DGK Ausschuss Geodäsie (DGK)
der Bayerischen Akademie der Wissenschaften

Reihe C Dissertationen Heft Nr. 981

Dominik Ernst

Uncertainty-Aware Positioning for LiDAR-Based Multi-Sensor Systems

München 2026

Verlag der Bayerischen Akademie der Wissenschaften, München

ISSN 0065-5325 ISBN 978 3 7696 5393 9

DGK Ausschuss Geodäsie (DGK)
der Bayerischen Akademie der Wissenschaften

Reihe C Dissertationen Heft Nr. 981

Uncertainty-Aware Positioning for LiDAR-Based Multi-Sensor Systems

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

von

Dominik Ernst, M. Sc.

München 2026

Verlag der Bayerischen Akademie der Wissenschaften, München

ISSN 0065-5325 ISBN 978 3 7696 5393 9

Adresse des Ausschusses Geodäsie (DGK)
der Bayerischen Akademie der Wissenschaften:

Ausschuss Geodäsie (DGK) der Bayerischen Akademie der Wissenschaften

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

Telefon +49 – 89 – 23 031 1113 ● Telefax +49 – 89 – 23 031 - 1283 / - 1100
e-mail post@dgk.badw.de ● http://www.dgk.badw.de

Prüfungskommission:

Vorsitzende:	Prof. Dr.-Ing. habil Monika Sester
Referent:	apl. Prof. Dr.-Ing. Hamza Alkhatib
Korreferenten:	Prof. Dr.-Ing. habil Dr. h.c. Volker Schwieger, Universität Stuttgart
	Prof. Dr.-Ing. Steffen Schön

Tag der mündlichen Prüfung: 30.03.2026

:

Diese Dissertation ist auf dem Server des Ausschusses Geodäsie (DGK)
der Bayerischen Akademie der Wissenschaften, München unter <http://dgk.badw.de/>
sowie unter Wissenschaftliche Arbeiten der Fachrichtung Geodäsie und Geoinformatik
der Leibniz Universität Hannover (ISSN 0174-1454), Nr. 422,
<https://repo.uni-hannover.de/handle/123456789/22327>, Hannover 2026, elektronisch publiziert

ISSN 0065-5325 ISBN 978 3 7696 5393 9

Für meine Mutter

Abstract

In many applications, system positioning is crucial for executing tasks effectively. In particular, autonomous systems must accurately determine their position while simultaneously monitoring it under consideration of the estimated uncertainties to ensure safe operation. However, achieving accurate positioning can be challenging depending on the environment and specific requirements. Although Inertial Navigation Systems (INS) are well-established in numerous applications, their limitations can impede further adoption. Without clear sky visibility, Global Navigation Satellite System (GNSS) observations can yield inaccurate position solutions, potentially compromising safety and performance. In the field of robotics, alternative approaches are widespread. Inertial Measurement Units (IMUs) and odometry are commonly employed with additional sensors such as cameras and Light Detection and Ranging (LiDAR) used to enhance position estimation. To fully leverage the advantages of all sensors, data fusion processes must accurately weight information from each sensor. For example, the measurement quality of LiDARs is influenced by atmospheric conditions, the characteristics of the measured object, and the sensor itself. Consequently, when integrating LiDAR data into the positioning process, measurement uncertainty directly impacts positioning accuracy.

This cumulative dissertation addresses the challenge of fusing LiDAR data with other sensor measurements at various stages of the processing chain. Sensor data fusion requires extrinsic calibration. When LiDAR measurements are involved in the calibration process, the inherent measurement uncertainty influences the calibration results. During sensor data fusion, achieving an optimal combination of information is crucial for deriving the best positioning solution. This involves two key aspects: maintaining time synchronization throughout the fusion process and appropriately weighting observations according to their uncertainties. For LiDAR specifically, the material properties of measured objects significantly impact measurement uncertainty. Therefore, adjusting the weighting of observations during positioning is essential to maximize accuracy. Finally, the results must be evaluated against ground truth data to validate their accuracy and reliability with respect to their uncertainties. The contributions of this dissertation are outlined as follows:

1. A comprehensive procedure for both intrinsic and extrinsic LiDAR calibration is presented. This procedure includes determining the transformation parameters between the LiDAR and its platform, applying corrections to the LiDAR observations, and utilizing Variance Component Estimation (VCE) to estimate the variance-covariance structure of the sensor.
2. Enhancements to sensor data fusion within an Error State Kalman Filter (ESKF) are achieved by implementing pose interpolation. This technique reduces errors associated with map matching by deskewing LiDAR point clouds. Additionally, an alternative procedure is introduced for propagating measurement uncertainties throughout the entire processing chain.
3. A dual Kalman filter approach is introduced to update the stochastic model of observations during operation. The second state vector tracks estimated variance components, serving as a regularization mechanism during the initial epochs of estimation.

The experiments demonstrate that incorporating the estimation of intrinsic parameters significantly enhances the calibration process. Further improvements are reached by implementing pose interpolation, nearly halving deviations depending on the system's speed. Furthermore, the estimated stochastic model provides valuable information for further steps. While the uncertainties for the angle measurements from the calibration prove applicable to other contexts, the corrections and uncertainties for the distance measurements are estimated in a dual filtering approach. The evaluation is performed in simulation and using real datasets, for which obtaining sufficiently accurate reference data proves challenging. The dual filtering approach successfully reduces the deviations in the simulations by up to 50% and consistency measure in the real dataset by up to 30%. For the real dataset, the recursive estimation of uncertainties and corrections yields plausible results.

Keywords: Gauß-Helmert model, Kalman filter, variance component estimation, adjustment, sensor data fusion

Kurzfassung

In vielen Anwendungen ist die genaue Positionierung von Systemen entscheidend für die erfolgreiche Ausführung von Aufgaben. Besonders autonome Systeme benötigen eine genaue Positionsschätzung unter Beachtung der Unsicherheiten, um einen sicheren Betrieb zu gewährleisten. Dabei kann es je nach Umgebung und spezifischen Anforderungen eine Herausforderung sein, die benötigte genaue Positionierung zu erreichen. Obwohl Inertiale Navigationssysteme (INS) in zahlreichen Anwendungen bereits etabliert sind, behindern ihre Einschränkungen die umfängliche Nutzung. Ohne uneingeschränkte Sicht auf den Himmel können Beobachtungen von Globalen Navigationssatellitensystemen (GNSS) zu ungenauen Positionslösungen führen, was die Sicherheit und Zuverlässigkeit des Systems beeinträchtigen kann. Eine Verbesserung der Positionsschätzung kann erreicht werden, indem Methoden aus der Robotik zum Einsatz kommen, bei denen zusätzliche Sensoren wie Kameras und Light Detection and Ranging (LiDAR) als Unterstützung zu Odometrie und Inertialen Messeinheiten (IMU) eingesetzt werden. Um die Vorteile aller Sensoren voll auszuschöpfen, müssen Datenfusionsprozesse die Informationen jedes Sensors korrekt gewichten. So wird beispielsweise die Messqualität von LiDARs durch atmosphärische Bedingungen, die Eigenschaften des gemessenen Objekts und den Sensor selbst beeinflusst. Dementsprechend haben die Messunsicherheiten der LiDAR-Daten einen direkten Einfluss auf die Genauigkeit im Positionierungsprozess.

Diese kumulative Dissertation widmet sich der Herausforderung, die Verarbeitungskette für die Fusion von LiDAR-Daten mit anderen Sensoren zu verbessern. Die Sensordatenfusion erfordert eine extrinsische Kalibrierung. Wenn LiDAR-Messungen im Kalibrierungsprozess einbezogen werden, beeinflusst die Sensorunsicherheit die Kalibrierungsergebnisse. Während der Sensordatenfusion ist es entscheidend, eine optimale Kombination der Informationen zu erreichen, um die bestmögliche Positionierung zu erzielen. Dies umfasst zwei Schlüsselaspekte: die Zeitsynchronisation über den gesamten Fusionsprozess hinweg und die angemessene Gewichtung der Beobachtungen gemäß ihrer Unsicherheiten. Speziell für LiDAR wirken sich die Materialeigenschaften der gemessenen Objekte erheblich auf die Messunsicherheit aus. Daher ist es wichtig, die Gewichtung der Beobachtungen während der Positionierung anzupassen, um die Genauigkeit zu maximieren. Abschließend müssen die Ergebnisse mit Referenzdaten verglichen werden, um ihre Genauigkeit und Zuverlässigkeit in Bezug auf ihre Unsicherheiten zu evaluieren. Die wesentlichen Beiträge dieser Dissertation können wie folgt zusammengefasst werden:

1. Ein Verfahren für die intrinsische und extrinsische LiDAR-Kalibrierung wird vorgestellt. Es umfasst die Bestimmung der Transformationsparameter zwischen LiDAR und Plattform, die Korrektur von LiDAR-Beobachtungen sowie die Nutzung einer Varianzkomponentenschätzung, um die Varianz-Kovarianz-Struktur des Sensors zu schätzen.
2. Verbesserungen der Sensordatenfusion innerhalb eines Error-State-Kalman-Filters (ESKF) werden durch die Implementierung der Posen-Interpolation erreicht. Diese Technik reduziert Fehler, die bei der Positionierung mit Lokalisierungskarten verbunden sind, indem sie LiDAR-Punktwolken entzerrt. Zudem wird ein alternatives Verfahren zum Fortpflanzen von Messunsicherheiten über die gesamte Verarbeitungskette eingeführt.
3. Ein dualer Kalman-Filter-Ansatz wird eingeführt, um das stochastische Modell der Beobachtungen während der Filterung zu aktualisieren. Der zweite Zustandsvektor umfasst die geschätzten Varianzkomponenten und dient als Regularisierung zu Beginn der Schätzung.

