Furthermore, removing the outliers in case of the R-PFI enhances its precision. This improvement arises from the diminished influence of outliers on calculating the importance weight of each particle. Doing so, the resampled particles are scattered closer to the estimated state.
Finally, modifying the particles in the R-EKPFI has led to a precision close to that of the versatile IEKF. In this case, unlike the versatile IEKF, an irregular pattern for the precision can be seen. This irregularity is attributed to the use of only 20 particles in the R-EKPFI, which are not only modified by means of the available observations in each epoch, but they are also regenerated around the modified sample. Consequently, due to the regeneration, their scattering around the mean (estimated state) fluctuates from one epoch to the other. However, since in each epoch they are modified based on the available observations, the resulting STD of the estimated state remains in the same order of magnitude over time. Conversely, according to the precision of the estimated d state, it can be seen that the R-EKPFI has the lowest precision compared to the other filters. Having such a result directly reflects the effect of uncertain initialized d state on the R-EKPFI. In this case, although the mean of the resampled particles in each epoch is close to the true value (see Figure 3.2), their dispersion is large. Such an effect is claimed to be overcome by using more particles in case of the R-EKPFI. However, doing so, the computation time will increase.
Furthermore, it can be seen that the other filters, namely the PFI, R-PFI and R-EKPFI have a comparable precision to one another. Having such results further proves that in case of wrong initialized values, the versatile IEKF results in a precision similar to the PFI and R-PFI.

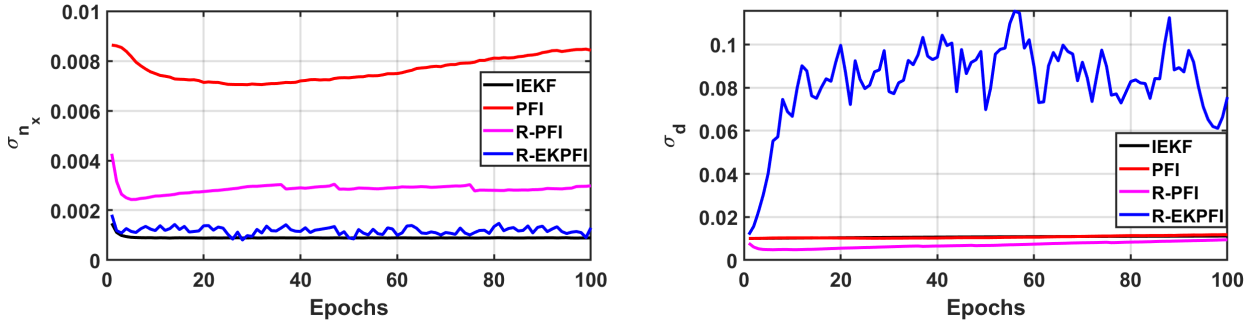

Figure 3.5: *Average precision of the estimated n_x (left plot) and d (right plot) over 50 MC runs.*

Similar to figures 3.3 and 3.4, figures 3.6 and 3.7 present the corresponding statistical values related to the precision of the estimated n_x and d states over multiple MC runs. According to these figures the versatile IEKF exhibits a narrow CI for both the resulting σ_{n_x} and σ_d, confirming the stability of the precision estimations obtained by this filter across the MC runs. For σ_d, an increasing pattern is observed over time, suggesting that minimization of the objective function over time is not optimal. Such a pattern results from the framework of the versatile IEKF in which the estimated states and their corresponding VCM of epoch $k-1$ are directly used in epoch k. Consequently, any unsatisfactory precision of the d state (σ_d) will accumulate, resulting in an overall larger σ_d over time. In contrast, for σ_{n_x}, a decreasing pattern over time is observed, indicating that the initialization for this state was appropriately selected. For both the PFI and R-PFI, the estimated precision values for σ_{n_x} and d appear unstable, as indicated by the wide CI region for both states. This instability is mitigated in the R-EKPFI framework by modifying the particles. The resulting

CI for both the estimated σ_{n_x} and σ_d indicates the stability of the precision values over the MC runs. However, the wider CI regions in this instance compared to those derived by the versatile IEKF, are due to the random effect inherent in the R-EKPFI's the particle-based approach. For a better comparison of the numerical results, the corresponding values to the last epoch of figures 3.6 and 3.7 is given in tables 3.3 and 3.4, respectively.

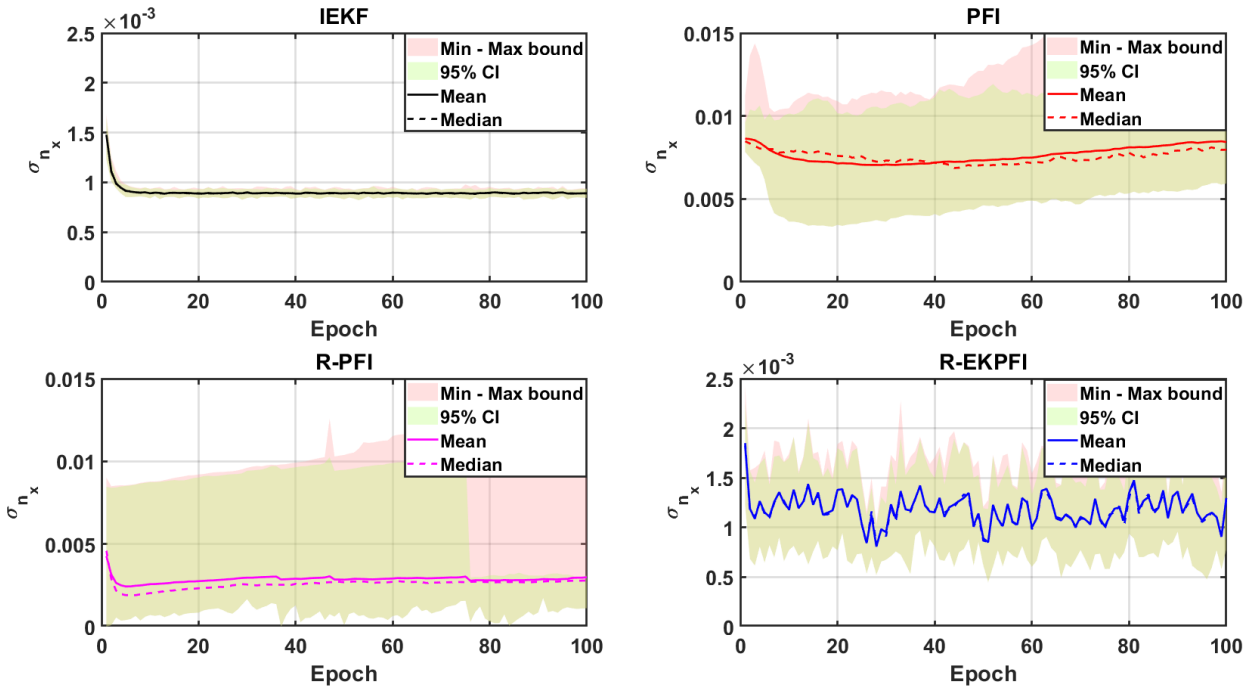

Figure 3.6: *Statistical values of the precision of the estimated n_x over 50 MC runs.*

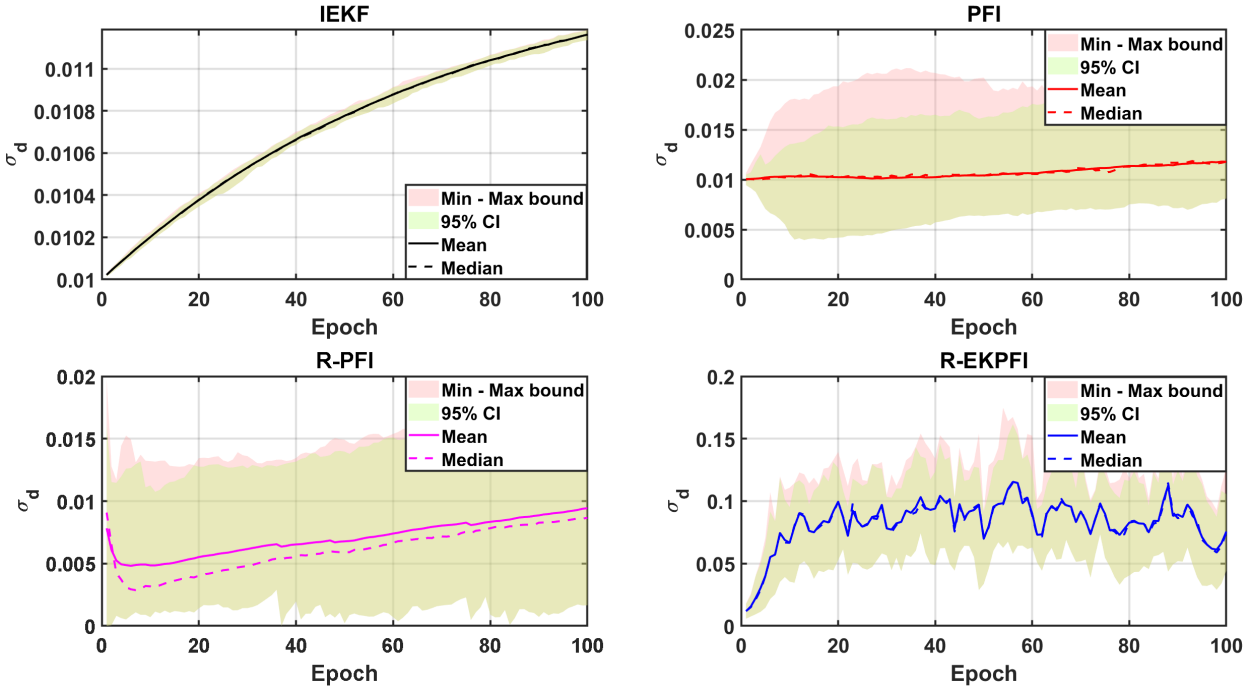

Figure 3.7: *Statistical values of the precision of the estimated d over 50 MC runs.*

Table 3.3: *Statistical values of the precision of the estimated n_x in the last epoch over the 50 MC runs by means of the different filters. Red and green colors indicate the minimum and maximum values in each column, respectively.*

	σ_{n_x}					
	Min	Max	Mean	Median	↑ CI (95%)	↓ CI (95%)
versatile IEKF	$8.44 \cdot 10^{-4}$	$9.37 \cdot 10^{-4}$	$8.91 \cdot 10^{-4}$	$8.88 \cdot 10^{-4}$	$9.35 \cdot 10^{-4}$	$8.44 \cdot 10^{-4}$
PFI	0.0059	0.0135	0.0084	0.0080	0.0130	0.0059
R-PFI	0.0011	0.0147	0.0030	0.0028	0.0031	0.0011
R-EKPFI	$7.79 \cdot 10^{-4}$	0.0019	0.0013	0.0013	0.0018	$7.79 \cdot 10^{-4}$

Table 3.4: *Statistical values of the precision of the estimated d in the last epoch over the 50 MC runs by means of the different filters. Red and green colors indicate the minimum and maximum values in each column, respectively.*

	σ_d					
	Min	Max	Mean	Median	↑ CI (95%)	↓ CI (95%)
versatile IEKF	0.0111	0.0112	0.0112	0.0112	0.0112	0.0111
PFI	0.0081	0.0211	0.0118	0.0118	0.0182	0.0081
R-PFI	0.0016	0.0179	0.0094	0.0087	0.0172	0.0016
R-EKPFI	0.0443	0.1266	0.0756	0.0751	0.1056	0.0443

Figure 3.8 shows the KLD estimate for each epoch, averaged over the MC runs. As explained in Section 3.4.2.2, the purpose of the KLD measure is to compare a desired distribution with a reference PDF. Ideally, this reference PDF would be the true one, but in the context of this work, the true distributions are unknown and beyond the scope of discussion. As explained in Section 3.4.2, the primary goal is to compare the performance of PFI, R-PFI, and R-EKPFI versus the versatile IEKF. Consequently, the latter serves as the reference distribution against which the PF-based frameworks are evaluated.

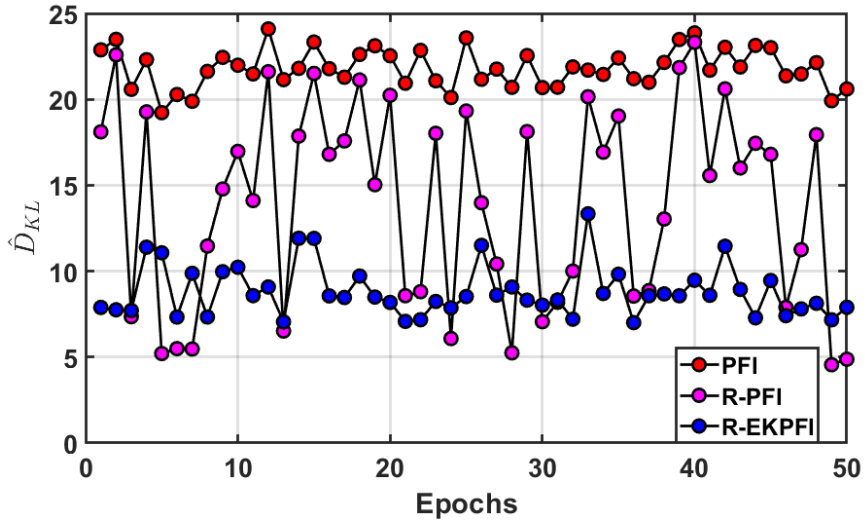

Figure 3.8: *Average of the KLD estimate $\left(\hat{D}_{KL}\right)$ in each epoch over 50 MC runs.*

For this comparison and in order to have the reference distribution, 1000 samples are generated in

each epoch from a multivariate Gaussian distribution with mean $\boldsymbol{\mu}_k$ and VCM $\boldsymbol{\Sigma}_k$. The reason for generating 1000 samples is to avoid high computation times when making comparisons involving the PFI framework across MC runs. While the current work claims that using 1000 samples is sufficient for such a relative comparison, it is essential to note that this claim lacks empirical support. Without results showing the performance stability over different sample sizes, this assertion remains speculative. To strengthen this point, sensitivity analyses showing the impact of varying the number of samples on the results would be beneficial. Additionally, equation (3.23) is employed to compute the KLD measure, considering the entire dimensionality of the state vector. Therefore, if due to any reason, one of the dimensions is poorly estimated, it will adversely affect the measure of KLD. An example of such influencing factors that can negatively affect the performance of the algorithm is uncertain initialization. Uncertain initialization refers to the practice of starting the algorithm with either random or poorly estimated initial values, which can lead to inaccurate results or even failure of the algorithm to converge to an optimal solution.

According to the Figure 3.8, it can be seen that for all of the three PF-based algorithms and as expected, the KLD measure is greater than zero $\left(\hat{D}_{KL} > 0\right)$. As elaborated in Section 3.4.2.2, a $\hat{D}_{KL} > 0$ signifies dissimilarity between two distributions. Consequently, it can be concluded that the calculated posterior distributions in PF-based frameworks deviate from those obtained through the KF-based approach. Examining the magnitude of the KLD measures reveals that the degree of this dissimilarity diminishes when a KF step is integrated into the filter. In other words, since each particle is modified by the available observations in the R-EKPFI methodology, the resulting resampled particles are more inclined towards a Gaussian distribution than the other frameworks. This non-zero KLD measure is due to to regeneration of each sample following the modification step, as indicated in line 11.

In the case of the R-PFI, employing the IQR method to estimate the likelihoods leads to having smaller KLD measures compared to the PFI algorithm. Applying this approach to remove the outliers and reduce the dimensionality of the implicit residual vector leads to having resampled particles that have less variations in their importance weights compared to the PFI framework. Consequently, the resulting posterior distributions by R-PFI are more similar to the Gaussian distributions derived by the versatile IEKF. It is even observed that at some epochs, the similarity of the R-PFI results to the Gaussian distributions is more than those derived by the R-EKPFI. However, such sudden decreases of the KLD measure are random, and thus, no solid statement can be given as to under which circumstances such an effect can be observed. Finally, the greatest dissimilarity belongs to the PFI results. As expected, this is because neither outliers are removed nor are the samples modified in this approach, resulting in resampled particles with highly variable importance weights.

Another aspect to consider when comparing different algorithms is the computational time. Since the MC runs are executed on a cluster system, averaging over the computation time across all MC runs does not provide a fair comparison between the filters. Therefore, to do such a comparison, the filters are compared in a single MC run. The results indicate that the versatile IEKF takes approximately 2 seconds to analyze all epochs, while the PFI, R-PFI and R-EKPFI require around 20, 80, and 24 seconds, respectively. The computation time in the case of the PF-based frameworks can potentially be improved by parallelizing the operations across multiple processors, which is beyond the scope of the current work. The given results for the computational time are intended to show that in the case of incorporating the Kalman gain concept, the computational time of the resulting R-EKPFI can be reduced by 70% compared to the R-PFI due to 98% fewer samples required. Another way to reduce the computation time of the R-EKPFI is to use fewer particles. In our experiments, reducing the number of particles to 10 instead of 20 affected the performance of the filter. While it may seem that fewer particles could speed up the computation, it is important to recognize that the trade-off is a reduction in the representational power of the filter. When using a PF-based approach such as R-EKPFI, the particles serve as discrete points that approximate a continuous distribution. A smaller particle set may lack the granularity required for accurate

representation, resulting in unreliable or unstable estimates. In challenging scenarios or high-dimensional state spaces, this can lead to filter divergence or failure.
In case of the PFI, the computation time is not comparable to the other two PF-based methodologies. The reason is due to the IQR method that is used in the R-PFI and R-EKPFI for each particle, which increases the computation time compared to the PFI.
In summary, the R-EKPFI framework shows promise in addressing challenges that the PFI and R-PFI methods struggle with. Among these challenges is the large number of observations, which introduces complexity to the PFI framework. In the case of R-PFI, the high computational time can be mentioned as its limiting aspect. In addition, it is confirmed that the R-EKPFI provides more robust performance in situations of uncertain initialization, which adds to its performance compared to the other frameworks including the versatile IEKF. Furthermore, the KLD measures also show that the derived R-EKPFI posterior distributions deviate from those obtained by the versatile IEKF, providing potential flexibility for complex applications. Finally, the implementation of constraints is simpler with the R-EKPFI than with the versatile IEKF. Therefore, for complex applications that require greater adaptability, the R-EKPFI is a better choice than the versatile IEKF.

4 Advanced Information-Based Georeferencing by Particle Filter

In Chapter 3, the performance of the developed methods – specifically PFI, R-PFI, and R-EKPFI – was demonstrated using a simple numerical example in comparison to the versatile IEKF. However, it is crucial to validate their performance in complex applications, such as localization, which is the focus of this work. Accordingly, this chapter is devoted to exploring the georeferencing of MSSs through the application of these proposed PF-based methods.
To achieve this objective, the chapter is organized as follows: The core principles of georeferencing are first outlined in Section 4.1. Subsequently, the setup required to apply the filters is described in Section 4.2. Furthermore, a simulated environment and a real-world georeferencing application are considered, which are described in detail in Section 4.3. Finally, the results obtained from the filters and their interpretation are presented in Section 4.4.

4.1 Core Principle

As suggested by Vogel et al. (2019), the MSS georeferencing can be enhanced by incorporating useful environmental information. Such information can include the geometric shapes of infrastructures, traffic lights and traffic signs. The main idea is to consider any additional knowledge that can be mathematically linked to the 6-DoF of the MSS. In this way, and assuming the information is reliable, the MSS can be georeferenced more accurately than by neglecting such evidence. Bureick et al. (2019) applied this principle of MSS georeferencing in urban environments to mitigate the limitations of IMU and GNSS data. In this case, the geometric information of the surrounding buildings and ground are leveraged to estimate the 6-DoF using the versatile IEKF. To use such information, first LiDAR sensors are used to capture the environment. Then, the scanned data are utilized together with the 3D city and digital terrain models to enhance georeferencing accuracy. Their results underscore the positive effect of considering reliable additional knowledge provided by the environment for improving the estimated 6-DoF. Therefore, a similar strategy is integrated in the framework of PF in this thesis to georeference MSSs in urban environments.
In general, 3D digital city and surface models provide georeferenced spatial data for a variety of urban elements, such as buildings and sites (Döllner et al., 2006). Depending on the granularity of the geometric representation, the 3D city models exist in different levels of detail. In this thesis, as in Bureick et al. (2019), LoD-2 models are used, which are at least freely available throughout Germany. These models provide the basic shapes of infrastructure elements but lack detailed features such as doors and windows. They also do not contain any information regarding the construction materials. Furthermore, the building facades are represented by planes for which the information about their plane parameters and vertices is available in a global coordinate system.
To establish a link between the LoD-2 models and the 6-DoF of the MSS, this thesis employs LiDAR sensors, following the approach given by Bureick et al. (2019). Using such sensors, the environment can be captured in the form of 3D point clouds (Fowler, 2000). Usually, the resulting 3D data are in the local coordinate system of the scanner, specified by its corresponding manufacturer. However, in case of a defined platform for the MSS, these data are usually obtained in the coordinate system of the platform. In any case, by using the 6-DoF of the MSS, the scanned data can be transformed into the global coordinate system. Then, they can be assigned to the facades of the buildings in the LoD-2 models using different approaches. In this thesis such an assignment is done based on the minimum 3D distance of the points to the buildings. Since the pose of the MSS is unknown,

the main idea is to achieve an acceptable assignment, which in turn yields a reliable 6-DoF for the MSS. Consequently, it is possible to benefit from the additional information provided by the building models in the localization of the MSS.
Besides LoD-2 models, DTM offer another source for extracting additional information from the environment. A DTM provides elevations, enabling the estimation of the height component of the MSS during georeferencing. These digital models usually consist of a grid of cells that have 2D coordinates, each of which provides height information in a vertical datum. This datum is alternatively refereed to as a zero height surface (Smith et al., 1999) or a zero height reference (Smith and Zuber, 1998).
While digital city and terrain models serve as static environmental elements that can be useful, other MSSs in the same environment can also provide valuable data for georeferencing. In such scenarios, using the right sensors to establish robust communication between the target MSS (referred to as the *primary* MSS) and the other systems (referred to as *secondary* MSSs) is advantageous. In this work, the UWB units are employed to link the primary MSS and the secondary MSSs. A UWB is a range sensor that derives the relative distance between these units based on the two way time of flight (TW-ToF) principle applied to the received and transmitted radio frequency signals (Retscher et al., 2019).
In line with Vogel et al. (2020), the process of georeferencing reliant on object information is referred to as *information-based georeferencing*, a term also utilized in this thesis. Figure 4.1 illustrates this principle. The main goal is to locate the blue vehicle, which is assumed to be equipped with a 3D scanner (the red box on its roof). The red and green vehicles serve as secondary MSSs providing useful information for georeferencing the blue vehicle. The green areas on either side of the primary vehicle represent the measured point clouds from the 3D scanner that are transformed into the global coordinate system. The red segments on the two adjacent buildings represent the facades to which a part of the scanned data can be assigned. The remaining data points can be assigned to the ground. Furthermore, the blue lines connecting the MSSs represent a communication links that can be established between the vehicles. The main idea of establishing these connections to the surrounding infrastructures or other MSSs is to use their potential geometrical information in order to assist georeferencing of the target MSS (the blue vehicle).

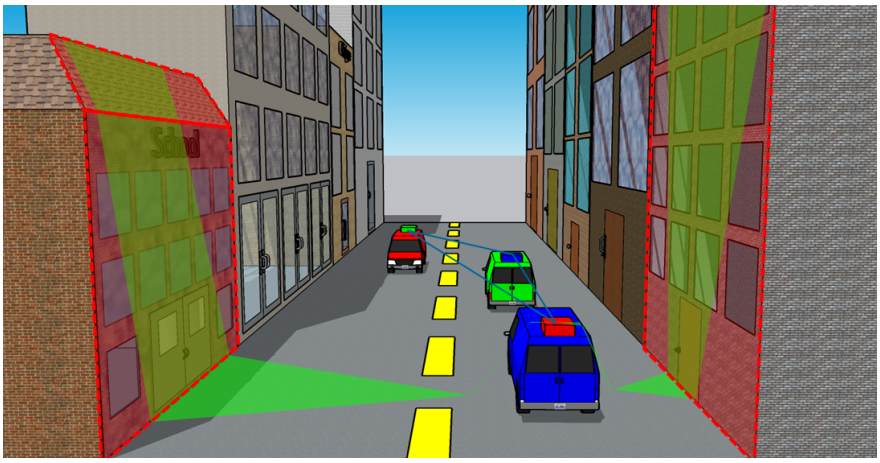

Figure 4.1: *Scheme of the information-based georeferencing.*

4.2 Methodological Setup

In this part, the methodological setup of the filters based on the core principle explained in Section 4.1 is explained. Since the workflow of information-based georeferencing in this thesis is similar to that of Bureick et al. (2019), the described setup is similar to the latter work. However, in sections

4.2.4 and 4.2.6, which concern the prediction step of the filters and the fusion of data from different sensors, two strategies are proposed that are achievements of the current work.

4.2.1 State Vector

The state vector comprises the 6-DoF of the MSS at each epoch in time k as follows:

$$\boldsymbol{\mu}_k = \begin{bmatrix} t_{x,k} & t_{y,k} & t_{z,k} & \omega_k & \phi_k & \kappa_k \end{bmatrix}^T, \tag{4.1}$$

where $t_{x,k}$, $t_{y,k}$, and $t_{z,k}$ denote the 3D positions of the MSS. In this work, these 3D positions are also termed the *translation* parameters because they signify the shifts of the MSS with respect to the global coordinate system. In addition, ω_k, ϕ_k, and κ_k represents the roll, pitch and yaw angles of the MSS with respect to the global coordinate system.

4.2.2 Assignment

As explained in Section 4.1, this works employs the LiDAR and UWB units to mitigate the limitations of the GNSS and IMU sensors. To assign the scanned data to either the building models or the DTM, it is crucial to transform these data into the same global coordinate system in which this additional information is available. In this thesis, both the building models and the DTM heights are in the same coordinate system in which the pose parameters of the MSS are determined. Consequently, the 6-DoF can be directly applied to transform the scanned data into the target coordinate system. The transformation in each epoch (k) can be executed as follows:

$$\boldsymbol{P}_{glo,k} = \boldsymbol{t}_k + \boldsymbol{R}_k \cdot \boldsymbol{P}_{loc,k}, \tag{4.2}$$

Here, $\boldsymbol{P}_{glo}$ represents the transformed scanned data in the global coordinate system, while $\boldsymbol{P}_{loc}$ are the 3D scanned data in the local coordinate system of the scanner. It is notable that in case of a defined platform coordinate system, the sensor data should first be transformed into this coordinate system before transformation into the global coordinate system. In addition, $\boldsymbol{t}$ denotes the translation vector, comprising the 3D translations. Furthermore, $\boldsymbol{R}$ is the rotation matrix, which is defined based on the 3D orientations in each epoch as follows:

$$\boldsymbol{R} = \boldsymbol{R}_\omega \cdot \boldsymbol{R}_\phi \cdot \boldsymbol{R}_\kappa. \tag{4.3}$$

In this equation $\boldsymbol{R}_\omega$, $\boldsymbol{R}_\phi$, and $\boldsymbol{R}_\kappa$ represent the rotations with respect to the main axes of the global coordinate system. In other words, if the axes of the global coordinate system are defined as X, Y, and Z, $\boldsymbol{R}_\omega$, $\boldsymbol{R}_\phi$, and $\boldsymbol{R}_\kappa$ correspond to the rotations around the X, Y, and Z axes, respectively. These rotations correspond to the Easting, Northing, and Altitude axes if the global coordinate system is universal transverse mercator (UTM). To derive these rotation matrices, the following equations apply:

$$\boldsymbol{R}_\omega = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\omega & -sin\omega \\ 0 & sin\omega & cos\omega \end{bmatrix}, \quad \boldsymbol{R}_\phi = \begin{bmatrix} cos\phi & 0 & sin\phi \\ 0 & 1 & 0 \\ -sin\phi & 0 & cos\phi \end{bmatrix}, \quad \boldsymbol{R}_\kappa = \begin{bmatrix} cos\kappa & -sin\kappa & 0 \\ sin\kappa & cos\kappa & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.4}$$

The transformed data $\boldsymbol{P}_{glo}$ at each epoch should be assigned to either the LoD-2 models or the DTM, in accordance with the principle outlined in Section 4.1. Various methods can be used to perform such an assignment; this work adopts the strategy proposed by Unger et al. (2016) and

Unger et al. (2017). This method assigns 3D scanned data to either the LoD-2 models or the DTM based on their minimum 3D distance. Figures 4.2 and 4.3 depict the applied assignment procedure. In Figure 4.2, the side view facade of a building model is represented by a gray box, and the vertical dashed line in black indicates a user-defined threshold for assignment. The 3D scanned data is also shown as circles. The main idea of the proposed procedure is to assign to the facade those scanned data whose distance to the facade is less than a defined threshold. Such assigned points are represented by green circles in the figure. On the contrary, those scanned data that have a greater distance than the threshold are either completely neglected or assigned to the cells of the DTM. The assignment to the DTM cells requires that the distance threshold is satisfied. The scanned data not assigned to the building model are marked by red circles in the figure. The mentioned threshold for the assignment is user-defined and depends on the application. In the current work and for each of the case studies, a simple grid search over a limited range of values is used to derive this threshold. To be more precise, a set of thresholds is first defined. Then, the filters are applied using these thresholds. Finally, the threshold that yielded the most reliable georeferencing solutions was used as the basis for further analysis.

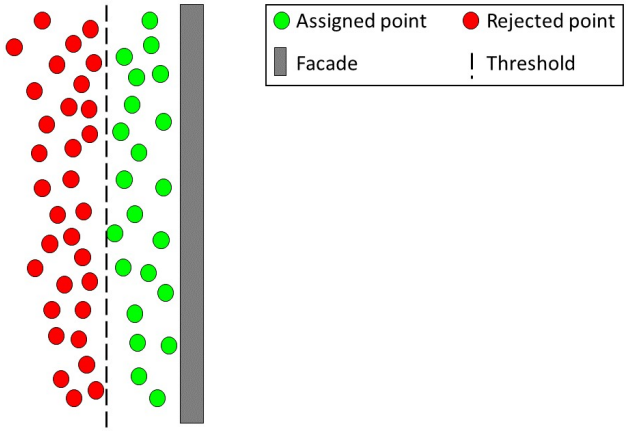

Figure 4.2: *Scheme of the scanned data assignment to the facade of a building.*

In Figure 4.3, the black grid represents a DTM with defined 2D cells, each having a specific height. The vertical plane represents a building facade to which a set of scanned data is assigned (green dots). As explained earlier, some of the scanned data cannot be assigned to a building facade because their distance to the plane is greater than a specified threshold (the dashed black line in Fig. 4.2). The red points are then checked to see if they can be assigned to a cell of the DTM or if they should be completely neglected. The procedure is similar to the one used for building planes, where a specified distance to the cells of the DTM must be maintained. In the figure, the red points represent the scanned data that should be assigned to the DTM cells shown in yellow.

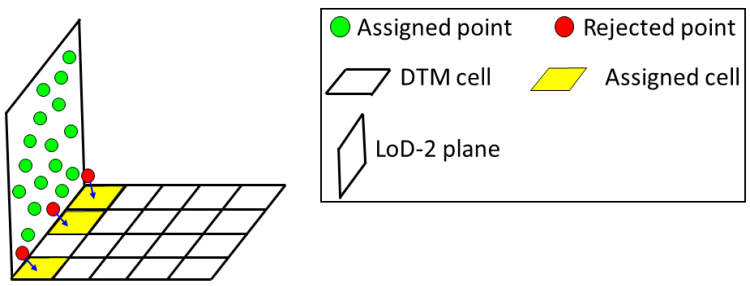

Figure 4.3: *Scheme of the scanned data assignment to the cells of a DTM.*

4.2.3 Observation Vector

The observation vector at each epoch k can be written as follows:

$$\boldsymbol{l}_k = \left[\boldsymbol{l}^{loc}_{LOD,k}; \boldsymbol{l}^{loc}_{DTM,k}; \boldsymbol{l}^{glo}_{pose,k}; \boldsymbol{l}_{dist,k}\right], \tag{4.5}$$

where $\boldsymbol{l}^{loc}_{LOD,k}$ are the scanned data in epoch k assigned to the facades of the buildings using the LoD-2 model. Furthermore, $\boldsymbol{l}^{loc}_{DTM,k}$ are the scanned data assigned to the DTM cells. In addition, $\boldsymbol{l}^{glo}_{pose,k}$ is a vector containing the 3D positions and 3D orientations of the MSS derived from the GNSS and IMU data, respectively. This vector is derived using techniques known to obtain 3D data from the raw measurements of the GNSS and IMU sensors. Note that this work does not cover the use of raw GNSS and IMU data in filters. Finally, $\boldsymbol{l}_{dist,k}$ contains the relative distance between the primary and secondary units, which corresponds to the UWB data in the current work.
In equation (4.5), the scanned data in $\boldsymbol{l}^{loc}_{LOD,k}$ and $\boldsymbol{l}^{loc}_{DTM,k}$ are in the local coordinate system of the scanner. These vectors are derived after applying the assignment procedure explained in Section 4.2.2. Depending on the acquired facades and the assigned DTM cells, these vectors may vary in size over time. At each epoch a total number of planes (E) of the building model are detected. Each of these planes is assigned a number of scanned data $\left(\boldsymbol{l}^{loc}_{LOD}\right)$, which can be written as follows:

$$\boldsymbol{l}^{loc}_{LOD,k} = \left[\boldsymbol{l}^{loc}_{LOD,1,k}; \boldsymbol{l}^{loc}_{LOD,2,k}; \ldots; \boldsymbol{l}^{loc}_{LOD,e,k}; \ldots; \boldsymbol{l}^{loc}_{LOD,E,k}\right]. \tag{4.6}$$

Equation (4.6) can be further decomposed for a better view of the 3D scanned data, e.g., for the e^{th} plane:

$$\begin{aligned} \boldsymbol{l}^{loc}_{LOD,e,k} &= \left[\boldsymbol{P}^{loc}_{e,1,k}; \boldsymbol{P}^{loc}_{e,2,k}; \ldots; \boldsymbol{P}^{loc}_{e,i,k}; \ldots; \boldsymbol{P}^{loc}_{e,N_e,k}\right] \\ \boldsymbol{P}^{loc}_{e,i,k} &= \begin{bmatrix} x^{loc}_{e,i,k} & y^{loc}_{e,i,k} & z^{loc}_{e,i,k} \end{bmatrix}^T, \end{aligned} \tag{4.7}$$

where $\boldsymbol{l}^{loc}_{LOD,e,k}$ represents the scanned points in epoch k that belong to the e^{th} plane, and $\boldsymbol{P}^{loc}_{e,i,k}$ is the i^{th} 3D point – with x, y, and z coordinates in the local coordinate system of the scanner – that lies on the e^{th} plane and N_e is the total number of assigned 3D points to plane e.
A similar representation holds for the assigned scanned data to the DTM cells $\left(\boldsymbol{l}^{loc}_{DTM,k}\right)$ as well:

$$\begin{aligned} \boldsymbol{l}^{loc}_{DTM,k} &= \left[\boldsymbol{l}^{loc}_{DTM,1,k}; \boldsymbol{l}^{loc}_{DTM,2,k}; \ldots; \boldsymbol{l}^{loc}_{DTM,c,k}; \ldots; \boldsymbol{l}^{loc}_{DTM,C,k}\right] \\ \boldsymbol{l}^{loc}_{DTM,c,k} &= \boldsymbol{P}^{loc}_{c,k} = \begin{bmatrix} x^{loc}_{c,k} & y^{loc}_{c,k} & z^{loc}_{c,k} \end{bmatrix}^T, \end{aligned} \tag{4.8}$$

where the assigned scanned points to the DTM cells in epoch k $\left(\boldsymbol{l}^{loc}_{DTM,k}\right)$ are divided into smaller groups each belonging to a specific cell. Each group contains one 3D point $\boldsymbol{l}^{loc}_{DTM,c,k}$ – with x, y, and z coordinates in the local coordinate system of the scanner – that belongs to the c^{th} cell. C shows the total number of DTM cells in epoch k to which a scanned point is assigned.
Furthermore, vector $\boldsymbol{l}^{glo}_{pose,k}$ is as follows:

$$\boldsymbol{l}^{glo}_{pose,k} = \left[\underbrace{\begin{matrix} t^{GNSS}_{x,k} & t^{GNSS}_{y,k} & t^{GNSS}_{z,k} \end{matrix}}_{\text{translations: GNSS data}} \quad \underbrace{\begin{matrix} \omega^{IMU}_{k} & \phi^{IMU}_{k} & \kappa^{IMU}_{k} \end{matrix}}_{\text{orientations: IMU data}}\right]^T, \tag{4.9}$$

where $t_{x,k}^{GNSS}$, $t_{y,k}^{GNSS}$ and $t_{z,k}^{GNSS}$ indicate the 3D positions of the MSS based on the GNSS data. Similarly, ω_k^{IMU}, ϕ_k^{IMU} and κ_k^{IMU} represent the 3D orientations of the MSS based on the IMU data.
Finally, vector $\boldsymbol{l}_{dist,k}$ is as follows:

$$\boldsymbol{l}_{dist,k} = \begin{bmatrix} l_{dist,k}^{p21} & l_{dist,k}^{p22} & \cdots & l_{dist,k}^{p2s} & \cdots & l_{dist,k}^{p2S} \end{bmatrix}^T, \tag{4.10}$$

where $l_{dist,k}^{p2s}$ is the relative distance between the primary MSS (p) and the s^{th} MSS (indicated by the superscript $p2s$). Furthermore, S in the superscript of the last entry indicates the last secondary MSS in the environment from which a connection to the primary one can be established.

4.2.4 Adaptive Kinematic Model

As explained in Section 4.2.1, this thesis employs a state vector limited to the 6-DoF of the MSS for use within filtering frameworks. As can be seen in equation (4.1), no motion-related parameters such as velocities and accelerations are included in the state vector. The reason for this is to keep the size of the state vector as small as possible in order to reduce the computation time. Including more states requires more observations, which in turn necessitates more particles to reliably estimate the states. As a result, the computation time increases and the filter becomes inefficient. However, the disadvantage of not considering the velocities in the state vector is the transition of the states from one epoch to another. In the case of considering the velocities in the state vector, a simple kinematic model such as the *constant white noise acceleration model* (Bar-Shalom et al., 2004) – which is also used in Bureick et al. (2019) – can be applied. For a state vector like the following, such a model has a general form as given by equation (4.12):

$$\boldsymbol{\mu}_k = \begin{bmatrix} t_{x,k} & t_{y,k} & t_{z,k} & \omega_k & \phi_k & \kappa_k & V_{x,k} & V_{y,k} & V_{z,k} & \Omega_{\omega,k} & \Omega_{\phi,k} & \Omega_{\kappa,k} \end{bmatrix}^T, \tag{4.11}$$

where $V_{x,k}$, $V_{y,k}$, and $V_{z,k}$ are the directional velocities corresponding to $t_{x,k}$, $t_{y,k}$ and $t_{z,k}$, respectively.. Furthermore, $\Omega_{\omega,k}$, $\Omega_{\phi,k}$, and $\Omega_{\kappa,k}$ are the angular velocities corresponding to ω_k, ϕ_k and κ_k, respectively.

$$\begin{aligned} \boldsymbol{\mu}_k &= \boldsymbol{F}_k \cdot \boldsymbol{\mu}_{k-1} + \boldsymbol{w}_k\ , \qquad \boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_k) \\ \boldsymbol{F}_k &= \begin{bmatrix} \boldsymbol{I}_{[3\times3]} & \boldsymbol{0}_{[3\times3]} & \text{diag}([\Delta\tau, \Delta\tau, \Delta\tau]) & \boldsymbol{0}_{[3\times3]} \\ \boldsymbol{0}_{[3\times3]} & \boldsymbol{I}_{[3\times3]} & \boldsymbol{0}_{[3\times3]} & \text{diag}([\Delta\tau, \Delta\tau, \Delta\tau]) \\ \boldsymbol{0}_{[3\times3]} & \boldsymbol{0}_{[3\times3]} & \boldsymbol{I}_{[3\times3]} & \boldsymbol{0}_{[3\times3]} \\ \boldsymbol{0}_{[3\times3]} & \boldsymbol{0}_{[3\times3]} & \boldsymbol{0}_{[3\times3]} & \boldsymbol{I}_{[3\times3]} \end{bmatrix}, \end{aligned} \tag{4.12}$$

where $\Delta\tau$ represents the time interval between two consecutive epochs. Also, $\text{diag}([\Delta\tau, \Delta\tau, \Delta\tau])$ denotes a matrix with $[\Delta\tau, \Delta\tau, \Delta\tau]$ on its diagonal. As can be seen, by using the directional and angular velocities, the 3D positions and orientations of the previous epoch can be propagated to the current epoch.
To compensate for the transition of states due to the absence of velocities in the state vector, the current work proposes a so-called *adaptive kinematic model*. In such a model, the directional and angular velocities are taken as control inputs that have to be estimated based on the derived states. The proposed kinematic model has the following notation:

$$
\begin{aligned}
&\boldsymbol{\mu}_k = \boldsymbol{F}_k \cdot \boldsymbol{\mu}_{k-1} + \boldsymbol{G}_k \cdot \boldsymbol{u}_k + \boldsymbol{w}_k \ , \qquad \boldsymbol{w}_k \ \sim \ \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_k) \\
&\boldsymbol{F}_k = \begin{bmatrix} \boldsymbol{I}_{[3\times3]} & \boldsymbol{0}_{[3\times3]} \\ \boldsymbol{0}_{[3\times3]} & \boldsymbol{I}_{[3\times3]} \end{bmatrix}, \qquad \boldsymbol{G}_k = \begin{bmatrix} \boldsymbol{I}_{[3\times3]} & \boldsymbol{0}_{[3\times3]} \\ \boldsymbol{0}_{[3\times3]} & \boldsymbol{I}_{[3\times3]} \end{bmatrix} \\
&\boldsymbol{u}_k = \begin{bmatrix} \hat{V}_{x,k} & \hat{V}_{y,k} & \hat{V}_{z,k} & \hat{\Omega}_{\omega,k} & \hat{\Omega}_{\phi,k} & \hat{\Omega}_{\kappa,k} \end{bmatrix}^T,
\end{aligned}
\tag{4.13}
$$

where $\hat{V}_{x,k}$, $\hat{V}_{y,k}$ and $\hat{V}_{z,k}$ are the estimated 3D directional velocities as control input. Similarly, $\hat{\Omega}_{\omega,k}$, $\hat{\Omega}_{\phi,k}$ and $\hat{\Omega}_{\kappa,k}$ are the estimated 3D angular velocities as control input.
The control input ($\boldsymbol{u}_k$) in equation (4.13) should either be provided by an external source or must be estimated. In practice, the IMU data can be used for this purpose in the prediction step. In this thesis, however, all sensor data are considered only in the update step. The reason for this is to eliminate uncertainties from the observations in the prediction step. Therefore, the current work recommends to estimate the velocities as follows:

$$
\begin{aligned}
\hat{V}_{x,k} &= \frac{\Delta t_x}{\Delta\tau} &&= \frac{t_x^* - t_x^{\tau_{k-k_b}}}{\Delta\tau} \\
\hat{V}_{y,k} &= \frac{\Delta t_y}{\Delta\tau} &&= \frac{t_y^* - t_y^{\tau_{k-k_b}}}{\Delta\tau} \\
\hat{V}_{z,k} &= \frac{\Delta t_z}{\Delta\tau} &&= \frac{t_z^* - t_z^{\tau_{k-k_b}}}{\Delta\tau} \\
\hat{\Omega}_{\omega,k} &= \frac{\Delta\omega}{\Delta\tau} &&= \frac{\omega^* - \omega^{\tau_{k-k_b}}}{\Delta\tau} \\
\hat{\Omega}_{\phi,k} &= \frac{\Delta\phi}{\Delta\tau} &&= \frac{\phi^* - \phi^{\tau_{k-k_b}}}{\Delta\tau} \\
\hat{\Omega}_{\kappa,k} &= \frac{\Delta\kappa}{\Delta\tau} &&= \frac{\kappa^* - \kappa^{\tau_{k-k_b}}}{\Delta\tau},
\end{aligned}
\tag{4.14}
$$

where, in the case of the versatile KF, t_x^*, t_y^* and κ^* correspond to t_x^{k-1}, t_y^{k-1}, κ^{k-1}, respectively. In the case of the PF-based frameworks, these terms correspond to the predicted states, namely $\overline{t}_x^k$, $\overline{t}_y^k$, $\overline{\kappa}^k$, respectively.
The main principle behind the equation (4.14) is to calculate the velocities based on the current states and the estimated states at some instant in the past. These derived states in the past are indicated by the superscript τ_{k-k_b} in the equation. Furthermore, τ_{k-k_b} is the epoch in the past from which the estimated states are to be extracted. To derive τ_{k-k_b}, the following relation applies:

$$
\tau_{k-k_b} = \begin{cases} k_1, & \text{if } k - k_b < k_1. \\ k - k_b, & \text{otherwise.} \end{cases}
\tag{4.15}
$$

In equation (4.15), k_1 is the first epoch and k_b is the number of epochs to pass before τ_{k-k_b} is changed. In other words, all terms specified by the superscript τ_{k-k_b} in the equation (4.14) are held constant until k_b number of epochs have passed. The current work specifies k_b as a design parameter that is user-defined and exclusive to the case study.

4.2.5 Observation Model

Some of the observations in the observation vector given in equation (4.5) have implicit observation models, while others have explicit observation models. The corresponding observation model for

the $\boldsymbol{l}^{loc}_{LOD,k}$ is the Hesse normal form (Bôcher, 1915) given by equation (3.17). The reason for this is that the facades in the LoD-2 models are defined by planes. The expansion of the equation (3.17) leads to the following relation:

$$\boldsymbol{n}_{x,k} \cdot \boldsymbol{X}^{glo}_{k} + \boldsymbol{n}_{y,k} \cdot \boldsymbol{Y}^{glo}_{k} + \boldsymbol{n}_{z,k} \cdot \boldsymbol{Z}^{glo}_{k} - \boldsymbol{d}_k = \boldsymbol{0}, \tag{4.16}$$

where $\mathbf{n}_k = \left[\mathbf{n}_{x,k}, \mathbf{n}_{y,k}, \mathbf{n}_{z,k}\right]$ and $\mathbf{d}_k$ are the normal vectors and distances to the origin of the detected planes at epoch k, respectively. This information can be extracted from the LoD-2 models. In addition, $\mathbf{P}_{glo,k} = \left[\mathbf{X}^{glo}_{k}, \mathbf{Y}^{glo}_{k}, \mathbf{Z}^{glo}_{k}\right]$ are the transformed scanned data to the global coordinate system derived by equation (4.2). Note that the measured scanned data are subject to uncertainties. In equation (3.17) the uncertainty of the sensor data is indicated by $\boldsymbol{v}_k$. Since the scanned data in equation (4.16) are in the global coordinate system, such an uncertainty is implicitly considered in the terms $\boldsymbol{X}^{glo}_{k}$, $\boldsymbol{Y}^{glo}_{k}$, and $\boldsymbol{Z}^{glo}_{k}$.
The observation model corresponding to the $\boldsymbol{l}^{loc}_{DTM,k}$ is the height difference between the scanned data transformed to the global coordinate system and the DTM. This observation model, which is of the explicit type, can be written as follows:

$$\boldsymbol{Z}^{glo,DTM} - \boldsymbol{Z}_{DTM} = 0, \tag{4.17}$$

where $\mathbf{Z}^{glo,DTM}$ are the global heights of the scanned data assigned to the DTM cells. Furthermore, $\mathbf{Z}_{DTM}$ are the height information extracted directly from the DTM. In addition, similar to equation (4.16), the given $\boldsymbol{Z}^{glo,DTM}$ is in the global coordinate system. Therefore, its corresponding uncertainty is implicitly included in this equation.
As indicated in equation (4.9), the post-processed GNSS and IMU data provide an estimate of the 6-DoF of the MSS. Consequently, the observation models for $\boldsymbol{l}^{glo}_{pose,k}$ follow an explicit form as follows:

$$\boldsymbol{l}^{glo}_{pose,k} = \boldsymbol{\mu}_k + \boldsymbol{\nu}^{INS}_{k}, \tag{4.18}$$

where $\boldsymbol{\nu}^{INS}_{k}$ corresponds to the accuracy of the 3D positions and 3D orientations of the MSS based on the GNSS and IMU data, respectively. As mentioned earlier in Section 4.2.3, $\boldsymbol{l}^{glo}_{pose,k}$ is derived from the raw GNSS and IMU data based on known navigation techniques. Therefore, $\boldsymbol{\nu}^{INS}_{k}$ corresponds to the accuracy of such derived solutions.
Since $\boldsymbol{l}_{dist,k}$ provides relative distances between the primary and secondary MSSs, the observation model for it is defined in terms of Euclidean distances. Therefore, the observation model for the relative distance between the primary p and the s^{th} MSS is as follows:

$$l^{p2s}_{dist,k} = \sqrt{\left(\boldsymbol{t}_k - \boldsymbol{t}^{p2s}_{k}\right)^T \left(\boldsymbol{t}_k - \boldsymbol{t}^{p2s}_{k}\right)}. \tag{4.19}$$

In this explicit observation model, $\boldsymbol{t}^{p2s}_{k}$ is a vector containing the 3D positions of the secondary MSSs in the global coordinate system. In the current work, this vector is always assumed to be known and its values are considered to be deterministic within the filters. The reason for this is that considering such information as stochastic steers the filters into the domain of cooperative positioning. However, this aspect, in the sense of forming a dynamic network to localize all the MSSs involved is beyond the scope of this thesis. The main purpose is to localize only the primary vehicle. Therefore, although the secondary MSSs are dynamic, they are treated as static objects with deterministic information at any instant of time.

4.2.6 Sensor Fusion

In the case of having more than one sensor for georeferencing a MSS, it is essential to properly fuse the observations. Within KF-based frameworks, sensor fusion is inherently accomplished through the Kalman gain, which, in the versatile IEKF model, follows equation (2.49). In the general framework of PF, the fusion of multiple sensor data is considered according to equation (2.68). Applying this equation to the observation vector given by equation (4.5) leads to the following relation:

$$\begin{aligned}\boldsymbol{\omega}_k^{[1:S]} &= \boldsymbol{\eta}\ \boldsymbol{p}\left(\boldsymbol{l}_k \mid \boldsymbol{x}_k\right) \\ &= \boldsymbol{\eta}\ \underbrace{\boldsymbol{p}\left(\boldsymbol{l}_{LOD,k}^{loc} \mid \boldsymbol{x}_k\right)\ \boldsymbol{p}\left(\boldsymbol{l}_{DTM,k}^{loc} \mid \boldsymbol{x}_k\right)}_{\text{scanner}}\ \underbrace{\boldsymbol{p}\left(\boldsymbol{l}_{pose,k}^{glo} \mid \boldsymbol{x}_k\right)}_{\text{GNSS and IMU}}\ \underbrace{\boldsymbol{p}\left(\boldsymbol{l}_{dist,k} \mid \boldsymbol{x}_k\right)}_{\text{UWB}}.\end{aligned} \tag{4.20}$$

As previously discussed in Section 3.2, utilizing equation (2.68) – corresponding to the "scanner" part of equation (4.20) – may yield unreliable estimates when dealing with a large number of observations. To address this issue, equation (3.11) is proposed, aiming to reduce the dimensionality of such a high number of observations. Such a procedure, as verified by a simple numerical example in Chapter 3, leads to reliable results when the filter employs only one type of sensor. However, in the scope of this work, the application of the proposed strategy to equation (4.20) results in filter instability. This instability occurs because the approach down-weights the scanner data relative to data from other sensors. This leads to downgrading the additional information gained from the environment. Therefore, in the context of PF-based methods developed in this work, a new strategy for fusing different sensor data has been proposed, inspired by the *Kalman fuser with scalar weights* from Deng et al. (2013).
According to Deng et al. (2013), when dealing with a total number of L sensors, the state vector $\boldsymbol{\mu}_{f,k}$ is determined as follows:

$$\boldsymbol{\mu}_{f,k} = \sum_{i=1}^{L} w_i \cdot \boldsymbol{\mu}_{i,k}, \tag{4.21}$$

where $\boldsymbol{\mu}_{i,k}$ represents the estimated state vector from each sensor i, and w_i is the corresponding weight assigned to it. Furthermore, the corresponding VCM of the estimated fused state vector $\left(\boldsymbol{\mu}_{f,k}\right)$ is calculated as follows:

$$\boldsymbol{\Sigma}_{f.k} = \sum_{i=1}^{L}\sum_{j=1}^{L} w_i \cdot w_j \cdot \boldsymbol{\Sigma}_{ij,k}, \tag{4.22}$$

where $\boldsymbol{\Sigma}_{ij,k}$ is the estimated VCM of the derived state vector by each sensor $\left(\boldsymbol{\mu}_{i,k}\right)$.
In the current work, the strategy outlined by Deng et al. (2013) is adapted to equation (4.20) as follows:

$$\begin{aligned}\omega_k^{[s]} &= p\left(\boldsymbol{l}_k \mid \boldsymbol{x}_k^{[s]}\right) \\ &= p\left(r_{f,k}^{[s]}\right) \cdot p\left(r_{o,k}^{[s]}\right),\end{aligned} \tag{4.23}$$

where $r_{f,k}^{[s]}$ and $r_{o,k}^{[s]}$ represent the so-called *fused residual* and *split residual*, respectively. Based

on the equations (3.1) and (3.4), using each particle $\left(\boldsymbol{x}_k^{[s]}\right)$ in the observation models leads to a set of residual vectors $\left(\hat{\boldsymbol{v}}_k^{[s]} \text{ and } \hat{\boldsymbol{r}}_k^{[s]}\right)$. The *fused residual* is a residual derived by combining those residuals that are of the same type. Conversely, the *split residual* is derived based on those residuals that cannot be combined due to their different unit. In the current work, the residuals derived from the scanner, GNSS and UWB data are all distance measures that have the same unit (e.g. meters). Therefore, they can be combined. On the contrary, the residuals based on the IMU data are of the angular type and therefore cannot be combined with those from the scanner, GNSS and UWB data.
The fused residual $\left(r_{f,k}^{[s]}\right)$ is derived as follows:

$$\begin{aligned} r_{f,k}^{[s]} &= \sum_{i=1}^{L} w_i \cdot r_{i,k}^{[s]} \\ &= \underbrace{w_1 \cdot r_{1,k}^{[s]}}_{\text{scanner}} + \underbrace{w_2 \cdot r_{2,k}^{[s]}}_{\text{GNSS}} + \underbrace{w_3 \cdot r_{3,k}^{[s]} + \ldots + w_L \cdot r_{L,k}^{[s]}}_{\text{UWB}}, \end{aligned} \tag{4.24}$$

where $r_{i,k}^{[s]}$ corresponds to the residuals obtained from the scanner, GNSS and UWB data along with their corresponding weights (w_i). The following explains how to obtain the fused residual in equation (4.24).
According to the observation vector given by equation (4.5), the total number of observations is as follows:

$$n_k = \underbrace{n_{LOD,k} + n_{DTM,k}}_{\text{scanner}} + \underbrace{n_{pose,k}}_{\text{GNSS}} + \underbrace{n_{dist,k}}_{\text{UWB}}, \tag{4.25}$$

where $n_{LOD,k}$ and $n_{DTM,k}$ denote the number of scanned data points in the $\boldsymbol{l}_{LOD,k}^{loc}$ and $\boldsymbol{l}_{DTM,k}^{loc}$ vectors, respectively. Additionally, $n_{pose,k}$ is the number of data in the $\boldsymbol{l}_{pose,k}^{glo}$ vector and $n_{dist,k}$ is the number of data in the $\boldsymbol{l}_{dist,k}$ vector. For the likelihood estimation in this work, only the LoD-2 models are considered ($\boldsymbol{l}_{LOD,k}^{loc}$). This is done to mitigate additional uncertainties introduced by the DTM. In other words, both the LoD-2 models and the DTM are subject to uncertainties. In the context of the present framework, which treats these two models as deterministic, any uncertainty inherently affects the filter solutions. Consequently, it is suggested to neglect the DTM in the likelihood estimation.
To estimate $r_{1,k}^{[s]}$ in equation (4.24) the following holds:

$$\begin{aligned} r_{1,k}^{[s]} &= \frac{1}{n_{LOD,k}} \cdot \tilde{r}_k^{[s]} \\ &= \frac{1}{q \cdot n_{LOD,k}} \cdot \sum_{j=1}^{q} \left| \hat{r}_{k,j}^{[s]} \right|, \end{aligned} \tag{4.26}$$

where $\tilde{r}_k^{[s]}$ should be calculated based on equation (3.10), which is already expanded to the second line of the above equation. $\hat{\boldsymbol{r}}_k^{[s]}$ is derived by applying equation (3.4) to the observation model given by equation (4.16) as follows:

$$\hat{\boldsymbol{r}}_k^{[s]} = \boldsymbol{n}_{x,k} \cdot \boldsymbol{X}_k^{glo,[s]} + \boldsymbol{n}_{y,k} \cdot \boldsymbol{Y}_k^{glo,[s]} + \boldsymbol{n}_{z,k} \cdot \boldsymbol{Z}_k^{glo,[s]} - \boldsymbol{d}_k, \tag{4.27}$$

where the superscript $[s]$ indicates that the particle $\boldsymbol{x}_k^{[s]}$ is considered as a possible solution to the given state vector in equation (4.1).
The factor $\frac{1}{n_{LOD,k}}$ in equation (4.26) gives a weight to the estimated $\tilde{r}_k^{[s]}$. This gives a higher weight to those particles that lead to a higher number of points assigned to the LoD-2 models.
To derive $r_{2,k}^{[s]}$, the following formulation holds:

$$\begin{aligned} r_{2,k}^{[s]} &= \bar{v}_{GNSS,k}^{[s]} \\ &= \frac{1}{3} \cdot \sum_{j=1}^{3} \left| \hat{v}_{pose,k,j}^{[s]} \right|, \end{aligned} \tag{4.28}$$

where $\bar{v}_{GNSS,k}^{[s]}$ is the average of the resulting absolute of $\hat{\boldsymbol{v}}_{\boldsymbol{pose,k}}^{[\boldsymbol{s}]}$ for the GNSS data. The $\hat{\boldsymbol{v}}_{\boldsymbol{pose,k}}^{[\boldsymbol{s}]}$ is derived by applying the equation (3.1) on the corresponding observation model to the GNSS and IMU data as follows:

$$\hat{\boldsymbol{v}}_{\boldsymbol{pose,k}}^{[\boldsymbol{s}]} = \boldsymbol{l}_{pose,k}^{glo} - \boldsymbol{x}_k^{[s]} - \boldsymbol{\nu}_k^{INS}. \tag{4.29}$$

Note that according to the vector $\boldsymbol{l}_{pose,k}^{glo}$, the first three entries indicate the GNSS data from which $\bar{v}_{GNSS,k}^{[s]}$ is calculated.
Similar to the scanner and GNSS data, to derive $r_{3,k}^{[s]}$ to $r_{L,k}^{[s]}$, the equation (3.1) should be applied to the corresponding observation model on the UWB data (equation (4.19)) as follows:

$$\hat{v}_{p2s,k}^{[s]} = l_{dist,k}^{p2s} - \sqrt{\left(\boldsymbol{t}_k - \boldsymbol{t}_k^{p2s}\right)^T \left(\boldsymbol{t}_k - \boldsymbol{t}_k^{p2s}\right)}. \tag{4.30}$$

Applying equation (4.30) by considering all the secondary MSSs yields $\hat{\boldsymbol{v}}_{\boldsymbol{dist,k}}^{[\boldsymbol{s}]}$:

$$\hat{\boldsymbol{v}}_{\boldsymbol{dist,k}}^{[\boldsymbol{s}]} = \begin{bmatrix} \hat{v}_{p21,k}^{[s]} & \hat{v}_{p22,k}^{[s]} & \dots & \hat{v}_{p2s,k}^{[s]} & \dots & \hat{v}_{p2S,k}^{[s]} \end{bmatrix}^T. \tag{4.31}$$

The $r_{3,k}^{[s]}$ to $r_{L,k}^{[s]}$ values in equation (4.24) correspond directly to the absolute of the entries of the $\hat{\boldsymbol{v}}_{\boldsymbol{dist,k}}^{[\boldsymbol{s}]}$:

$$\begin{bmatrix} r_{3,k}^{[s]} & \dots & r_{L,k}^{[s]} \end{bmatrix}^T = \left| \hat{\boldsymbol{v}}_{\boldsymbol{dist,k}}^{[\boldsymbol{s}]} \right|. \tag{4.32}$$

Inspired by equation (4.22), the uncertainty of the fused residual is calculated as follows:

$$\begin{aligned} \sigma_{r_f}^2 &= \sum_{i=1}^{L} w_i^2 \cdot \sigma_{r_i}^2 \\ &= \underbrace{w_1^2 \cdot \sigma_{r_1}^2}_{\text{scanner}} + \underbrace{w_2^2 \cdot \sigma_{r_2}^2}_{\text{GNSS}} + \underbrace{w_3^2 \cdot \sigma_{r_3}^2 + \ldots + w_L^2 \cdot \sigma_{r_L}^2}_{\text{UWB}}, \end{aligned} \tag{4.33}$$

where $\sigma_{r_i}^2$ is the corresponding variance for $r_{i,k}^{[s]}$. Note that $\sigma_{r_i}^2$ is similar for all $r_{i,k}^{[s]}$ estimated by different particles. As mentioned earlier, the $r_{i,k}^{[s]}$ values are derived based on one of the equations

(4.27), (4.29), or (4.30). $\sigma^2_{r_i}$ indicates the uncertainty of these observation models, from which the importance weight of the particles is derived. Except for $\sigma^2_{r_1}$, the other variance values can be set directly to the accuracy of the corresponding sensors. In other words, for the explicit observation models given by equations (4.18) and (4.19), those particles that lead to a mean value in the range of the sensor noise can be proper solutions of the state vector and should therefore be given a higher weight. However, for the implicit observation model given by equation (4.16) such a claim does not hold. The reason is the uncertainty of the LoD-2 model, which should be taken into account. However, there is no uncertainty information about the plane parameters in the LoD-2 models. Consequently, $\sigma^2_{r_1}$ is claimed to be a design parameter that should be specified based on the application. It is suggested to design this variance based on the accuracy of the 3D scanner and potential errors of the LoD-2 model in the area of study, e.g. due to generalization.
For the sensor fusion strategy proposed in the current work, the corresponding weights of the sensors (w_i in equations (4.24) and (4.33)) are considered equal $\left(w_i = \frac{1}{L}\right)$. Since the accuracy of the sensors is taken into account when estimating $\sigma^2_{r_f}$, unreliable sensor data is taken into account when deriving the importance weight of the particles. The reason for this is to avoid down-weighting the impact of each sensor data on the estimated fused residual. This can happen when the number of observations from one sensor is significantly higher than the others. In this case, if w_i is not considered equal in the equations (4.24) and (4.33), the influence of the sensor with fewer observations will disappear over time. Depending on the application, this may result in missing informative observations. If the full correlation structure between the data of a sensor is available, the effective observations can be extracted. In such a case, the weights can be defined based on such observations. However, in the current work, since the correlation structure of the observations is not available, the weights are assumed to be equal. Nevertheless, the proposed PF-based frameworks can deal with any other weights if they are properly derived. However, considering various weights is beyond the scope of the current work, which requires further investigation in the future.
In the current work, the split residual holds only for the IMU data. Since the last three entries of the derived $\hat{\boldsymbol{v}}^{[s]}_{\boldsymbol{pose,k}}$ in equation (4.29) correspond to the IMU data, the split residual $r^{[s]}_{o,k}$ is calculated as follows:

$$\begin{aligned} r^{[s]}_{o,k} &= \bar{v}^{[s]}_{IMU,k} \\ &= \frac{1}{3} \cdot \sum_{j=4}^{6} \left| \hat{v}^{[s]}_{pose,k,j} \right|, \end{aligned} \tag{4.34}$$

where $\bar{v}^{[s]}_{IMU,k}$ is the mean of the absolute value of the last three entries of the vector $\hat{\boldsymbol{v}}^{[s]}_{\boldsymbol{pose,k}}$. Similar to $r^{[s]}_{2,k}$ for $r^{[s]}_{L,k}$, the corresponding uncertainty for $r^{[s]}_{o,k}$ (σ_{r_o}) should be set to the accuracy of the IMU sensor.
The current work claims that the fused and split residuals belong to the normal distributions given by $p\left(r^{[s]}_{f,k}\right) \sim \mathcal{N}\left(r^{[s]}_{f,k};\, 0,\, \sigma_{r_f}\right)$ and $p\left(r^{[s]}_{o,k}\right) \sim \mathcal{N}\left(r^{[s]}_{o,k};\, 0,\, \sigma_{r_o}\right)$, respectively. The reason for such a claim in the case of the explicit observation models is the normal distribution of the GNSS, IMU, and UWB data. For the implicit observation model given by equation (4.16), the uncertainty of the LoD-2 models may violate the normally distributed implicit residuals $\hat{\boldsymbol{r}}^{[s]}_k$. However, as mentioned earlier, the approximate errors from the LoD-2 models can be taken into account in the $\sigma^2_{r_1}$. Therefore, the distribution of $\hat{\boldsymbol{r}}^{[s]}_k$ can be considered Gaussian. As a result, since the resulting residuals derived from all sensors are normally distributed, it can be claimed that the fused residual also follows a Gaussian distribution.
In Algorithm 11 and Algorithm 12, the pseudo-codes of R-PFI and R-EKPFI for MSS georeferencing are provided. They illustrate the methodological setup discussed in Algorithms 9 and 10.

Algorithm 11: Pseudo-code of the R-PFI for information-based georeferencing.

1 $\left[\boldsymbol{\chi}_k,\ \boldsymbol{\mu}_k,\ \boldsymbol{\Sigma}_k\right] = \textbf{R-PFI}\left(\boldsymbol{\chi}_{k-1},\ \boldsymbol{l}_k,\ \boldsymbol{u}_k\right)$

2 $\overline{\boldsymbol{\chi}}_k = \boldsymbol{\chi}_k = \emptyset$

3 **for** $s = 1 : S$ **do**

Prediction step

4 $\boldsymbol{\mu}_k = \boldsymbol{F}_k \cdot \boldsymbol{\mu}_{k-1} + \boldsymbol{G}_k \cdot \boldsymbol{u}_k + \boldsymbol{w}_k\ , \qquad \boldsymbol{w}_k\ \sim\ \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_k)$ `equation (4.13)`

Assignment step

5 $\boldsymbol{P}_{glo,k} = \boldsymbol{t}_k + \boldsymbol{R}_k \cdot \boldsymbol{P}_{loc,k}$ `equation (4.2)`

Update step

6 For the scanner data:

7 $\hat{\boldsymbol{r}}_k^{[s]} = \boldsymbol{n}_{x,k} \cdot \boldsymbol{X}_k^{glo} + \boldsymbol{n}_{y,k} \cdot \boldsymbol{Y}_k^{glo} + \boldsymbol{n}_{z,k} \cdot \boldsymbol{Z}_k^{glo} - \boldsymbol{d}_k$ `equation (4.27)`

8 Applying the IQR method based on lines 6 to 9 of Algorithm 9

9 $r_{1,k}^{[s]} = \dfrac{1}{q \cdot n_{LOD,k}} \cdot \sum_{j=1}^{j=q} \left|\hat{r}_{k,j}^{[s]}\right|$ `equation (4.26)`

10 For the GNSS data:

11 $\hat{\boldsymbol{v}}_{\boldsymbol{pose,k}}^{\boldsymbol{[s]}} = \boldsymbol{l}_{pose,k}^{glo} - \boldsymbol{x}_k^{[s]} - \boldsymbol{\nu}_k^{INS}$ `equation (4.29)`

12 $r_{2,k}^{[s]} = \dfrac{1}{3} \cdot \sum_{j=1}^{j=3} \left|\hat{v}_{pose,k,j}^{[s]}\right|$ `equation (4.28)`

13 For the UWB data:

14 $\hat{v}_{p2s,k}^{[s]} = l_{dist,k}^{p2s} - \sqrt{\left(\boldsymbol{t}_k - \boldsymbol{t}_k^{p2s}\right)^T \left(\boldsymbol{t}_k - \boldsymbol{t}_k^{p2s}\right)}$ `equation (4.30)`

15 $\left[r_{3,k}^{[s]} \quad \dots \quad r_{L,k}^{[s]}\right]^T = \left|\hat{\boldsymbol{v}}_{\boldsymbol{dist,k}}^{\boldsymbol{[s]}}\right|$ `equation (4.32)`

16 $r_{f,k}^{[s]} = w_1 \cdot r_{1,k}^{[s]} + w_2 \cdot r_{2,k}^{[s]} + w_3 \cdot r_{3,k}^{[s]} + \ldots + w_L \cdot r_{L,k}^{[s]}$ `equation (4.24)`

17 $\sigma_{r_f}^2 = w_1^2 \cdot \sigma_{r_1}^2 + w_2^2 \cdot \sigma_{r_2}^2 + w_3^2 \cdot \sigma_{r_3}^2 + \ldots + w_L^2 \cdot \sigma_{r_L}^2$ `equation (4.33)`

18 For the IMU data:

19 $r_{o,k}^{[s]} = \dfrac{1}{3} \cdot \sum_{j=4}^{j=6} \left|\hat{v}_{pose,k,j}^{[s]}\right|$ `equation (4.34)`

20 $\omega_k^{[s]} = p\left(r_{f,k}^{[s]}\right) \cdot p\left(r_{o,k}^{[s]}\right)$ `equation (4.23)`

21 add $\boldsymbol{x}_k^{[s]}$ to $\overline{\boldsymbol{\chi}}_k$

22 **end**

(Re-) Sampling step

23 Lines 14 to 17 of Algorithm 9

State estimation step

24 Lines 18 to 19 of Algorithm 9

25 return $\boldsymbol{\chi}_k$ and $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$

Algorithm 12: Pseudo-code of the R-EKPFI for information-based georeferencing.

1 $\left[\boldsymbol{\chi}_k,\ \boldsymbol{\mu}_k,\ \boldsymbol{\Sigma}_k\right] = \textbf{R-EKPFI}\left(\boldsymbol{\chi}_{k-1},\ \boldsymbol{l}_k,\ \boldsymbol{u}_k\right)$

2 $\overline{\boldsymbol{\chi}}_k = \boldsymbol{\chi}_k = \emptyset$

3 **for** $s = 1 : S$ **do**

Prediction step

4 $\boldsymbol{\mu}_k = \boldsymbol{F}_k \cdot \boldsymbol{\mu}_{k-1} + \boldsymbol{G}_k \cdot \boldsymbol{u}_k + \boldsymbol{w}_k\ , \qquad \boldsymbol{w}_k\ \sim\ \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_k)$ `equation (4.13)`

Assignment step

5 $\boldsymbol{P}_{glo,k} = \boldsymbol{t}_k + \boldsymbol{R}_k \cdot \boldsymbol{P}_{loc,k}$ `equation (4.2)`

Modification step

6 Lines 5 to 11 of Algorithm 10

Update step

7 For the scanner data:

8 $\hat{\boldsymbol{r}}_k^{[s]} = \boldsymbol{n}_{x,k} \cdot \boldsymbol{X}_k^{glo} + \boldsymbol{n}_{y,k} \cdot \boldsymbol{Y}_k^{glo} + \boldsymbol{n}_{z,k} \cdot \boldsymbol{Z}_k^{glo} - \boldsymbol{d}_k$ `equation (4.27)`

9 Applying the IQR method based on lines 6 to 9 of Algorithm 9

10 $r_{1,k}^{[s]} = \dfrac{1}{q \cdot n_{LOD,k}} \cdot \sum_{j=1}^{j=q} \hat{r}_{k,j}^{[s]}$ `equation (4.26)`

11 For the GNSS data:

12 $\hat{\boldsymbol{v}}_{\boldsymbol{pose,k}}^{\boldsymbol{[s]}} = \boldsymbol{l}_{pose,k}^{glo} - \boldsymbol{x}_k^{[s]} - \boldsymbol{\nu}_k^{INS}$ `equation (4.29)`

13 $r_{2,k}^{[s]} = \dfrac{1}{3} \cdot \sum_{j=1}^{j=3} \left|\hat{v}_{pose,k,j}^{[s]}\right|$ `equation (4.28)`

14 For the UWB data:

15 $\hat{v}_{p2s,k}^{[s]} = l_{dist,k}^{p2s} - \sqrt{\left(\boldsymbol{t}_k - \boldsymbol{t}_k^{p2s}\right)^T \left(\boldsymbol{t}_k - \boldsymbol{t}_k^{p2s}\right)}$ `equation (4.30)`

16 $\left[r_{3,k}^{[s]} \quad \ldots \quad r_{L,k}^{[s]}\right]^T = \left|\hat{\boldsymbol{v}}_{\boldsymbol{dist,k}}^{\boldsymbol{[s]}}\right|$ `equation (4.32)`

17 $r_{f,k}^{[s]} = w_1 \cdot r_{1,k}^{[s]} + w_2 \cdot r_{2,k}^{[s]} + w_3 \cdot r_{3,k}^{[s]} + \ldots + w_L \cdot r_{L,k}^{[s]}$ `equation (4.24)`

18 $\sigma_{r_f}^2 = w_1^2 \cdot \sigma_{r_1}^2 + w_2^2 \cdot \sigma_{r_2}^2 + w_3^2 \cdot \sigma_{r_3}^2 + \ldots + w_L^2 \cdot \sigma_{r_L}^2$ `equation (4.33)`

19 For the IMU data:

20 $r_{o,k}^{[s]} = \dfrac{1}{3} \cdot \sum_{j=4}^{j=6} \left|\hat{v}_{pose,k,j}^{[s]}\right|$ `equation (4.34)`

21 $\omega_k^{[s]} = p\left(r_{f,k}^{[s]}\right) \cdot p\left(r_{o,k}^{[s]}\right)$ `equation (4.23)`

22 add $\boldsymbol{x}_k^{[s]}$ to $\overline{\boldsymbol{\chi}}_k$

23 **end**

(Re-) Sampling step

24 Lines 21 to 24 of Algorithm 10

State estimation step

25 Lines 25 to 26 of Algorithm 10

26 return $\boldsymbol{\chi}_k$ and $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$

4.3 Case Studies

To investigate the explained methodological setup explained in Section 4.2, two case studies are considered. One is a simulated environment that allows a controlled setup to realize the functionality of the proposed filters. The other case is a real-world application that includes the unavoidable complications of the environment and sensor setups. The transition from the simulated environment to the real-world application requires careful consideration of such complications. Real-world applications introduce challenges related to data completeness and time synchronization. Unlike simulations with perfectly synchronized, complete sensor data, real-world scenarios may have missing data points or data gaps due to varying sensor sampling rates. Additionally, ensuring accurate time alignment between different sensors becomes crucial. Another challenge in the transition from the simulated environment to the real-world application is the kinematic behavior of the MSS. While in the simulated environment, the speed of the vehicle can be considered to be constant, in the real world application, the vehicle may take on different speeds over time. Therefore, even if the kinematic model can be considered the same over time, the selection of the optimized VCM – $\boldsymbol{R}_k$ in equation (4.38) – becomes challenging. These challenges, along with the unexpected situations that may arise in real-world case studies, require that the developed methodologies be examined not only in the simulated case, but also in the real-world application.
The simulated and real-world case studies deal with the localization of a car in an urban environment with insignificant height changes. Furthermore, in both cases only the heading of the car – corresponding to κ_k – and the 2D positions change over time. Therefore, the current work proposes to reduce the given state vector in equation (4.1) to only three states as follows:

$$\boldsymbol{\mu}_k = \begin{bmatrix} t_{x,k} & t_{y,k} & \kappa_k \end{bmatrix}^T \tag{4.35}$$

For the states $t_{z,k}$, ω_k and ϕ_k constant values over time can be considered. In the current work, $t_{z,k}$ is set to a constant height in both the simulated and real environment. To do this, the average height extracted from the DTM is added to the height of the vehicle. Furthermore, in the simulated case ω_k and ϕ_k are set to zero. These values in the real environment are extracted from the IMU data in the first epoch. The main idea is to reduce the size of the state vector as much as possible. By doing so, the uncertainties of the estimated states can be reduced by having fewer degrees of freedom. To initialize $\boldsymbol{\mu}_k$ in both the simulated environment and the real case, the GNSS and IMU data are used. The reason for this is to have a good initialization for the investigated case studies. Unlike the numerical example in Chapter 3, it is beyond the scope of this chapter to consider incorrect initialization for the filters.
Due to the consideration of deterministic values for $t_{z,k}$, ω_k and ϕ_k and in case of the simulated environment, the given observation by equation (4.5) changes to the following vector:

$$\begin{aligned} \boldsymbol{l}_k &= \begin{bmatrix} \boldsymbol{l}^{loc}_{LOD,k}; \boldsymbol{l}^{glo}_{pose,k}; \boldsymbol{l}_{dist,k} \end{bmatrix} \\ \boldsymbol{l}^{glo}_{pose,k} &= \begin{bmatrix} t^{GNSS}_{x,k} & t^{GNSS}_{y,k} & \kappa^{IMU}_k \end{bmatrix}^T \end{aligned} \tag{4.36}$$

Investigations on the derived 3D orientations based on the IMU data in the real environment showed an unreliability of the resulting κ_k values over the epochs. Therefore, the IMU data are only considered in the first epoch. This means that these data are only used to initialize κ_k and to set ω_k and ϕ_k to constant values for the whole trajectory. Consequently, the following observation vector applies in this case:

$$\begin{aligned} \boldsymbol{l}_k &= \left[\boldsymbol{l}^{loc}_{LOD,k}; \boldsymbol{l}^{glo}_{pose,k}; \boldsymbol{l}_{dist,k}\right] \\ \boldsymbol{l}^{glo}_{pose,k} &= \begin{bmatrix} t^{GNSS}_{x,k} & t^{GNSS}_{y,k} \end{bmatrix}^T \end{aligned} \tag{4.37}$$

Furthermore, the given kinematic model by equation (4.13) changes to the following equation:

$$\begin{aligned} &\boldsymbol{\mu}_k = \boldsymbol{F}_k \cdot \boldsymbol{\mu}_{k-1} + \boldsymbol{G}_k \cdot \boldsymbol{u}_k + \boldsymbol{w}_k \ , \qquad \boldsymbol{w}_k \ \sim \ \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_k) \\ &\boldsymbol{F}_k = \boldsymbol{I}_{[3\times3]}, \qquad \boldsymbol{G}_k = \boldsymbol{I}_{[3\times3]}, \qquad \boldsymbol{u}_k = \begin{bmatrix} \hat{V}_{x,k} & \hat{V}_{y,k} & \hat{\Omega}_{\kappa,k} \end{bmatrix}^T \end{aligned} \tag{4.38}$$

In addition, for sensor fusion and in the case of the simulated environment, the equations (4.28) and (4.34) change as follows:

$$\begin{aligned} r^{[s]}_{2,k} &= \bar{v}^{[s]}_{GNSS,k} \\ &= \sum_{j=1}^{j=2} \left| \hat{v}^{[s]}_{pose,k,j} \right| \end{aligned} \tag{4.39}$$

$$r^{[s]}_{o,k} = \left| \hat{v}^{[s]}_{pose,k,3} \right| \tag{4.40}$$

In the real case, since IMU data are not used as observations, equation (4.40) is removed for the PF-based algorithms.
Furthermore, only the R-PFI and R-EKPFI frameworks from Chapter 3 are considered for the following case studies. The reason for this – as also shown in Chapter 3 – is the better performance of these two filters compared to the PFI. Moreover, same as in the plane estimation example, the versatile IEKF is also considered in this chapter to compare its results with those of the PF-based frameworks.
The target global coordinate system in both case studies is UTM and the reference frame is ETRS89, in which both the LoD-2 model and the DTM are defined. These two models are open source and can be accessed using Landesamt für Geoinformation und Landesvermessung Niedersachsen (LGLN), Landesvermessung und Geobasisinformation (2023) and Landeshauptstadt Hannover, FB Planen und Stadtentwicklung, Bereich Geoinformation (2023), respectively.