Die Experimente zeigen, dass die Einbeziehung der intrinsischen Parameter in die Schätzung den Kalibrierungsprozess erheblich verbessert. Die Implementierung der Posen-Interpolation führt zu einer verbesserten Positionierungsgenauigkeit, bei der die Abweichungen in Abhängigkeit von der Geschwindigkeit nahezu halbiert werden. Darüber hinaus erweist sich das geschätzte stochastische Modell als wertvoll für die weitere Verarbeitung. Während die Unsicherheiten für die Winkelmessung aus der Kalibrierung übertragbar sind, werden die Korrekturen und Unsicherheiten für die Distanzmessung in einem dualen Filteransatz weitergeschätzt. Die Evaluation wird mittels Simulationen und Echtdaten durchgeführt, wobei bei den Echtdaten keine Referenzdaten in der notwendigen Genauigkeit zur Verfügung stehen. Der duale Filteransatz verringert die Abweichungen in den Simulationen um bis zu 50% und verbessert die Konsistenz in den Echtdaten um bis zu 30%.

Die rekursive Schätzung von Unsicherheiten und Korrekturen liefert plausible Ergebnisse bei den Echtdaten.

Schlagwörter: Gauß-Helmert Modell, Kalman Filter, Varianzkomponentenschätzung, Ausgleichung, Sensordatenfusion

Abbreviations

6-DoF	six degrees of freedom	12
ALKIS	Amtliche Liegenschaftskatasterinformationssystem	49
AOI	angle of incidence	9
C2C	cloud to cloud	59
CCR	corner cube reflector	11
DHHN	Deutsches Haupthöhennetz	49
DTM	digital terrain model	9
EKF	extended Kalman filter	7
EM	expectation maximization	16
ESKF	error state Kalman filter	16
FLU	forward-left-up	60
FoV	field of view	20
FGO	factor graph optimization	9
GCP	ground control point	59
GNSS	Global Navigation Satellite System	1
GPS	Global Positioning System	10
GHM	Gauß-Helmert model	10
GIS	geographic information system	9
GUM	Guide to the Expression of Uncertainty in Measurement	23
GMM	Gauß-Markov model	14
HD	high definition	10
ICP	iterative closest point	10
IMU	inertial measurement unit	1
INS	inertial navigation system	1
KF	Kalman filter	10
LiDAR	light detection and ranging sensor	1
LOD	level of detail	9
LUCOOP	Leibniz University Cooperative Perception and Urban Navigation Dataset	17
LT	laser tracker	8
M3C2	multiscale model-to-model cloud comparison	11
MAP	maximum a-posteriori estimator	16
MC	Monte Carlo	23
ML	machine learning	69
MMS	mobile mapping system	9

Contents

1 Introduction

1.1 Motivation

Modern society relies on numerous systems that either assist human operators or function with full autonomy. These systems range from vacuum cleaners in homes to logistic robots in warehouses and large vehicles in industrial or public spaces. This wide array of applications highlights their significance. In each case, accurately positioning these systems is crucial to their successful operation. While industries continue to drive automation forward to reduce costs and workforce demands, ensuring safety remains a paramount concern. Larger systems, or those operating in environments shared with humans and other actors, must ensure the safety of both the system and its surroundings. Safety can only be assured if the systems' positions and orientations are known at all times. This is particularly evident in public traffic scenarios, where different transportation modes interact. Recently, there has been a significant push towards autonomous driving, with many manufacturers enhancing assistance functions and offering some fully autonomous driving options (Mallozzi et al., 2019; Othman, 2021). Although fully autonomous driving in Germany is currently restricted to highways (Mercedes-Benz Group, 2022), other countries permit autonomous vehicles in diverse areas, including urban centers (Waymo, 2025). This presents a host of challenges. Urban environments yield a wide variety of potential scenarios involving other traffic participants at relatively low speeds. Conversely, roads outside urban areas pose increased dangers due to higher speeds. In any scenario, vehicles must adhere to their lanes and follow applicable traffic regulations while avoiding other participants.

To address these challenges, combining different sensors has become a preferred strategy. As a result, multi-sensor systems (MSSs) are commonly employed for both positioning systems and detecting obstacles in kinematic systems. The selection of sensors for positioning depends on the specific application. In outdoor environments, inertial navigation system (INS) are prevalent. By integrating inertial measurement unit (IMU) with Global Navigation Satellite System (GNSS), a balance is struck between the inherent limitations of each sensor type. IMUs provides high measurement rates but suffers from sensor drift over extended periods (Titterton & Weston, 2004). Conversely, GNSS delivers precise global positioning solutions, with newer systems offering high-frequency position updates (ublox, 2025; JAVAD, 2025). The primary limitation of GNSS lies in its requirement for unobstructed sky visibility. Positional accuracy diminishes due to obstructions or reflections, whether from natural or man-made structures (Hsu, 2018), complicating the use of traditional INSs in urban settings. Indoors, and increasingly in other applications, many systems incorporate cameras, light detection and ranging sensors (LiDARs), or a combination of both to facilitate positioning (Pöppl et al., 2023). Their independence of external information and comparatively high temporal stability make these sensors valuable for this purpose. These exteroceptive sensors are naturally well-suited for detecting obstacles and other traffic participants as well. LiDAR, in particular, offers a direct method for acquiring 3D environmental information. Its active measurement principle enables functionality independent of external lighting conditions. In controlled settings like warehouses or offices, point clouds generated by LiDAR can create maps or position systems with sufficient accuracy for most tasks (Nüchter et al., 2018). However, transferring these systems to vehicles operating in outdoor environments introduces new challenges. To enhance positional reliability, pre-constructed maps serve as a fixed reference for estimation. Vehicles travel at higher speeds in more variable conditions, influenced by diverse weather factors. LiDARs are sensitive to the surfaces they measure and the geometry involved. Highly reflective or wet surfaces present issues (Velodyne, 2019b), as do measurements taken at steep angles (Soudarissanane et al., 2009). These factors elevate the requirements for data fusion. Achieving lane-level accuracy of 10 cm with 95% confidence (Reid et al., 2019) necessitates not only high sensor precision but also the proper synchronization and weighting of data from multiple sensors.

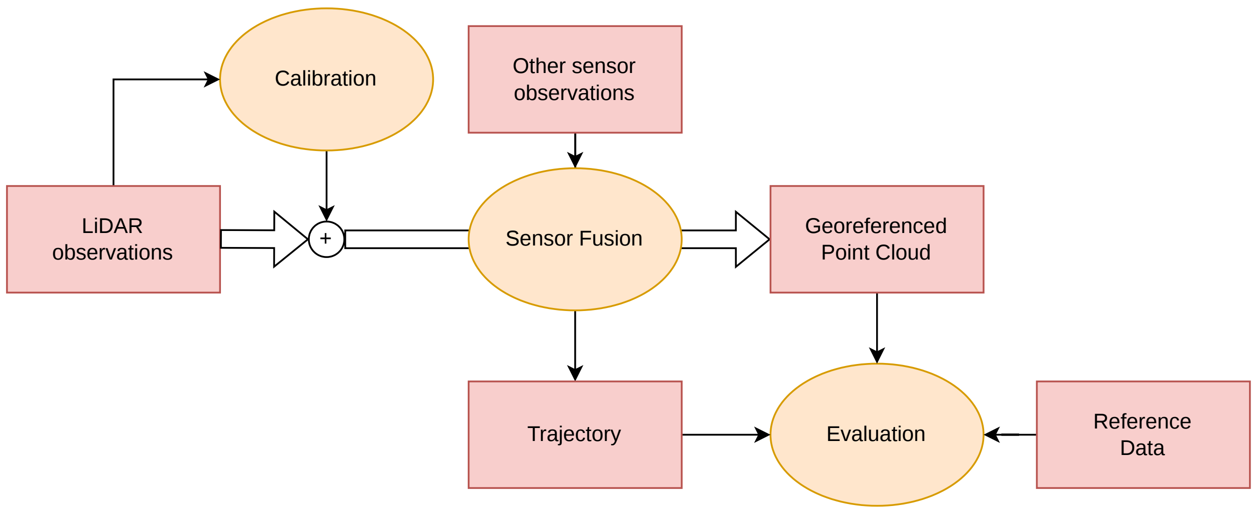

Figure 1.1: *Overview of sensor fusion process for LiDAR-based MSS.*

To enable sensor data fusion that satisfies the outlined requirements, several essential steps must be carried out. These steps are illustrated in Figure 1.1. With a focus on LiDARs, calibration is necessary before deploying the system. During the data fusion, LiDAR observations are fused with data from other sensors, resulting in an estimated trajectory. This trajectory or a georeferenced point cloud derived from it can be used to evaluate the results. Numerous studies have explored this issue (Vennegeerts, 2011; Durrant-Whyte & Henderson, 2016; Heinz, 2021; Hartmann, 2023), each contributing to various aspects of the processing chain.