4.3.1 Simulated Environment

The simulated case study in the current work consists of three cars following each other in an urban environment modeled by the LoD-2 and DTM. A 3D overview of this environment is given in Figure 4.4.

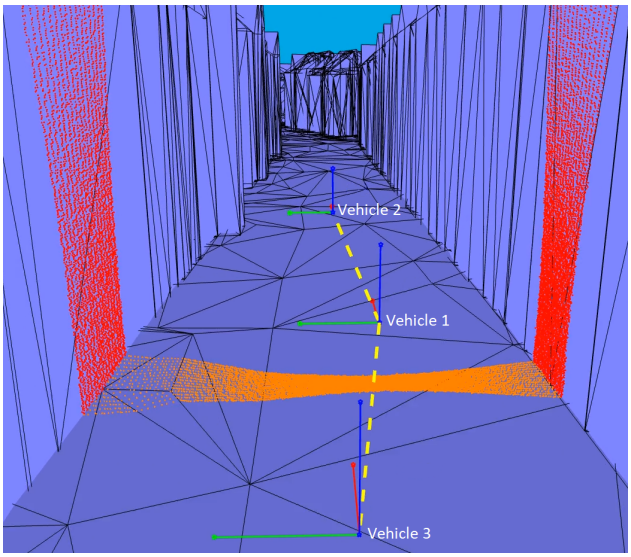

Figure 4.4: *3D overview of the simulated environment. The body frame of the vehicles are depicted by red (along the x-axis), green (along the y-axis) and blue (along the z-axis) arrows.*

The main purpose is to georeference vehicle 1 (V1) – as the primary MSS – which has connections to vehicle 2 (V2) and vehicle 3 (V3) as secondary MSSs. In this environment, LoD-2 and DTM are real models of the environment used to simulate the sensor data. Moreover, it is also assumed that V1 is equipped with a 3D scanner, a GNSS, an IMU and an UWB. V2 and V3 are assumed to have only GNSS and UWB. As shown by the red dots in Figure 4.4, the 3D scanner is assumed to be installed vertically on V1. This allows more vertical information to be extracted from the surrounding facades. Furthermore, the yellow dotted lines represent the connections that is established between V1 to V2 and V3 by means of the UWB sensors. In Table 4.1 the accuracy and sampling rate (f_s) of the sensors is given, on the basis of which their data is simulated.

Table 4.1: *Accuracy and sampling rate of the sensors in the simulated environment.*

	σ	f_s $[Hz]$
scanner	0.02 [m]	10
GNSS	0.5 [m]	10
UWB	0.5 [m]	10
IMU	0.2 [°]	10

The simulated data of the 3D scanner represents the data of the Velodyne VLP-16 (VLP-16) (Velodyne LiDAR, Inc., 2019). To simulate these data, after triangulation of the LoD-2 models and the DTM, the *ray-triangle intersection* method is used, which is given, e.g. by Curless (2020). The frequency given for this scanner in Table 4.1 indicates how many times its lasers rotate per second to scan the environment. In practice, each scanned point has its own time stamp. However, these data are considered as segments at each epoch of the filters based on this specified frequency. To be more concise, the time stamp of the first scanned point in each segment and the time stamp of the first scanned point in the subsequent segment results in a frequency rate of 10 Hz. Figure 4.5 shows a top view of the simulated environment. The red line in this figure represents the true trajectory of V1 along with its start and end points. This trajectory has a length of approximately 1 kilometer and it is derived based on the GNSS and IMU data of the measurement campaign conducted by Axmann et al. (2023). However, it should be noted that these data are not directly taken as the true pose of V1. This is due to the noisy observations of the GNSS data and the unreliability of the heading angles obtained from the IMU. Therefore, these data are adequately

pre-processed to represent a reasonable trajectory for V1. The resulting trajectory is considered the true trajectory for further analysis. The full trajectory consists of 3670 epochs, of which the first 1400 epochs are selected for the MC simulations. The reason for not using the full trajectory is to avoid high computational cost. Besides, the trajectory selected for this purpose consists of both straight parts and curves of the routes. Furthermore, it is used to obtain the design parameters for the filters. Therefore, it is claimed that the functionality of the filters can be confirmed by using it. In total, the results of 100 MC simulations are given in Section 4.4.1.

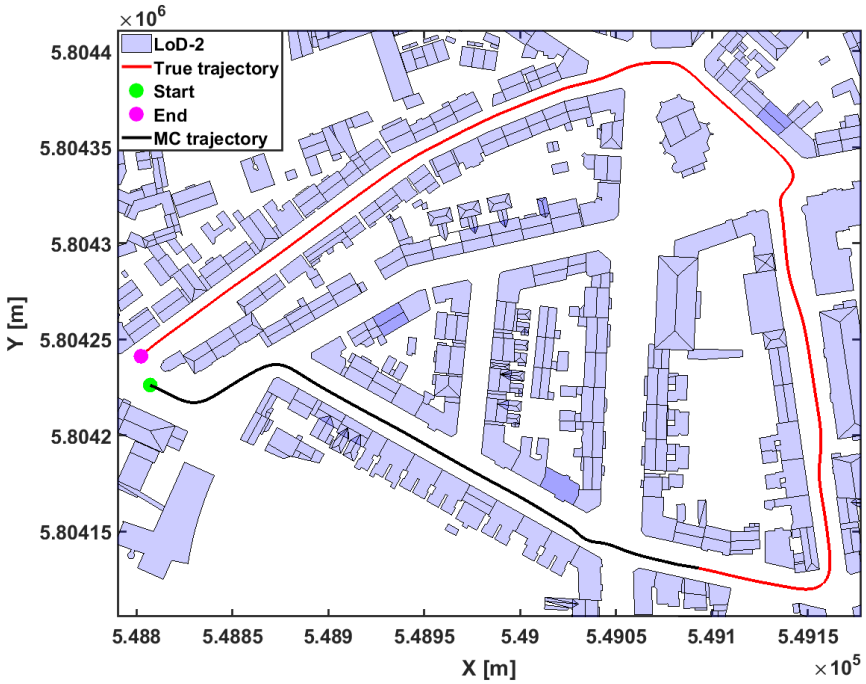

Figure 4.5: *Top view of the simulated environment.*

The values of the optimized VCM of the kinematic model – $\boldsymbol{R}_k$ in equation (4.38) – are given in Table 4.2. These values are derived by applying the *grid search* method, which is explained, for example by Liashchynskyi and Liashchynskyi (2019). The reason for not considering similar design parameters for all filters is their different performance. To be more specific, these values are used to compensate for the uncertainties that cannot be modeled by the adaptive kinematic model. Furthermore, in the prediction step, the estimates of the previous epoch are used to predict the states in the current epoch. Depending on these estimates, the aforementioned uncertainties vary in each filter. Consequently, different design parameters are required to properly compensate for them. Note that it is not impossible to consider similar values for each filter. However, the by-product may be unreliable predictions. In such a case, considering the observations in the update step of the versatile IEKF may not compensate for the incorrect predictions. As a result, this filter may diverge over time or its estimates may not be reliable. In the case of the PF-based frameworks, a larger number of particles may be required to compensate for the unreliable predictions. This will lead to higher computational cost, which is avoided in this thesis.

Table 4.2: *Design parameters of the adaptive kinematic model for the different filters in the simulated environment.*

	σ_{t_x} [m]	σ_{t_y} [m]	σ_κ [°]	k_b
versatile IEKF	0.1	0.1	0.5	8
R-PFI	0.5	0.5	0.5	8
R-EKPFI	0.05	0.05	0.5	8

For the sensor fusion in the case of the R-PFI and R-EKPFI algorithms, w_i is considered to be $\frac{1}{4}$ in the equations (4.24) and (4.33). The reason for this is that there are four sensors at each epoch. These sensors are one scanner, one GNSS and two UWB sensors, whose data can be fused with each other. The corresponding uncertainties to the resulting residuals of the sensors, namely σ_{r_i} in equation (4.33), are given in Table 4.3. In the same table, σ_{r_o} associated with $r_{o,k}^{[s]}$ in equation (4.34) that is used for the IMU data is also given.

Table 4.3: *Uncertainty of the estimated residuals for the matter of sensor fusion in case of the R-PFI and R-EKPFI in the simulated environment.*

	σ_{r_1} [m]	σ_{r_2} [m]	σ_{r_3} [m]	σ_{r_4} [m]	σ_{r_o} [°]
R-PFI	0.1	0.5	0.5	0.5	0.2
R-EKPFI	0.1	0.5	0.5	0.5	0.2

For this case study, the data is simulated with synchronized time stamps. As a result, the data from all of the sensors is available at each epoch. Therefore, the time synchronization aspect is not required as an additional step before applying the filters.
Furthermore, the sensor data is subsampled at each epoch for all filters. This is due to the large number of these data, which leads to high computation time without necessarily adding more information to the filters. Therefore, a subset of these data can be considered for state estimation. Throughout this thesis, a built-in library in MATLAB – called *pcdownsample* is used for this purpose. The main principle in this library is to partition the observations and extract the average of the data in each partition as a representative of the whole. Therefore, the subsampled data are not the original observations. However, the current work claims that considering these data as such is sufficient for the main purpose of this thesis. The reason is that these observations are derived based on the original sensor data. Further investigation of the impact of different subsampling approaches on the developed PF-based frameworks is beyond the main scope of the current work and should be further explored in the future.

4.3.2 Real Environment

The real environment is the Leibniz University Cooperative Perception and Urban Navigation Dataset (LUCOOP) published by Axmann et al. (2023) and available at `https://data.uni-hannover.de/dataset/lucoop-leibniz-university-cooperative-perception-and-urban-navigation-dataset` (DOI: https://doi.org/10.25835/75o9yrc0) under the Creative Commons Attribution-NonCommercial 3.0 License. This dataset is obtained from a large measurement campaign in which three vans are equipped with multiple sensors for different purposes. A schematic overview of these vans and their sensors is given in Figure 4.6. In this figure, the different shapes and colors represent the different sensors on the vehicles. Also, the coordinate frames shown represent the body frame of the vehicles as defined by Axmann et al. (2023). Such frames are needed to combine data from different sensors. In the current work, only the 3D scanner (green rectangle

on the back of the V1), the GNSS (purple circles) and the UWB sensors (represented by orange rectangles) are of interest. To bring the data of these sensors into the origin of the defined body frames, the transformation parameters provided by Axmann et al. (2023) are used. After bringing the 3D scanned data to the origin of the body frame, the transformation given by the equation (4.2) holds for the purpose of assignment.

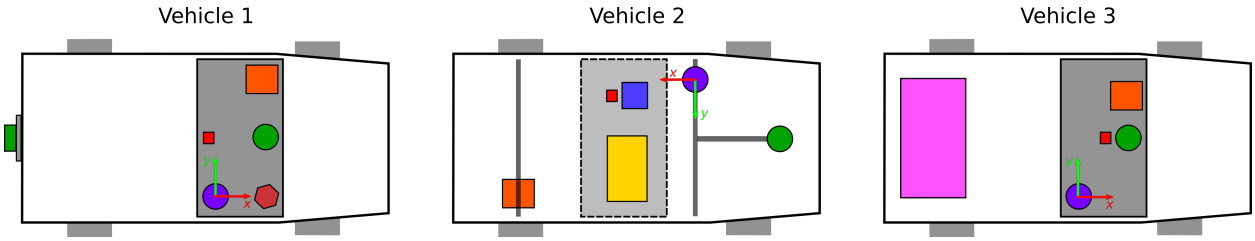

Figure 4.6: *Schematic overview of the three vehicles from top view taken from Axmann et al. (2023).*

The 3D scanned data are derived from the VLP-16 sensor which, due to its vertical installation at the back of V1, captures the environment vertically. To derive the GNSS data on V1, a Javad G3T-JS antenna (JAVAD GNSS Inc., 2020) is used together with a Septentrio PolaRx5TR receiver (Septentrio, 2020). For V2, a NovAtel VEXXIS GNSS-850 antenna (NovAtel Inc., 2017) is used together with a Septentrio PolaRx5e receiver (Septentrio, 2020). The V3 uses a Javad G3T-JS antenna (JAVAD GNSS Inc., 2020) together with a Septentrio PolaRx5e receiver (Septentrio, 2020). The UWB data for all vehicles is derived from PulsON 440 units (Cummings Research Park, 2015). In addition to the sensors mounted on the vehicles, a Leica MS60 total station (Leica Geosystems, 2023) is used to track V1 along part of the route. Figure 4.7 shows how this tracking is done. The left sub-figure shows the detection of the 360° prism on the roof of V1 by the total station. The right sub-figure shows the detected prism as seen by the total station.

Figure 4.7: *An overview of the measurement of the total station taken from Axmann et al. (2023).*

The MS60 total station has an angular and distance accuracy of (1σ) of $1''$ and $1\ \text{mm} + 1.5$ ppm (Leica Geosystems, 2023). The accuracy provided by manufacturers is usually for static measurements. In kinematic cases, the accuracy of the measurements is expected to be in 1-2 centimeter range due to the movement. The current work claims that this level of accuracy is sufficient to evaluate the performance of the filters. Therefore, the data of this sensor is taken as ground truth, as also suggested by (Axmann et al., 2023). For this thesis, these data are used to evaluate the estimated $t_{x,k}$ and $t_{y,k}$ from the filters. To evaluate the estimated κ_k, the given iterative closest

point (ICP)-based trajectory in Axmann et al. (2023) is used as a reference solution. In principle, the derived $t_{x,k}$ and $t_{y,k}$ from the ICP-based trajectory can also be used as ground truth. However, to derive this trajectory, the total station is used to obtain the coordinates of the ground control point. Therefore, the total station is the leading sensor and the trajectory derived from it is given priority over the ICP trajectory in this thesis. This sensor does not provide observations for the κ angle. Therefore, the ICP-based values are used to evaluate this state.
Similar to the simulated environment, V1 is the primary vehicle in the real case. This vehicle is followed by V2 and V3 throughout the measurement scenario. A 3D overview of the real environment is shown in Figure 4.8. The surrounding buildings are from the LoD-2 model. The red dots are a merged point cloud from all available LiDAR sensors. The blue cylinders are the static UWB units and the blue lines are their range measurements. The yellow cylinder is the total station along with its measurement – shown by the yellow line – to the primary vehicle. The secondary vehicles are shown with their body frames behind and in front of the primary vehicle. The green boxes are the 3D bounding box annotations of the static and dynamic objects in the scene.

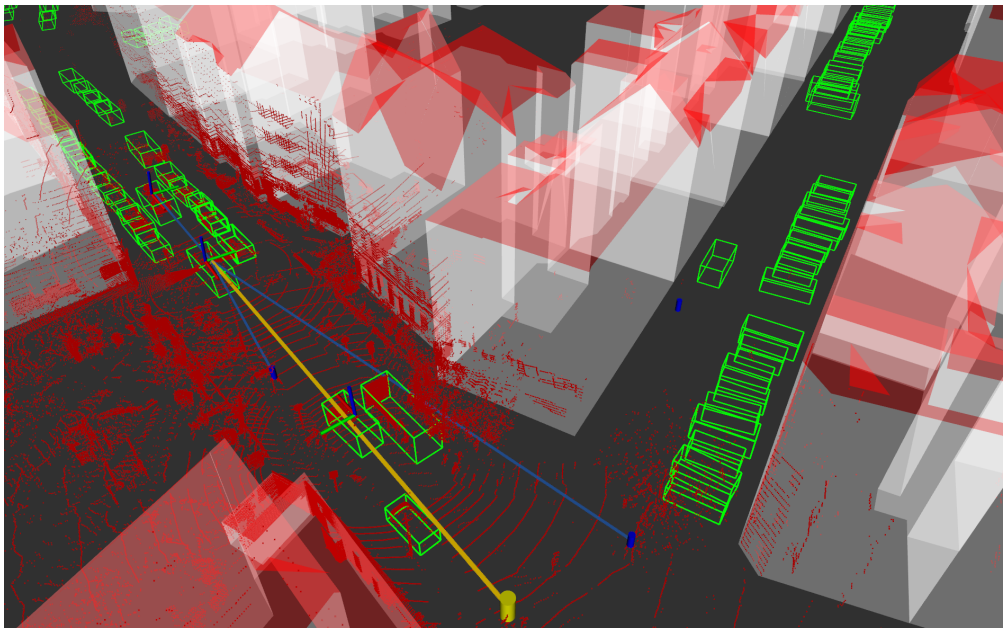

Figure 4.8: *3D overview of the real environment taken from Axmann et al. (2023).*

A top view of the trajectory taken by V1 is shown in Figure 4.9. Similar to the simulated environment, this trajectory has a length of about 1 kilometer. The vehicle V1 has covered this trajectory with an average speed of about 10 kilometers per hour during the measurement campaign. In this figure, the red line is the GNSS data. Furthermore, the blue line is the measured data by the MS60 total station. Since the ground truth data is only available for this patch, the filters are applied to only a selected part of the trajectory. The black line shows this selected part for which the states are estimated using the R-PFI, R-EKPFI and the versatile IEKF algorithms. Note that the ground truth data is not available from the first epoch, but about 17 seconds after the vehicle starts moving. However, the vehicle stands still for several epochs in the beginning before it starts moving. The trajectory for the analysis is chosen to take this stationary state into account. This makes the prediction step of the filters more reliable. The reason for this is on the one hand the VCM of the initialized states. If the filter starts at the epoch where the ground truth data is available, this uncorrelated VCM is used to predict the kinematic motion. This is an unrealistic assumption that may lead to an unreasonable prediction due to the correlation of states in reality. By considering the stationary state, the VCM is eventually fully populated, which in turn leads to a reasonable prediction when the vehicle starts moving. On the other hand, according to the adaptive kinematic model and depending on k_b, the velocities are estimated after several epochs have

passed. Therefore, neglecting the stationary state leads to incorrect estimation of the velocities. This leads to lagging behind the real motion due to not properly propagate the states over time. In addition, the ground truth trajectory is available for about 2 minutes (about 1000 epochs). To ensure that the filters do not diverge, the selected trajectory includes more epochs. However, due to the unavailability of the ground truth, only the estimated κ can be evaluated for this additional part of the trajectory. The main purpose is to assure that the filters do not fail.

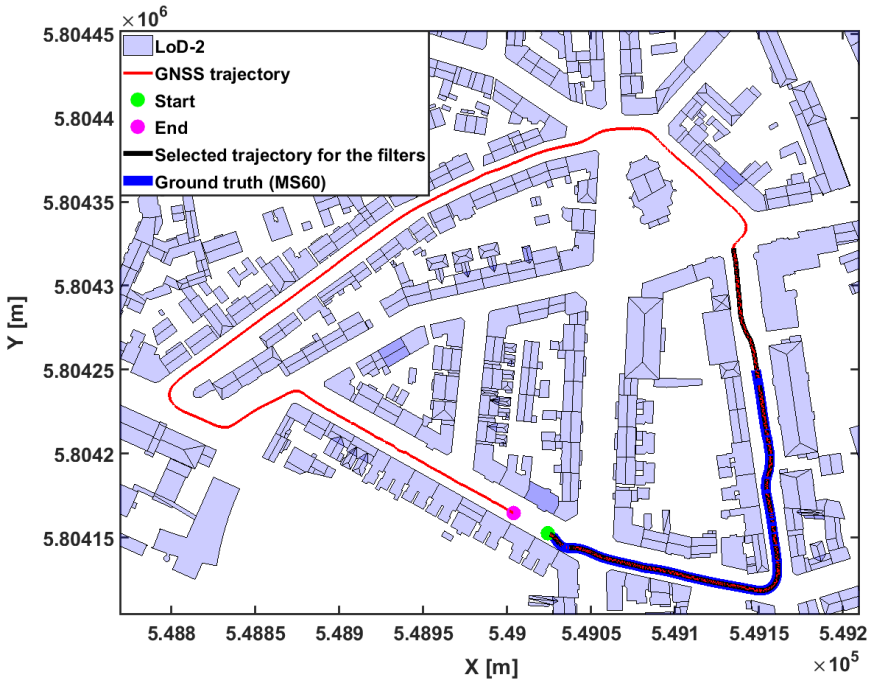

Figure 4.9: *Top view of the real environment.*

Table 4.4 shows the accuracies and sampling rates of the sensors used in the filters. In this table, the term "epoch-wise" indicates whether the accuracy changes over time or remains constant. Furthermore, the term "component-wise" means whether the accuracy is different for each 3D component or not. As can be seen, for the scanner data, the accuracy values are constant over time. Moreover, one accuracy value is considered for all 3D components. On the contrary, for the GNSS data, the accuracy values are both epoch and component specific. The σ_{min}, σ_{mean} and σ_{max} are the minimum, average and maximum accuracy of the derived 3D components over all epochs. These values are based on the techniques used to derive the 3D positions from the GNSS data. In addition, the accuracy of the UWB data changes over time. For these sensors, the output is only one range measurement per UWB sensor. Therefore, the "component-wise" accuracy does not apply to them. In this case, σ_{min}, σ_{mean} and σ_{max} are the minimum, average and maximum accuracy of the derived ranges over all epochs. These values are based on the sensor outputs. $\mathrm{UWB}_{\mathrm{V1V2}}$ and $\mathrm{UWB}_{\mathrm{V1V3}}$ are the derived UWB data from V1 to V2 and V1 to V3, respectively. It should be noted that in the current work the 3D positions of V2 and V3 are taken as deterministic values. Therefore, considering their accuracy within the filters is not relevant.

Table 4.4: *Accuracy and sampling rate of the sensors in the real environment.*

	epoch-wise	component-wise	σ_{min} $[m]$	σ_{mean} $[m]$	σ_{max} $[m]$	f_s $[Hz]$
scanner	no	no	0.02	0.02	0.02	10
GNSS	yes	yes	0.01	0.02	0.1	10
UWB_{V1V2}	yes	no	0.02	0.06	0.3	1
UWB_{V1V3}	yes	no	0.02	0.06	0.3	1

Comparing Figure 4.5 with Figure 4.9, it can be seen that the region in which the georeferencing problem is investigated is similar. In the case of the simulated environment, the sensor data are derived based on the specifications of the sensors used in the LUCOOP dataset. In other words, the simulated environment is a simplified version of the complicated real-world scenario. Due to the similarity of the environment and the accuracy of the sensors, the current work considers the same uncertainty for the kinematic model in both cases. In practice, it is suggested to determine these uncertainties individually for each application. However, due to the high computational time, this is avoided in the current work. Therefore, the same values as given in the Table 4.2 are used for the optimized VCM ($\boldsymbol{R}_k$). Furthermore, for the likelihood estimation in the case of the R-PFI and R-EKPFI algorithms, the same values as in the Table 4.3 are used. For sensor fusion in this case, w_i should be set epoch-wise in the equations (4.24) and (4.33). The reason for this is that the sensors have different sampling rates, which results in a different number of sensors in each epoch. To be more precise, in some epochs all four sensors are used, while in other epochs only some of them are considered.
Furthermore, similar to the simulated environment, the scanned data is considered as segments in each epoch. These segments are based on the frequency of the sensor, which is 10 Hz according to the Table 4.4. To be more precise, the scanned data are cut in such a way that the first point of a segment is 0.1 seconds behind the first point of the previous segment.
In order to apply sensor fusion, it is essential that the sensors are synchronized in time. To do this, a filter time is defined that is common to all filters. Then, based on the filter time, the corresponding sensor data are obtained in each epoch. For that, the observations that are closest to the filter time in each epoch are taken into account.
To define the filter time, the first epoch is derived from the first segment of the scanned data. Then, the entire filter time is derived by successively adding 0.1 seconds. The filter time can also be set to the timestamp of the midpoint in each segment. However, this may result in a filter time with varying time lags. This is due to the fact that the point cloud is sliced based on the sampling rate of the scanner. More specifically, the time difference between the first and last point of the segments may not be the same. This can lead to incorrect time synchronization of the sensors. This means that the considered sensors in each epoch may not be synchronized in time. Therefore, the filters become unstable and diverge after some epochs.
Furthermore, in order to have the ground truth of the total station in each epoch, its data are interpolated based on the filter time. Note that due to the high accuracy of this sensor, no error propagation is considered for the interpolated data.
Additionally, in the same way as in the simulated environment, the scanned data are subsampled in each epoch to avoid high computational time. This is done using the same MATLAB library (*pcdownsample*) as in the simulated case.

4.4 Results and Discussion

In this section the results of R-PFI, R-EKPFI and versatile IEKF on the simulated environment and the real-world case are presented. The main advantage of the simulated scenario is its simplicity compared to the real-world environment. By using this case, it is possible to assure the capability

of the developed PF-based algorithms. Therefore, the first part of this section presents the results of the analysis with respect to the simulated environment.
After verifying the viability of the algorithms in the simulated scenario, it is essential to realize their performance in real-world applications. In such cases, there are usually unpredictable circumstances that can cause complications for the filters. Therefore, it is vital to ascertain that the developed frameworks are capable of reliably mastering such situations. The results of the analysis with respect to this case study are presented in the second part of the current section. In the end, the performance of the algorithms in the two case studies is summarized in order to better reflect their strengths and weaknesses.

4.4.1 Simulated Environment

In this section, two sets of results related to the simulated case are given in two separate parts. These parts are referred to as "Setup 1" and "Setup 2". In Setup 1, the results of the filters on the first 1400 epochs of the trajectory are given. As mentioned earlier, 100 MC runs are applied to this segment of the trajectory to verify the operability of the filters. Furthermore, the correct design parameters to be used in the filters in the simulated environment (Table 4.2) are derived based on this patch of the trajectory. The reason for not considering the whole trajectory to obtain these parameters is the high computation time. Besides, this selected piece of the trajectory contains curves and straight parts. Therefore, in the current work, it is considered sufficient to obtain the design parameters, since it can be a representative of the whole trajectory. In this part, a total number of 1000 and 20 particles in each epoch are considered for the R-PFI and R-EKPFI, respectively. For the R-PFI, more than 1000 particles are not selected due to computational cost. In the case of R-EKPFI, the use of 20 particles is found to be sufficient to compare its performance with the other filters over the MC runs.
In addition, Setup 2 analyzes the entire trajectory consisting of 3670 epochs. In this part, the same filters as in Setup 1 are applied to obtain the trajectory. However, in the case of the R-EKPFI, four realizations with different numbers of particles (10, 20, 50 and 100) are considered. The reason for this is to explore the effect of sample size on the accuracy of the estimated states. Furthermore, the trajectory is analyzed 10 times by the filters. This is to assure their stability due to the random nature of these frameworks. Note that unlike the MC runs, the observations remain the same throughout these 10 runs. For the analyzes in Setup 2, the obtained design parameters in Setup 1 are used.
As mentioned throughout the thesis, the underlying basis of the filters is to benefit from the geometric information of the building models. Therefore, the number of planes detected at each epoch plays a considerable role in the performance of the filters. Figure 4.10 shows the number of planes assigned to the scanned data at each epoch by each filter over the entire trajectory (3670 epochs). This figure is deemed helpful to better inspect the given results obtained from Setup 1 and Setup 2 in the following parts. The colored bars indicate the trajectory curves, which are further shown in Figure 4.11. Note that the versatile IEKF is referred to as IEKF in the legend of all the plots in the following parts.

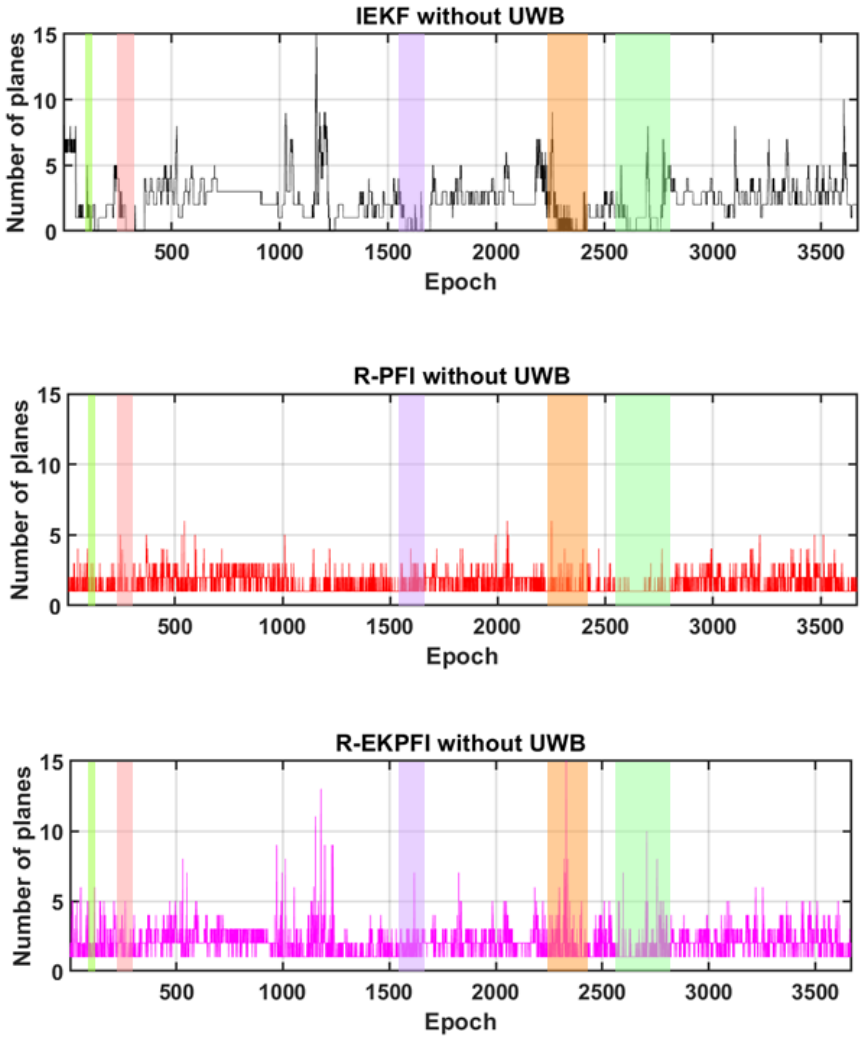

Figure 4.10: *Number of detected planes over the while trajectory by means of different filters. The colored bars indicate the curves of the trajectory.*

The plots in Figure 4.10 are derived from the results of the assignment step. For this, and in the case of the versatile IEKF, the predicted states in each epoch are used. In the case of R-PFI and R-EKPFI, one of the predicted particles is considered for this visualization. These plots are given as a general overview of the planes identified by each filter. Therefore, in the current work, one particle is claimed to be representative of the detection capability of the corresponding PF-based method.

Figure 4.11: *Top view of the simulated environment with the curves of the trajectory depicted by colored boxes.*

From Figure 4.10 it can be seen that in the case of R-PFI and R-EKPFI at least 1 plane is detected in each epoch. On average, 2 planes are detected in each epoch by these two filters. Furthermore, the maximum number of planes detected by the R-PFI over the epochs is shown to be 6. In the case of the R-EKPFI, the maximum is derived to be 15 planes. For the versatile IEKF it can be seen that in several epochs, including the curves, no planes are detected. This means that in these epochs the state estimation is done by using other sensors than the scanner. In addition, the average and maximum number of planes detected by this filter over time are obtained to be 2 and 15, respectively. In general, the comparison of the detected planes in the case of the versatile IEKF with the other two filters shows a considerable variation in the detected planes in a short period of time. It can be seen that this variation in some of the epochs can lead to the absence of planes in the subsequent epochs. Such a pattern is reflected in the estimated states using this filter. As will be shown later, there is a high variation between the accuracy of the estimated states by this filter over time compared to the others. In the case of R-PFI and R-EKPFI, it can be seen that for a large part of the trajectory, the number of planes detected varies between 1 and 5, with less variation over consecutive epochs. As will be discussed later, this leads to a more stable accuracy of the resulting estimates by these filters.
Besides the number of planes detected, their configuration in each epoch is also important. Detecting planes on only one side of the street will result in a trajectory that is drifted. In contrast, if the planes on both sides are correctly detected, a rational trajectory is estimated that remains between the buildings without dragging to one side. In this case, by detecting more planes on both sides, it is more likely that the effect of the uncertainty of the planes on the estimated trajectory is randomized. Such an aspect becomes critical at intersections where there are several planes on different sides of the road. In these situations, a correct detection of the planes with a proper configuration is essential. However, to detect the planes in each epoch, the predicted states (in the versatile KF) or particles (in the R-PFI and R-EKPFI) are used. Therefore, a correct assignment of the planes requires a reliable prediction. Nevertheless, for a reliable state estimation, correctly detected planes with appropriate configuration are required. Consequently, the detected planes and the estimated states influence each other over the epochs. This is an important aspect to bear in mind when evaluating the performance of the filters.

4.4.1.1 Setup 1

Figure 4.12 shows the average of the accumulated RMSE of the estimated states by the filters over the MC runs. To derive these values, equation (3.20) is used. In general, obtaining the accumulative RMSE does not evaluate the performance of the filters at each epoch. However, it does give a pattern of performance over time. To be more precise, if the state estimation is improved, a decreasing pattern for the accumulative RMSE should be observed. Conversely, if the performance is degraded, this measure will show an increasing pattern. Such a clear ascending and descending pattern is usually observed for the simulated scenarios where unexpected effects are avoided. Therefore, this thesis considers the accumulative RMSE as a suitable measure to monitor the performance of the filters over the epochs in the case of the simulated environment. In real applications, a number of effects can interfere with the state estimation. Since these effects are not necessarily similar in all epochs, a clear pattern for the accumulative RMSE over time may not be obtained. To evaluate the individual performance of each epoch, the absolute error should be calculated separately for each epoch without including the estimated states in the preceding epochs.
The results of the analyses over the MC runs show that in the case of the R-EKPFI with and without the UWB data, 75% and 93% of the runs cover the 1400 epochs. Therefore, the diverged runs before reaching the 1400^{th} epoch are not taken into account in obtaining the given results. In general, it is concluded that the inclusion of the UWB data does not have a strong impact on the estimated states. The current work gives two reasons for such an effect. First, according to Table 4.1, the same accuracy is considered for the simulated UWB and GNSS data. Therefore, the presence of two additional measurements with a similar accuracy to the GNSS data does not necessarily have to result in an improvement of the estimated states. Second, as explained in Section 4.3, in the current work the positions of V2 and V3 are considered deterministic. These positions are simulated with the same accuracy as the GNSS data of V1. Therefore, considering them as deterministic values does not guarantee an enhancement of the estimated states.
In addition, it can be seen that the estimated t_x and t_y by the R-PFI are similar to those derived by the versatile IEKF. However, the results of the κ estimates show a better performance of the R-PFI compared to the versatile IEKF. At this stage, the advantage of the R-PFI over the versatile IEKF can be confirmed. Based on the current work, it is claimed that increasing the number of particles in the R-PFI can help improve the estimated t_x and t_y states. The reason for the better performance in estimating κ compared to the other two states by the R-PFI is the less changes in the heading angle over time during the ride. This means that the heading angle is not subject to constant changes over time, except in the parts where the vehicle is supposed to turn. Conversely, $t_{x,y}$ and t_y change constantly over time.
From Figure 4.12 it can be seen that the best performance for all estimated states belongs to the R-EKPFI. In this case, the computations are on average 12 times faster than those derived by the R-PFI, which is the main advantage of the R-EKPFI compared to the R-PFI. Moreover, for the current example, the results of R-PFI and R-EKPFI show an average outlier removal of about 1.5% and 2% over the epochs, respectively.
Note that in the simulated case, the vehicle stands still for about 5 seconds before starting to move. Such a situation, together with the good initialized state values, has caused the RMSE results of the versatile IEKF to be close to zero in the first few epochs. For the R-PFI and R-EKPFI such an effect is not observed. The reason for this is the stochastic nature of these filters. In other words, in the case of the versatile IEKF, the good initialization is used directly in the filter to estimate the states. In the case of the R-PFI and R-EKPFI, however, several samples are randomly generated around the initialized state. This leads to an increased uncertainty of the estimated states.

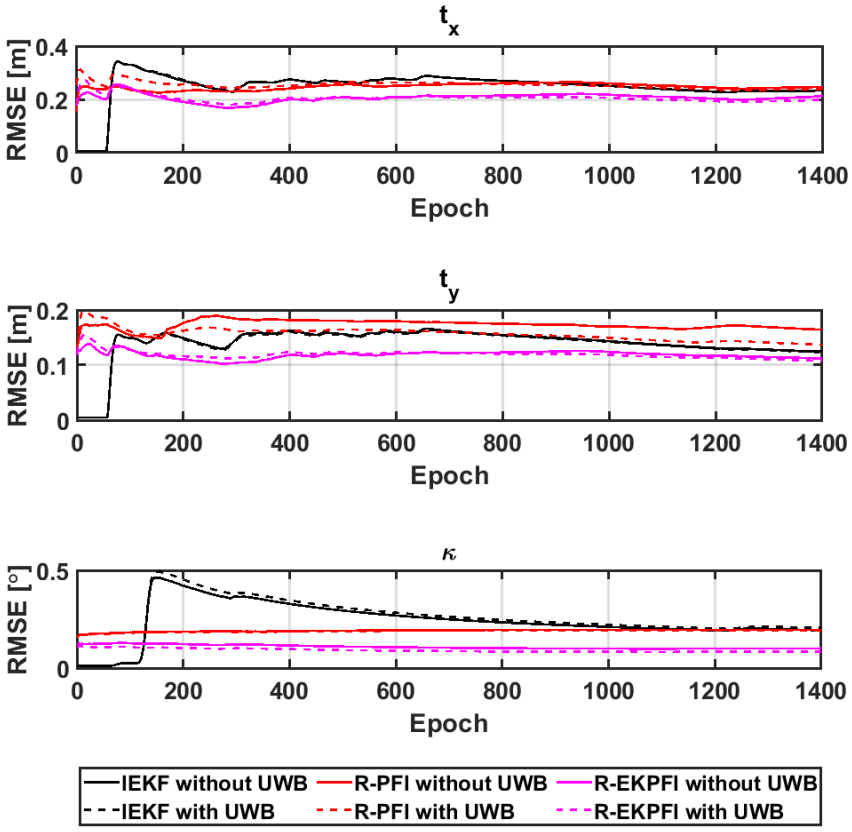

Figure 4.12: *Average accumulative RMSE of the estimated states over MC runs.*

To get a better impression of the performance of the versatile IEKF, R-PFI and R-EKPFI over the MC runs, figures 4.13, 4.14 and 4.15 are given. These figures show the statistical values of the accumulative RMSE for t_x over the MC runs. The corresponding figures for t_y and κ are given in the Appendix (A.2). In these figures it can be seen that the 95% CI is acutely small, indicating that the estimates over the MC runs are similar. In Figure 4.13, it can be seen that in the case of the versatile IEKF and for the first few epochs, the minimum and maximum bounds are close to the mean and median values. Such a result is explained by the fact that the vehicle is stationary at the beginning. In this case, since the filter is initialized similarly over the MC runs, the resulting estimates are the same. As the vehicle begins to move, the minimum and maximum bounds become larger. The reason for this is that we have different observations over the MC runs, which leads to different estimates. The minimum and maximum bounds shown are caused by those estimates that are outside the 95% CI. Furthermore, it can be seen that the mean and the median confirm each other, indicating that there is no bias in the estimates over the MC runs. By having a similar pattern in the upper and lower plots, the previous conclusion regarding the insignificant impact of the UWB data on the estimated states is further confirmed.

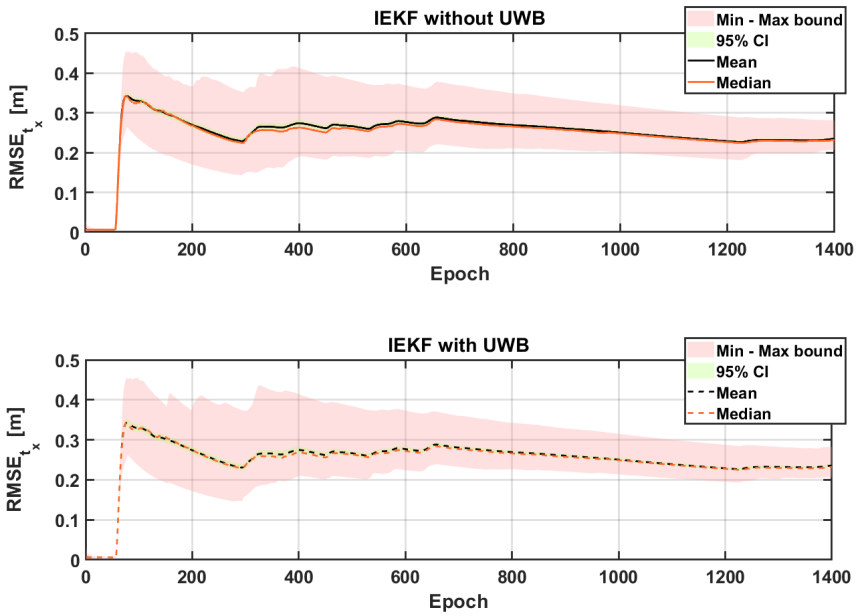

Figure 4.13: *statistical values of the accumulative RMSE of the estimated t_x by the versatile IEKF over MC runs.*

According to Figure 4.14, the minimum and maximum bounds appear large at the beginning and become smaller over time. Such an effect is in contrast to the case of the versatile IEKF. In the R-PFI, since new samples are generated over the MC runs, the minimum and maximum bounds initially have a large area. Over time, as the precision of the estimates improves due to the resampled particles, the boundary becomes smaller. Comparing the top and bottom plots, it can be seen that considering the UWB data leads to a higher maximum and minimum boundary, which is more substantial in the beginning. In other words, by considering the UWB data, those estimated states over the MC runs that are outside the 95% CI have a lower accuracy than in the case where no UWB data is considered. To identify the reason for this, the residuals resulting from the UWB data are inspected. These residuals are confirmed to be meaningful, which justifies the correct use of such data. Based on further investigation, the current work argues that the larger maximum and minimum bounds in the case of including the UWB data are due to the sensor fusion strategy explained in Section 4.2.6. As mentioned earlier, the weight given to the sensors during the fusion is considered equal in the current work. According to equation (4.33), this leads to an increased $\sigma^2_{r_f}$ per sensor. Initially, when the vehicle is stationary, fusing the sensors as described will result in a resampled particle set with more variability. Considering this variation in the MC runs results in a wider minimum and maximum bound compared to the case where no UWB data is considered. Similar to the versatile IEKF, no bias is observed in the estimates over the MC runs with and without consideration of the UWB.

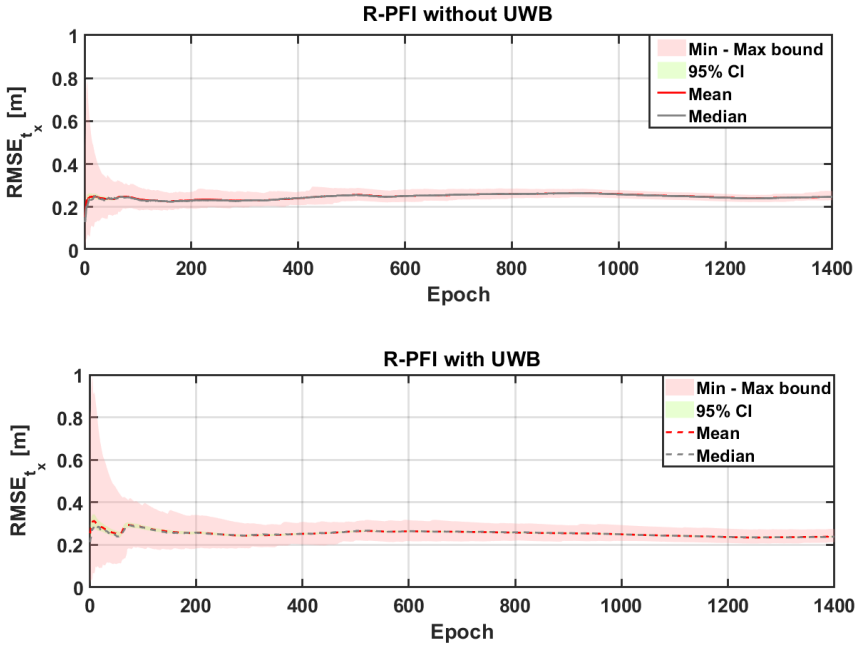

Figure 4.14: *statistical values of the accumulative RMSE of the estimated t_x by the R-PFI over MC runs.*

According to Figure 4.15, the same conclusion can be drawn for the R-EKPFI as for the R-PFI. However, in this case the RMSE values are higher compared to the MC runs due to the significantly lower number of particles (20 versus 1000). In addition, according to the lower plot, the inclusion of the UWB data leads to a slight bias in the estimates. The current work asserts that the double consideration of the UWB data in the "Modification step" and "Update step" (lines 6, and 13 to 16 of Algorithm 12) for each particle is the reason for such an effect. This, together with considering the position of the secondary vehicles as deterministic values, leads to such a bias in the MC runs. It should also be noted that these figures show the accumulative RMSE. Therefore, a bias in one epoch affects the upcoming RMSE values.

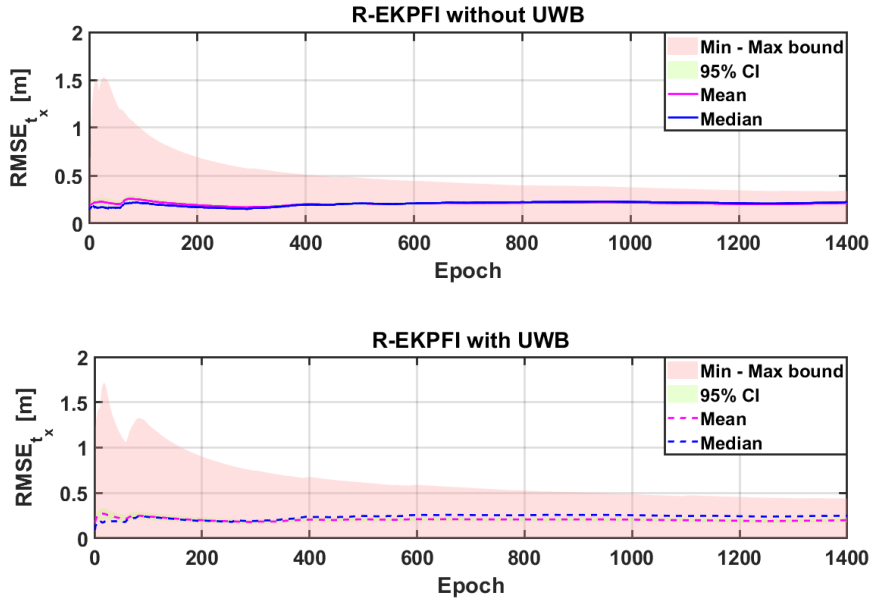

Figure 4.15: *statistical values of the accumulative RMSE of the estimated t_x by the R-EKPFI over MC runs.*

To get an impression of the statistical values, Table 4.5 is given below. In this table, the corresponding statistical values to the $RMSE_{t_x}$ in the last epoch are given for all filters. The related tables to the statistical values of $RMSE_{t_y}$ and $RMSE_{\kappa}$ are given in the Appendix (A.2).

Table 4.5: *statistical values of the average accumulative RMSE of the estimated t_x in the last epoch over the MC runs by means of the different filters. Red and green colors indicate the minimum and maximum values in each column, respectively.*

	$RMSE_{t_x}$ $[m]$					
	Min	Max	Mean	Median	↑ CI (95 %)	↓ CI (95 %)
versatile IEKF without UWB	0.2071	0.2810	0.2352	0.2325	0.2386	0.2318
versatile IEKF with UWB	0.2069	0.2827	0.2357	0.2326	0.2388	0.2326
R-PFI without UWB	0.2317	0.2750	0.2471	0.2470	0.2485	0.2456
R-PFI with UWB	0.2124	0.2742	0.2396	0.2386	0.2420	0.2371
R-EKPFI without UWB	0	0.3386	0.2139	0.2232	0.2264	0.2014
R-EKPFI with UWB	0	0.4394	0.1996	0.2504	0.2231	0.1762

The accumulative RMSE values do not reveal the epochs in which the states were poorly estimated. Therefore, Figure 4.16 shows the mean absolute error (MAE) of the estimated states. The MAE is derived by calculating the average of the absolute error of the estimate states in each epoch k over the MC runs as follows:

$$\boldsymbol{MAE}_k = \frac{1}{S} \cdot |\boldsymbol{\mu}_k - \tilde{\boldsymbol{\mu}}_k| \tag{4.41}$$

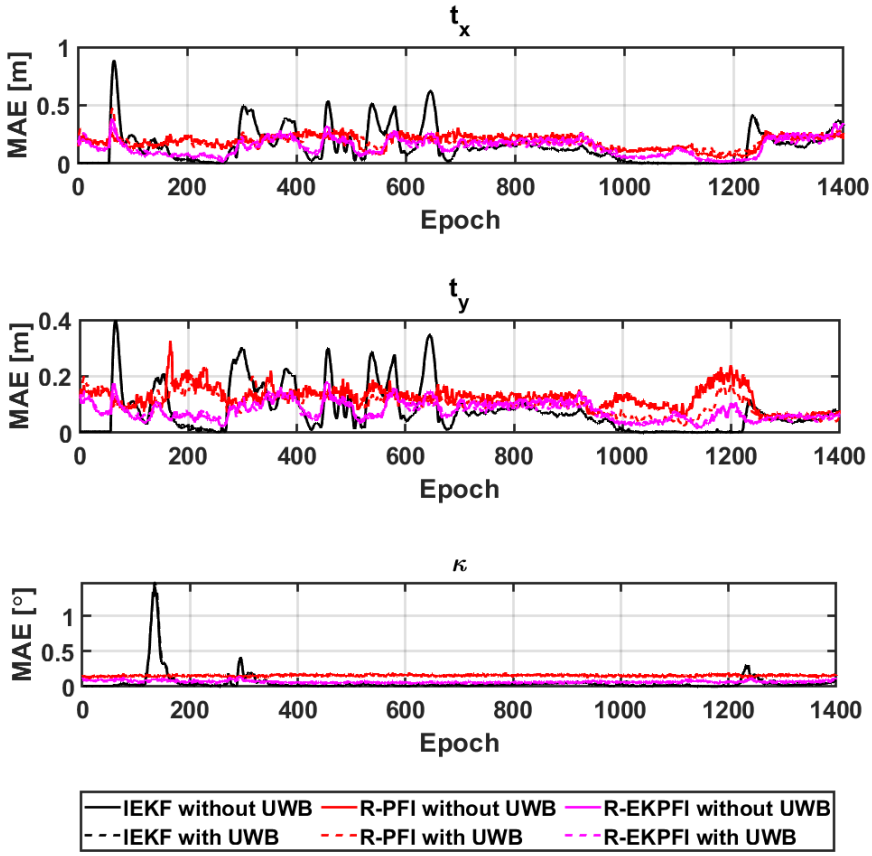

Figure 4.16: *Absolute error of the estimated states over MC runs.*

The results of the versatile IEKF according to Figure 4.16 display several peaks that occur in some of the epochs. The first sudden increase of the MAE values is due to the movement of the vehicle, which starts after about 5 seconds. Comparing the first peak that appears in the corresponding plots of t_x and t_y with that of κ, it can be seen that in the latter plot the peak appears with a delay. Inspecting the simulated environment, it can be seen that such an increase in the estimated κ is due to the first curve that the vehicle should take. According to Figure 4.10, the number of planes detected by this filter on the first turn decreases to 1, and in several epochs along the curve no planes are detected. This directly affects the estimated states including κ. In other words, as the vehicle begins to move, only t_x and t_y change significantly, while changes in κ are not noticeable. The first significant change in κ is on the first curve, leading to the sudden increase in its corresponding MAE value. By further inspecting the simulated environment, it is argued that the increase in the estimated MAE values of the states in the case of the versatile IEKF at some epochs is due to two reasons. One is the curves on which the vehicle should turn. In this case, the detection of the facades of the building models is essential. However, as explained earlier and according to Figure 4.10, there are a number of epochs, including on the curves, in which the versatile IEKF is unable to detect any planes. If such a situation occurs in the first epochs, the state estimation is directly affected. As already mentioned, the detection of the planes and the state estimation owe their reliability to each other. Therefore, the negatively influenced states can lead to a false detection of planes in the subsequent epochs. Instead of a wrong assignment, it can also happen that no plane is detected due to a wrong estimation of the states. Such an influence can be repeated over the whole trajectory, resulting in a less reliable state estimation compared to the other two filters.

The other reason is those locations where not enough LoD-2 models are detected. According to Figure (4.5) or Figure 4.11, it can be seen that due to intersecting streets in some locations, the LoD-2 models are more available on one side of the vehicle. Furthermore, at the intersections, the

configuration of the detected LoD-2 models may lead to an estimated pose that is not close to the ground truth. It can be seen that the R-PFI and the R-EKPFI are less affected by such situations. In other words, no clear peak can be detected for the resulting MAE values of these filters. In addition, this figure shows that the R-EKPFI results in smaller MAE values of the states compared to the R-PFI. Finally, according to this figure, the insignificant influence of the UWB data on the estimated states is further verified. The corresponding figures to the statistical values of these MAE values are provided in the Appendix (A.2).
To have a comparison of the precision of the estimated states between different filters, Figure 4.17 is given. The main influence of the UWB data on the R-PFI and R-EKPFI is shown in this figure. As can be seen, the inclusion of the UWB data in these two filters leads to a higher precision for the estimated t_x and t_y. The precision of the estimated κ remains unaffected. The reason for this is the observation model – given by the equation (4.19) – of the UWB data, which applies only to the translation parameters. In the case of the versatile IEKF, considering these data does not lead to a significant change in the derived precision. Therefore, it can be concluded that considering the UWB data reduces the precision of the proposed PF-based methods; while it has no significant influence on the precision of the versatile IEKF.

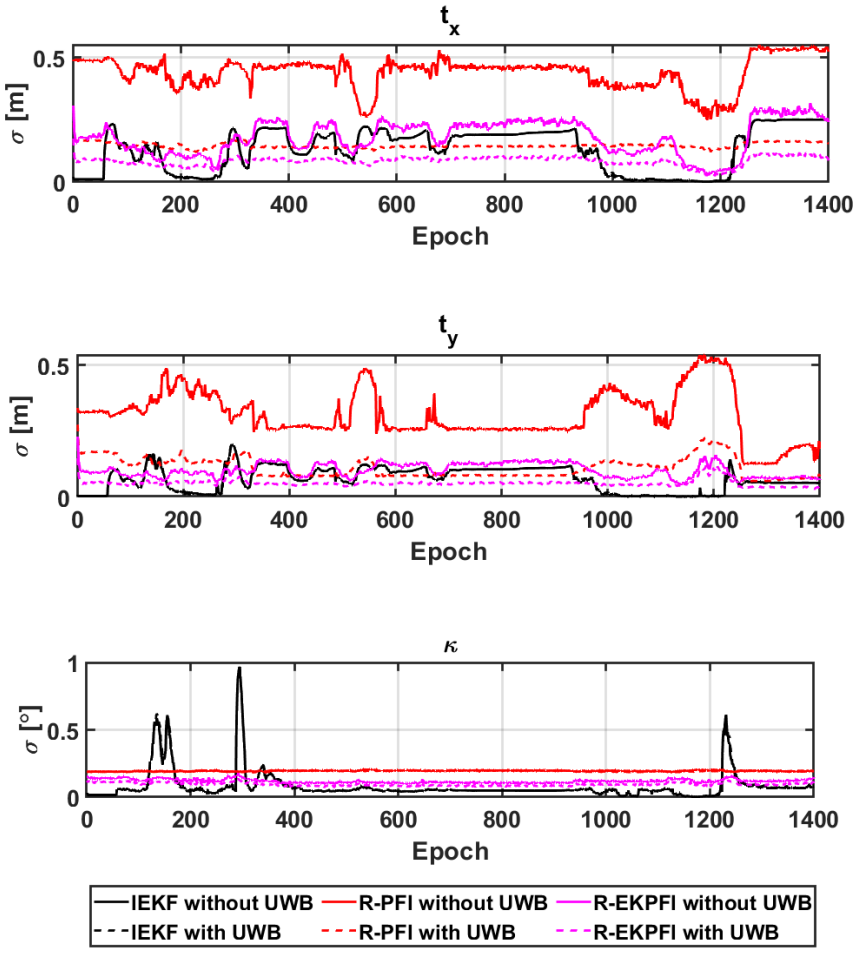

Figure 4.17: *Average precision of the estimated states over MC runs.*

To have a better overview of the statistical values of the precision of the estimated t_x over the MC runs, figures 4.18, 4.19 and 4.20 are given, corresponding to the versatile IEKF, R-PFI and R-EKPFI, respectively. The figures related to the precision values of t_y and κ are given in the appendix.

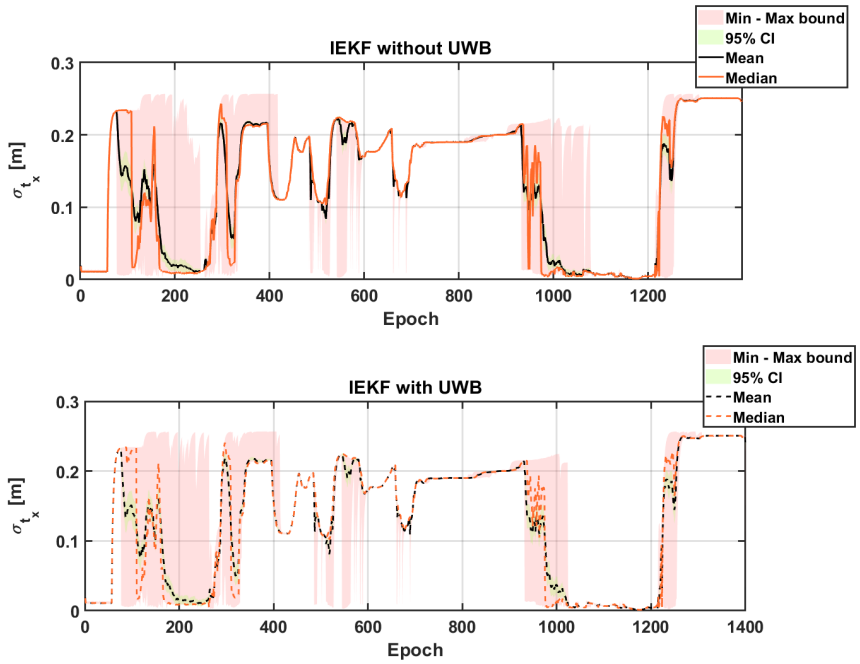

Figure 4.18: *statistical values of the precision of the estimated t_x by the versatile IEKF over MC runs.*

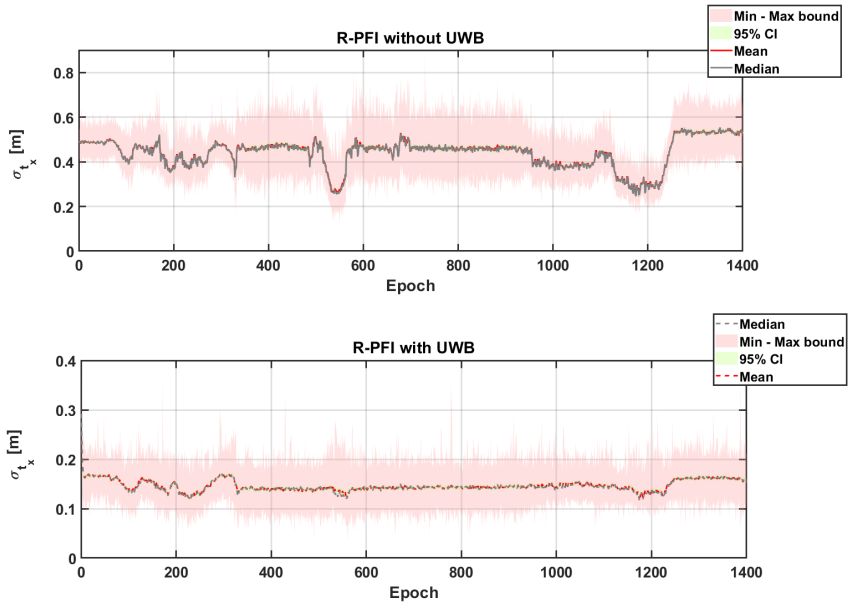

Figure 4.19: *statistical values of the precision of the estimated t_x by the PFI over MC runs.*

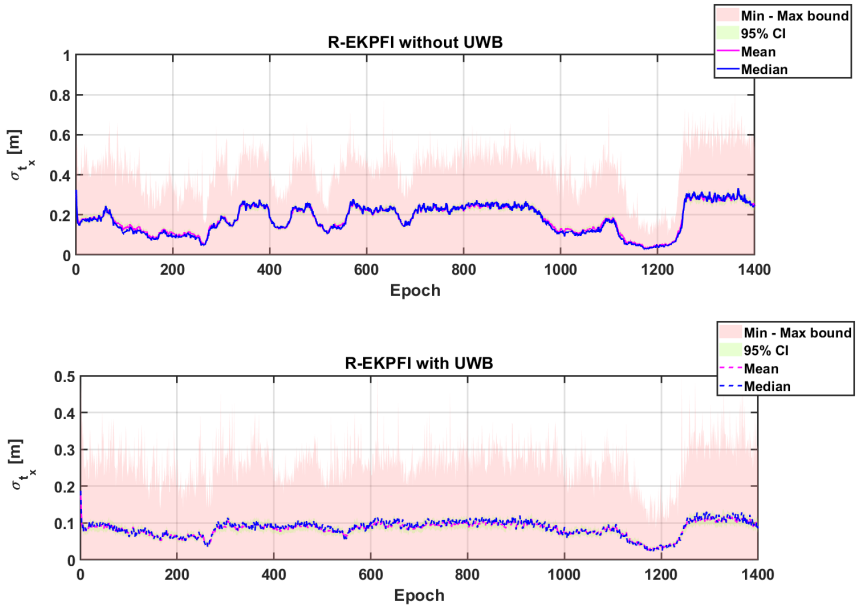

Figure 4.20: *statistical values of the precision of the estimated t_x by the R-EKPFI over MC runs.*

The small 95% CI regions indicate the stable precision of these filters over the MC simulations. In Figure 4.18 it can be seen that the minimum and maximum bounds appear only in some of the epochs. In other words, only in some of the epochs do the precision results vary over the MC runs. The current work argues that these epochs are those in which either a curve is reached or the configuration of the detected LoD-2 models is not suitable. It can be seen that the consideration of the UWB data does not lead to a significant improvement of the derived statistical values.
In the case of the R-PFI, it can be seen that the consideration of the UWB data leads to a smaller minimum and maximum bounds. In other words, the estimated precision values over the MC runs are in agreement. A similar interpretation can also be concluded from Figure 4.20 for the R-EKPFI. However, due to the significantly lower number of particles in this case, the resulting boundaries are larger. In other words, the precision of the R-EKPFI for t_x is not as stable as that of the R-PFI. Table 4.6 gives the statistical values of σ_{t_x} in the last epoch, which helps to have a better numerical comparison between the performance of the filters.

Table 4.6: *statistical values of the average precision of the estimated t_x in the last epoch over the MC runs by means of the different filters. Red and green colors indicate the minimum and maximum values in each column, respectively.*

	σ_{t_x} $[m]$					
	Min	Max	Mean	Median	↑ CI (95 %)	↓ CI (95 %)
versatile IEKF without UWB	0.0140	0.2507	0.2437	0.2481	0.2501	0.2372
versatile IEKF with UWB	0.0146	0.2507	0.2392	0.2481	0.2482	0.2303
R-PFI without UWB	0.3589	0.6734	0.5221	0.5237	0.5332	0.5110
R-PFI with UWB	0.1108	0.2125	0.1555	0.1533	0.1595	0.1514
R-EKPFI without UWB	0	0.5984	0.2464	0.2485	0.2705	0.2224
R-EKPFI with UWB	0	0.2927	0.0861	0.0799	0.0996	0.0726

To have a comparison between the estimated posterior distributions by the proposed PF-based algorithms and the Gaussian distributions derived by the versatile IEKF, the KLD measure is used according to equation (3.23). For this, the versatile IEKF is considered as a reference against which

the estimated posterior distributions by R-PFI and R-EKPFI are compared. The main purpose is to realize whether the obtained posterior distributions are similar between the developed PF-based approaches and the versatile IEKF. In order to have the reference posterior distribution, 1000 samples are generated by using the estimated states of the versatile IEKF and its corresponding VCM in each epoch. In the current work, 1000 samples are considered sufficient to compare the posterior distributions. However, it is worth noting that due to the high computational cost, the effect of using more samples is not investigated. This fact should be borne in mind when interpreting the following results.
In addition, to have a fair comparison between the filters, the cases with and without the UWB data are compared separately. In other words, two sets of 1000 samples are generated from the versatile IEKF. One set belongs to the case in which the UWB data are considered. These samples are taken as reference solutions against which the resampled particles from the R-PFI and R-EKPFI with the UWB data are compared. The other set is derived from the versatile IEKF solutions in which no UWB data are considered. The results of the R-PFI and R-EKPFI without UWB data are then compared to these samples. Furthermore, to avoid high computational times, the KLD measure is computed every 100 epochs, results of which, after averaging over the MC runs, are shown in Figure 4.21. The main purpose is to ensure that the posterior distributions of the R-PFI and R-EKPFI are not the same Gaussian distributions as in the versatile IEKF.

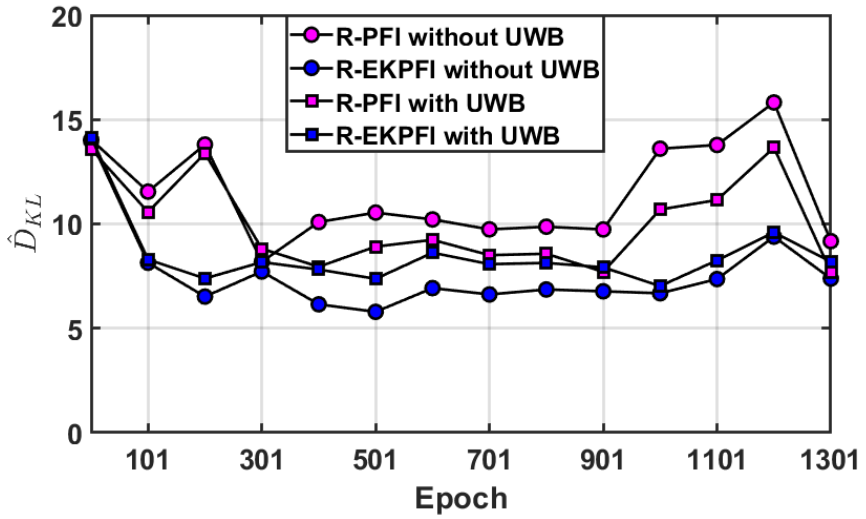

Figure 4.21: *Average of the KLD estimate* $\left(\hat{D}_{KL}\right)$ *between the versatile IEKF and the PF-based algorithms in each epoch over MC runs.*

Since the estimated KLD measures have large values, the current work claims that the posterior distributions derived by the PF-based methods differ from those obtained by the versatile IEKF. However, this claim requires further exploration beyond the scope of this work. To the best of our knowledge at the time of writing this dissertation and according to Chou et al. (2011), there is no significance test that can be applied to the obtained KLD estimates. Therefore, no solid conclusion with analytical support regarding the similarity or dissimilarity of the distributions can be carried out.
In addition, it can be seen that for the cases where no UWB data are considered, the posterior distributions derived by the R-EKPFI have smaller distances to the reference solutions compared to those derived by the R-PFI. The reason for such a result is argued to be the modification of the particles by the Kalman gain. This results in shifting the particles to the regions where the versatile IEKF estimates are. For the cases where the UWB data are considered, it can be seen that the resulting distances to the reference solutions by the R-PFI and R-EKPFI are in a similar range in most of the epochs. The reason is thought to be the constraining effect of the UWB data,

which compels the particles into the regions where the corresponding observation model to these data is satisfied. To be more precise, considering the UWB data in both the R-PFI and R-EKPFI results in resampled particles that have less variability compared to the case where no UWB data is considered.
To have a comparison between the posterior distributions derived by the R-PFI and R-EKPFI, the KLD measure is calculated for the results of these two algorithms. For this, the resampled particles of the R-PFI in each epoch are taken as a reference against which those of the R-EKPFI are compared. Similar to Figure 4.21, cases with and without the UWB data are compared separately. To better compare this figure with Figure 4.21, the scale of the y axis is considered similar. According to Figure 4.22, it can be seen that the derived KLD measures for the two cases are close to each other and smaller than those shown in Figure 4.21. Therefore, it is concluded that the resulting posterior distributions from the R-EKPFI are more similar to the R-PFI than to the versatile IEKF. This thesis contends that these results confirm the similar underlying framework of these two filters.

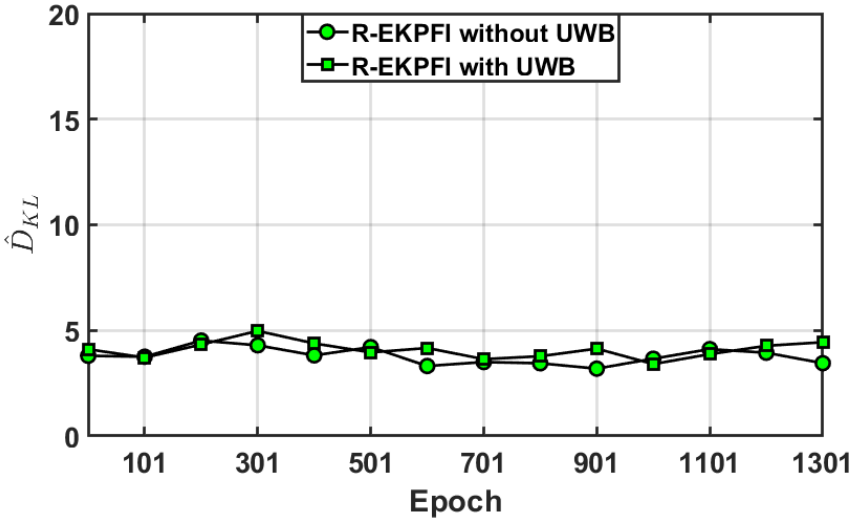

Figure 4.22: *Average of the KLD estimate* $\left(\hat{D}_{KL}\right)$ *between the R-PFI and the R-EKPFI algorithms in each epoch over MC runs.*

The overall conclusion based on the given results is that the proposed PF-based methods have a better performance than the versatile IEKF framework. It is also found that modifying the particles based on the Kalman gain within the R-EKPFI leads to better results than the R-PFI with substantially less computational time. The main challenge in the R-EKPFI is the number of samples to be used in each epoch. As explained earlier, the R-EKPFI results given in this section are based on the use of 20 particles, which has led to the divergence of the filter in some MC simulations. In the second part of the results related to the simulated environment, the effect of using fewer and more particles on the estimates is shown.

4.4.1.2 Setup 2

In this part, the entire trajectory (3670 epochs) is analyzed by the filters to ensure their ability to estimate the states in all the epochs. In this case, the filters are applied 10 times to ensure that no divergence occurs in any of them. The reason for applying R-PFI and R-EKPFI 10 times is to ensure that these filters do not diverge due to their random nature. The reason for applying the versatile IEKF 10 times is to be certain that the adaptive kinematic model explained in Section 4.2.4 does not cause this filter to diverge. In this part, a total number of 1000 particles in each epoch is considered for the R-PFI. In the case of the R-EKPFI, 10, 20, 50 and 100 particles in each epoch are considered. Furthermore, unlike the MC simulations, the same observations are used

over the 10 runs. Additionally, due to the marginal effect of the UWB data on the filters, their corresponding results are neglected in the following.
Figure 4.23 shows the average of the accumulative RMSE of the estimated states over all 3670 epochs. The main purpose of this figure is to show the ability of the aforementioned filters to georeference V1 along its entire trajectory. As can be seen, the filters show the same pattern as in Figure 4.12 over the whole trajectory. The corresponding figure to the MAE is given in the Appendix (A.3).