This dissertation examines the entire processing chain and investigates how the process can be improved, particularly focusing on the uncertainties introduced by the LiDAR. The widely used low-cost LiDARs introduce significant uncertainties of unknown magnitude. Addressing these uncertainties is essential to achieve optimal results. Where applicable, insights from higher accuracy terrestrial laser scanners (TLSs) and MSSs are applied.

In the initial step, **calibration**, the first uncertainties are accumulated, which affect the entire remaining processing. Therefore, accurate calibration of the system, encompassing all sensors, is a fundamental prerequisite for effective fusion (Glennie, 2007). Specifically, it is necessary to determine the transformation parameters between the sensors and the platform. Depending on the sensors used, manufacturer-provided information may not suffice for achieving optimal fusion results. Thus, intrinsic calibration of the sensor (Glennie et al., 2016; Bergelt et al., 2017; Sánchez & Pany, 2019) becomes crucial, as it enhances the outcomes of extrinsic calibration when the sensor's measurements are incorporated in the process (Heinz et al., 2015; Kümmerle & Kuhner, 2020; Lv et al., 2022). The weighting of observations from different sensors is typically based on their random measurement uncertainties, making the stochastic model of these sensors a crucial component in the uncertainty modeling. Consequently, procedures for a combined calibration of extrinsic and intrinsic parameters, including uncertainties, are necessary.

This information is subsequently utilized in **sensor data fusion**. Here, the motion of the system in combination with the various uncertainties poses challenges. Given the system's movement during operation, compensating for this motion is essential when georeferencing the LiDAR points. Various deskewing techniques for LiDAR are available, each offering different levels of flexibility depending on the application (Qin et al., 2022; Mounier et al., 2025; Hartmann, 2023). As sensors might exhibit temporal instability and scanning behavior influenced by the geometry and material of scanned objects, methods to adapt corrections and the stochastic model during operations are essential. There are existing strategies for elements of uncertainty modeling during operation, such as identifying issues between stochastic and functional models (Ettlinger et al., 2018), detecting reflected observations (Kim & Chung, 2018), or updating the stochastic model (Hu et al., 2025). A comprehensive modeling approach, updated during operation, is missing so far.

Lastly, the **evaluation** of the results is crucial to acquire quality measures. Various techniques are documented in the literature: some apply to point clouds generally, and others specifically address kinematically acquired point clouds (Stenz et al., 2020; Hartmann, 2023). Introducing the uncertainties of the entire processing chain strengthens the significance of the results, although it is often not performed rigorously. In the context of low-cost LiDARs, it is crucial to assess the accuracy of results relative to sensor and system uncertainties. This dissertation demonstrates the entire processing chain based on two datasets, one from a robot in a laboratory environment and one from a vehicle in an urban environment (Axmann et al., 2023). Section 1.2 will outline the research questions that underpin this thesis.

1.2 Research Questions

The objective of this thesis is to achieve a consistent end-to-end sensor fusion chain for kinematic LiDAR-based MSSs, covering all relevant steps. This is demonstrated through a position tracking application that matches LiDAR observations with a pre-existing environmental map while accounting for all system uncertainties. Current literature presents unresolved issues relevant for this application, which this work addresses. One significant issue is the calibration of an MSS, where the integration of extrinsic and intrinsic calibration for LiDARs often tends to omit either the stochastic model or the sensor-specific corrections. A consistent data fusion process depends on several critical factors. Firstly, the asynchronously acquired data must be processed accurately, necessitating a strategy to compensate for movement during the scanning process and to deskew the point clouds in a recursive state estimation appropriately. Secondly, the measurement uncertainty of LiDARs is influenced by environmental conditions. A particularly notable factor is the nature of the scanned object and its material. Different sensors can respond differently to the same material and may even exhibit temporal drift in their measurements, thus requiring an online estimation process to appropriately weight and correct observations. Finally, the outcomes need to be evaluated with an understanding of the estimated uncertainties inherent in the fusion process.

This leads to the following research questions:

- **Research question 1**: How can the consistent estimation for the calibration parameters of a LiDAR be accomplished?
- **Research question 2**: How can the processing pipeline for referencing kinematically acquired point clouds be modeled to provide consistent predictions of point uncertainties?
- **Research question 3**: How can remaining systematic deviations of an MSS be estimated from kinematically acquired point clouds, and can these deviations be assigned to specific components of the model?

An overview of the relevant publications, including their context with respect to the research questions, is given in Figure 1.2. While this dissertation is cumulative, summarizing publications **A** to **D**, Chapter 4 includes additional investigations contributing to answering the research questions **3**.

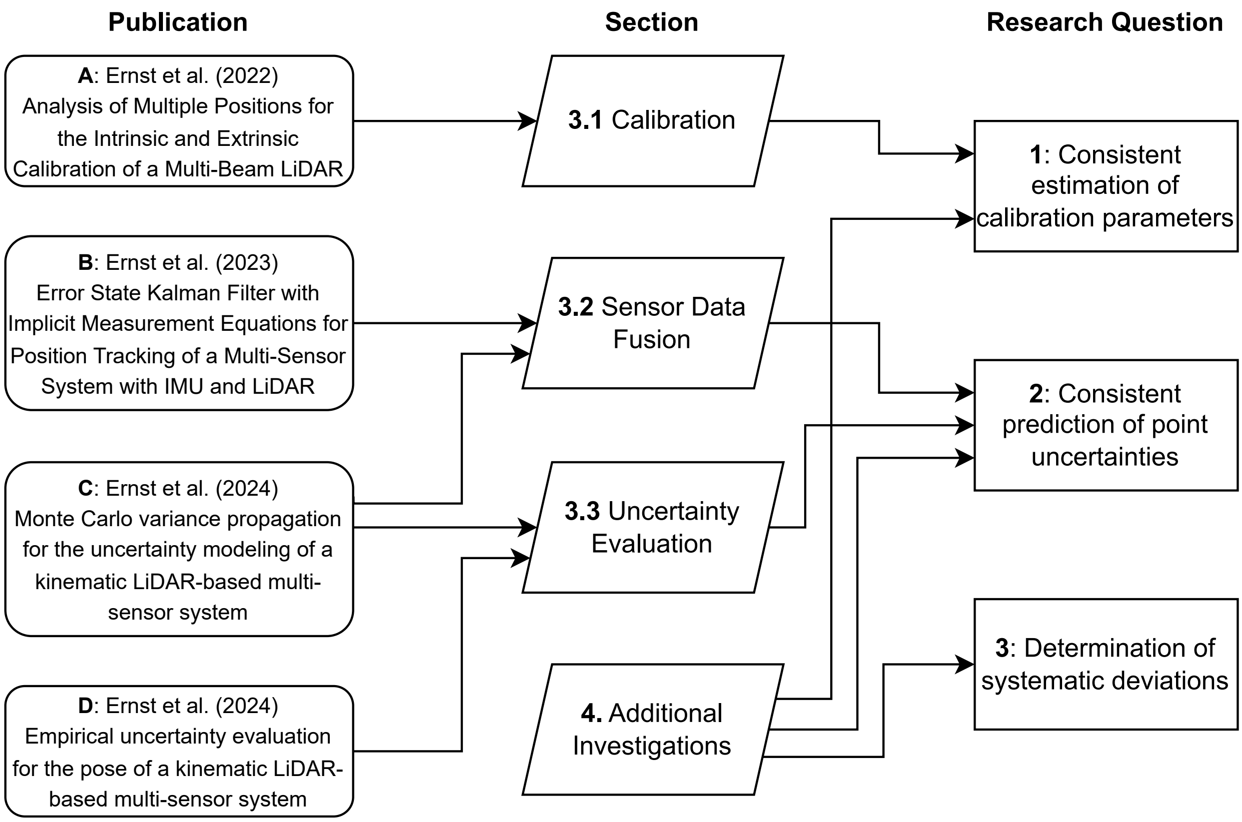

Figure 1.2: *Overview of the publications for this cumulative thesis. The flowchart shows how each publication addresses specific research questions. Arrows indicate the content of the publications contributing to the sections in this chapter and the next.*

1.3 List of Publications and Outline

The presented cumulative dissertation contains methodological contributions and investigations concerning the uncertainty modeling of kinematic LiDAR-based MSSs. The relevant publications of the author are listed in Chapter 7. These publications were all checked by peer-review and published in well known journals or conferences. The relation with respect to the research questions in presented in Figure 1.2. Four publications set the foundation for this dissertation:

- **Publication A:**
  Ernst, D.; Alkhatib, H., Neumann, I. & Vogel, S. (2022). Analysis of Multiple Positions for the Intrinsic and Extrinsic Calibration of a Multi-Beam LiDAR. In *2022 25th International Conference on Information Fusion (FUSION)*, Linköping, Sweden, 1 - 8.
  `https://doi.org/10.23919/FUSION49751.2022.9841366`

- **Publication B:**
  Ernst, D., Vogel, S., Neumann, I. & Alkhatib, H. (2023). Error State Kalman Filter with Implicit Measurement Equations for Position Tracking of a Multi-Sensor System with IMU and LiDAR. In *2023 13th International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Nuremberg, Germany, 1 - 6.
  `https://doi.org/10.1109/IPIN57070.2023.10332480`

- **Publication C:**
  Ernst, D.; Vogel, S., Alkhatib, H. & Neumann, I. (2024). Monte Carlo variance propagation for the uncertainty modeling of a kinematic LiDAR-based multi-sensor system. *Journal of*

Applied Geodesy, 18(2), 237–252.
https://doi.org/10.1515/jag-2022-0033

- **Publication D:**
  Ernst, D., Vogel, S., Neumann, I. & Alkhatib, H. (2024). Empirical uncertainty evaluation for the pose of a kinematic LiDAR-based multi-sensor system. *Journal of Applied Geodesy,* 18(4), 629–642.
  https://doi.org/10.1515/jag-2023-0098

The individual contributions for these publications are as follows:

Publication A

- **D. Ernst (main author)**: Conception and design of study, conducting the experiment, data analysis, writing the manuscript, correspondence during review process.
- **H. Alkhatib**: Review and checking of manuscript, methodological advice.
- **I. Neumann**: Review and checking of manuscript.
- **S. Vogel**: Review and checking of manuscript, methodological advice and technical expertise.

Publication B

- **D. Ernst (main author)**: Conception and design of study, coding of simulation environment, conducting the experiment, data analysis, writing the manuscript, correspondence during review process.
- **S. Vogel**: Review and checking of manuscript, methodological advice and technical expertise.
- **I. Neumann**: Review and checking of manuscript.
- **H. Alkhatib**: Review and checking of manuscript, methodological advice.

Publication C

- **D. Ernst (main author)**: Conception and design of study, conducting the experiment, data analysis, writing the manuscript, correspondence during review process.
- **S. Vogel**: Review and checking of manuscript, methodological advice and technical expertise.
- **H. Alkhatib**: Review and checking of manuscript, methodological advice.
- **I. Neumann**: Review and checking of manuscript, methodological advice and technical expertise.

Publication D

- **D. Ernst (main author)**: Conception and design of study, conducting the experiment, data analysis, writing the manuscript, correspondence during review process.
- **S. Vogel**: Review and checking of manuscript, methodological advice and technical expertise.
- **I. Neumann**: Review and checking of manuscript, methodological advice and technical expertise.
- **H. Alkhatib**: Review and checking of manuscript, methodological advice.

These publication set the foundation to answering the first two research questions (c.f. Section 1.2). Publication **A** addresses the calibration of LiDARs on MSS platforms. Publications **B** and **C** deal with the sensor data fusion necessary for the second research questions. Additionally, publications **C** and **D** expand on the uncertainty evaluation to check the consistency of the results. In Chapter 3, the contents of the four listed publications will be summarized and the main aspects identified. Remaining issues of the relevant publications and the third research question are addressed in the additional investigations (c.f. Chapter 4). The thesis continues with the conclusion and outlook in Chapter 5. Chapter 6 lists more publications, which are not directly relevant to this thesis. In Chapter 7, the four relevant publications are included. The appendix in Chapter 8 includes the algorithms used.

2 Scientific Context

This chapter situates the thesis within its broader scientific context and provides the conceptual basis for the subsequent analyses. As multi-sensor fusion is a widely used strategy, a wide array of works can be found in the literature, focusing on different aspects of the process. Section 2.1 provides an overview of relevant works highlighting aspects related to this thesis. Afterwards, the necessary prerequisites for understanding this thesis are introduced. First, Section 2.2 introduces the necessary notation and terms used. Finally, the utilized datasets are presented in Section 2.3, ranging from a fully simulated dataset to two real datasets from an indoor and an urban dataset.

2.1 State of the Art

Multi-sensor fusion is a complex task with numerous applications, alongside significant challenges. Consistent with the overall focus of this dissertation, the emphasis is on LiDAR-based MSSs. While systems incorporating other exteroceptive sensors, such as cameras or radar, are acknowledged, they are discussed only when relevant. This section aims to provide an overview of the current state of the art concerning the aspects of this thesis. While the relevant papers each contain their respective state of the art, an overview is provided here, with a special focus on the additional investigations in Chapter 4. Due to the complexity of the topic, the overview here has no claim to be exhaustive. Initially, the spotlight is on the calibration of MSSs in Subsection 2.1.1. Subsequently, Subsection 2.1.2 examines data fusion for georeferencing systems. The final Subsection 2.1.3 addresses the evaluation of MSSs.

2.1.1 Calibration

The calibration of MSSs can be approached in various ways. Given that an MSS comprises multiple sensors mounted on a shared platform or positioned in proximity, their relative translation (lever arms) and orientation (bore angles) must be known to process sensor data consistently. Determining the transformation parameters between individual sensors is essential for every MSS and constitutes the extrinsic calibration. Additionally, intrinsic calibration can be conducted to correct the measurements of individual sensors. This process is common across various sensor types, such as cameras (Luhmann, 2000) or GNSS antennas (Kröger et al., 2021), and is equally applicable to LiDARs, which are of particular interest in this dissertation. This section addresses both extrinsic and intrinsic calibrations, as well as their combination.

Extrinsic calibration can be conducted either between the frames of two sensors or between a sensor and a designated platform frame. Utilizing a separate platform frame can be beneficial, especially when sensors are frequently swapped or are inaccessible due to the platform's configuration. Extrinsic calibration between sensors can involve various sensor combinations, such as LiDAR-camera (Zhang et al., 2025) or LiDAR-IMU. The LiDAR-camera scenario is not covered here, as cameras are not consistently included in kinematic systems, unlike IMUs, which are integral to such systems. For smaller setups, Mishra et al. (2021) propose a method to kinematically calibrate a LiDAR and an IMU using an extended Kalman filter (EKF), where the system moves through a targetless environment on a trajectory, and both the trajectory and transformation parameters are optimized simultaneously. In larger environments, Li et al. (2023) implement a vehicle-mounted configuration, incorporating additional raw GNSS observations and performing kinematic measurements to jointly estimate trajectory and transformation parameters. In this instance, the height component is disregarded as the vehicle operates on a horizontal plane. For calibrating a LiDAR with a platform frame, several approaches exist. In static scenarios, where the platform is oriented in specific ways within a calibration environment containing dedicated reference geometries, Strübing & Neumann (2013) presents a method for a rail surveying vehicle, demanding high precision profile scanners due to the application's stringent requirements. The reference geometries are recorded using a hand scanner to meet these standards. Other researchers adopt similar strategies with varied types of reference geometries: Hartmann et al. (2017) use a laser tracker (LT) to capture reference panels, while Vennegeerts (2011) employs a close-range photogrammetry setup for reference information. This calibration principle demands careful design of the calibration field to achieve optimal results (Hartmann et al., 2019). Kinematic calibration between LiDAR and a platform is also possible. Hillemann et al. (2019) use geometric features as a measure of point cloud consistency in targetless environments, optimizing by computing a georeferenced point cloud based on a predefined trajectory and minimizing feature values through adjustments in extrinsic calibration. Esser et al. (2024) use a similar approach, specifically focusing on omnivariance, catering to applications in agricultural environments where stable features are absent, and thus, reference geometries are strategically positioned in the scene. In this scenario, additional observations from a total station and TLS enhance trajectory estimates. In all these instances, the precision of trajectory estimates significantly influences the quality of the calibration results.

Intrinsic calibration is essential for applications with stringent accuracy requirements or for sensors lacking manufacturer-provided calibration. Geodetic sensors can undergo calibration to meet high accuracy standards, whether for all measurement components, as with TLS (Medic, 2021), or specifically for distance measurement in kinematic tasks (Hartmann et al., 2023). In kinematic applications, calibrating smaller LiDARs is generally beneficial, as manufacturers often do not offer calibration, or the calibration must be performed regularly. Glennie et al. (2016), Bergelt et al. (2017) and Sánchez & Pany (2019) explore the intrinsic calibration of Velodyne Pucks. Sánchez & Pany (2019) employ a laboratory setup for precise reference data, whereas Glennie et al. (2016) utilize an open environment with planar structures and various positions to derive corrections. Both studies confirm the manufacturer's specifications for distance measurements, yet Glennie et al. (2016) note instability in the distance offset over time. Similar intrinsic calibration methodologies exist for both older (Glennie & Lichti, 2010) and newer scanner models, such as the approach presented by Brazeal et al. (2021) for corrections using Ripley prisms. In addition to functional corrections, a sensor's stochastic model is crucial for subsequent processing tasks. For geodetic sensors, the intensity-based stochastic model for distance measurements is well-documented and broadly applicable, ranging from single TLSs (Wujanz, 2016) to calibrations across various sensor types (Schill et al., 2025) and other scanners like small profile laser scanners for unmanned aerial vehicles (UAVs) (Dreier et al., 2024). However, for lower-cost mobile LiDARs, these models are less applicable due to uncalibrated and quantized intensity information (Velodyne, 2019b). Studies focusing on calibrating LiDAR intensity data typically emphasize high-contrast scenes for object distinction without relying on prior measurements (Levinson & Thrun, 2010), thus hindering the creation of stochastic models akin to those used for TLSs.