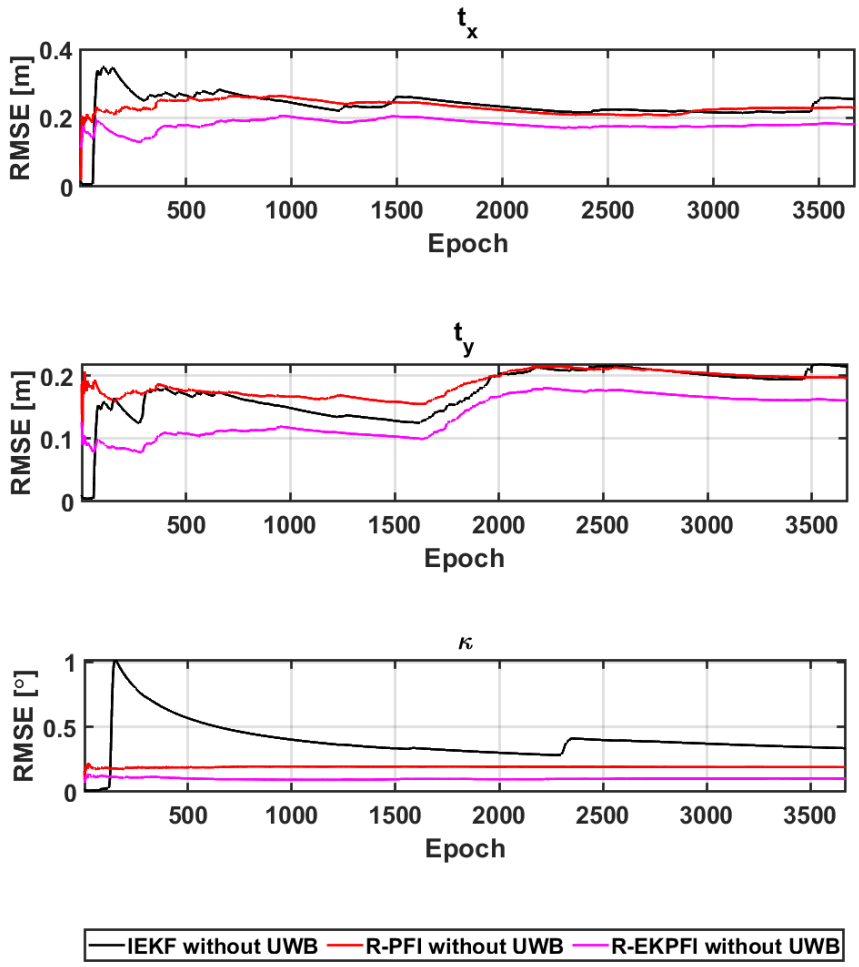

Figure 4.23: *Average accumulative RMSE of the estimated states over multiple runs.*

As mentioned before, the R-EKPFI algorithm is applied with 10, 20, 50 and 100 particles to realize the effect of the number of samples on its performance. The analysis results for the variant with 10 particles showed its divergence after some epochs. Therefore, its results are not shown in the following figures. Figure 4.24 depicts the average accumulative RMSE of the remaining R-EKPFI variants, namely with 20, 50 and 100 particles. As can be seen, increasing the number of particles has led to a better estimation of t_x and t_y. However, the κ estimates are slightly worse when the number of particles is increased. Considering the results derived from Setup 1 over the MC runs, the current work sees the reason in the considerably poor performance of the versatile KF in estimating κ in several epochs compared to the other filters. As explained in Section 3.3, the versatile KF is used to modify the particles in the R-EKPFI. Therefore, as the number of particles increases, the degradation effect of the versatile KF on the κ estimation becomes more visible. Furthermore, the plots given in this figure are related to the accumulative RMSE. Thus, the effect of inaccurate estimates accumulates over time.
Moreover, as can be seen in Figure 4.24, the results of the R-EKPFI with 50 and 100 particles are

not substantially worse than the case with 20 particles. This is due to the likelihood estimation and resampling steps in the R-EKPFI. These two steps effectively control those particles that can have a significantly high worsening effect on the κ estimation. In addition, the results of the analysis show a success rate of 100% for the cases where 50 and 100 particles are used. In the case of using 20 particles, the success rate is shown to be 90%. Thus, despite the worse κ estimates in the 50 and 100 particle cases compared to the 20 particle case, there is always a guarantee of convergence for these variants. To have a better overview of the RMSE of the estimated states in each epoch, the corresponding plots to the MAE of the estimated states by different variants of the R-EKPFI are given in the Appendix (A.3).

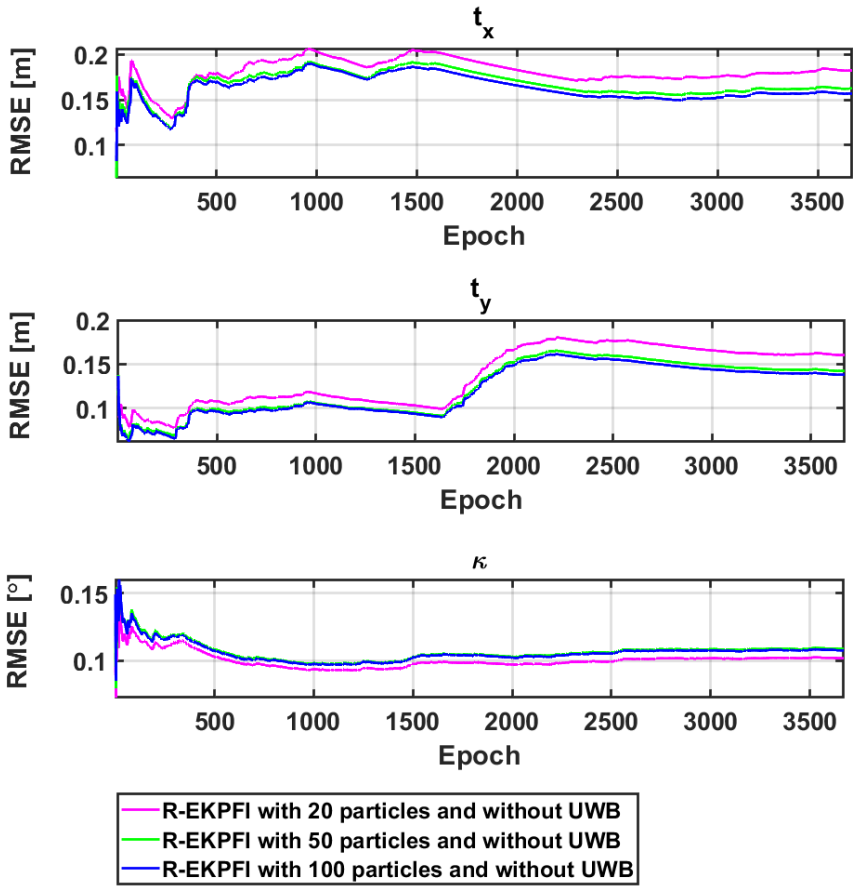

Figure 4.24: *Average accumulative RMSE of the estimated states over multiple runs by the R-EKPFI algorithm with various number of particles.*

Figure 4.25 shows the average precision of the estimated states over the multiple runs of the R-EKPFI variants. It can be seen that in cases with more particles the precision of the estimation is lower. Such an effect is attributed to the higher variation between the samples, which in turn leads to a lower precision of the estimated states. However, as can be seen from the figure, these precision values are not substantially lower than those of the R-EKPFI with 20 particles. This is due to the underlying KF step, which prevents the samples from varying considerably from each other. Note the area in the figure marked by red rectangles. Inspecting the simulated environment, it is noticed that during these epochs the vehicle is moving along the Y axis. This means that the building models are on the sides of the vehicle without any information along the driving direction. This leads to a lower precision for the t_y estimates. On the contrary, the t_x estimates have a higher precision due to the detected planes on both sides of the vehicle. The corresponding figure for the average precision of the versatile IEKF and the R-PFI compared to the R-EKPFI with 20 particles is given in the Appendix (A.3).

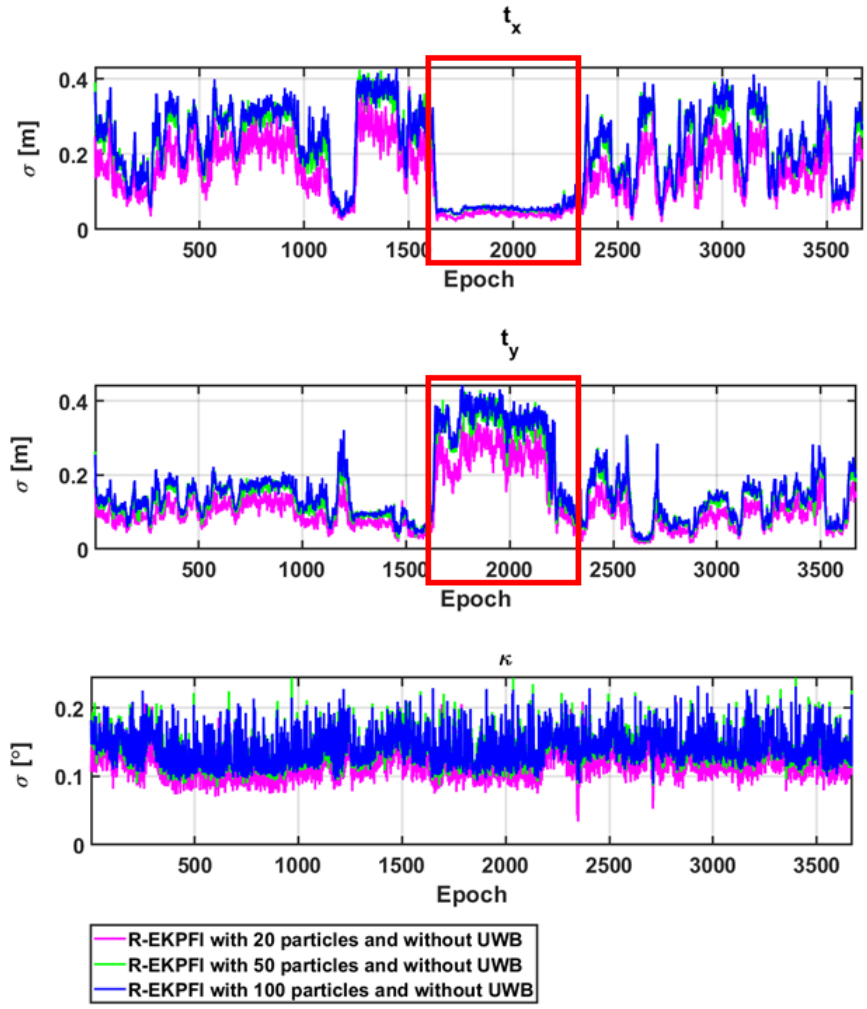

Figure 4.25: *Average precision of the estimated states over multiple runs by the R-EKPFI algorithm with various number of particles.*

In Figure 4.26, the estimated KLD measures between the versatile IEKF and the suggested PF-based methodologies is depicted. Note that in all the following plots related to the KLD measures, the scale of the y axis is considered similar to figures 4.21 and 4.22. This is done to give a general impression of the relative comparison between the obtained KLD measures. As mentioned before, no significance test can be applied to realize the degree of similarity or dissimilarity between the obtained posterior distributions. The current work suggests that keeping the scale of the plots similar helps to have a better interpretation of the results on a general level. Similar to Figure 4.21, the 1000 generated samples from the versatile IEKF estimations are taken as the reference to which the resampled particles from the other filters are compared. In this case as well, to avoid high computation times, the KLD measure is calculated every 100 epochs. As it can be seen, the posterior distributions of the R-EKPFI are more similar to the reference distribution due to having closer KLD measures to zero than those derived for the R-PFI estimations. It can be seen that increasing the number of the particles in the R-EKPFI has led to more similarity to the reference distribution. Such an effect is due to higher effect of the KF step on the R-EKPFI when the number of the particles increases.

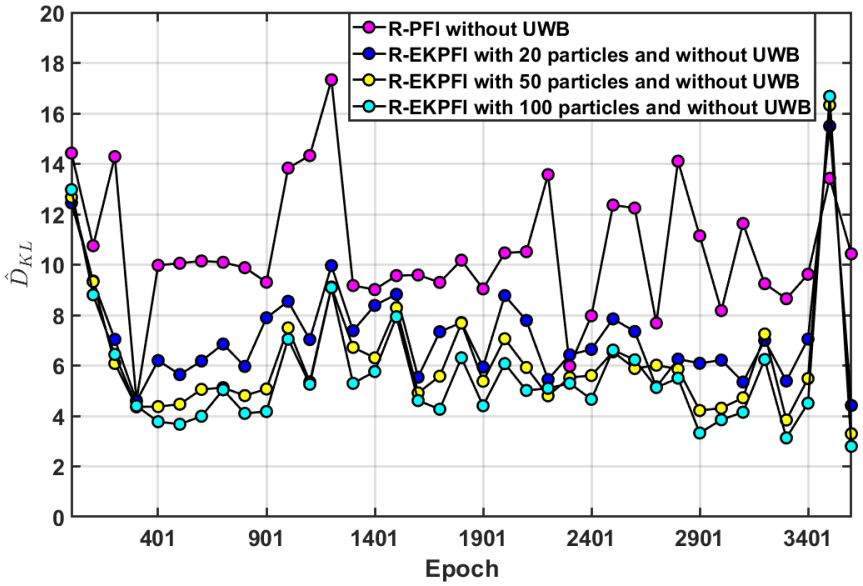

Figure 4.26: *Average of the KLD estimate* $\left(\hat{D}_{KL}\right)$ *between the versatile IEKF and the PF-based algorithms in each epoch over multiple runs.*

In Figure 4.27, similar to Figure 4.22, the resampled particles of the R-PFI are taken as a reference against which the resampled particles of the R-EKPFI variants are compared. It can be seen that the derived KLD measures for all these variants are in a similar range. However, as the number of particles is increased, the similarity between the resampled particles of the R-EKPFI and the reference particles increases.

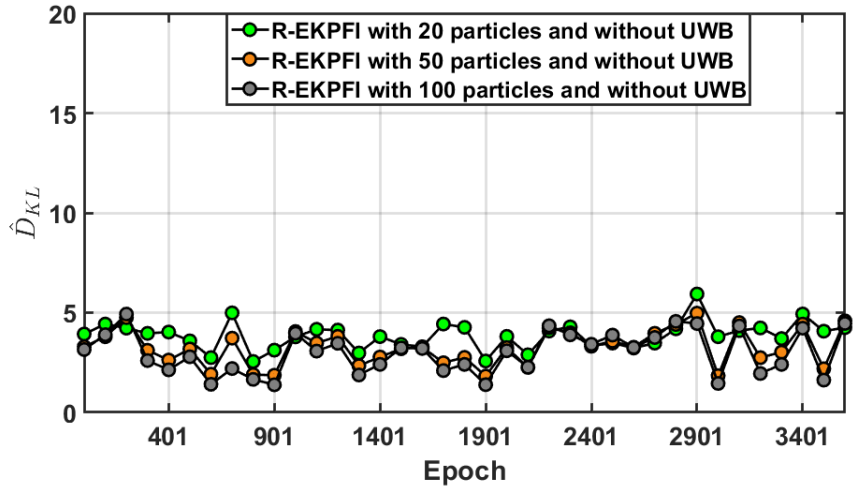

Figure 4.27: *Average of the KLD estimate* $\left(\hat{D}_{KL}\right)$ *between the R-PFI and the R-EKPFI with various number of particles in each epoch over multiple runs.*

A comparison of the average computation times shows that the R-EKPFI with 20 particles is about twice and three times faster than the cases with 50 and 100 particles, respectively. Such a comparison with the R-PFI shows that the R-EKPFI with 20, 50 and 100 particles is about 11, 6 and 4 times faster than the R-PFI.
According to the given results, it is concluded that the R-EKPFI is prone to divergence when the number of particles is too low (e.g. 10 as explained above). On the other hand, increasing the number of samples in the R-EKPFI causes the estimates to be influenced not only by the KF

framework, but also by the principle behind the PF. In the present work such an effect is called the *interdisciplinary effect*. In other words, the R-EKPFI is influenced by the characteristics of both PF and KF-based methods. Such an effect can become critical in cases with highly multimodal distributions and should be adequately addressed. However, as confirmed by the analysis results, the advantage of the R-EKPFI methodology is its ability to perform computations notably faster than the R-PFI framework. Such a feature is advantageous when it comes to georeferencing problems where computation time plays an important role. Furthermore, in the R-PFI framework, depending on the number of samples, the state estimation is strongly affected. Consequently, depending on the application, such an attribute can cause complications due to the unknown sufficient number of samples that can lead to reliable estimates. However, in the R-EKPFI framework, by modifying the particles using the observations, such an effect is markedly reduced, which highlights the benefit of this framework even more.

4.4.2 Real Environment

In this section, the results of the R-PFI, R-EKPFI and the versatile IEKF on the real environment explained in Section 4.3.2 are given. As in the simulated environment, 1000 particles are used for the R-PFI and different variants of 10, 20, 50 and 100 particles are used for the R-EKPFI. Furthermore, as explained in Section 4.3.2, the evaluation of the estimated t_x and t_y is done using the total station sensor. For the derived κ_k the provided data based on the ICP approach given by Axmann et al. (2023) is used. Additionally, to ensure the stability of the filters, similar to the simulated environment, the case study is analyzed 10 times. The following results are averaged over these multiple runs.

Figure 4.28 shows the absolute error of the estimated states using the different filters over multiple runs. To avoid confusion, only the results of the versatile IEKF, the R-PFI and the R-EKPFI with 20 particles are shown. Furthermore, as explained in Section 4.3.2, only 1500 epochs of the entire trajectory are analyzed by the filters. However, as shown in Figure 4.9, the ground truth data provided by the total station is only available for a part of this selected trajectory. Therefore, unlike the κ_k, the corresponding plots to the RMSE of t_x and t_y are not available for the entire trajectory. From the given plots in this figure, the considerably poor performance of the filters in estimating κ can be confirmed. The current work argues two reasons for such a result. First, as explained in Section 4.3, no κ values based on the IMU data are available in this case. Consequently, the predicted heading angle at each epoch cannot be modified using the IMU data. Second, as mentioned in Section 4.3.2, the design parameters for the adaptive kinematic model ($\boldsymbol{R}_k$ and k_b) are taken from the simulated environment. However, the current results indicate that the design parameters given in Table 4.2 cannot account for the lack of IMU data in each epoch. In such a case, it is necessary to perform a hyperparameter optimization or a detailed grid search to derive the optimal design parameters for the adaptive kinematic model that can best overcome such a data deficiency. In general, as explained earlier, the scanned data is divided based on its corresponding sampling rate. Then, at each filter time, the corresponding segment is used to estimate the states. However, in practice, although the segmented data is considered to be at one instance in time, it is obtained over a period of time. This means that the scanned data used to update the predicted states at a particular epoch are not captured exactly at that time. This aspect is highly dependent on the speed of the vehicle and may not be thoroughly overcome by optimizing the design parameters. To reduce such an effect, the vehicle speed must be taken into account while synchronizing the sensors in time and defining the filter time. However, this aspect is beyond the scope of this work and should be investigated in the future.

Furthermore, it can be seen that the R-PFI is prone to more divergence compared to the other two filters. This thesis asserts that this is due to the insufficient number of samples used by the R-PFI, in addition to the causes mentioned above.

In addition, it can be seen that in some epochs the estimated states by the filters have higher deviations compared to the other epochs. Inspection of the environment, similar to the simulated

case, has shown that such poor estimations occur on the curves and at the locations where the configuration of the LoD-2 models is not suitable. Based on the corresponding plots of t_x and $t_{y,k}$, it can be seen that the R-PFI and R-EKPFI can better withstand such situations.
Furthermore, from this figure it can be seen that for most of the epochs the UWB data has not improved the performance of the versatile IEKF. This is due to the sampling rate of these sensors, which according to Table 4.4 is considerably lower compared to the GNSS and the 3D scanner. Therefore, in contrast to the simulated environment, these data are only considered in some of the epochs. In a few epochs a sudden increase of the estimated RMSE values of t_x and t_y can be seen. The current work suggests that this is due to the uncertainty in the position of the secondary vehicles in these epochs. Since the position of these vehicles is considered to be deterministic, any uncertainty in their position can adversely affect the filters. In the case of the R-PFI, a random effect from these data can be seen. In other words, in some epochs the performance of the filter is improved by taking into account the UWB data, while in some other epochs it is worsened. It is claimed that the given reason about the deterministic position of the secondary vehicles affects the sensor fusion and consequently the resampling step of the R-PFI. In other words, the resampled particles are demanded to satisfy the observation model of the UWB data. In the case of high uncertainty in the position of the secondary vehicles, such a requirement causes the resampled particles to deviate from the ground truth. This thesis suggests that such an effect can be resolved by increasing the number of particles, at the cost of a substantial increase in computational time. Finally, it can be seen that the UWB data has no noticeable effect on the performance of the R-EKPFI. This further proves the effect of modifying the particles using the KF step. This, in combination with the PF principle, leads to the reduction of the negative effect of the UWB data resulting from the uncertain position of the secondary vehicles.

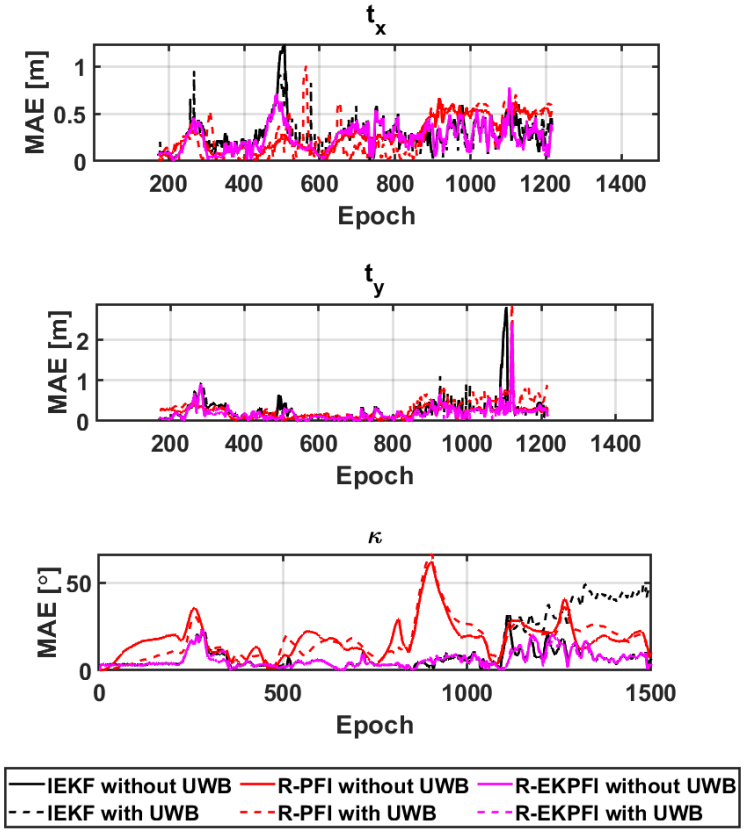

Figure 4.28: *Absolute error of the estimated states over multiple runs.*

Figure 4.29 shows the average precision of the estimated states using the versatile IEKF, the R-PFI, and the R-EKPFI. Note that the y axis is in logarithmic scale. This is due to the notably lower

precision values of the versatile IEKF compared to the other filters. Furthermore, in all three plots, the cases with the UWB data in R-PFI and R-EKPFI overlap with the plots obtained for the R-EKPFI when the UWB data are not considered. Therefore, they are not clearly visible. In the case of the versatile IEKF, the cases with and without the UWB data lead to close results. Therefore, their corresponding plots coincide.

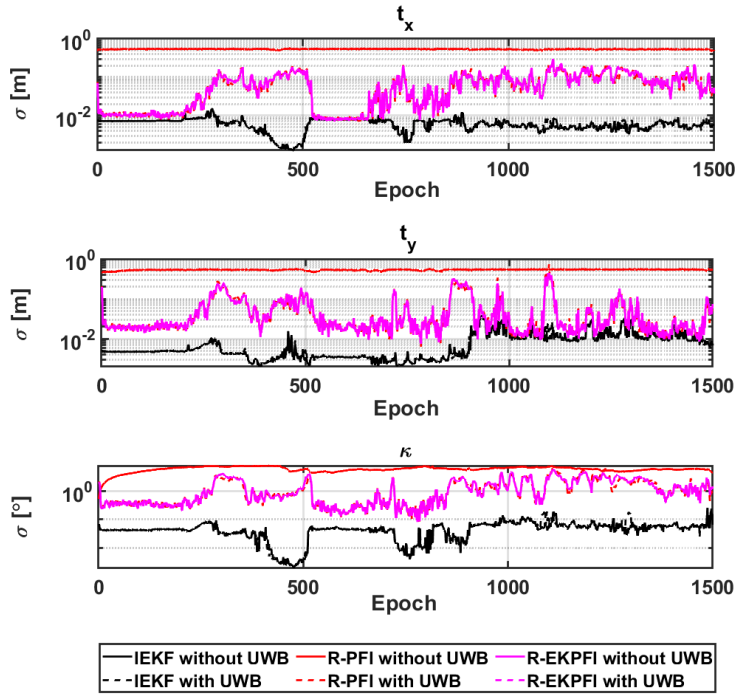

Figure 4.29: *Average precision of the estimated states over multiple runs runs.*

The analysis results show a precision in centimeters for the estimated t_x and t_y by this filter. The precision of the estimated κ is derived to be less than 0.5°. The reason for such small precision values is the assumption that the scanner data are uncorrelated. According to the law of variance propagation, such an assumption leads to small values for the precision values. To explain more, these small values result from inverting a large matrix $\left(\boldsymbol{O}_{k,c} + \boldsymbol{S}_{k,c}\right)^{-1}$ in the context of the versatile IEKF (line 9 that affects line 16). This leads to the inconsistency of the versatile IEKF, which is thoroughly investigated by Vogel (2020). Since this aspect is beyond the scope of this thesis, it will not be discussed further below.

In the case of the R-PFI, the precision of the filter increases when the UWB data are taken into account. Such a result further confirms the derived results from the MC simulations shown in Figure 4.17. The inclusion of the UWB data, and thus the fact that the resampled particles are in the same interval, increases the precision of the filter. In the R-EKPFI, due to the modification step by considering the observations, the particles fall in the same range. Consequently, considering or neglecting the UWB data does not considerably affect the resampled particles and therefore does not improve the precision. As can be seen, the assumption of uncorrelated scanned data has not affected the precision values in the R-EKPFI. This is despite the fact that in this filter the versatile IEKF is used to modify each particle. The current work asserts the reason to be the regeneration of each particle around the modified sample. In addition, the uncertainty of the detected planes is considered in the likelihood estimation step. In the versatile IEKF this uncertainty is not included. These two reasons cause the resulting precision values to be in a higher order of magnitude compared to those from the versatile IEKF.

Figure 4.30 shows the MAE of the estimated states using different numbers of particles in the R-EKPFI. Since the UWB data have no noticeable effect on the estimates, they are neglected for the results given in this figure. As can be seen, using a minimum and maximum of 10 and 100

particles, respectively, does not lead to a considerable change in the estimated states. However, based on the further investigations of the current work, it is found that the 10-particle variant is prone to instability and divergence when applied to the entire trajectory. This further confirms the investigations of the simulated environment.

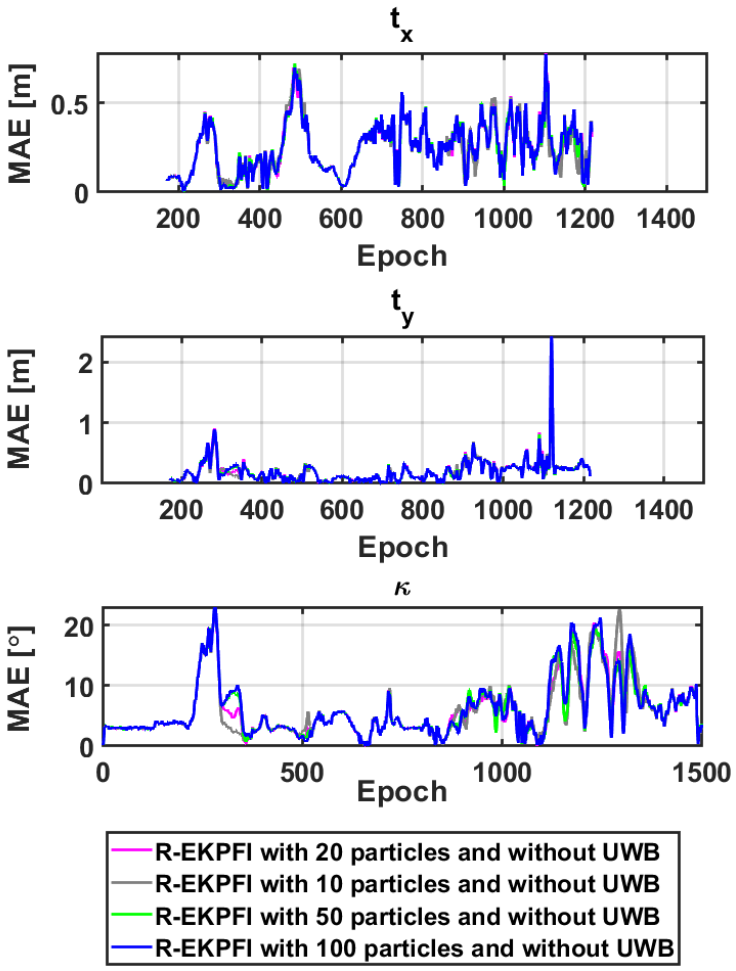

Figure 4.30: *Absolute error of the estimated states over multiple runs by the R-EKPFI algorithm with various number of particles.*

Figure 4.31 shows the corresponding average precision to the estimated states by the R-EKPFI filters with different particles. This figure further confirms that changing the particles within the R-EKPFI filter does not lead to significant changes in the results.

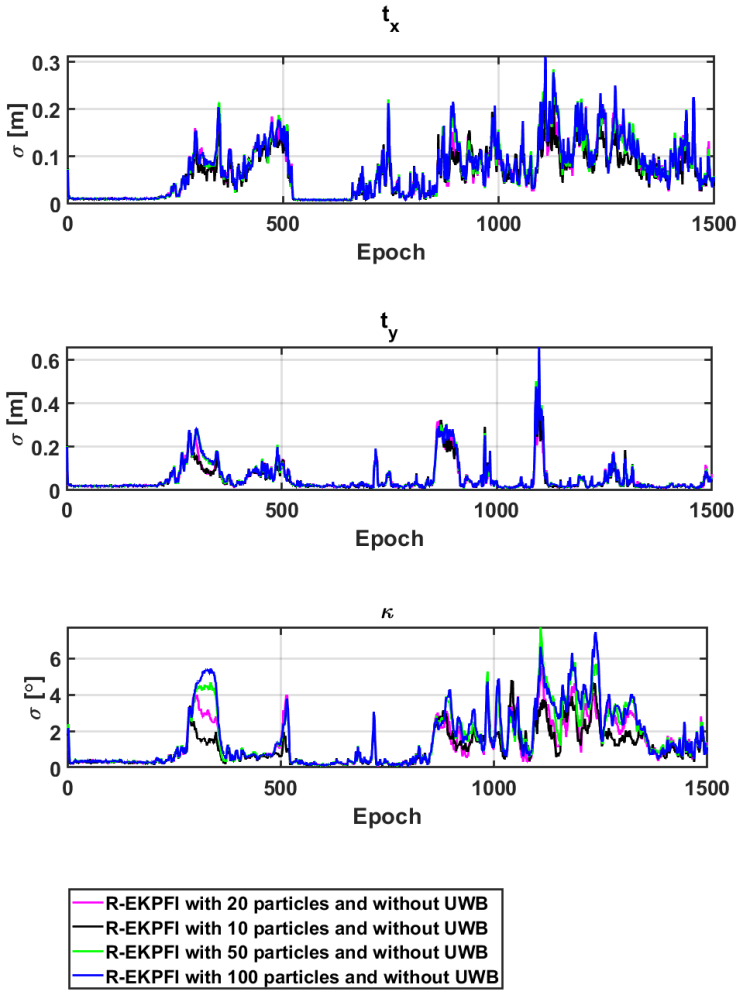

Figure 4.31: *Average precision of the estimated states over multiple runs by the R-EKPFI algorithm with various number of particles.*

Figure 4.32 shows the estimated KLD measures between the versatile IEKF – as a reference – and the R-PFI and R-EKPFI with 20 particles. Note that in all the following plots related to the KLD measures, the scale of the y axis is considered similar. As explained in Section 4.4.1, this is done to provide a better general overview of the obtained KLD measures. Furthermore, similar to the other KLD plots, the KLD measures are derived every 100 epochs to avoid high computational times. It can be seen that the posterior distributions derived from the R-EKPFI with and without the UWB data are more similar to those obtained from the versatile IEKF. The reason for the lack of substantial changes in the estimated KLD measures with and without the UWB data is the aforementioned lower sampling rate of this sensor compared to the others. To be more concise, the UWB data are considered only in some epochs. As confirmed by the simulated environment, even considering these data in every epoch does not lead to a considerable improvement of the results. Therefore, it is not unusual to obtain such results when considering these data only in some epochs.

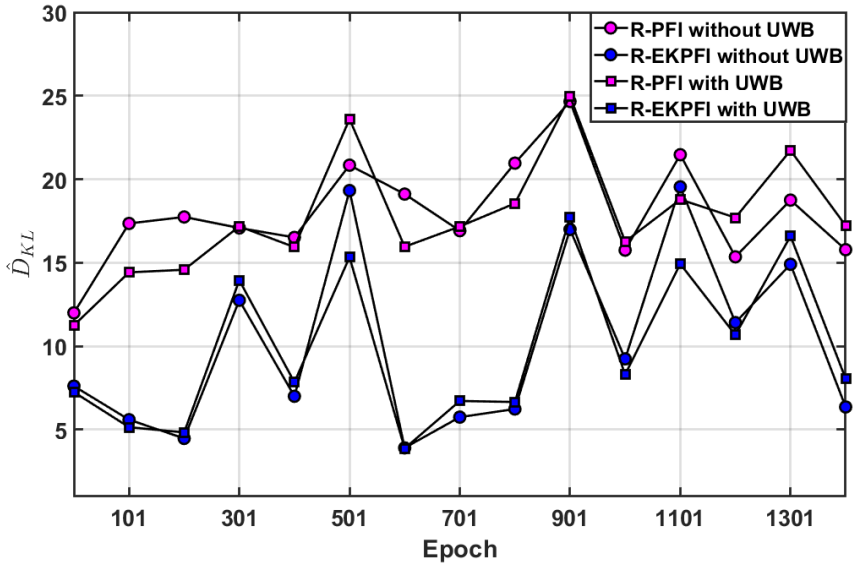

Figure 4.32: *Average of the KLD estimate* $\left(\hat{D}_{KL}\right)$ *between the versatile IEKF and the PF-based algorithms in each epoch over multiple runs.*

Figure 4.33 shows a comparison between the posterior distributions derived by the versatile IEKF – as a reference – and the R-EKPFI with different numbers of particles. As can be seen, changing the number of particles in the R-EKPFI does not lead to a considerable change in the KLD measures. These results also confirm the conclusion obtained on the basis of the average RMSE values that considering the UWB data only in some epochs is ineffective.

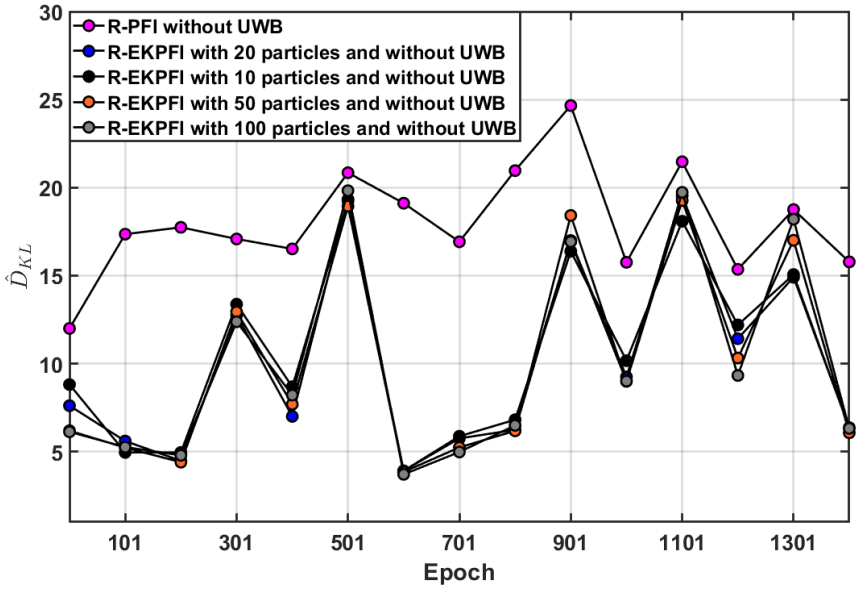

Figure 4.33: *Average of the KLD estimate* $\left(\hat{D}_{KL}\right)$ *between the versatile IEKF, the R-PFI and the R-EKPFI with various number of particles in each epoch over multiple runs.*

Figure 4.34 shows the KLD measures between the R-PFI as reference and the R-EKPFI with different numbers of particles. The general conclusion from this figure, similar to the simulated environment, is that the derived posterior distributions from these two filters are similar. However, when comparing the general pattern with figure 4.27 of the simulated environment, an opposite effect is observed. By increasing the number of particles, the dissimilarity between the posterior

distributions of R-PFI and R-EKPFI increases. Having a true posterior distribution, both in the simulated environment and in the real case study, helps to better evaluate the performance of the filters and to draw a more solid conclusion in this regard. Finally, a comparison between the computation times of the filters for 1500 epochs shows that the 20, 50 and 100 particles perform the analysis 5, 4 and 3 times faster than the R-PFI, respectively. When using 10 particles, the computation time is in the same range as the case of 20 particles.