To conclude this section, combined calibration approaches are considered. For calibrating two LiDARs, Kim et al. (2021) propose a stop-and-go method that estimates intrinsic parameters simultaneously during optimization. This process involves capturing a trajectory and using single frames to determine parameters based on planar structures in the environment. Similarly, Glennie & Lichti (2010) calibrates a Velodyne HDL-64E on a platform while concurrently estimating the intrinsic parameters. This is achieved from multiple positions to enhance the adjustment's constellation. Without addressing corrections, Heinz et al. (2015) calibrates a platform with a Hokuyo profile laser scanner using a static setup, where variance component estimation (VCE) is employed due to the absence of sensor stochastic model information. Similarly, Khami et al. (2025) calibrate a UAV platform, including the distance offset of the used Velodyne Puck. Heinz (2021) conducts a combined kinematic calibration, utilizing a vehicle-mounted platform in an outdoor calibration field to estimate both extrinsic and intrinsic parameters. In this case, the stochastic model is fixed because the profile laser scanner employed is surveying-grade. Lv et al. (2022) present an observability-aware intrinsic and extrinsic calibration procedure for LiDAR-based MSS. A larger trajectory is analyzed, and only relevant parts are used to estimate the parameters. Levinson & Thrun (2014) perform kinematic extrinsic and intrinsic calibration for a Velodyne HDL-64E, including remittance calibration per scan line, but omit the stochastic model estimation. In summary, there appears to be no existing calibration procedure that simultaneously estimates extrinsic and intrinsic parameters while including a stochastic model, whether by VCE or another method. This presents challenges for systems using low-cost LiDARs, as manufacturers do not provide detailed information regarding their measurement uncertainties. Additionally, LiDAR distance measurements are affected by the reflective properties of the measured surfaces, typically yielding higher uncertainties for darker objects (Wujanz et al., 2017). Since the intensity-based stochastic model cannot be applied to low-cost LiDARs, as the intensity information is scaled and quantized (usually to 8 bits), there is a necessity for an alternative approach. Soudarissanane et al. (2009) and Zámečníková & Neuner (2018) show deviations based on the measurement constellation, i.e. angle of incidence (AOI). In further investigations, Hartmann et al. (2023) demonstrate material-dependent systematic deviations. Accounting for these effects during positioning is an open issue, too.

2.1.2 Sensor Data Fusion

Once the necessary parameters for sensor data fusion are established, the positioning of the system can be computed. This dissertation focuses on LiDAR-based MSSs that track their position over time, meaning the system's initial state is known and the pose is continuously updated; in short, position tracking. Even when the original position is known, GNSS remains valuable, as its frequent absolute position updates help mitigate sensor drift, especially in IMUs. In the absence of reliable GNSS data, alternative techniques become essential. The reliability of GNSS data can be compromised by different influences, such as occlusions or reflections (Hsu, 2018). These influences often occur in urban environments, in which accurate positioning is crucial to ensure the safety of operation.

These techniques can be broadly classified into simultaneous localization and mapping (SLAM) and map matching. SLAM involves detecting environmental features and building a map to locate the system within this construct. Geneva et al. (2018) utilizes planes detected in kinematic LiDAR scans. Their solution relies on factor graph optimization (FGO), minimizing the closest point-plane distances between the point cloud and the pre-existing map. Other methods, such as the one by Zuo et al. (2019), integrate multiple sensors, such as LiDAR, cameras, and IMU, to track sparse environmental features. Their integration of asynchronous data results in an optimal outcome. SLAM can reduce drift through additional sensor inputs and possible loop closure (Debeunne & Vivet, 2020), yet it remains limited to tracking from the initial position. This can present issues if the original position's accuracy is compromised (Ghallabi et al., 2018). Map matching involves using prior environmental information to guide the localization process, serving as an absolute reference that enhances accuracy and reduces drift. The type of map can vary. For instance, Toth et al. (2009) leverage different data sources to locate an aerial system by matching airborne laser scans and aerial images. The images are compared with data from geographic information system (GIS) databases, while the scans are matched to digital terrain models (DTMs) of the area. This fusion is conducted within an EKF, which integrates the results back into the assignment steps for the individual sensor data. Alternatively, city models can be employed to facilitate localization. Klein & Filin (2011) use building edges detected in mobile mapping system (MMS) scans to enhance system translation during GNSS outages. Bureick et al. (2019) utilize the planar features of a level of detail (LOD)2 city model and a DTM to locate a UAV equipped with LiDAR, IMU, and GNSS. Their work uses a simulated environment to test the algorithm and demonstrate the LiDAR data's benefits in georeferencing. The problem of LOD2 city model is the strong generalization of the geometries. For the processing of real data, the differences between the actual environment captured by LiDAR and the models can lead to significant deviations (Wahbah et al., 2025). Though all previously mentioned applications employ Kalman filters (KFs) for estimation, other methods exist. Moftizadeh (2024) employs a particle filter (PF) to compute robust localization for a vehicle in an urban environment, similar to Bureick et al. (2019), with a particular focus on robust updates to exclude scatter points common in LiDAR data. Additionally, ultra wide band (UWB) observations are included to incorporate data from other vehicles. Lucks et al. (2021) use iterative closest point (ICP) on segments of larger point clouds acquired by an MMS, estimating transformation parameters that are interpolated between segments to compute a continuous trajectory. This work emphasizes point cloud segmentation, assigning only facade and ground points. Esser et al. (2023) use FGO to improve MMS trajectories by detecting georeferenced targets, with detected target coordinates serving as additional range and bearing observations for georeferencing. Meanwhile, Zhang et al. (2021) employs a sliding window adjustment to locate vehicles using high definition (HD) vector maps, varying the window size from two to five LiDAR frames.

Some studies concentrate on enhancing data fusion by improving specific aspects, such as time synchronization. This element has both a technical and a methodological dimension. Technically, sensor measurements must be timestamped correctly within the appropriate time system, which could be a local system time or a universal system like Global Positioning System (GPS) time, especially when integrating data from various platforms (Stempfhuber, 2004). Numerous modern sensors are equipped with interfaces for GNSS antennas or receivers to include GPS timestamps within their observations (Velodyne, 2019b). Methodologically, the synchronization process depends on the sensors involved. For cameras, the rolling shutter effect is well known, with numerous methods available to counteract this effect and enhance results (Oth et al., 2013). Rotating LiDARs require similar remedies to prevent distortions in the georeferenced point clouds due to movement during scanning. Mounier et al. (2025) address this by dividing LiDAR scans into subscans, then computing interpolated poses for each subscan. The size of these subscans can be adjusted to manage computational effort. Qin et al. (2022) linearly extrapolated prior poses to generate deskewed scans. Hartmann et al. (2023) georeferences points measured using kinematic TLS by computing individual poses for each point, predicting the system's pose for every timestamp. Similarly, Lee et al. (2022) utilize pose interpolation for GNSS pseudo-range observations to enhance data fusion with IMU and cameras. Delays in timestamps or measurement processing lead to increased deviations in estimations, becoming more pronounced at higher velocities or rotation rates. For instance, a system moving at 1 $^{m}/_{s}$ with a timing accuracy of 1 ms results in a deviation of 1 mm (Thalmann & Neuner, 2021). Higher speeds exacerbate these deviations, underscoring the necessity for precise timestamps and time-sensitive processing. Although integral in LiDAR data processing, the combined approach of using individual poses for each LiDAR point, synchronized with pose interpolation for estimation, is not currently documented in the literature.

The uncertainty modeling of observations significantly impacts localization performance. Levinson & Thrun (2010) utilize probabilistic maps for vehicle localization, where their SLAM technique requires calibrated intensities to match points from different scan lines accurately. This calibration is self-supervised and applicable only to individual runs. Analyzing observations offers another strategy through various methodologies. Ettlinger et al. (2018) examine the reliability of the update step in a KF using a Gauß-Helmert model (GHM). Their objective is to distinguish between issues in the observations, such as outliers, and estimation constellation problems. Conversely, Hu et al. (2025) carry out VCE in a LiDAR SLAM context to assign appropriate weights to the observations from different sensors Without an absolute reference, SLAM cannot estimate corrections for the sensor data. Another source of uncertainty lies in the maps used. Probabilistic maps, as used by Levinson & Thrun (2010), assign a probability of being occupied to each cell. By aligning all LiDAR observations with these cells, probabilities for estimated positions can be calculated. Notably, this is only done for 2D localization. Alternative approaches, such as normal distance transforms (NDTs), are also viable. Park & Chung (2024) employ NDT maps for uncertainty-aware map matching, where localization is conducted using a histogram filter that calculates the probability for positions at various grid points concerning the horizontal position and heading. Currently, the literature lacks studies that integrate the estimation of corrections for LiDARs with the weighting of observations. This presents an area for further exploration, as such a combination could enhance the accuracy and reliability of sensor data fusion and localization.

2.1.3 Uncertainty Evaluation

As with other geodetic measurements, the final stage of sensor data fusion involves evaluating the results to provide quality measures. This evaluation can be performed using either the estimated trajectory or the georeferenced point cloud, with the choice often depending on the specific application. Evaluating using the trajectory necessitates additional sensors during the experiment. Frequently, GNSS-based solutions are employed, including post-processed GNSS (Axmann et al., 2023) or real time kinematic (RTK) solutions (Lee et al., 2022; Schlichting et al., 2019). These solutions face challenges, particularly in urban or otherwise obstructed environments. Alternative methods involve using robotic total stations (RTSs) to track systems during experiments. For example, Tombrink et al. (2023) compared an INS estimated trajectory on a rail-bound track to one measured by an RTS. The system changes orientation along the trajectory to study the impact of different orientations. Similarly, Moftizadeh (2024) employs RTS measurements to assess a vehicle's trajectory under realistic conditions, tracking the vehicle in an urban setting before, during, and after turning. Vogel (2020) utilizes a LT to track a corner cube reflector (CCR) mounted on a vehicle for evaluation purposes. The LT offers higher measurement rates and improved accuracy. Unlike RTSs measurements, which involve 360° prisms, the LT measures a standard CCR, further restricting the field of view. All these applications share a common requirement for a direct line of sight, which limits their usability. Furthermore, only a position on the system is measured, evaluating just three degrees of freedom, despite estimating a full six degrees of freedom.

LiDAR-based MSSs generate georeferenced point clouds, which are valuable for assessing localization accuracy. Given a LiDAR with high accuracy and resolution, targets can be utilized for evaluation. Dreier et al. (2021) employ ground control points and tables as planar surfaces to assess the trajectory of an UAV. The control points serve to evaluate the horizontal position, while tables are sensitive to deviations in the estimated height. Similarly, Stenz et al. (2017) use various targets to analyze scans from different sensors. By using reference spheres and photogrammetric targets, they are able to compare kinematic scans to static ones. For any LiDAR, approaches for comparing point clouds can be applied. Shi et al. (2020) implement a plane-based ICP method for strip adjustment of data captured by a MMS, thereby evaluating the data quality. The estimated uncertainty of georeferenced points is incorporated into this estimation process. Furthermore, Hofmann (2017) introduces object-based evaluation methods utilizing environmental planes or cylinders. Multiple evaluation methods are also proposed by Heinz (2021). Adding to the aforementioned works, a parameter-based method is demonstrated here for road surface monitoring. A comparable approach is applied by Kalenjuk & Lienhart (2022) for monitoring retaining walls in alpine areas through data collected via an MMS. With the aim of deformation monitoring, point cloud discrepancies are utilized to refine transformation parameters between distinct runs along the wall. Another viable option is employing multiscale model-to-model cloud comparison (M3C2) for conducting patch-based comparisons. Hartmann et al. (2023) applies this technique to compare georeferenced point clouds against a reference point cloud obtained from a more precise sensor. This methodology is particularly suited for evaluation of the correctness, as it averages the noise within a radius around the core points selected by the method. Problems near edges can be addressed using the approaches suggested by Yang (2023), where the switch from circular to supervoxel-based patch definition enhances adaptation to the point cloud's structure, leading to a more accurate portrayal of its quality.

In summary, evaluating kinematic MSSs remains a relevant yet challenging issue. Difficult environments can limit the use of conventional reference sensors, such as post-processed GNSS or tracking RTSs. Evaluations based on point clouds are feasible, but when low-cost LiDARs are used, measurement uncertainties affect the assessment. This limitation impedes some evaluation methods, particularly those relying on targets. Other methods yield less significant results because the assumptions concerning measurement uncertainties, while important, are not well understood.

2.2 Mathematical Fundamentals

This section covers the essential mathematical concepts required for the remainder of the thesis, introducing symbols and terms for future reference. The goal is not to provide an exhaustive overview of all concepts, but to clarify the methods used in this thesis. The content is organized into three parts. The first part discusses the fundamental principles of sensor data fusion. Following this, the focus shifts to adjustment computation, encompassing both batch adjustments and filtering techniques. Lastly, quality measures are presented for subsequent application.

Sensor Data Fusion

The processing of data acquired by MSSs involves two key components: spatial referencing and time synchronization. **Spatial referencing** involves determining the transformation parameters between the different sensor frames. An additional platform frame may be introduced as an intermediary step. Generally, the necessary transformations in 3D are represented by a rigid body transformation with six degrees of freedom (6-DoF) that maps from one sensor frame, s, to another sensor frame, t, as follows:

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{\mathrm{t}} = \mathbf{H}_{\mathrm{s}}^{\mathrm{t}} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{\mathrm{s}} = \begin{bmatrix} \mathbf{R}_{\mathrm{s}}^{\mathrm{t}} & \mathbf{t}_{\mathrm{s}}^{\mathrm{t}} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{\mathrm{s}} . \tag{2.1}$$

Here, the translation vector $\mathbf{t}$ represents the shift between the frames, while the rotation matrix $\mathbf{R}$ is an orthonormal matrix. This matrix can be derived in various ways, such as using quaternions or Euler angles (Luhmann, 2000). When using Euler angles, the rotation matrices can be composed using multiple rotation matrices corresponding to rotation around single axes:

$$\mathbf{R}_{xyz} = \mathbf{R}_z \mathbf{R}_y \mathbf{R}_x \quad \text{with} \tag{2.2}$$

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix}, \tag{2.3}$$

$$\mathbf{R}_y = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \quad \text{and} \tag{2.4}$$

$$\mathbf{R}_z = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.5}$$

The homogeneous transformation matrix $\mathbf{H}$ is advantageous because its inverse, $\mathbf{H}^{-1}$, reverses the direction of the transformation. In this thesis, the pose is represented by $\boldsymbol{\theta}$, a vector for transformation parameters:

$$\boldsymbol{\theta}_{\mathrm{s}}^{\mathrm{t}} = {}^{\mathrm{t}}\begin{bmatrix} t_x \\ t_y \\ t_z \\ \omega \\ \varphi \\ \kappa \end{bmatrix}_{\mathrm{s}} \tag{2.6}$$

where ω, φ, and κ are the Euler angles extracted from the rotation matrix. When working with quaternions

$$\mathbf{q} = \begin{bmatrix} q_w & q_x & q_y & q_z \end{bmatrix}^T, \tag{2.7}$$

the multiplication is defined using the symbol $\otimes$ as

$$\mathbf{q}_1 \otimes \mathbf{q}_2 = \begin{bmatrix} q_{1,w}q_{2,w} - q_{1,x}q_{2,x} - q_{1,y}q_{2,y} - q_{1,z}q_{2,z} \\ q_{1,w}q_{2,x} + q_{1,x}q_{2,w} + q_{1,y}q_{2,z} - q_{1,z}q_{2,y} \\ q_{1,w}q_{2,y} - q_{1,x}q_{2,z} + q_{1,y}q_{2,w} + q_{1,z}q_{2,x} \\ q_{1,w}q_{2,z} + q_{1,x}q_{2,y} - q_{1,y}q_{2,x} + q_{1,z}q_{2,w} \end{bmatrix} \tag{2.8}$$

For vectors containing quaternions, the addition is written as $\boxplus$ and the subtraction as $\boxminus$, performing the normal vector addition and subtraction operations, while multiplying the quaternions to combine the rotations.

The second aspect is **time synchronization**, which involves ensuring the availability of timestamps with reference to a specific time system. This time system could either be a local system time or an external reference time, such as GPS time (Stempfhuber, 2004). Depending on the sensor, each time stamp can be stored with individual time stamps, the sensor can be triggered by a signal based on defined times, or the observation can be time-dependent, i.e., GNSS.

Adjustment Computation

In most applications, more observations are recorded than are strictly necessary to compute the desired values. This practice ensures overdetermination, which enhances the reliability of the results and facilitates the detection of outliers within the observations. The random nature of the observations can be represented using a stochastic model. For this, an assumption about the distribution of the observations to be made. In many instances, including this thesis, the solution to the overdetermination is computed using least squares optimization, commonly referred to in geodetic terms as an adjustment (Niemeier, 2008). This computation can be performed by processing all observations simultaneously in a batch adjustment or sequentially, where observations are not available all at once.