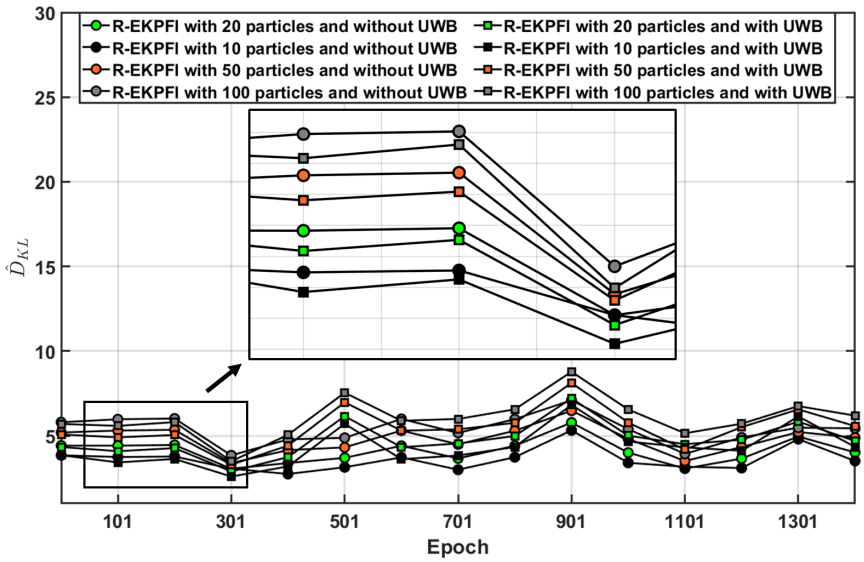

Figure 4.34: *Average of the KLD estimate* $\left(\hat{D}_{KL}\right)$ *between the R-PFI and the R-EKPFI with various number of particles in each epoch over multiple runs.*

4.4.3 Discussion

As a general conclusion, applying the filters to both a simulated environment and a real-world application allowed the strengths and weaknesses of each filter to be recognized in different situations. As explained, the simulated case is used to simplify the real world in such a way that unexpected situations causing complications are avoided. The purpose of using this environment was to verify the ability of the developed PF-based frameworks to estimate the states accurately. In this setting, data from all sensors were available in each epoch. These data sets included information from GNSS, IMU, UWB and 3D scanner. Furthermore, the sampling rates of the sensors were configured to obviate the need for time synchronization. This is the most important aspect that was not fulfilled in the real scenario. In the latter case, a time synchronization step was required due to different sampling rates that caused different time stamps for the data of each sensor. In real-world applications, the time synchronization step is crucial and must be performed carefully. Any error in this step will directly affect the performance of the filters. Depending on the application, there are different ways to synchronize the sensors in time. In this work, this was done by first defining a filter time based on the scanned data. Subsequently, corresponding sensor data were matched to this filter time, based on the closest available time stamps. Due to the different time stamps of each data point, the considered data in each epoch still have time shifts with respect to each other. Therefore, if the kinematic model in the prediction step and the design parameters of the filters are not suitable, it can adversely affect the performance of the filters.
Aside from the time-synchronization aspect, using the sensors in practice can lead to unavailable or unreliable data due to practical problems. For example, as explained in this thesis, no IMU data for estimating the heading angle was available in the real-world scenario. This was due to the unreliable obtained κ angles from this sensor. Additionally, due to different sampling rates, the UWB data was only available in some of the epochs. However, according to the results of the filters in the simulated

environment, it was confirmed that the UWB data do not have a strong impact on the estimations. Consequently, their absence in some of the epochs in the real-world scenario was not critical. In contrast, the unavailable IMU data proved to have a negative effect on the estimated headings by the different filters. Such an effect was confirmed to be most evident in the case of R-PFI. In general, as stated throughout the thesis, the main purpose of the information-based georeferencing by means of the versatile IEKF and the developed PF-based frameworks is to overcome unreliable or unavailable GNSS and IMU data. However, to reach such an integrity level, it is essential to select the design parameters in a way that such a data deficiency can be well overcome. To be more concise, the current frameworks of these filters required a careful setup depending on each application. As stated in this thesis, the design parameters related to the kinematic model in the real-world application are directly taken from the simulated environment. Investigating the results, it was realized that doing so does not lead to optimal results. It was concluded that as much as these two scenarios have common characteristics, there are still considerable differences that their effect should be considered in the design parameters. Of such effects, the different sampling rate of the sensors can be mentioned.

The performance of the filters in the two case studies was evaluated from four aspects. First, the accuracy of their estimations was examined by obtaining the RMSE values. Second, the precision of these estimates over time was analyzed. Then, the KLD measure between the different filters was used to explore the similarity of the obtained posterior distributions. Finally, the time required by each filter was characterized as an important feature for its suitability in real-time applications. Considering the above criteria, it was realized that in the simulated case, the R-EKPFI framework has the best performance in terms of accuracy and precision. On the contrary, it was realized that the versatile IEKF has a considerable variation in the RMSE measures over time. The precision of this filter was found to be at the same level as that of the R-EKPFI. In the case of the R-PFI, the RMSE measures were noted to have less change compared to the versatile IEKF. However, the precision of the estimates was high. By inspecting the simulated environment, it was realized that the versatile IEKF does not detect any planes in some of the epochs. Therefore, the estimations in those epochs were based on other sensors than the 3D scanner. Such a performance – which was also observed for the curves of the trajectory – was shown to be the reason for the inaccurate estimations of this filter in some epochs. For R-EKPFI and R-PFI, it was shown that at least one plane was detected in each epoch, which led to a constant consideration of the additional information from the environment over time.

In the case of the real environment, it has been shown that the estimated 2D positions by the filters are in the same level of accuracy. However, in some epochs these estimates by the versatile IEKF are worse, which is presumed to be due to the configuration of the detected planes in these epochs. Furthermore, similar to the simulated environment, the R-PFI was shown to have the worst precision. Unlike the simulated environment, the precision of the R-EKPFI was shown to be worse than the versatile IEKF. However, due to the inconsistency of the versatile IEKF proven by Vogel (2020), the current work does not consider this as a negative characteristic for the R-EKPFI. More precisely, the uncorrelated structure of the VCM of the observations causes the resulting precision values of the versatile IEKF to be unrealistically small. This leads to an inconsistent characteristic of the filter, which is more evident in the real-world application of this thesis than in the simulated case study. However, it has been shown that the uncertainty of the R-EKPFI is of a similar magnitude as in the simulated case study. Investigating the KLD measures in both applications confirmed that the posterior distributions derived by the PF-based frameworks are not similar to the Gaussian distributions derived by the versatile IEKF. This conclusion was reached by making a relative comparison between the filters and noting that the KLD measures obtained were greater than zero. However, this conclusion lacks sufficient analytical support due to the absence of the true posterior distributions. The current work contends that a non-zero KLD measure between two filters is expected. This is due to the different setup of each filter, which is exclusive. However, if the resulting RMSE values are in the same order of magnitude and the posterior distributions

are similar, it is not unrealistic to expect KLD measures to be close to zero. Based on such an attitude, and by comparing the KLD measures with the obtained RMSE values of different filters, it is claimed that the posterior distributions of the PF-based approaches are different than those that are derived by the versatile IEKF. Further analysis is required to reach a firm conclusion in this regard.
And lastly, both case studies confirmed the significantly faster performance of the R-EKPFI compared to the other two filters.
In conclusion, the current work claims that the R-EKPFI methodology is a more suitable state estimator compared to the versatile IEKF and R-PFI for several reasons. First, it can always take into account a minimum of additional information from the environment. Second, the precision of its estimates is realistic. In addition, it does not necessarily have to yield similar Gaussian distributions to the versatile IEKF, which can be a confirmation of its potential ability to handle multi-modal distributions. Finally, its fast performance makes this framework a potential candidate for real-time applications such as autonomous localization.

5 Conclusion and Future Work

5.1 Summary

The current work focuses on developing a reliable and efficient PF-based methodology capable of handling both explicit and implicit observation models. The PF framework allows for solving filtering problems without imposing any prior assumptions, thus more effectively addressing the global uncertainties inherent in real-world applications than KF-based approaches. In traditional KF-based approaches, many of these uncertainties are eliminated by adopting simplifying assumptions. The primary motivation for developing such a filter is the need for reliable georeferencing of MSSs in urban environments, which has gained significant interest in the last decades. For autonomous systems such as cars or drones in outdoor environments, georeferencing can usually be accomplished using GNSS and IMU data alone. In such cases, a framework capable of handling only explicit observation models is sufficient. However, these data may not always be available or reliable, and thus additional sensors and information may be needed to compensate for the errors. Incorporating additional sensor data may require the use of implicit observation models. In such instances, a filtering framework free from requirements regarding the nature of the underlying observation models proves invaluable. Consequently, any useful data can be included in the filter, which in turn leads to a more reliable state estimation.
The proposed PF framework, known as the R-EKPFI, is designed to handle a large number of observations that comply with their corresponding implicit observational models. The development process of the methodology is explained in three parts in Chapter 3. First, the work introduces how implicit observation models can be accommodated within the PF framework. Implicit residuals are employed to derive the importance weight of the particles. This resulting strategy, referred to as PFI, can suffer from divergence due to the presence of a large amount of data. Second, a strategy to mitigate this divergence is discussed. This approach considered the second novelty of the work, identifies observations causing divergence in the PFI framework. The IQR method is employed to detect these problematic observations as outliers. The resulting framework, named R-PFI, requires high computational time due to the need for a sufficient number of particles. The third novelty addresses this computational burden. The R-EKPFI modifies the particles within the R-PFI by incorporating the available observations in a KF step. This modification allows for using fewer particles, leading to a significant reduction in the computational time of the filter. However, it is important to note that the applications considered in this thesis were not intended for real-time processing. Therefore, the current framework is not yet optimized for real-time applications. Nevertheless, transitioning to real-time performance is an achievable next step. Achieving such performance positions the filter as a potential candidate for the autonomous driving industry, where successful implementation and adherence to safety standards could lead to significant advancements. Also, the ability to handle uncertainties in urban localization positions the framework as a potentially transformative solution for the autonomous driving industry, addressing a key challenge that has been actively pursued for decades.
To demonstrate the effectiveness of the proposed R-EKPFI methodology and to expose the limitations of the PFI and R-PFI frameworks, a straightforward numerical example related to model parameter estimation is employed in Chapter 3. In this example, two factors – constraints and improper initialization – are also considered to more comprehensively evaluate the performance of each filter. The consideration of these two aspects stems from the fact that constraints provide supplementary information that aids in state estimation. Hence, exploring how to effectively integrate them into the filter is crucial. Moreover, in scenarios where initialization is flawed, as often occurs

in real-world applications with high initialization uncertainty, the filter must retain the capability to accurately estimate states. Compared to the ground truth, the results of the analysis show that the PFI diverges over time due to the presence of outliers within the likelihood estimation step. Furthermore, the R-EKPFI is shown to estimate states approximately three times faster than the R-PFI algorithm, confirming its superior computational efficiency. Moreover, the derived results are compared with the results obtained from the versatile IEKF, which further confirms the better performance of the R-EKPFI algorithm.
The fourth novelty of this work is adapting the proposed R-EKPFI for MSS georeferencing. The core idea leverages surrounding environmental information to overcome the limitations of the GNSS and IMU sensors in urban areas. This approach focuses on integrating data from 3D scans, GNSS, IMU, and UWB sensors. The adapted R-EKPFI, tailored for MSS georeferencing, comprises two primary components: the first is the prediction step of the filter, proposed to be performed using an adaptive kinematic model; the second is the fusion of data from multiple sensors, which is crucial for assessing likelihood. The effectiveness of the R-EKPFI is verified in two scenarios: a simulated environment and a real-world urban setting. To demonstrate the advantages of the R-EKPFI, a comparison was made with the versatile IEKF and the R-PFI developed in this thesis. Comparing the results with the ground truth and investigating the precision of each filter reveal that the R-EKPFI outperforms both the R-PFI and the versatile IEKF, particularly in handling highly nonlinear paths, such as curves, and in scenarios where environmental information is limited. Notably, the accuracy of the estimated 2D positions with R-EKPFI ranges from centimeter to decimeter levels, validating its applicability in fields like autonomous vehicle localization. Furthermore, the better results of this framework in the real case scenario, even when IMU data are not available, confirm its capability to deal with data deficiency of sensors. Such a feature is of high importance due to the possibility of having data gaps or unavailability of sensor data in real-world applications. Additionally, this thesis compares the posterior distributions obtained through the proposed PF-based methods against those from IEKF using the KLD metric. The results of such an investigation in the analyzed cases show a dissimilarity between the PF-based methods and the versatile IEKF. Such a result further confirms the necessity of using a filtering framework where simplifying assumptions, such as the Gaussianity of the states, do not form the basis of the analysis. This is helpful in applications such as autonomous vehicle localization, where a multi-modal posterior distribution is more realistic than a uni-modal one. In these cases, knowing the probabilistic aspect of the estimated states is essential to realize the level of confidence that can be placed in the georeferencing solutions.

5.2 Outlook

Despite the proper performance of the developed R-EKPFI methodology, several aspects still require refinement.
The R-EKPFI framework relies heavily on detecting outliers in implicit residuals, using the IQR method. This approach is effective only when the symmetry of the implicit residuals is ascertainable. In all cases investigated in this work, implicit residuals were assumed to exhibit symmetric distribution, a premise considered reasonable for practical purposes. However, should this underlying assumption not hold, the application of the IQR method must be adapted based on the actual distribution of the residuals. It is, therefore, suggested that future work explores outlier detection for various distributions, especially those that are non-symmetric, such as skewed or bimodal distributions, including but not limited to Poisson, exponential, and Skewed Generalized T distributions.
Additionally, the current georeferencing sensor fusion strategy assigns equal weights to all sensors. However, it is proposed to develop a strategy that incorporates both sensor accuracy and particle dispersion. The decision to not base sensor weights on accuracy in the current work aims to retain information from sensors with lower accuracy. This approach mitigates the degeneracy problem in the proposed PF-based methods arising from insufficient number of samples, a problem

potentially more critical in the case of R-EKPFI, where the number of particles is inherently limited. Nonetheless, it is also important to consider the accuracy of the sensors; it aids in compensating for unreliable sensor data, while increasing the impact of accurate observations, optimizing the reliability and robustness of the state estimates.

In georeferencing, it is crucial to incorporate as much additional environmental information as possible. Such data aids in leveraging existing knowledge within the filter, thus contributing to more reliable state estimation.

However, since incorporating extensive data often increases computational demands, it is advisable to focus on observations that are not large in size. For example, position data extracted from pedestrians serves as valuable input, playing a crucial role in likelihood estimation and leading to more reliable state estimations.

When considering additional information such as pedestrians, it is essential to establish a connection between the MSS and them. This parallels the approach used in this dissertation for secondary vehicles, which have been considered using UWB data. However, unlike this work, it is crucial to account for the uncertainty associated with such observations within the R-EKPFI filter. Neglecting this uncertainty, as indicated by the analysis results of the current work, could lead to unimproved state estimation.

To address this, employing a parallel PF-based or KF-based filter for the state estimation of such secondary nodes is proposed. This approach facilitates the establishment of a dynamic network comprising the primary MSS and its surrounding dynamic objects (e.g., other MSSs or pedestrians). Consequently, by sequentially estimating all states, the configuration between the nodes of the network can be preserved, allowing for more effective utilization of additional information provided by neighboring objects.

The assignment algorithm used in this work relies on thresholds derived from a hyperparameter optimization process, making the filters application-dependent. Such a parameter selection process introduces an undesired preprocessing step to the filters in georeferencing applications. Additionally, coarse hyperparameter optimization enhances the uncertainty of the filters due to possible misalignments arising from non-optimal assignment thresholds. Therefore, exploring an assignment strategy that is either independent of such parameter selection process or minimizes its influence is suggested.

Similarly, the adaptive kinematic model proposed in this work includes a design parameter, (k_b), necessitating derivation through hyperparameter optimization. A suboptimal choice of this parameter leads to an incorrect estimation of the control inputs, which in turn negatively affects the prediction step. Using direct measurements, such as odometer and IMU data in vehicles, can provide a more reliable estimation of movement, thus eliminating the need for parameter optimization.

Moreover, the proposed information-based georeferencing relies on the use of LoD-2 city models. However, the uncertainties inherent in these models, e.g. the generalization error, introduce unavoidable uncertainties in the estimated states. To address this, information extracted from these models can be treated as additional states within the filtering procedure. While incorporating this information as additional states within the filter could improve accuracy, it also increases computational time. Therefore, utilizing sources of environmental information that are more reliable than LoD-2 models is suggested, such as level of detail 3 (LoD-3) models, which are more detailed.

Finally, to confirm the capability of the developed R-EKPFI framework, further exploration into real-world applications is essential. Each application has an exclusive setup. It is essential to verify the applicability of the algorithm to new setups. This gives the possibility to extend its use, which in turn increases its reliability.

A Appendix

A.1 Additional Results of the Plane Estimation

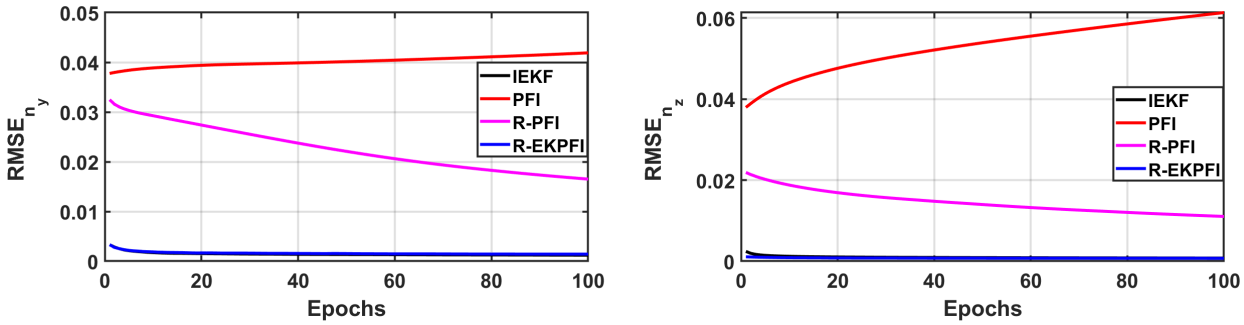

Figure A.1: *Average accumulative RMSE of the estimated n_y (left plot) and n_z (right plot) over 50 MC runs.*

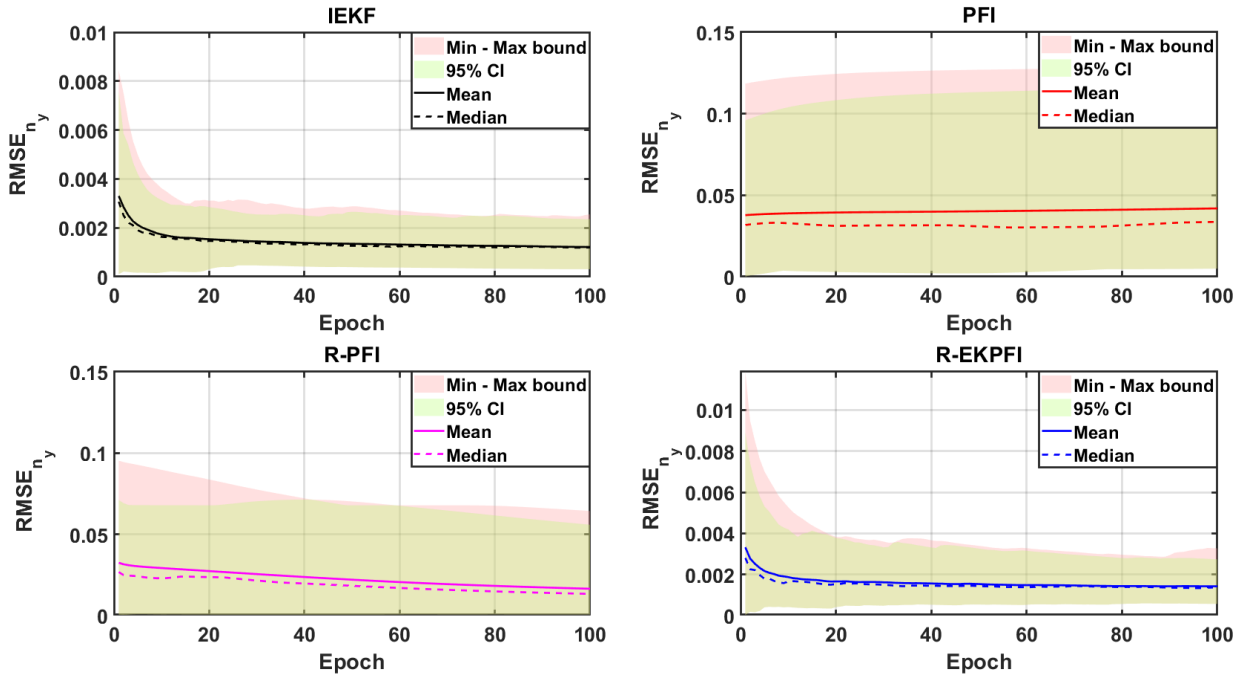

Figure A.2: *Statistical values of the accumulative RMSE of the estimated n_y over 50 MC runs.*

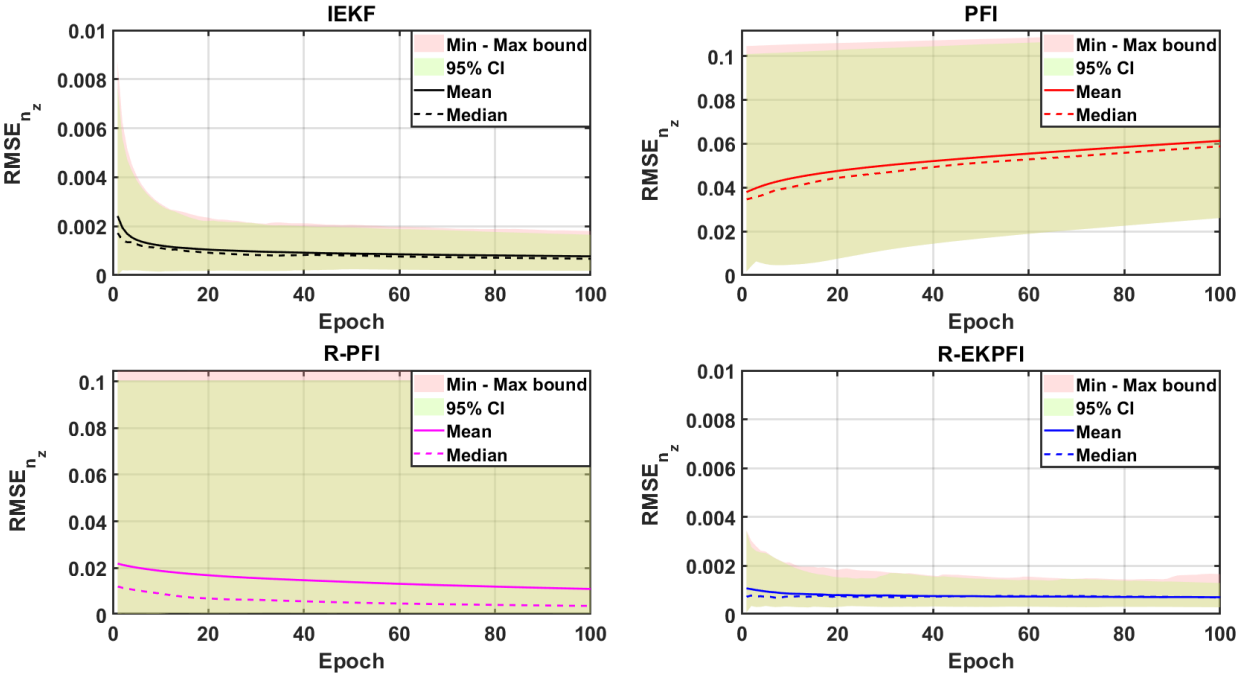

Figure A.3: *Statistical values of the accumulative RMSE of the estimated n_z over 50 MC runs.*

Table A.1: *Statistical values of the accumulative RMSE of the estimated n_y in the last epoch over the 50 MC runs by means of the different filters. Red and green colors indicate the minimum and maximum values in each column, respectively.*

	$RMSE_{n_y}$					
	Min	Max	Mean	Median	↑ CI (95%)	↓ CI (95%)
versatile IEKF	$3.07 \cdot 10^{-4}$	0.0026	0.0012	0.0012	0.0024	$3.07 \cdot 10^{-4}$
PFI	0.0050	0.1290	0.0419	0.0336	0.1163	0.0050
R-PFI	$6.29 \cdot 10^{-4}$	0.0643	0.0165	0.0133	0.0559	$6.29 \cdot 10^{-4}$
R-EKPFI	$5.67 \cdot 10^{-4}$	0.0033	0.0014	0.0014	0.0027	$5.67 \cdot 10^{-4}$

Table A.2: *Statistical values of the accumulative RMSE of the estimated n_z in the last epoch over the 50 MC runs by means of the different filters. Red and green colors indicate the minimum and maximum values in each column, respectively.*

	$RMSE_{n_z}$					
	Min	Max	Mean	Median	↑ CI (95%)	↓ CI (95%)
versatile IEKF	$1.78 \cdot 10^{-4}$	0.0018	$7.73 \cdot 10^{-4}$	$6.71 \cdot 10^{-4}$	0.0017	$1.78 \cdot 10^{-4}$
PFI	0.0261	0.1118	0.0613	0.0588	0.1093	0.0261
R-PFI	$5.55 \cdot 10^{-4}$	0.1003	0.0110	0.0038	0.0935	$5.55 \cdot 10^{-4}$
R-EKPFI	$3.05 \cdot 10^{-4}$	0.0017	$7.16 \cdot 10^{-4}$	$7.06 \cdot 10^{-4}$	0.0013	$3.05 \cdot 10^{-4}$

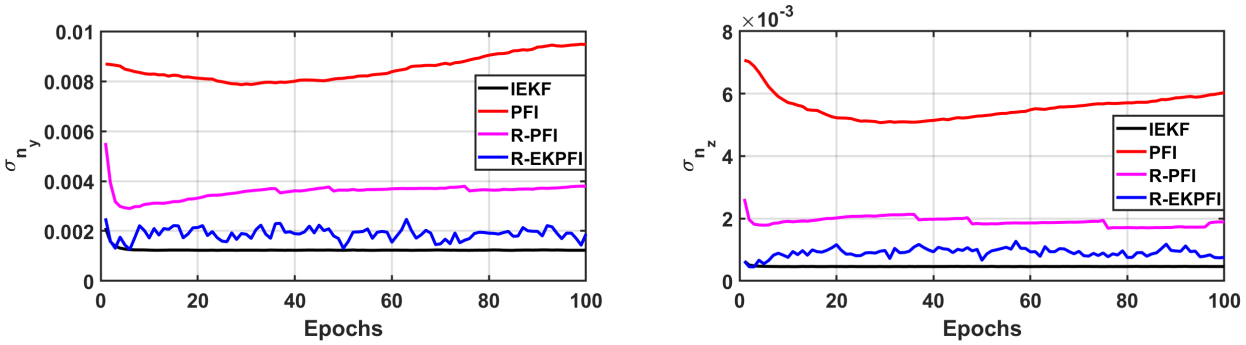

Figure A.4: *Average precision of the estimated n_y (left plot) and n_z (right plot) over 50 MC runs.*

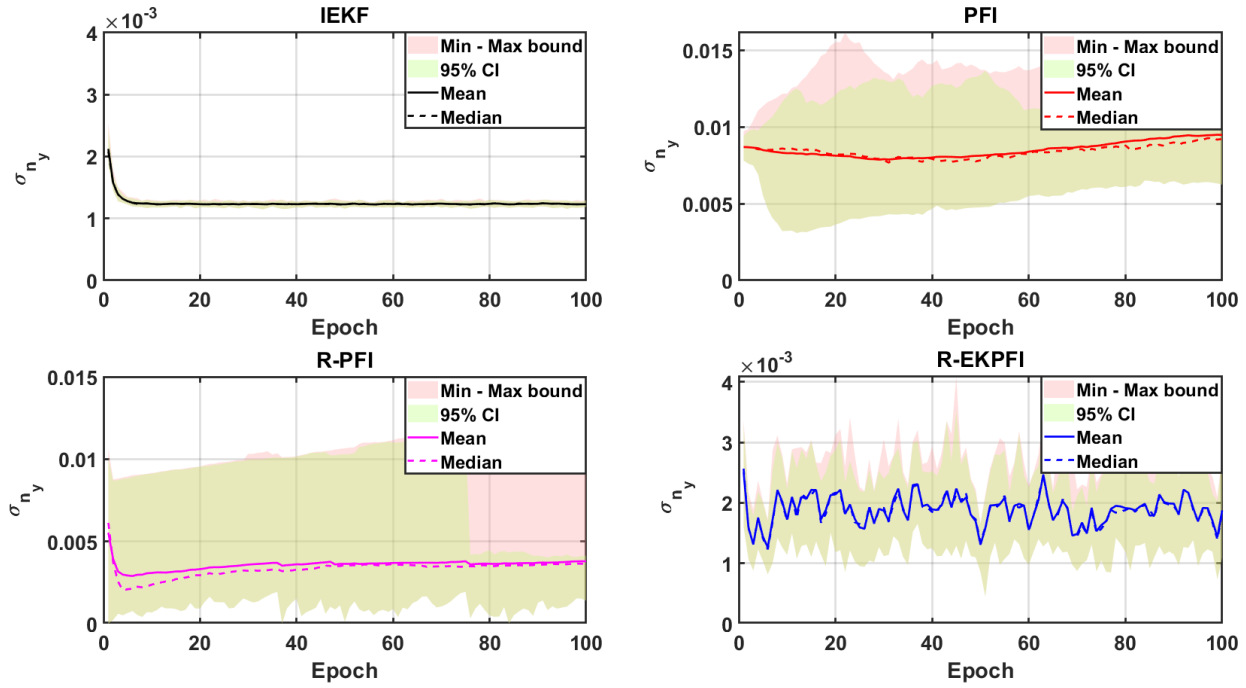

Figure A.5: *Statistical values of the precision of the estimated n_y over 50 MC runs.*

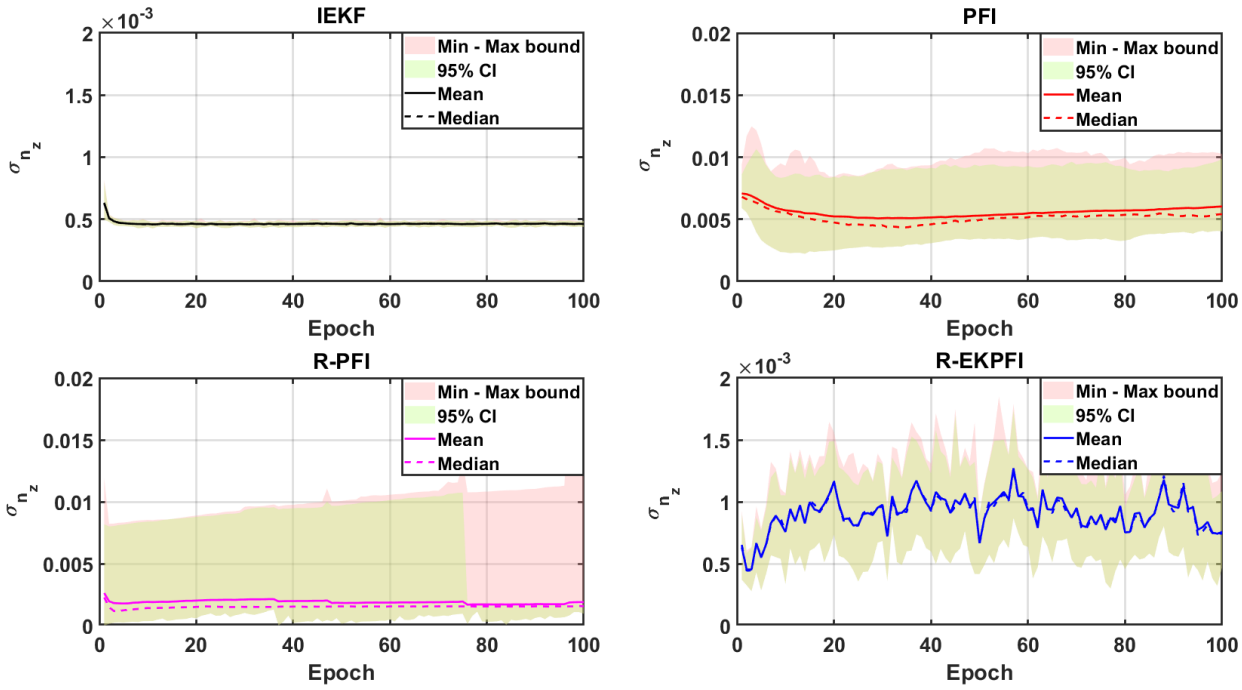

Figure A.6: *Statistical values of the precision of the estimated n_z over 50 MC runs.*

Table A.3: *Statistical values of the precision of the estimated n_y in the last epoch over the 50 MC runs by means of the different filters. Red and green colors indicate the minimum and maximum values in each column, respectively.*

	σ_{n_y}					
	Min	Max	Mean	Median	↑ CI (95%)	↓ CI (95%)
versatile IEKF	0.0012	0.0013	0.0012	0.0012	0.0013	0.0012
PFI	0.0062	0.0131	0.0095	0.0092	0.0128	0.0062
R-PFI	0.0014	0.0129	0.0038	0.0036	0.0041	0.0014
R-EKPFI	0.0011	0.0028	0.0019	0.0018	0.0027	0.0011

Table A.4: *Statistical values of the precision of the estimated n_z in the last epoch over the 50 MC runs by means of the different filters. Red and green colors indicate the minimum and maximum values in each column, respectively.*

	σ_{n_z}					
	Min	Max	Mean	Median	↑ CI (95%)	↓ CI (95%)
versatile IEKF	$4.38 \cdot 10^{-4}$	$4.86 \cdot 10^{-4}$	$4.62 \cdot 10^{-4}$	$4.61 \cdot 10^{-4}$	$4.86 \cdot 10^{-4}$	$4.38 \cdot 10^{-4}$
PFI	0.0040	0.0103	0.0060	0.0054	0.0098	0.0040
R-PFI	$9.54 \cdot 10^{-4}$	0.0187	0.0019	0.0016	0.0017	$9.54 \cdot 10^{-4}$
R-EKPFI	$4.69 \cdot 10^{-4}$	0.0012	$7.57 \cdot 10^{-4}$	$7.41 \cdot 10^{-4}$	0.0011	$4.69 \cdot 10^{-4}$

A.2 Additional Results of the Information-Based Georeferencing for the Simulated Case Study - Setup 1

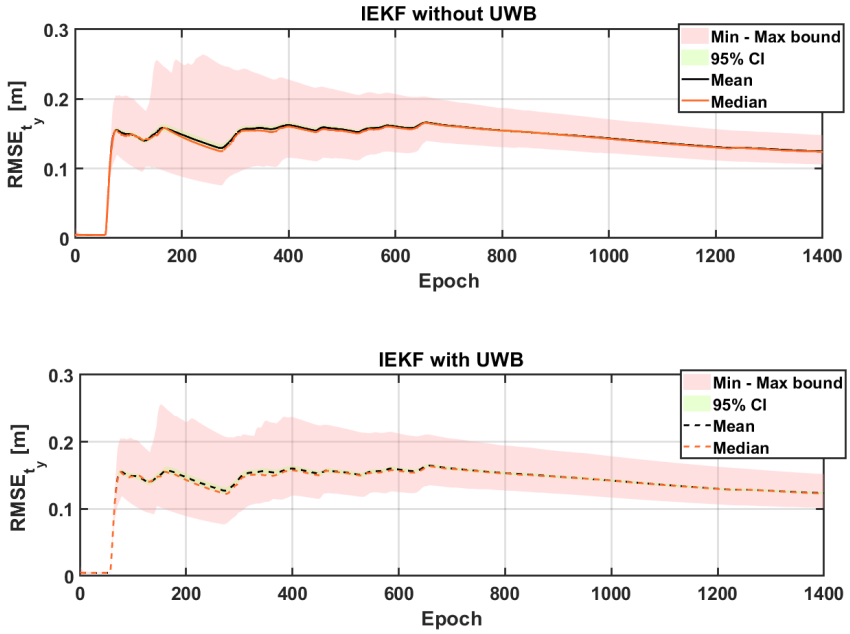

Figure A.7: *Statistical values of the accumulative RMSE of the estimated t_y by the versatile IEKF over MC runs.*

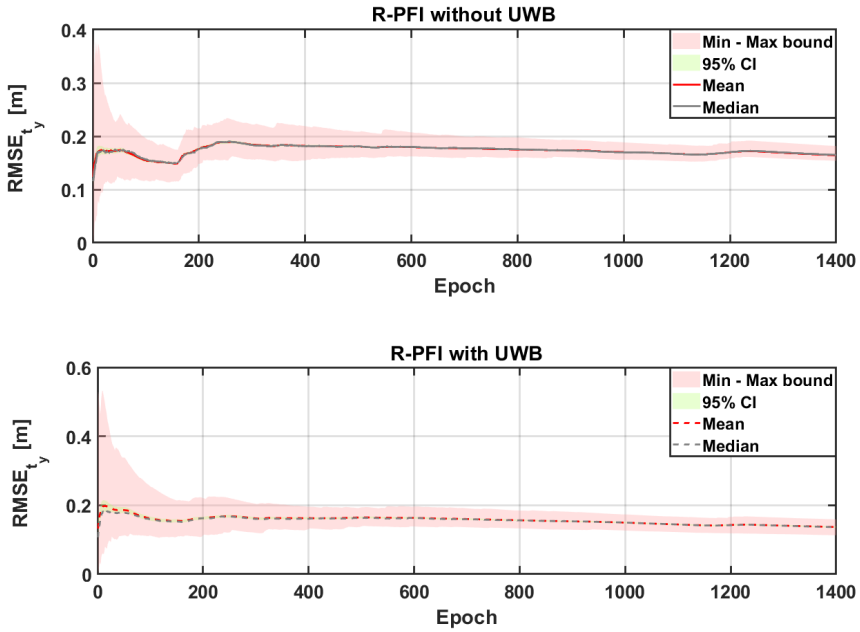

Figure A.8: *Statistical values of the accumulative RMSE of the estimated t_y by the R-PFI over MC runs.*

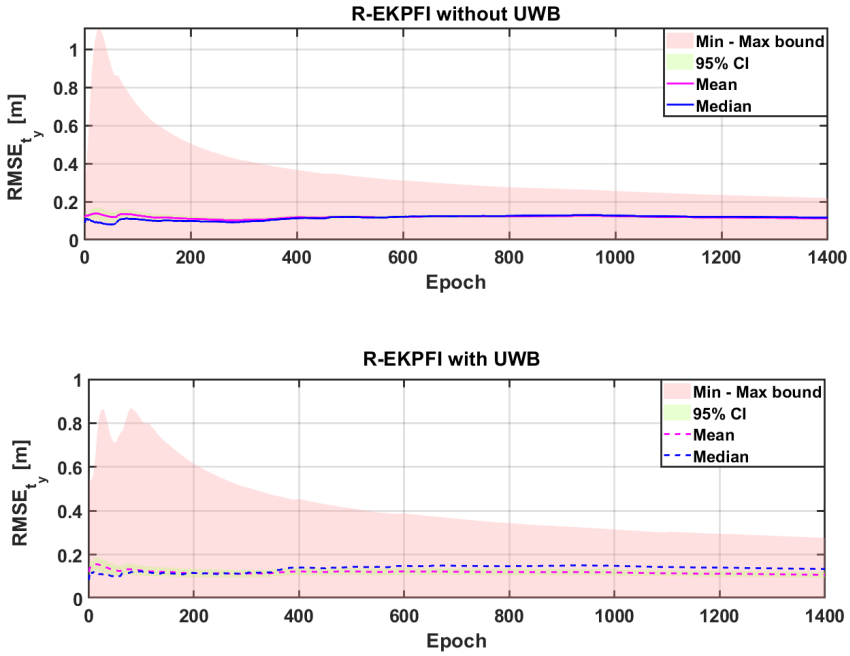

Figure A.9: *Statistical values of the accumulative RMSE of the estimated t_y by the R-EKPFI over MC runs.*

Table A.5: *Statistical values of the accumulative RMSE of the estimated t_y in the last epoch over the MC runs by means of the different filters. Red and green colors indicate the minimum and maximum values in each column, respectively.*

	$RMSE_{t_y}$ $[m]$					
	Min	Max	Mean	Median	↑ CI (95%)	↓ CI (95%)
versatile IEKF without UWB	0.1059	0.1481	0.1245	0.1239	0.1264	0.1227
versatile IEKF with UWB	0.1015	0.1519	0.1236	0.1228	0.1254	0.1218
R-PFI without UWB	0.1537	0.1816	0.1642	0.1648	0.1652	0.1632
R-PFI with UWB	0.1129	0.1589	0.1368	0.1364	0.1385	0.1351
R-EKPFI without UWB	0	0.2197	0.1123	0.1167	0.1192	0.1054
R-EKPFI with UWB	0	0.2764	0.1078	0.1343	0.1207	0.0950

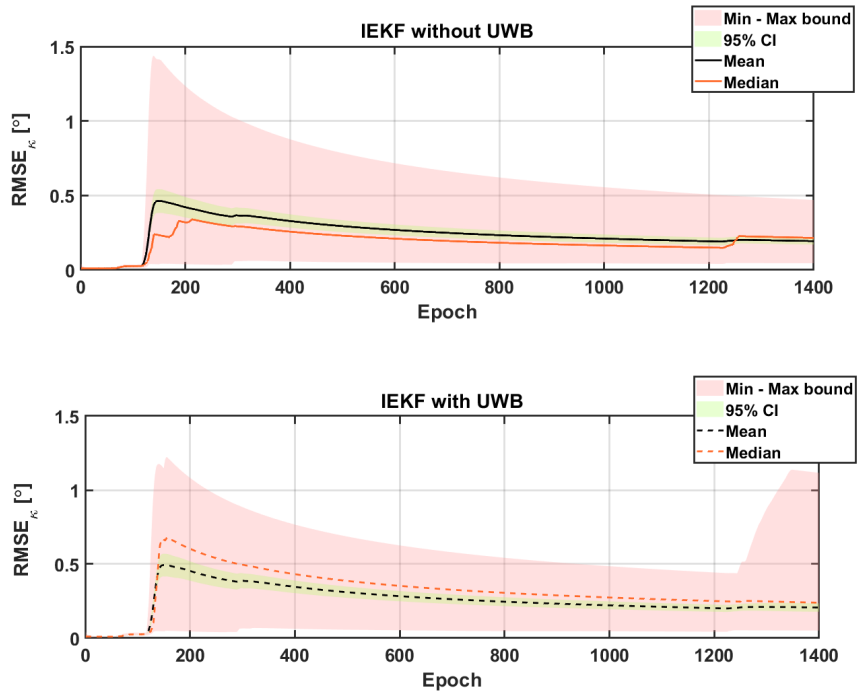

Figure A.10: *Statistical values of the accumulative RMSE of the estimated κ by the versatile IEKF over MC runs.*

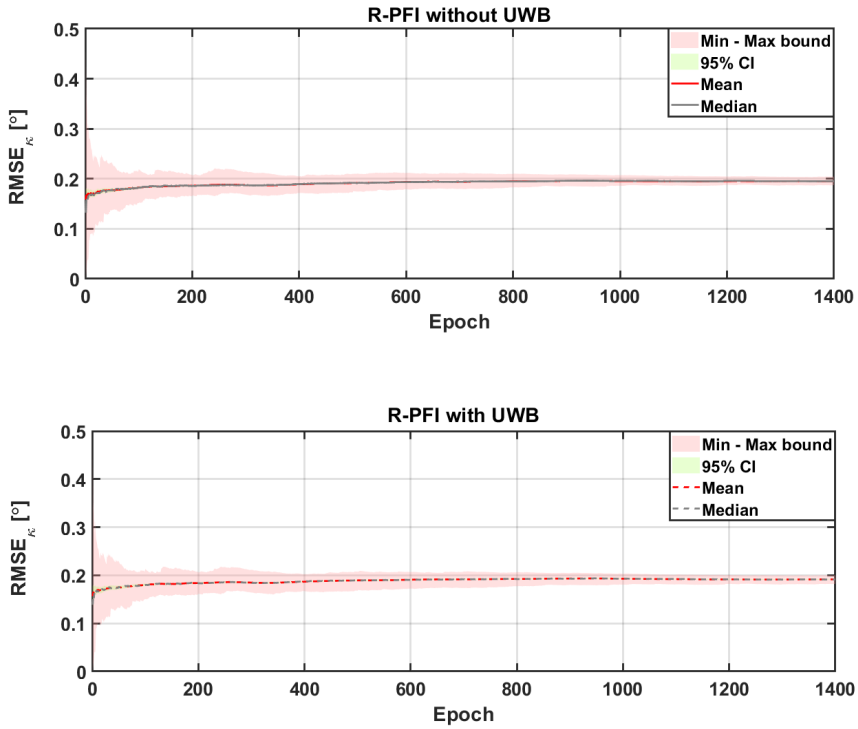

Figure A.11: *Statistical values of the accumulative RMSE of the estimated κ by the R-PFI over MC runs.*

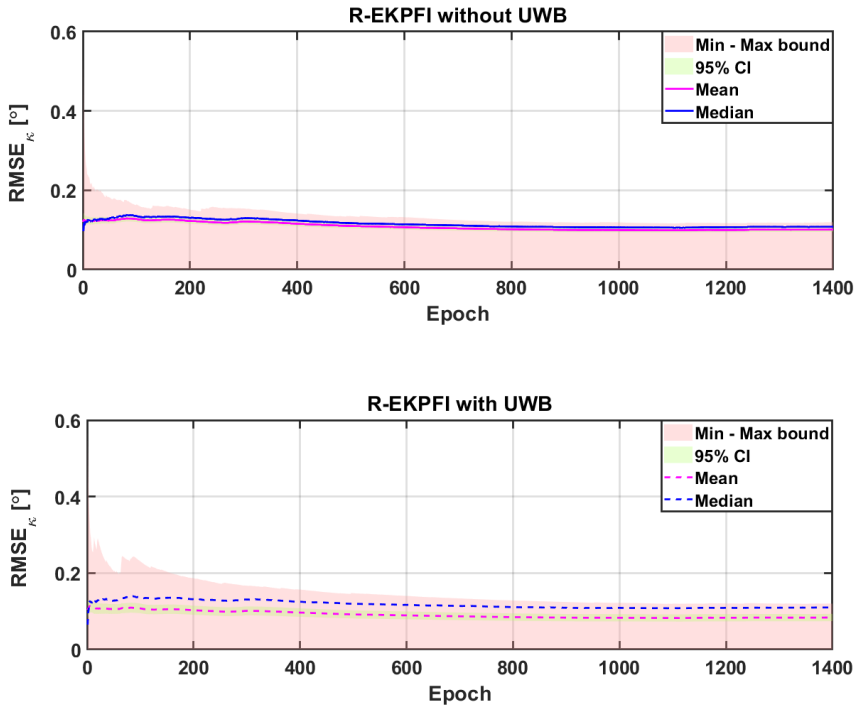

Figure A.12: *Statistical values of the accumulative RMSE of the estimated κ by the R-EKPFI over MC runs.*

Table A.6: *Statistical values of the accumulative RMSE of the estimated κ in the last epoch over the MC runs by means of the different filters. Red and green colors indicate the minimum and maximum values in each column, respectively.*

	$RMSE_\kappa$ [°]					
	Min	Max	Mean	Median	↑ CI (95%)	↓ CI (95%)
versatile IEKF without UWB	0.0453	0.4697	0.1939	0.2153	0.2172	0.1707
versatile IEKF with UWB	0.0509	1.1163	0.2067	0.2390	0.2351	0.1783
R-PFI without UWB	0.1874	0.2041	0.1951	0.1954	0.1957	0.1945
R-PFI with UWB	0.1822	0.2001	0.1915	0.1913	0.1922	0.1909
R-EKPFI without UWB	0	0.1192	0.1009	0.1078	0.1064	0.0954
R-EKPFI with UWB	0	0.1201	0.0838	0.1103	0.0934	0.0743

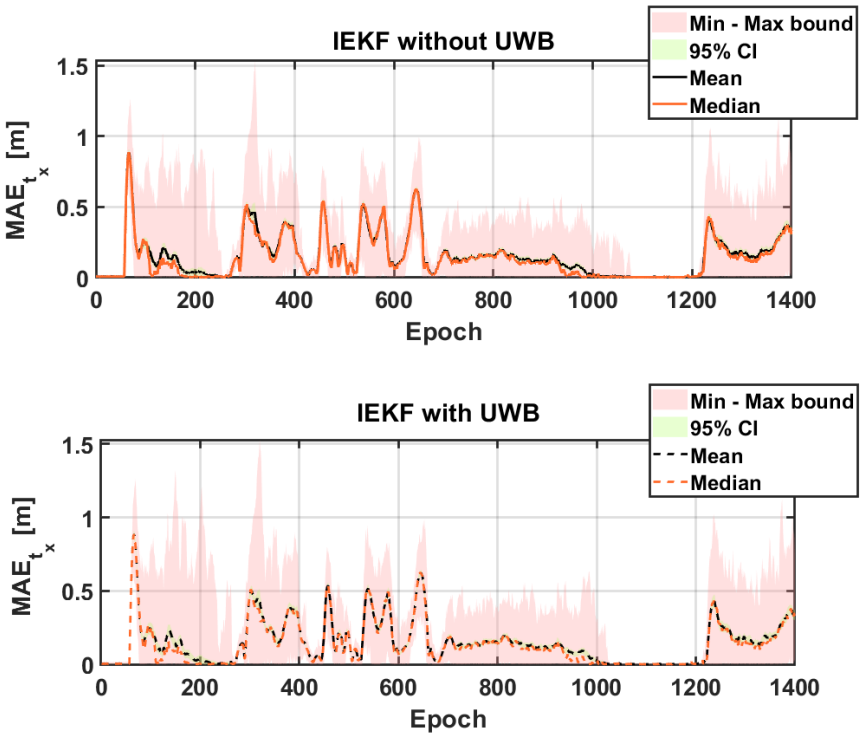

Figure A.13: *Statistical values of the absolute error of the estimated t_x by the versatile IEKF over MC runs.*

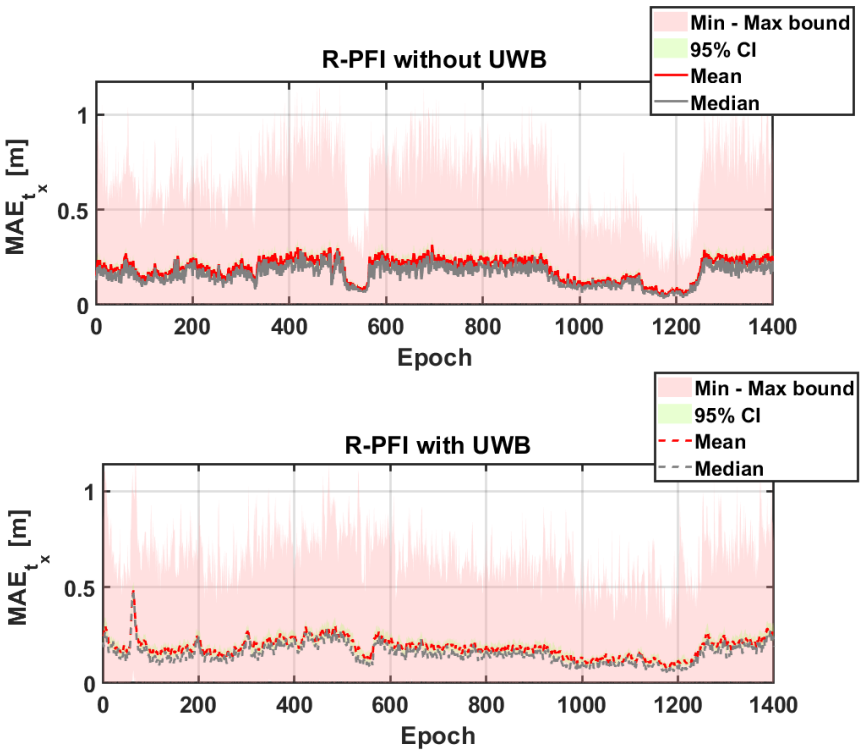

Figure A.14: *Statistical values of the absolute error of the estimated t_x by the R-PFI over MC runs.*

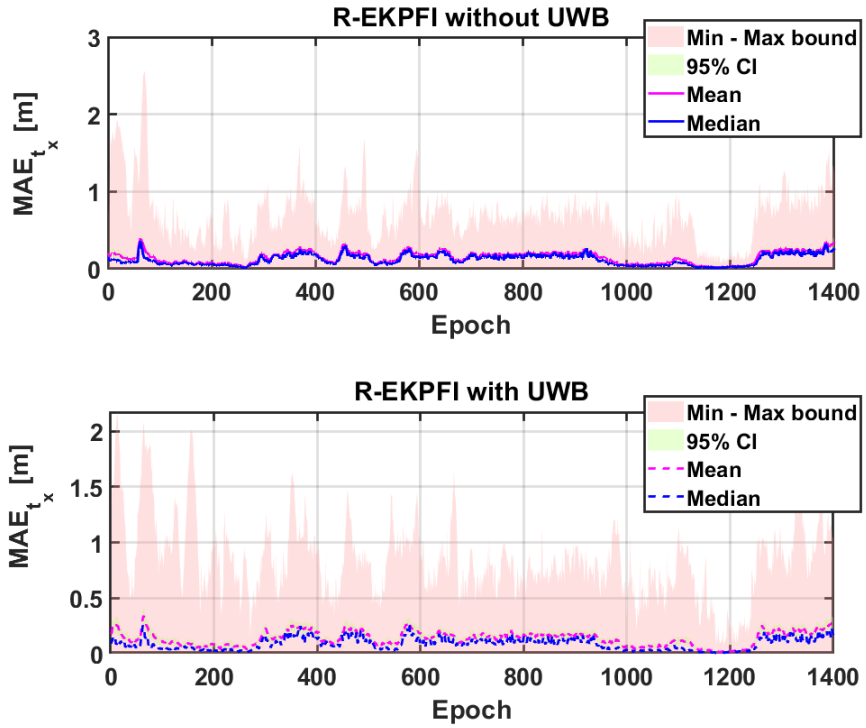

Figure A.15: *Statistical values of the absolute error of the estimated t_x by the R-EKPFI over MC runs.*

Table A.7: *Statistical values of the absolute error of the estimated t_x in the last epoch over the MC runs by means of the different filters. Red and green colors indicate the minimum and maximum values in each column, respectively.*

	$RMSE_{t_x}$ $[m]$					
	Min	Max	Mean	Median	↑ CI (95%)	↓ CI (95%)
versatile IEKF without UWB	0.0105	0.7172	0.3270	0.3028	0.3602	0.2939
versatile IEKF with UWB	0.0315	0.7511	0.3331	0.3008	0.3672	0.2990
R-PFI without UWB	0.0002	0.7956	0.2101	0.1752	0.2433	0.1769
R-PFI with UWB	0.0119	0.7387	0.2514	0.1893	0.2868	0.2160
R-EKPFI without UWB	0	1.3696	0.3354	0.2482	0.3937	0.2772
R-EKPFI with UWB	0	1.0491	0.2675	0.1693	0.3213	0.2138

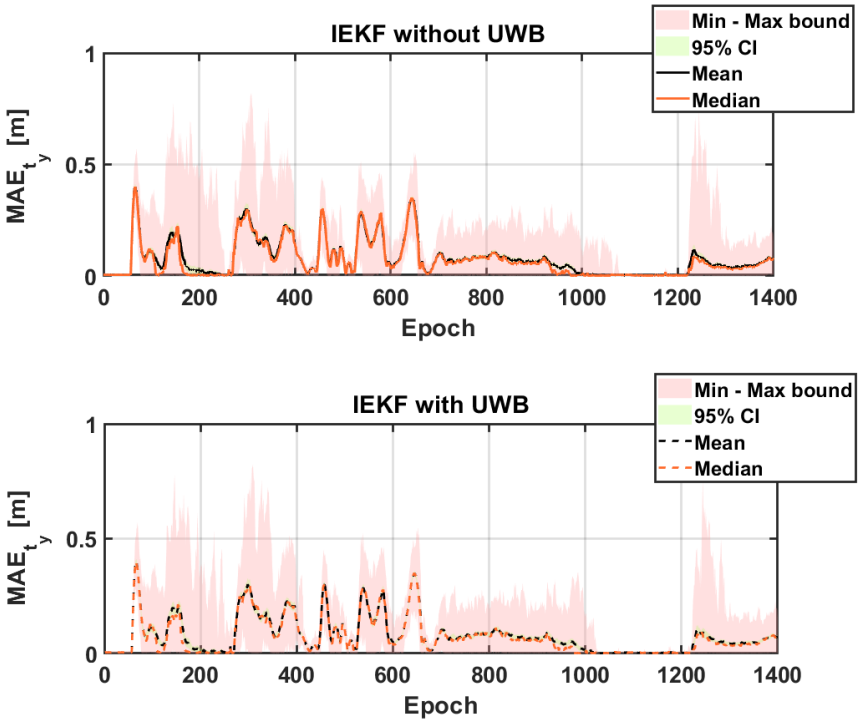

Figure A.16: *Statistical values of the absolute error of the estimated t_y by the versatile IEKF over MC runs.*

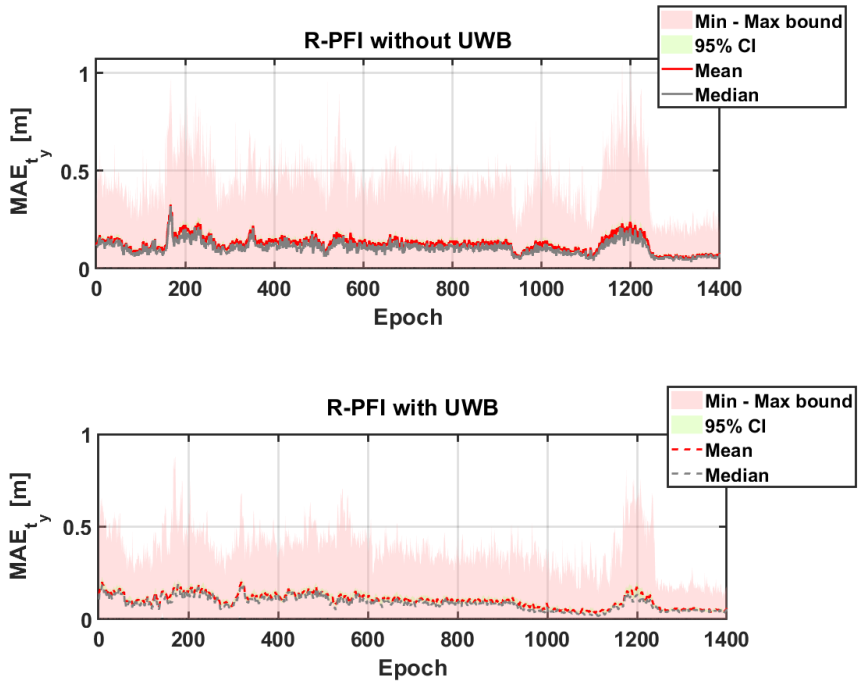

Figure A.17: *Statistical values of the absolute error of the estimated t_y by the R-PFI over MC runs.*

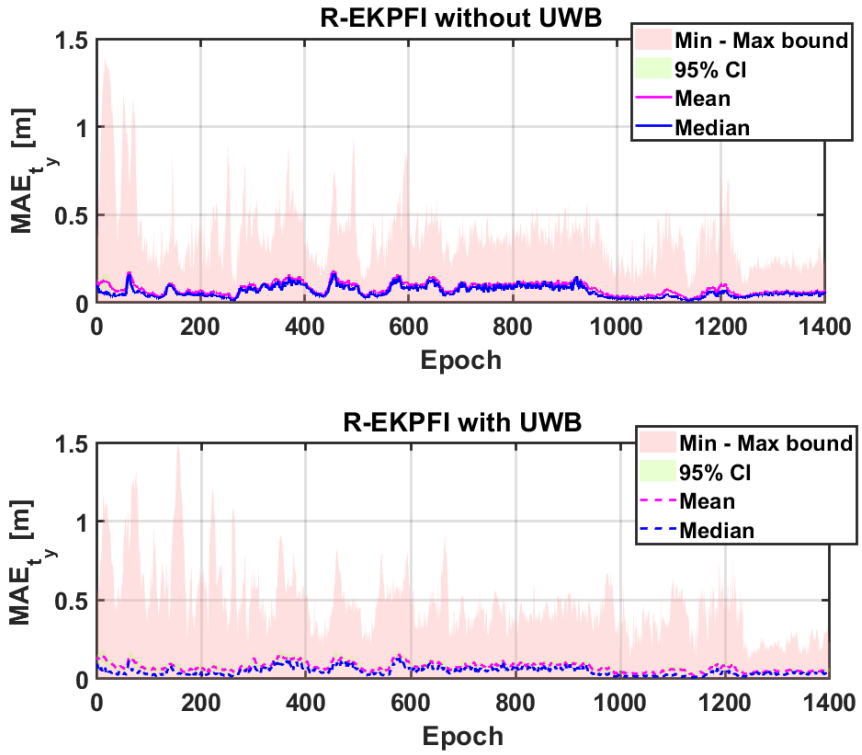

Figure A.18: *Statistical values of the absolute error of the estimated t_y by the R-EKPFI over MC runs.*

Table A.8: *Statistical values of the absolute error of the estimated t_y in the last epoch over the MC runs by means of the different filters. Red and green colors indicate the minimum and maximum values in each column, respectively.*

	$RMSE_{t_y}$ $[m]$					
	Min	Max	Mean	Median	↑ CI (95%)	↓ CI (95%)
versatile IEKF without UWB	0.0076	0.1565	0.0744	0.0760	0.0816	0.0672
versatile IEKF with UWB	0.0051	0.1577	0.0708	0.0691	0.0780	0.0637
R-PFI without UWB	0.0007	0.2547	0.0785	0.0744	0.0891	0.0680
R-PFI with UWB	0.0002	0.1776	0.0606	0.0561	0.0686	0.0526
R-EKPFI without UWB	0	0.2688	0.0695	0.0625	0.0802	0.0588
R-EKPFI with UWB	0	0.2396	0.0583	0.0383	0.0703	0.0463

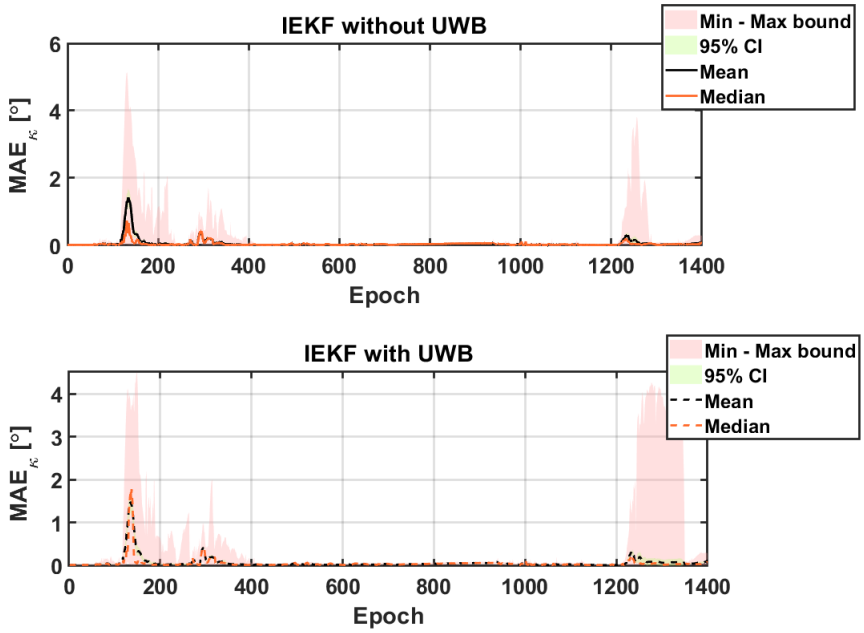

Figure A.19: *Statistical values of the absolute error of the estimated κ by the versatile IEKF over MC runs.*

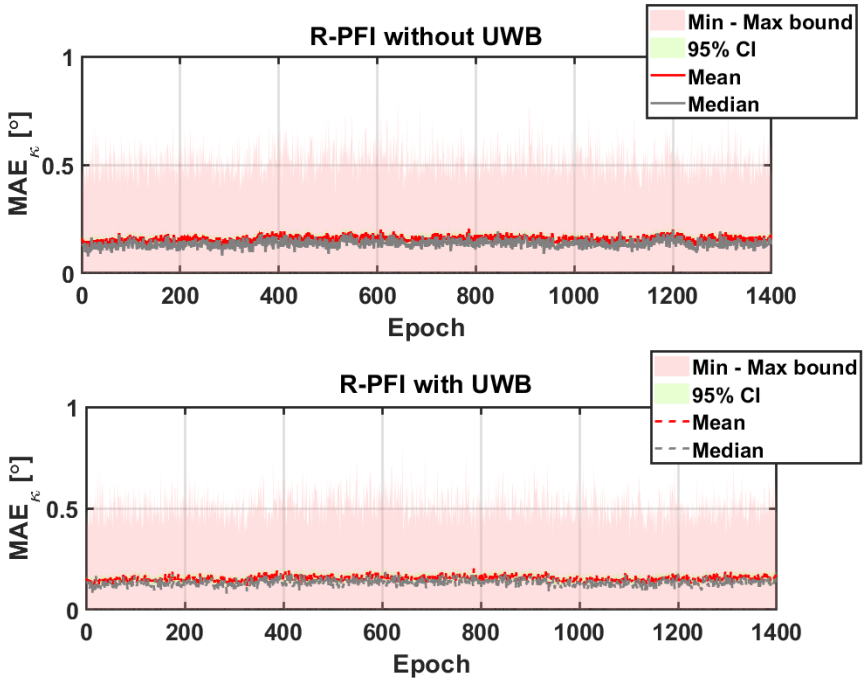

Figure A.20: *Statistical values of the absolute error of the estimated κ by the R-PFI over MC runs.*

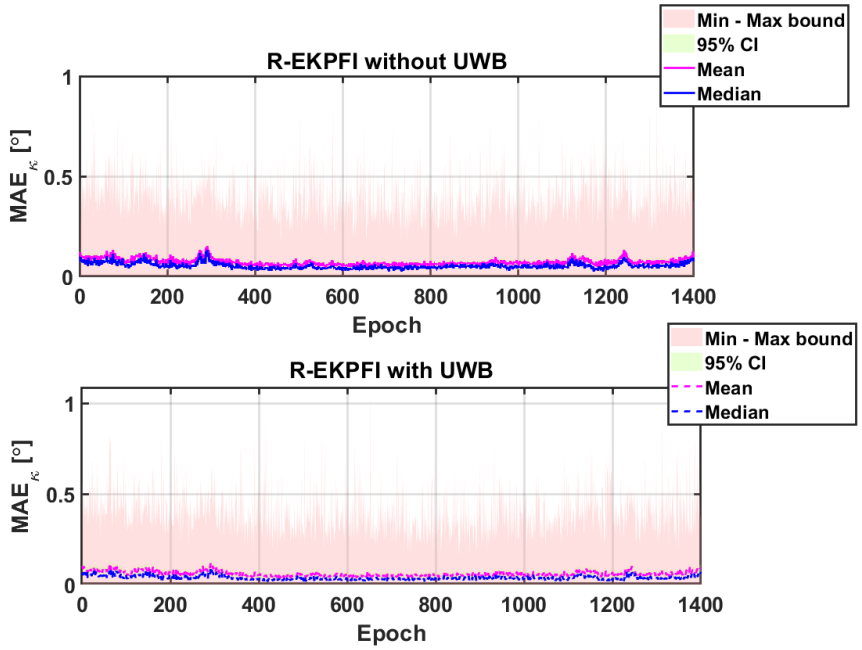

Figure A.21: *Statistical values of the absolute error of the estimated κ by the R-EKPFI over MC runs.*

Table A.9: *Statistical values of the absolute error of the estimated κ in the last epoch over the MC runs by means of the different filters. Red and green colors indicate the minimum and maximum values in each column, respectively.*

	$RMSE_\kappa$ [°]					
	Min	Max	Mean	Median	↑ CI (95%)	↓ CI (95%)
versatile IEKF without UWB	0.0016	0.2903	0.0893	0.0661	0.1029	0.0758
versatile IEKF with UWB	0.0027	0.2781	0.0988	0.0674	0.1141	0.0835
R-PFI without UWB	0.0008	0.4612	0.1630	0.1464	0.1858	0.1402
R-PFI with UWB	0.0014	0.4963	0.1649	0.1410	0.1881	0.1417
R-EKPFI without UWB	0	0.3737	0.0960	0.0758	0.1133	0.0787
R-EKPFI with UWB	0	0.3853	0.0906	0.0699	0.1093	0.0719

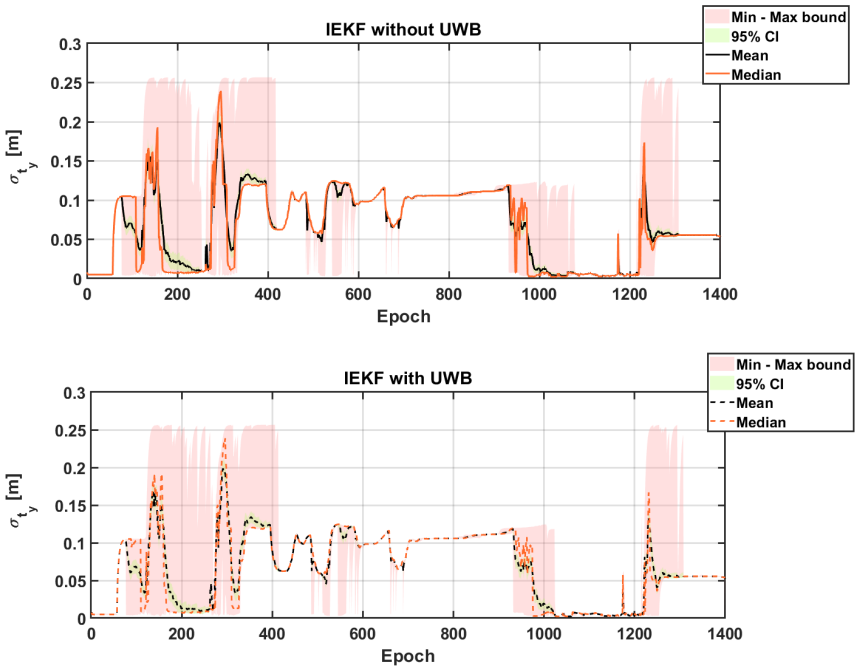

Figure A.22: *Statistical values of the precision of the estimated t_y by the versatile IEKF over MC runs.*

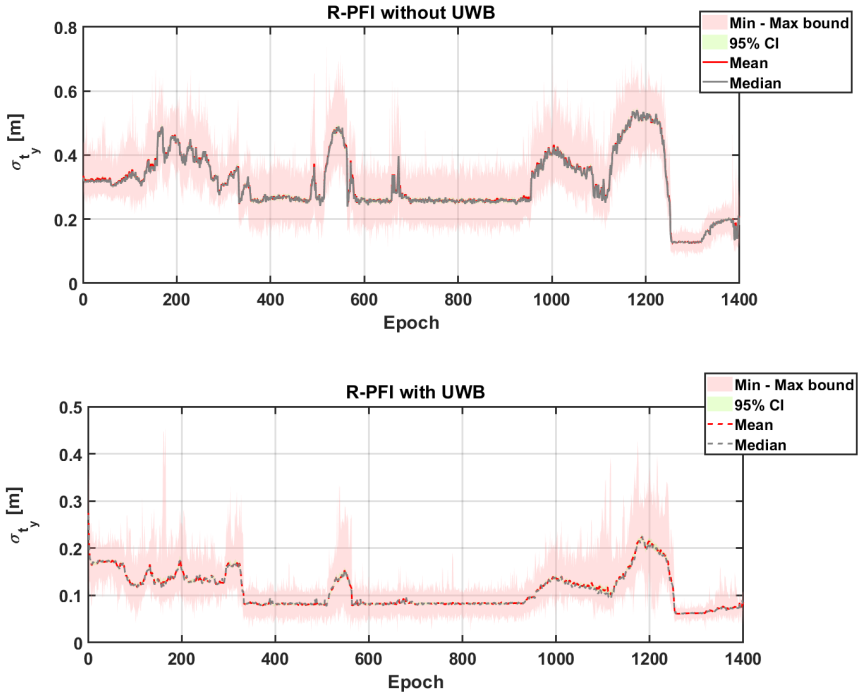

Figure A.23: *Statistical values of the precision of the estimated t_y by the PFI over MC runs.*

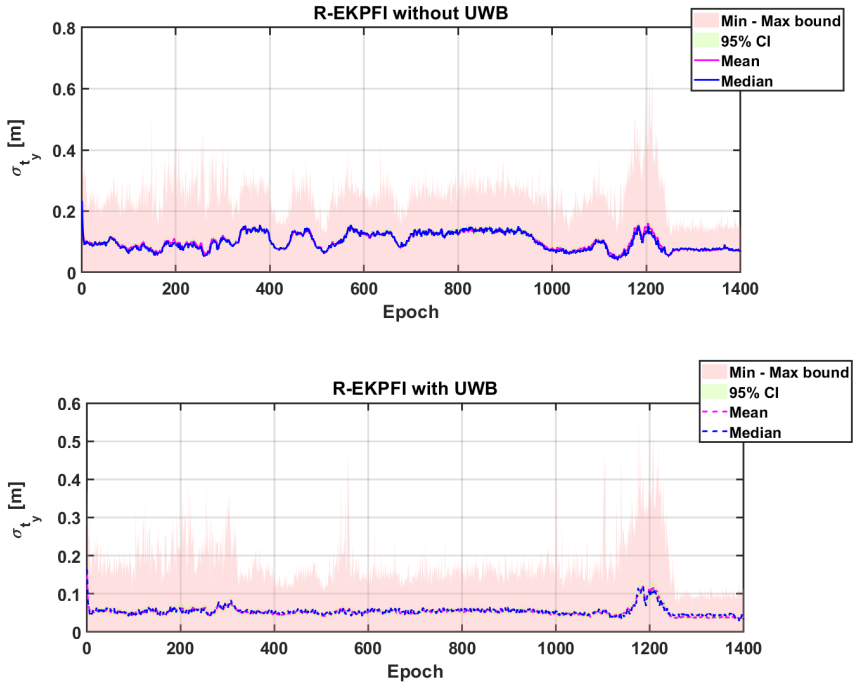

Figure A.24: *Statistical values of the precision of the estimated t_y by the R-EKPFI over MC runs.*

Table A.10: *Statistical values of the precision of the estimated t_y in the last epoch over the MC runs by means of the different filters. Red and green colors indicate the minimum and maximum values in each column, respectively.*

	σ_{t_y} $[m]$					
	Min	Max	Mean	Median	↑ CI (95%)	↓ CI (95%)
versatile IEKF without UWB	0.0042	0.0550	0.0532	0.0541	0.0546	0.0518
versatile IEKF with UWB	0.0040	0.0550	0.0523	0.0541	0.0542	0.0503
R-PFI without UWB	0.1290	0.2946	0.2097	0.2046	0.2163	0.2032
R-PFI with UWB	0.0601	0.1077	0.0874	0.0864	0.0892	0.0857
R-EKPFI without UWB	0	0.1512	0.0687	0.0694	0.0742	0.0631
R-EKPFI with UWB	0	0.1161	0.0358	0.0360	0.0415	0.0301

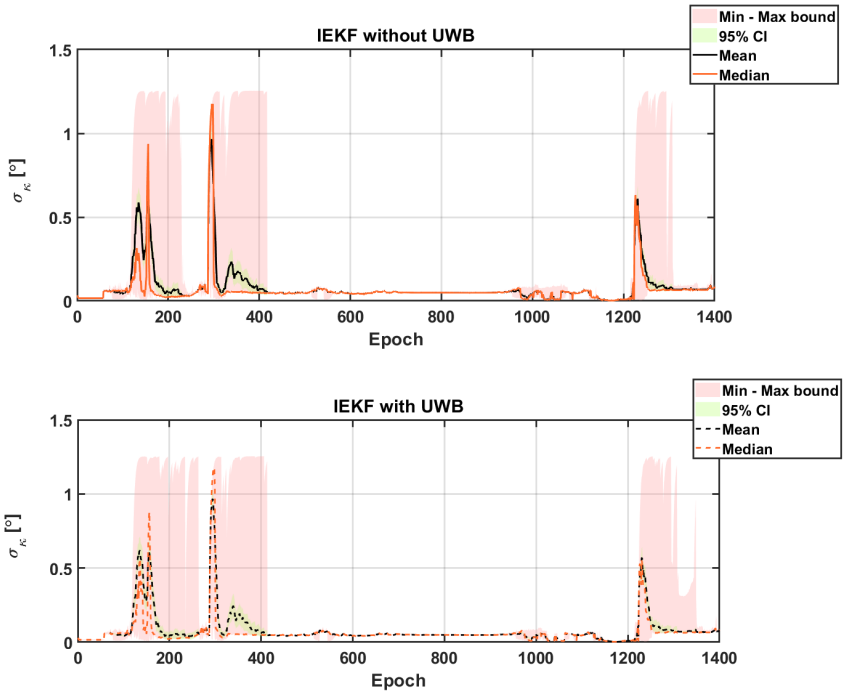

Figure A.25: *Statistical values of the precision of the estimated κ by the versatile IEKF over MC runs.*