Batch Adjustment

Batch adjustment refers to optimizations based on maximum likelihood estimation. Its goal is to determine the optimal estimate of u parameters $\boldsymbol{x}$ given a set of n observations $\boldsymbol{\ell}$. The redundancy r measures the degree of overdetermination. This optimization problem can be modeled using one of two approaches. The first approach involves explicit observation models:

$$E(\boldsymbol{\ell}) = \mathbf{f}(\boldsymbol{x}). \tag{2.9}$$

Here, the estimated residuals $\hat{\boldsymbol{v}}$ represent the inconsistencies in the observations, and the notation $\hat{\cdot}$ indicates estimated values. These observation equations are formulated within an Gauß-Markov model (GMM). The stochastic model is employed to weight the observations appropriately. Observations are assumed to follow a normal distribution:

$$\boldsymbol{\ell} \sim \mathcal{N}(E(\ell), \boldsymbol{\Sigma}_{\boldsymbol{\ell}\boldsymbol{\ell}}) \tag{2.10}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{\ell}\boldsymbol{\ell}}$ represents the variance-covariance matrix (VCM) of the observations. From this VCM, a weight matrix $\mathbf{P}$ is computed after extracting an a-priori variance factor σ_0^2 to form the cofactor matrix $\mathbf{Q}_{\boldsymbol{\ell}\boldsymbol{\ell}}$:

$$\boldsymbol{\Sigma}_{\boldsymbol{\ell}\boldsymbol{\ell}} = \sigma_0^2 \mathbf{Q}_{\boldsymbol{\ell}\boldsymbol{\ell}} = \sigma_0^2 \mathbf{P}^{-1}. \tag{2.11}$$

The weight matrix ensures that observations with lower uncertainties have a greater influence in the optimization process.

The objective of the adjustment is to minimize the sum of squared residuals:

$$\boldsymbol{v}^T \mathbf{P} \boldsymbol{v} \rightarrow \text{ min}. \tag{2.12}$$

For linear function models $\mathbf{f}$, the GMM yields optimal results (Förstner & Wrobel, 2016). In cases involving nonlinear models, the results can only be locally optimal, depending on the properties of the function. Observations with greater uncertainties or higher non-linearity in the functional model may cause the estimates to deviate further from the true values. The algorithm detailing the solution of the GMM is omitted here for brevity.

The second approach involves implicit observation models, formulated as constraints:

$$\mathbf{h}(E(\boldsymbol{\ell}), \boldsymbol{x}) = \mathbf{0}. \tag{2.13}$$

This model is known as the GHM and is also referred to as the general model, as any GMM can be expressed in the form of a GHM (Niemeier, 2008; Koch, 2004). The number of constraints, p, is independent of the number of observations. Estimation must be carried out iteratively, requiring initial values for the parameters $\boldsymbol{x}_0$. As the GHM is broadly applicable and utilized for calibration in this thesis, all relevant formulas are presented here, based on the solution provided by Lenzmann & Lenzmann (2004). Initially, the current parameters and observations are defined as:

$$\hat{\boldsymbol{x}} = \boldsymbol{x}_0 \qquad \text{and} \qquad \hat{\boldsymbol{\ell}} = \boldsymbol{\ell}. \tag{2.14}$$

For each iteration, the misclosures $\boldsymbol{w}$ are calculated based on the current parameters and observations:

$$\boldsymbol{w} = \mathbf{h}\left(\hat{\boldsymbol{\ell}}, \hat{\boldsymbol{x}}\right) + \mathbf{B}\left(\boldsymbol{\ell} - \hat{\boldsymbol{\ell}}\right). \tag{2.15}$$

Furthermore, the design matrix $\mathbf{A}$ and the condition matrix $\mathbf{B}$ are computed as partial derivatives of the constraints:

$$\mathbf{A} = \left.\frac{\partial \mathbf{h}}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{\ell}}, \hat{\boldsymbol{x}}} \qquad \text{and} \qquad \mathbf{B} = \left.\frac{\partial \mathbf{h}}{\partial \boldsymbol{\ell}}\right|_{\hat{\boldsymbol{\ell}}, \hat{\boldsymbol{x}}}. \tag{2.16}$$

To update the parameters and observations, cofactor matrices for the misclosures $\mathbf{Q}_{\boldsymbol{ww}}$ and the parameters $\mathbf{Q}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}}$ are computed as intermediate steps. Subsequently, the parameter vector update and residuals for the current iteration are determined:

$$\mathbf{Q}_{\boldsymbol{ww}} = \mathbf{B}\mathbf{Q}_{\boldsymbol{\ell\ell}}\mathbf{B}^T, \tag{2.17}$$

$$\mathbf{Q}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}} = \left(\mathbf{A}^T\mathbf{Q}_{\boldsymbol{ww}}^{-1}\mathbf{A}\right)^{-1}, \tag{2.18}$$

$$\mathbf{Q}_{\hat{\boldsymbol{k}}\hat{\boldsymbol{k}}} = \mathbf{Q}_{\boldsymbol{ww}}^{-1}\left(\mathbf{Q}_{\boldsymbol{ww}} - \mathbf{A}\mathbf{Q}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}}\mathbf{A}^T\right)\mathbf{Q}_{\boldsymbol{ww}}^{-1}, \tag{2.19}$$

$$\Delta\hat{\boldsymbol{x}} = -\mathbf{Q}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}}\mathbf{A}^T\mathbf{Q}_{\boldsymbol{ww}}^{-1}\boldsymbol{w}, \tag{2.20}$$

$$\hat{\boldsymbol{k}} = \mathbf{Q}_{\hat{\boldsymbol{k}}\hat{\boldsymbol{k}}}\boldsymbol{w}, \tag{2.21}$$

$$\hat{\boldsymbol{v}} = -\mathbf{Q}_{\boldsymbol{\ell\ell}}\mathbf{B}^T\hat{\boldsymbol{k}}. \tag{2.22}$$

Finally, the current values for the parameters and observations are updated as follows:

$$\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}} + \Delta\hat{\boldsymbol{x}} \qquad \text{and} \qquad \hat{\boldsymbol{\ell}} = \boldsymbol{\ell} + \hat{\boldsymbol{v}}. \tag{2.23}$$

The iteration process continues until the functional model $\mathbf{h}\left(\hat{\boldsymbol{\ell}}, \hat{\boldsymbol{x}}\right) = 0$ is satisfied to a suitable numerical accuracy, i.e. 10^{-10} depending on units and scale. The cofactor matrix of the parameters $\mathbf{Q}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}}$ can be scaled with either the a-priori or a-posteriori variance factor:

$$\hat{\sigma}_0^2 = \frac{\hat{\boldsymbol{v}}^T\mathbf{P}\hat{\boldsymbol{v}}}{p-u} \tag{2.24}$$

with the number of constraints p and the number of unknowns u, to compute the VCM of the estimated parameters $\boldsymbol{\Sigma}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}}$, which provides insights into the precision of the parameters, including their correlations.

Variance Component Estimation (VCE)

In most cases, the uncertainties of observations are defined a priori and remain constant throughout the adjustment process. Depending on the application, outliers can be removed (Baarda, 1967), observations can be reweighted robustly (Holland & Welsch, 1977; Barron, 2019), or the weights for entire groups of observations can be adjusted. Here, the focus is on the latter option. Weight adjustment for observation groups can be achieved through VCE. This concept is well established in batch adjustments, employing either explicit (Heinz et al., 2015; Amiri-Simkooei et al., 2009) or implicit observation models (Heiker, 2013; Fang et al., 2022). Notably, VCEs in implicit observation models are prone to estimation failures, as a single restriction often involves multiple observations. In scenarios where the geometric constellation is unsuitable or numerous observation groups are present, the estimated variance components (VCs) can become negative and, consequently, unusable. Each of the z observation groups starts with a VC, where γ is initialized as 1, which gets updated iteratively. The methodology is detailed by Yu (1992) and Heiker (2013). The stochastic model comprises an initial stochastic model $\mathbf{V}$ and the current VCs:

$$\boldsymbol{\Sigma}_{\boldsymbol{\ell\ell}} = \sum_{i=1}^{z}\gamma_i\mathbf{V}_i. \tag{2.25}$$

The VCs are updated using:

$$\boldsymbol{\gamma} = \hat{\boldsymbol{S}}^{-1}\hat{\boldsymbol{q}}. \tag{2.26}$$

Both necessary components, the $[z \times z]$ matrix $\boldsymbol{S}$ and the transformed residual sum $\mathbf{Q}$ with dimensions $[z \times 1]$, are calculated element-wise as follows:

$$\hat{q}_i = \hat{\boldsymbol{v}}^T \mathbf{B}^T \mathbf{Q}_{\boldsymbol{ww}}^{-1} \mathbf{B} \mathbf{V}_i \mathbf{B}^T \mathbf{Q}_{\boldsymbol{ww}}^{-1} \mathbf{B} \hat{\boldsymbol{v}}, \tag{2.27}$$

$$\hat{s}_{r,c} = \operatorname{trace}\left(\mathbf{Q}_{\hat{\boldsymbol{k}}\hat{\boldsymbol{k}}} \mathbf{B} \mathbf{V}_c \mathbf{B}^T \ \mathbf{Q}_{\hat{\boldsymbol{k}}\hat{\boldsymbol{k}}} \mathbf{B} \mathbf{V}_r \mathbf{B}^T \right). \tag{2.28}$$

Elements of $\boldsymbol{S}$ are indexed with r for rows and c for columns, while $\boldsymbol{q}$ consists of z elements. The estimation is performed in combination with a GHM within an expectation maximization (EM) scheme Koch (2007). Initially, a GHM adjustment is conducted. Upon convergence, VCE is applied. The updated stochastic model is then used to re-perform the adjustment via GHM. This iterative process continues until VCE convergence is achieved. In situations with an insufficient number of observations or when functional or stochastic models are incorrect, VCE might result in negative VCs (Heiker, 2013). To counter this, besides acquiring more observations or revising the stochastic or functional models, Amiri-Simkooei (2007) suggests formulating the VCE in a least squares manner to incorporate prior information as regularization. For recursive estimation tasks, VCE is feasible (Caspary & Wang, 1998; Wang et al., 2009), yet the challenges associated with estimation increase. The distribution of observations across multiple epochs results in incomplete constellations in individual epochs, requiring the tracking of estimated VCs information across epochs.