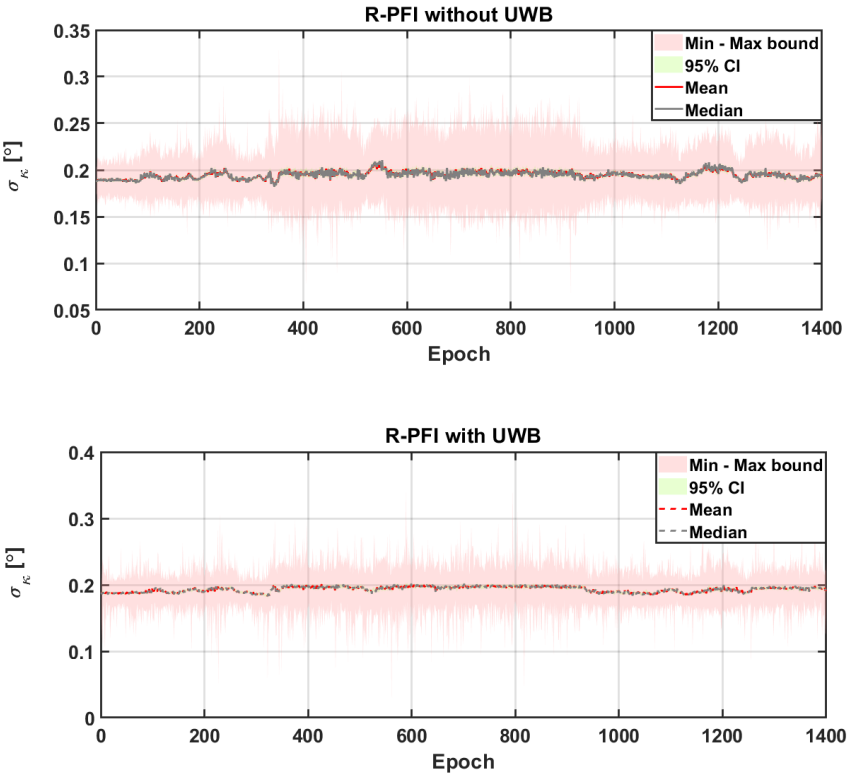

Figure A.26: *Statistical values of the precision of the estimated κ by the PFI over MC runs.*

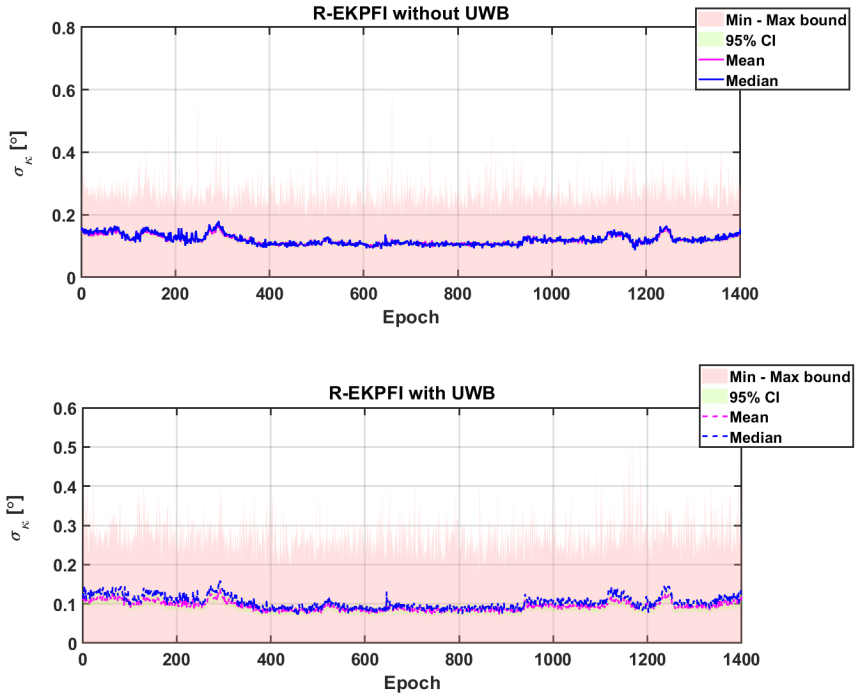

Figure A.27: *Statistical values of the precision of the estimated κ by the R-EKPFI over MC runs.*

Table A.11: *Statistical values of the precision of the estimated κ in the last epoch over the MC runs by means of the different filters. Red and green colors indicate the minimum and maximum values in each column, respectively.*

	σ_κ [°]					
	Min	Max	Mean	Median	↑ CI (95%)	↓ CI (95%)
versatile IEKF without UWB	0.0696	0.0984	0.0835	0.0841	0.0845	0.0824
versatile IEKF with UWB	0.0696	0.0926	0.0822	0.0841	0.0832	0.0811
R-PFI without UWB	0.1619	0.2355	0.1943	0.1934	0.1966	0.1921
R-PFI with UWB	0.1579	0.2350	0.1931	0.1926	0.1954	0.1908
R-EKPFI without UWB	0	0.2761	0.1477	0.1558	0.1594	0.1360
R-EKPFI with UWB	0	0.3730	0.1181	0.1275	0.1353	0.1009

A.3 Additional Results of the Information-Based Georeferencing for the Simulated Case Study - Setup 2

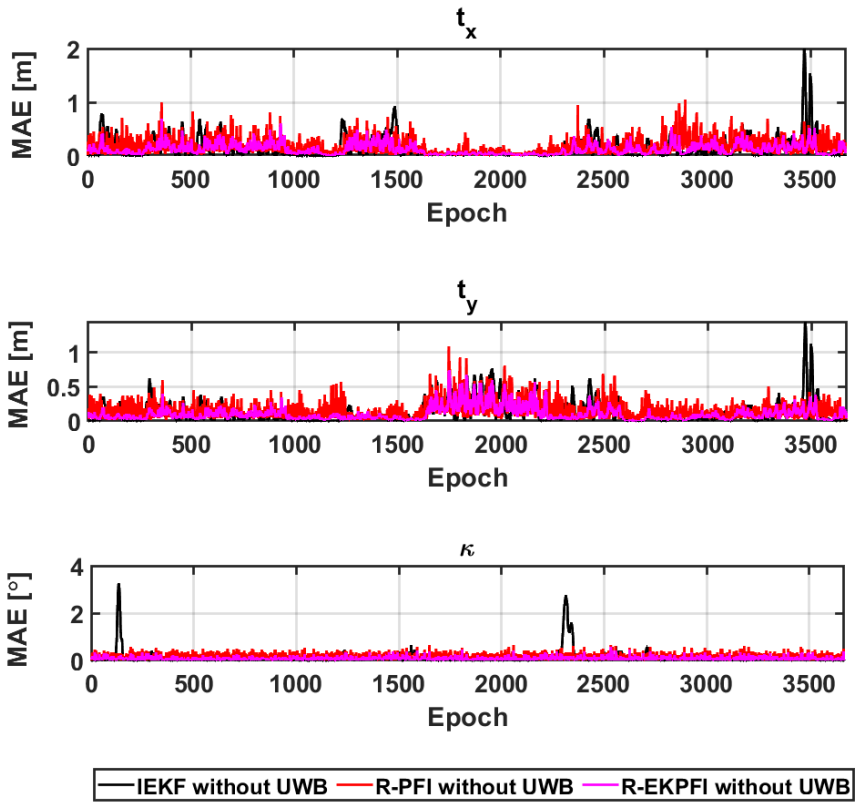

Figure A.28: *Absolute error of the estimated states over multiple runs.*

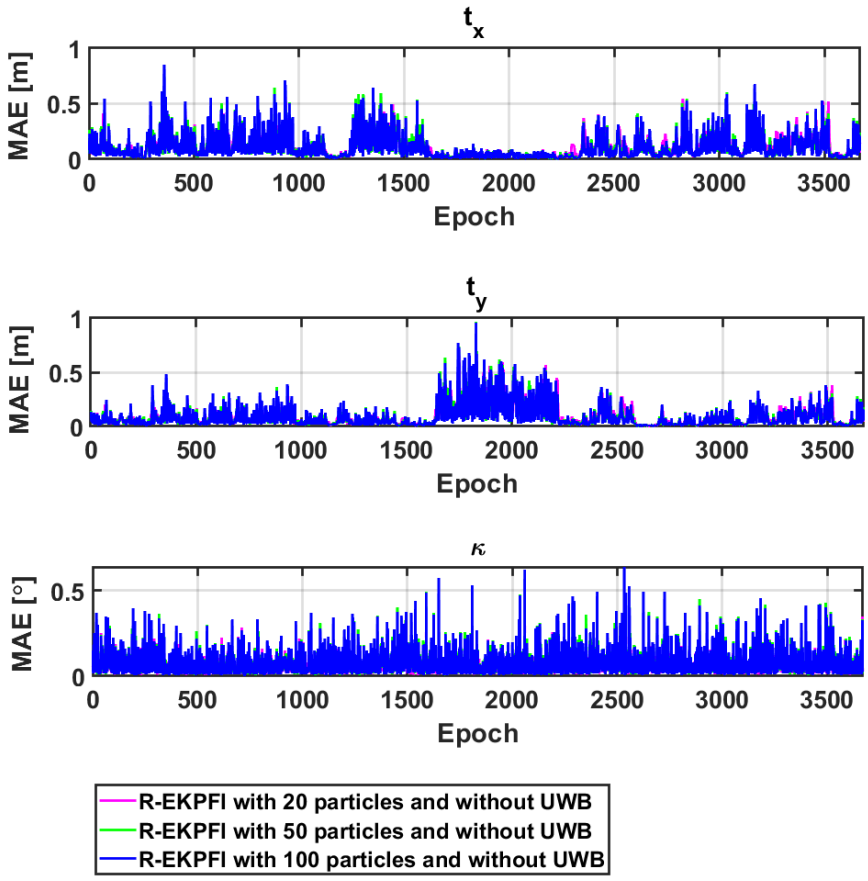

Figure A.29: *Absolute error of the estimated states over multiple runs by the R-EKPFI algorithm with various number of particles.*

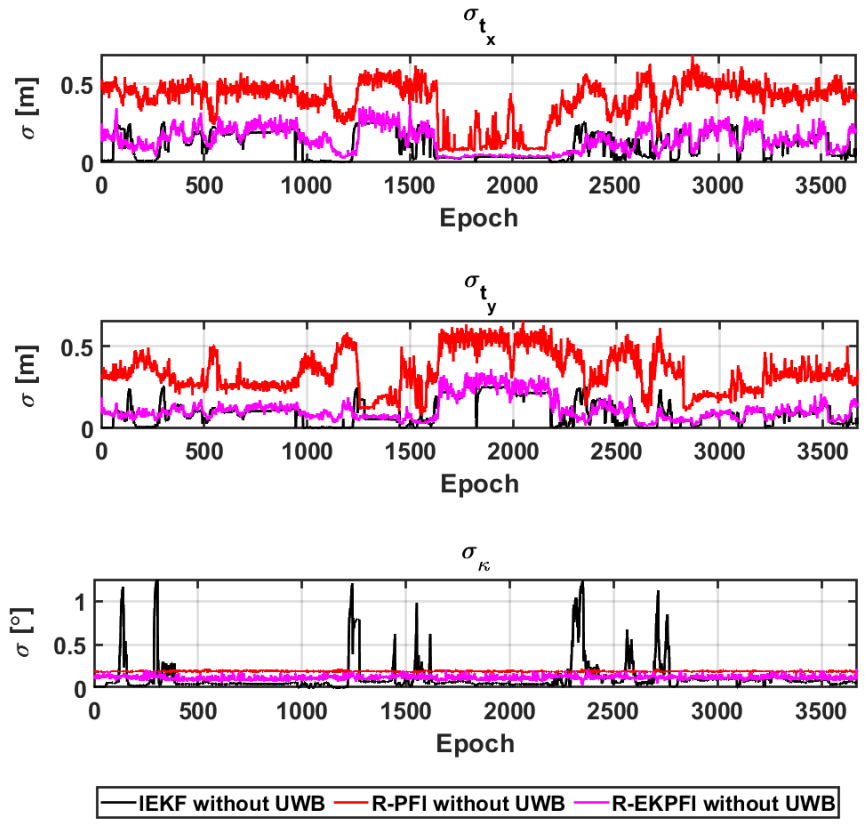

Figure A.30: *Average precision of the estimated states over multiple runs.*

Bibliography

T. Abmayr, F. Härtl, G. Hirzinger, D. Burschka, and C. Fröhlich. A correlation based target finder for terrestrial laser scanning. *Journal of Applied Geodesy 2*, 2008.

M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, volume 55. US Government printing office, 1968.

H. Alkhatib. Alternative nonlinear filtering techniques in geodesy for dual state and adaptive parameter estimation. In *The 1st International Workshop on the Quality of Geodetic Observation and Monitoring Systems (QuGOMS'11)*, pages 107–113. Springer, 2015.

H. Alkhatib, J.-A. Paffenholz, and H. Kutterer. Sequential monte carlo filtering for nonlinear gnss trajectories. In *VII Hotine-Marussi Symposium on Mathematical Geodesy*, pages 81–86. Springer, 2012.

B. Allotta, A. Caiti, R. Costanzi, F. Fanelli, D. Fenucci, E. Meli, and A. Ridolfi. A new auv navigation system exploiting unscented kalman filter. *Ocean Engineering*, 113:121–132, 2016.

E. Apriliani, H. Nurhadi, et al. Ensemble and fuzzy kalman filter for position estimation of an autonomous underwater vehicle based on dynamical system of auv motion. *Expert Systems with Applications*, 68:29–35, 2017.

J. Axmann, R. Moftizadeh, J. Su, B. Tennstedt, Q. Zou, Y. Yuan, D. Ernst, H. Alkhatib, C. Brenner, and S. Schön. Lucoop: Leibniz university cooperative perception and urban navigation dataset. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–8. IEEE, 2023.

Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. John Wiley & Sons, 2004.

G. Barbato, E. Barini, G. Genta, and R. Levi. Features and performance of some outlier detection methods. *Journal of Applied Statistics*, 38(10):2133–2149, 2011.

E. R. Beadle and P. M. Djuric. A fast-weighted bayesian bootstrap filter for nonlinear model state estimation. *IEEE Transactions on Aerospace and Electronic Systems*, 33(1):338–343, 1997.

B. M. Bell and F. W. Cathey. The iterated kalman filter update as a gauss-newton method. *IEEE Transactions on Automatic Control*, 38(2):294–297, 1993.

C. M. Bishop and N. M. Nasrabadi. *Pattern Recognition and Machine Learning*, volume 4. Springer, 2006.

M. Bôcher. *Plane Analytic Geometry: with Introductory Chapters on the Differential Calculus*. H. Holt, 1915.

J. Bureick, S. Vogel, I. Neumann, J. Unger, and H. Alkhatib. Georeferencing of an unmanned aerial system by means of an iterated extended kalman filter using a 3d city model. *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 87(5):229–247, 2019.

G. Casella and R. L. Berger. *Statistical Inference*. Cengage Learning, 2021.

F. Castanedo. A review of data fusion techniques. *The scientific world journal*, 2013, 2013.

T. Chai and R. R. Draxler. Root mean square error (rmse) or mean absolute error (mae). *Geoscientific model development discussions*, 7(1):1525–1534, 2014.

R. Chou, Y. Boers, M. Podt, and M. Geist. Performance evaluation for particle filters. In *14th International Conference on Information Fusion*, pages 1–7. IEEE, 2011.

L. Clarke. Probability and measure. *The Mathematical Gazette*, 64(430):293–294, 1980.

Cummings Research Park. Data Sheet/User Manual PulsON 440 TIME DOMAIN, 2015. URL `https://usermanual.wiki/Humatics/P440-A`. URL last accessed on 05.04.2024.

B. Curless. Ray-triangle intersection. 2006. *Acesso em*, 2, 2020.

J. Čurn, D. Marinescu, N. O'Hara, and V. Cahill. Data incest in cooperative localisation with the common past-invariant ensemble kalman filter. In *Proceedings of the 16th International Conference on Information Fusion*, pages 68–76. IEEE, 2013.

T. Dang. *Kontinuierliche Selbstkalibrierung von Stereokameras*. Universitätsverlag Karlsruhe, 2007.

Y. Dehbi, L. Lucks, J. Behmann, L. Klingbeil, and L. Plümer. Improving gps trajectories using 3d city models and kinematic point clouds. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4, 2019.

Z. Deng, P. Zhang, W. Qi, G. Yuan, and J. Liu. The accuracy comparison of multisensor covariance intersection fuser and three weighting fusers. *Information Fusion*, 14(2):177–185, 2013.

W. F. Denham and S. Pines. Sequential estimation when measurement function nonlinearity is comparable to measurement error. *AIAA journal*, 4(6):1071–1076, 1966.

D. Dennig, J. Bureick, J. Link, D. Diener, C. Hesse, and I. Neumann. Comprehensive and highly accurate measurements of crane runways, profiles and fastenings. *Sensors*, 17(5):1118, 2017.

P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez. Particle filtering. *IEEE signal processing magazine*, 20(5):19–38, 2003.

J. Döllner, K. Baumann, and H. Buchholz. *Virtual 3D City Models as Foundation of Complex Urban Information Spaces*. na, 2006.

A. Doucet, N. d. Freitas, and N. Gordon. An introduction to sequential monte carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer, 2001.

M. Elhousni, Z. Zhang, and X. Huang. Lidar-osm-based vehicle localization in gps-denied environments by using constrained particle filter. *Sensors*, 22(14):5206, 2022.

I. Elkhrachy and W. Niemeier. Stochastic assessment of terrestrial laser scanner. In *Proceedings of the ASPRS Annual Conference, Reno, Nevada, USA*, pages 1–5, 2006.

G. Evensen. The ensemble kalman filter: Theoretical formulation and practical implementation. *Ocean dynamics*, 53(4):343–367, 2003.

J. Fayyad, M. A. Jaradat, D. Gruyer, and H. Najjaran. Deep learning sensor fusion for autonomous vehicle perception and localization: A review. *Sensors*, 20(15):4220, 2020.

Á. F. G. Fernández. *Detection and Tracking of Multiple Targets Using Wireless Sensor Networks Detección y Seguimiento De Múltiples Blancos En Redes Inalámbricas De Sensores*. PhD thesis, Universidad Politécnica de Madrid, 2011.

N. G. Fernandez. *Simulation Framework for Collaborative Navigation: Development-Analysis-Optimization*. PhD thesis, Fachrichtung Geodäsie und Geoinformatik der Leibniz Universität Hannover, 2020.

R. A. Fowler. The lowdown on lidar. *Earth Observation Magazine*, 9(3):5, 2000.

M. Frigge, D. C. Hoaglin, and B. Iglewicz. Some implementations of the boxplot. *The American Statistician*, 43(1):50–54, 1989.

B. E. Fristedt and L. F. Gray. *A Modern Approach to Probability Theory*. Springer Science & Business Media, 2013.

A. Gelb et al. *Applied Optimal Estimation*. MIT press, 1974.

J. González, J.-L. Blanco, C. Galindo, A. Ortiz-de Galisteo, J.-A. Fernández-Madrigal, F. A. Moreno, and J. L. Martinez. Mobile robot localization based on ultra-wide-band ranging: A particle filter approach. *Robotics and autonomous systems*, 57(5):496–507, 2009.

N. J. Gordon, D. J. Salmond, and A. F. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE proceedings F (radar and signal processing)*, volume 140, pages 107–113. IET, 1993.

M. S. Grewal, A. P. Bartone, and P. Andrews. *Inertial Navigation Sensors and Systems*. Wiley, 2001.

J. Hammersley. *Monte Carlo Methods*. Springer Science & Business Media, 2013.

J. Hartmann, P. Trusheim, H. Alkhatib, J.-A. Paffenholz, D. Diener, and I. Neumann. High accurate pointwise (geo-) referencing of a k-tls based multi-sensor-system. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences IV-4 (2018)*, 4:81–88, 2018.

J. Hartzer and S. Saripalli. Vehicular teamwork: Collaborative localization of autonomous vehicles. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 1077–1082. IEEE, 2021.

T. Herlambang, H. Nurhadi, et al. Unusaits auv navigation and guidance system with nonlinear modeling motion using ensemble kalman filter. In *2019 International Conference on Advanced Mechatronics, Intelligent Manufacture and Industrial Automation (ICAMIMIA)*, pages 1–4. IEEE, 2019.

JAVAD GNSS Inc. GNSS Antenna GrAnt Datasheet, 2020. URL `https://download.javad.com/sheets/GrAnt_Datasheet.pdf`. URL last accessed on 05.04.2024.

S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

H. Kim, B. Liu, C. Y. Goh, S. Lee, and H. Myung. Robust vehicle localization using entropy-weighted particle filter-based data fusion of vertical and road intensity information for a large scale urban area. *IEEE Robotics and Automation Letters*, 2(3):1518–1524, 2017.

D. P. Kroese, T. Taimre, and Z. I. Botev. *Handbook of Monte Carlo Methods*. John Wiley & Sons, 2013.

S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Landesamt für Geoinformation und Landesvermessung Niedersachsen (LGLN), Landesvermessung und Geobasisinformation. 3D-gebäudemodell (LoD2), 2023. URL https://opengeodata.lgln.niedersachsen.de/#lod2. URL last accessed on 05.04.2024.

Landeshauptstadt Hannover, FB Planen und Stadtentwicklung, Bereich Geoinformation. Digitales geländemodell (DGM1), 2023. URL https://www.hannover.de/Leben-in-der-Region-Hannover/Verwaltungen-Kommunen/Die-Verwaltung-der-Landeshauptstadt-Hannover/Dezernate-und-Fachbereiche-der-LHH/Stadtentwicklung-und-Bauen/Fachbereich-Planen-und-Stadtentwicklung/Geoinformation/Open-GeoData/3D-Stadtmodell-und-Gel%C3%A4ndemodell/Digitales-Gel%C3%A4ndemodell-DGM1. URL last accessed on 05.04.2024.

Leica Geosystems. Leica nova ms60, 2023. URL https://leica-geosystems.com/products/total-stations/multistation/leica-nova-ms60. URL last accessed on 05.04.2024.

T. Li, M. Bolic, and P. M. Djuric. Resampling methods for particle filtering: Classification, implementation, and strategies. *IEEE Signal processing magazine*, 32(3):70–86, 2015.

J. Li-Chee-Ming and C. Armenakis. Determination of uas trajectory in a known environment from fpv video. *Int Arch Photogramm Remote Sens Spat Inf Sci*, 40:247–252, 2013.

L. Liang-Qun, J. Hong-Bing, and L. Jun-Hui. The iterated extended kalman particle filter. In *IEEE International Symposium on Communications and Information Technology, 2005. ISCIT 2005.*, volume 2, pages 1213–1216. IEEE, 2005.

P. Liashchynskyi and P. Liashchynskyi. Grid search, random search, genetic algorithm: a big comparison for nas. *arXiv preprint arXiv:1912.06059*, 2019.

J. S. Liu and R. Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American statistical association*, 93(443):1032–1044, 1998.

D. J. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge university press, 2003.

S. Malik, M. A. Khan, and H. El-Sayed. Collaborative autonomous driving—a survey of solution approaches and future challenges. *Sensors*, 21(11):3783, 2021.

L. Marchetti, G. Grisetti, and L. Iocchi. A comparative analysis of particle filter based localization methods. In *Robot Soccer World Cup*, pages 442–449. Springer, 2006.

C. Melendez-Pastor, R. Ruiz-Gonzalez, and J. Gomez-Gil. A data fusion system of gnss data and on-vehicle sensors data for improving car positioning precision in urban environments. *Expert systems with applications*, 80:28–38, 2017.

R. Metzler. Generalized chapman-kolmogorov equation: A unifying approach to the description of anomalous transport in external fields. *Physical Review E*, 62(5):6233, 2000.

R. Moftizadeh, S. Vogel, A. Dorndorf, J. Jüngerink, and H. Alkhatib. Information-based georeferencing of multi-sensor-systems by particle filter with implicit measurement equations. In *2021 IEEE 24th International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2021.

C. Z. Mooney. *Monte Carlo Simulation*, volume 116. Sage, 1997.

NovAtel Inc. VEXXIS Antennas GNSS-850, 2017. URL https://hexagondownloads.blob.core.windows.net/public/Novatel/assets/Documents/Papers/GNSS-850-Product-Sheet/GNSS-850-Product-Sheet.pdf. URL last accessed on 05.04.2024.

D. Orlando, F. Ehlers, and G. Ricci. A maximum likelihood tracker for multistatic sonars. In *2010 13th International Conference on Information Fusion*, pages 1–6. IEEE, 2010.

J.-A. Paffenholz. *Direct Geo-Referencing of 3D Point Clouds with 3D Positioning Sensors*. PhD thesis, Hannover: Gottfried Wilhelm Leibniz Universität Hannover, 2012.

J. Patoliya, H. Mewada, M. Hassaballah, M. A. Khan, and S. Kadry. A robust autonomous navigation and mapping system based on gps and lidar data for unconstraint environment. *Earth Science Informatics*, pages 1–13, 2022.

A. U. Peker, O. Tosun, and T. Acarman. Particle filter vehicle localization and map-matching using map topology. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 248–253. IEEE, 2011.

K. Potter, H. Hagen, A. Kerren, and P. Dannenmann. Methods for presenting statistical information: The box plot. In *VLUDS*, pages 97–106, 2006.

G. Retscher, V. Gikas, H. Hofer, H. Perakis, and A. Kealy. Range validation of uwb and wi-fi for integrated indoor positioning. *Applied Geomatics*, 11:187–195, 2019.

B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech house, 2003.

A. R. Rormero, P. V. K. Borges, A. Pfrunder, and A. Elfes. Map-aware particle filter for localization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2940–2947. IEEE, 2018.

F. Rosique, P. J. Navarro, C. Fernández, and A. Padilla. A systematic review of perception system and simulators for autonomous vehicles research. *Sensors*, 19(3):648, 2019.

N. Salameh, G. Challita, S. Mousset, A. Bensrhair, and S. Ramaswamy. Collaborative positioning and embedded multi-sensors fusion cooperation in advanced driver assistance system. *Transportation Research Part C: Emerging Technologies*, 29:197–213, 2013.

A. Satish and R. L. Kashyap. Multiple target tracking using maximum likelihood principle. *IEEE transactions on signal processing*, 43(7):1677–1695, 1995.

S. Schön, C. Brenner, H. Alkhatib, M. Coenen, H. Dbouk, N. Garcia-Fernandez, C. Fischer, C. Heipke, K. Lohmann, I. Neumann, et al. Integrity and collaboration in dynamic sensor networks. *Sensors*, 18(7):2400, 2018.

S. Schuhmacher and J. Böhm. Georeferencing of terrestrial laserscanner data for applications in architectural modeling 3d-arch 2005: Virtual reconstruction and visualization of complex architectures. *XXXVI, PART*, 5:W17, 2005.

Septentrio. PolaRx5 GNSS Reference Receivers, 2020. URL `https://www.septentrio.com/en/products/gnss-receivers/reference-receivers/polarx-5`. URL last accessed on 05.04.2024.

D. Simon. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. John Wiley & Sons, 2006.

D. Simon. Kalman filtering with state constraints: A survey of linear and nonlinear algorithms. *IET Control Theory & Applications*, 4(8):1303–1318, 2010.

D. Simon and T. L. Chia. Kalman filtering with state equality constraints. *IEEE transactions on Aerospace and Electronic Systems*, 38(1):128–136, 2002.

D. E. Smith and M. T. Zuber. The relationship between mola northern hemisphere topography and the 6.1-mbar atmospheric pressure surface of mars. *Geophysical Research Letters*, 25(24): 4397–4400, 1998.

D. E. Smith, M. T. Zuber, S. C. Solomon, R. J. Phillips, J. W. Head, J. B. Garvin, W. B. Banerdt, D. O. Muhleman, G. H. Pettengill, G. A. Neumann, et al. The global topography of mars and implications for surface evolution. *science*, 284(5419):1495–1503, 1999.

A. Soloviev, D. Bates, and F. Van Graas. Tight coupling of laser scanner and inertial measurements for a fully autonomous relative navigation solution. *Navigation*, 54(3):189–205, 2007.

J. K. Suhr, J. Jang, D. Min, and H. G. Jung. Sensor fusion-based low-cost vehicle localization system for complex urban environments. *IEEE Transactions on Intelligent Transportation Systems*, 18 (5):1078–1086, 2016.

M. Tailanián, S. Paternain, R. Rosa, and R. Canetti. Design and implementation of sensor data fusion for an autonomous quadrotor. In *2014 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, pages 1431–1436. IEEE, 2014.

J. Talaya, R. Alamus, E. Bosch, A. Serra, W. Kornus, and A. Baron. Integration of a terrestrial laser scanner with gps/imu orientation sensors. In *Proceedings of the XXth ISPRS Congress*, volume 35, pages 1049–1055, 2004.

S. Thrun, W. Burgard, and D. Fox. Probalistic robotics. *Kybernetes*, 2006.

J. W. Tukey et al. *Exploratory Data Analysis*, volume 2. Reading, MA, 1977.

J. Unger. *Integrated Estimation of UAV Image Orientation with a Generalised Building Model*. Hannover: Leibniz Universität Hannover, 2020.

J. Unger, F. Rottensteiner, and C. Heipke. Integration of a generalised building model into the pose estimation of uas images. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences; XLI-B1*, 41:1057–1064, 2016.

J. Unger, F. Rottensteiner, and C. Heipke. Assigning tie points to a generalised building model for uas image orientation. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42:385–392, 2017.

R. Van Der Merwe, A. Doucet, N. De Freitas, and E. Wan. The unscented particle filter. *Advances in neural information processing systems*, 13, 2000.

Velodyne LiDAR, Inc. VLP-16 User Manual, 2019. URL `https://velodynelidar.com/wp-content/uploads/2019/12/63-9243-Rev-E-VLP-16-User-Manual.pdf`. URL last accessed on 05.04.2024.

S. Vogel. *Kalman Filtering with State Constraints Applied to Multi-sensor Systems and Georeferencing*. PhD thesis, Fachrichtung Geodäsie und Geoinformatik der Leibniz Universität Hannover, 2020.

S. Vogel, H. Alkhatib, and I. Neumann. Iterated extended kalman filter with implicit measurement equation and nonlinear constraints for information-based georeferencing. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 1209–1216. IEEE, 2018.

S. Vogel, H. Alkhatib, J. Bureick, R. Moftizadeh, and I. Neumann. Georeferencing of laser scanner-based kinematic multi-sensor systems in the context of iterated extended kalman filters using geometrical constraints. *Sensors*, 19(10):2280, 2019.

P. Wu, J. Kong, and Y. Bo. Modified iterated extended kalman particle filter for single satellite passive tracking. *Turkish Journal of Electrical Engineering and Computer Sciences*, 21(1):120–130, 2013.

Z. Xian, J. Lian, M. Shan, L. Zhang, X. He, and X. Hu. A square root unscented kalman filter for multiple view geometry based stereo cameras/inertial navigation. *International Journal of Advanced Robotic Systems*, 13(5):1729881416664850, 2016.

S. Zair, S. Le Hégarat-Mascle, and E. Seignez. A-contrario modeling for robust localization using raw gnss data. *IEEE Transactions on Intelligent Transportation Systems*, 17(5):1354–1367, 2015.

R. Zhan and J. Wan. Iterated unscented kalman filter for passive target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 43(3):1155–1163, 2007.

Acknowledgments

As George Shinn said, "There is no such thing as a self-made man. You will reach your goals only with the help of others." Without others, this dissertation could not have been successfully completed.
First of all, I would like to express my gratitude to my supervisor PD Dr.-Ing. Hamza Alkhatib. Your constant support and constructive comments were the key to the successful completion of this dissertation. The belief you had in me paved the way for me to explore new ideas and find myself in the scientific world. Thank you for not only introducing me to new scientific ideas, but also for showing me new attitudes towards life.
I would also like to thank my boss, Prof. Dr.-Ing. Ingo Neumann, who was also my second supervisor and examiner for this thesis. I have always said that having a boss like you is the most valuable experience an employee can have. You have shown me how it is possible to be in the highest position and still be a friend to the member of staff. Thank you for always taking the time to carefully monitor my progress and thank you for all the valuable feedback you have given me during this time.
I would like to express my sincere gratitude to Prof. Allison Kealy, with whom I had the honor to have a period of collaboration during my doctoral studies as well as the examiner of this dissertation. Working with you has been an unforgettable experience that has made me, as a woman, even more confident to stand up for my ideas. Thank you for always believing in me and encouraging me to try new concepts.
I would also like to pay tribute to apl. Prof. Dr. techn. Franz Rottensteiner as the examiner of this dissertation. It was an honor to benefit from your constructive comments and to broaden my scientific attitude, which I will certainly benefit from in the future.
Special thanks to Prof. Dr.-Ing. habil. Christian Heipke, not only for the exchange of ideas during my doctoral thesis, but also for making my defense day a memorable one.
Every single colleague at the Geodetic Institute of Leibniz Universität Hannover (GIH) has had a considerable influence on the quality of my work. Thank you for the friendly atmosphere, which is more like a family than a strict working environment. Among my colleagues, a big thank you goes to Dr.-Ing. Sören Vogel, for whom I had the opportunity to work first as a student assistant and then to continue his research work. Your helpful guidance and feedback were always a big help and a key to making this thesis a success.
A warm thank you to Dr.-Ing. Johannes Bureick, who I had the opportunity to work with for a short time. Thank you for the helpful ideas and the provided scientific material, which played an important role in the next steps.
A big thank you to my colleague Dominik Ernst, from whom I learned (and still learn) many scientific topics. Thank you for always supporting me, especially during the writing phase of my thesis. Your helpful feedback played an important role in creating a successful product.
Special thanks to my colleague Johannes Link as IT administrator who provided the necessary tools and helpful guidance for my analyses. Thank you for always being patient with my questions and requests.
I would also like to express my endless gratitude to my family and friends who never left my side. With your constant support and love, I found the confidence and courage to take big steps and not lose sight of my ultimate goals. Thank you for always being there, for all the good moments you gave me and for standing by my side during the hard times. I love you and I am proud to have you. Thank you for everything.

Curriculum Vitae

Personal Information

Name:	Rozhin Moftizadeh
Date of Birth:	November 2nd , 1989
Place of Birth:	Kermanshah, Iran

Work Experience

since 10.2019	Geodetic Institute Leibniz Universität Hannover Research Assistant
11.2017 – 09.2019	Geodetic Institute Leibniz Universität Hannover Student Assistant

Education

12.2019 – 30.11.2023	DFG Research Training Group i.c.sens (RTG 2159) Leibniz Universität Hannover PhD student
10.2016 – 09.2019	Degree Course Geodesy and Geoinformatics Leibniz Universität Hannover Master of Science
10.2012 – 10.2014	Degree Course Earthquake Engineering Razi University of Kermanshah Master of Science
10.2008 – 08.2012	Degree Course Civil Engineering Razi University of Kermanshah Bachelor of Science

Wissenschaftliche Arbeiten der Fachrichtung Geodäsie und Geoinformatik der Leibniz Universität Hannover

(Eine vollständige Liste der Wiss. Arb. ist beim Geodätischen Institut, Nienburger Str. 1, 30167 Hannover erhältlich.)

Nr. 370	DBOUK, Hani:	Alternative Integrity Measures Based on Interval Analysis and Set Theory (Diss. 2021)
Nr. 371	CHENG, Hao:	Deep Learning of User Behavior in Shared Spaces (Diss. 2021)
Nr. 372	MUNDT Reinhard Walter:	Schätzung von Boden- und Gebäudewertanteilen aus Kaufpreisen bebauter Grundstücke (Diss. 2021)
Nr. 373	WANG, Xin:	Robust and Fast Global Image Orientation (Diss. 2021)
Nr. 374	REN, Le:	GPS-based Precise Absolute and Relative Kinematic Orbit Determination of Swarm Satellites under Challenging Ionospheric Conditions (Diss. 2021)
Nr. 375	XU, Wei:	Automatic Calibration of Finite Element Analysis Based on Geometric Boundary Models from Terrestrial Laser Scanning (Diss. 2021)
Nr. 376	FENG, Yu:	Extraction of Flood and Precipitation Observations from opportunistic Volunteered Geographic Information (Diss. 2021)
Nr. 377	YANG, Chun:	A hierarchical deep learning framework for the verification of geospatial databases (Diss. 2021)
Nr. 378	MEHLTRETTER, Max:	Uncertainty Estimation for Dense Stereo Matching using Bayesian Deep Learning (Diss. 2021)
Nr. 379	KAZIMI, Bashir:	Self Supervised Learning for Detection of Archaeological Monuments in LiDAR Data (Diss. 2021)
Nr. 380	PETERS, Torben:	Learning Multi-View 2D to 3D Label Transfer for Semi-Supervised Semantic Segmentation of Point Clouds (Diss. 2022)
Nr. 381	WASSINK, Martin:	Kommunal- und Regionalentwicklung durch Kooperation und Teilung von Verantwortung in ländlichen Räumen - eine multiperspektivische Untersuchung an Beispielen aus dem Raum Steinwald/Fichtelgebirge (Diss. 2022)
Nr. 382	GOLDSCHMIDT, Jürgen:	Die Berücksichtigung künftiger Entwicklungen bei der Verkehrswertermittlung (Diss 2022)
Nr. 383	KRUSE, Christian:	Impact maps from bomb craters detected in aerial wartime images using marked point processes (Diss. 2023)
Nr. 384	ZOURLIDOU, Stefania:	Traffic Regulation Recognition from GPS Data (Diss. 2023)
Nr. 385	SLEDZ, Artuom:	Thermal anomaly detection based on information fusion from optical and infrared images (Diss. 2023)
Nr. 386	WITTICH, Dennis:	Deep Domain Adaptation for the Pixel-wise Classification of Aerial and Satellite Images (Diss. 2023)
Nr. 387	SINGH, Vishwa Vijay:	Lunar Laser Ranging - Improved Modelling and Parameter Estimation (Diss. 2023)
Nr. 388	HARTMANN, Jens:	Hochgenaue 3D-Erfassung von Großstrukturen durch kinematisches terrestrisches Laserscanning (Diss. 2023)
Nr. 389	ZHUGE, Xia:	Characterizing slope instability kinematics by integrating multi-sensor satellite remote sensing observations (Diss. 2023)
Nr. 390	DOROZYNSKI, Mareike Marianne:	Image Classification and Retrieval in the Context of Silk Heritage using Deep Learning (Diss. 2023)
Nr. 391	KNABE, Annike:	New Concepts for Gravity Field Recovery using Satellites (Diss. 2023)
Nr. 392	KALIA, Andre:	Landslide activity detection based on nationwide Sentinel-1 PSI datasets (Diss. 2023)
Nr. 393	BROCKMEYER, Marco:	Modellierung von Bodenbewegungen anhand heterogener Messverfahren am Beispiel der niedersächsischen Landesfläche (Diss. 2023)
Nr. 394	ZHANG, Mingyue:	Characteristics and Benefits of Differential Lunar Laser Ranging (Diss. 2023)
Nr. 395	DENNIG, Dirk:	Entwicklung eines kinematischen Profilvermessungssystems am Beispiel Kranbahnvermessung (Diss. 2024)
Nr. 396	FUEST, Stefan:	Nudging travelers to societally favorable routes by means of cartographic symbolization (Diss. 2024)
Nr. 397	MOFTIZADEH, Rozhin:	Advanced Particle Filtering for Vehicle Navigation based on Collaborative Information (Diss. 2024)