Filtering

Filtering enables the recursive estimation of values in a least-squares manner that incorporates prior information, resulting in a maximum a-posteriori estimator (MAP) estimator. In filtering, parameters are considered time-variable and are thus referred to as states. States can represent the position of systems, evolving corrections, or other dynamic properties. In discrete filtering, epochs (k) are defined for the estimation process. State estimation typically involves two steps, which alternate over the processing.

The first step is prediction, which describes a state transition between epochs over a time interval Δt. States are influenced by a physical model, such as a motion model, control inputs $\boldsymbol{u}$, and process noise $\mathbf{w}$:

$$\boldsymbol{x}^{(k)} = \mathbf{g}\left(\boldsymbol{x}^{(k-1)}, \boldsymbol{u}^{(k)}, \mathbf{w}^{(k)} \right), \tag{2.29}$$

$$\mathbf{w} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{ww}}\right). \tag{2.30}$$

The process noise, which follows a normal distribution, models the uncertainties associated with the transition. These uncertainties may arise from imperfections in the physical model or from sensor inaccuracies affecting control inputs (Solà, 2017). Predicted quantities, such as the state, are denoted with a superscript $-$.

The second step is the state update, where observations are used to refine the state estimate. These updated, or filtered, quantities are denoted with a superscript $+$. Similar to batch adjustments, the observation model can be either explicit or implicit. The general algorithm for the KF with explicit observation models is detailed in Simon (2006) and Bar-Shalom et al. (2004). The EKF with implicit observation equations is thoroughly discussed by Dang (2008) and Vogel (2020). The remaining algorithm for an error state Kalman filter (ESKF) with implicit observations is provided in Section 3.2.

Quality

The quality of adjustment results can be represented in various ways for different contexts. Specifically, the accuracy is important to evaluate the usefulness of the data for their purpose. The accuracy consists of the precision, which is the distribution around a mean, and the trueness, which is the deviation of the mean from the true value (ISO 5725-1:2023, 2023). On one hand, accuracy is a requirement to evaluate the integrity of a system as the basis of the navigation performance pyramid (Zhu et al., 2018; Hofmann-Wellenhof et al., 2012). The compliance of the system with the maximum threshold for deviations from the estimated positions is of utmost importance to ensure safe operation. Also in other contexts, such as construction, accuracy is part of the quality modeling (Zhang et al., 2020; Balangé et al., 2021). In holistic modeling, accuracy is an important factor for the technical quality, together with reliability and sensitivity. For the evaluation of the accuracy, the root mean square error (RMSE) can be used. To compute the RMSE for a set of values $\hat{x}_i$, the true value $\tilde{x}$ is necessary

$$\Delta x_i = \tilde{x} - \hat{x}_i \tag{2.31}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \Delta x_i^2}. \tag{2.32}$$

Additionally, the deviations of the estimation can be weighted by their uncertainties to provide insights into how well the estimated uncertainties represent the accuracy. The VCM of the adjusted parameters $\boldsymbol{\Sigma}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}}$ includes information about the precision and correlations of the estimated parameters. One representation of results to consider the uncertainty is consistency. Consistency offers a statistical analysis to assess whether the estimated uncertainties of the parameters accurately reflect their actual deviations from a reference. Therefore, consistency is often applied in simulations or situations where an appropriate reference is available. Bar-Shalom et al. (2004) defines a consistent state error as "zero mean with a magnitude that is commensurate with the state covariance." Validation can be computed using the following equations:

$$\Delta \boldsymbol{x} = \tilde{\boldsymbol{x}} - \hat{\boldsymbol{x}}_i \tag{2.33}$$

$$d_m = \Delta \boldsymbol{x}^T \boldsymbol{\Sigma}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}}^{-1} \Delta \boldsymbol{x}, \tag{2.34}$$

with the quality parameter Mahalanobis distance d_m. This distance can be employed in a hypothesis test with the null hypothesis $E\left(d_m\right) = u$ (Bar-Shalom et al., 2004).

2.3 Data

This thesis utilizes three datasets to produce the presented results, each with different properties that highlight the necessary aspects of the research. First, a simulation framework is introduced. This framework simulates a fictional environment and the forthcoming two datasets to validate the developed algorithms. Next, a real dataset acquired in a laboratory setting is employed. This dataset involves data from an LT tracking a robot equipped with a platform that includes LiDAR, a camera, and an IMU. It serves to illustrate the differences between simulations and real measurements in a controlled environment. Finally, the Leibniz University Cooperative Perception and Urban Navigation Dataset (LUCOOP) dataset captured by vehicles equipped with platforms carrying MSSs is used to test the methods under real-world conditions (Axmann et al., 2023). This dataset reflects genuine conditions in real traffic situations, incorporating a wide array of sensors.

Simulated Environment

The goal of the simulation is the generation of IMU and LiDAR observations. The simulation is self-developed, as other frameworks either simplify the generation of LiDAR data for performance reasons (Carla (Dosovitskiy et al., 2017), Gazebo (Koenig & Howard, 2004)) or perform a realistic physical simulation (HELIOS (Winiwarter et al., 2022)). Both cases lead to a violation of the assumptions of the developed algorithms, which should be expanded in the future to account for realistic effects. To generate this data, two data sources must be defined: an environment and a trajectory along which the simulated system moves during data acquisition. For the generation of simplified IMU data, only the trajectory is required. The objective is to generate accelerometer and gyroscope measurements for all three axes. To compute the error-free sensor readings, the time derivatives of the trajectory are calculated. The first derivatives of the orientations are used for gyroscope readings, while the second derivatives are used for accelerometer readings. Other effects, like changes to the gravity or the Coriolis force, are neglected here, as the filter is currently only used for short trajectories in small areas. These error-free sensor readings are saved for later use in simulation studies. In each simulation realization, the sensor readings are perturbed by uncertainties corresponding to the assumed sensor characteristics. For the IMU, two main uncertainties are typically considered: noise density and bias instability. Noise density quantifies the white noise affecting all measurements, while bias instability describes the drift of the sensor's systematic offset over time, modeled as a random walk process.

LiDAR observations are generated by combining the trajectory with the environment, which is defined by a mesh of triangles. Initially, the assumed sensor measurement rate is used to calculate timestamps for all observations. For state-of-the-art LiDARs, measurement rates range from a few hundred kHz to a few MHz (Velodyne, 2019b). These timestamps are used to interpolate the poses along the trajectory. To derive the correct coordinates, multiple transformations must be sequentially applied. First, the extrinsic calibration of the LiDAR with respect to the body frame is applied to the ray originating in the sensor frame. Next, the transformation for the pose at the current timestamp of the LiDAR observation is applied. Subsequently, the ray-triangle intersection is computed by intersecting the ray with all triangles in the map and returning only the distance to the closest intersection. This distance is used along with the unit vector in the sensor frame to compute the LiDAR observation.

This simulation framework is employed three times: once for publication **B**, once for publication **D**, and once for additional investigations in this thesis. In the simulations for publication **B**, the environment is artificially crafted to evaluate the developed algorithm. A cuboid with dimensions of 10×8×6 m, featuring additional inner panels, is used. These inner panels are incorporated to ensure pose observability at each epoch of the filter, especially since the trajectory is generated randomly.

The generation process begins with the creation of random poses within the cuboid, maintaining a certain distance from the outer walls. The environment is designed to test the algorithm. Thus, the augmentation using the inner planes provides structure to estimate the pose at any given time. While not realistic, this allows us to check the implementation before testing the algorithm on more realistic environments based on real scenes. The initial five seconds of the trajectory are defined such that the system remains stationary, allowing the filter to initialize properly. Subsequently, random velocities and rotation rates are assigned to establish the time intervals between poses. Based on these poses and time intervals, spline interpolation is applied to generate a continuous trajectory that moderates changes in velocity without eliminating them. The trajectory is then cropped to 60 seconds to maintain a uniform length across all simulation runs. The assumed sensor uncertainties are detailed in Table 2.1. They are used for the simulation of sensor observations and as inputs for the estimation process.

Figure 2.1 illustrates the simulation. The 66 panels (6 outer walls and 60 small panels) form the environment and provide structure for the pose estimation. The trihedron shows the position and orientation of the simulated system, with the colors corresponding to red - X, green - Y and blue - Z. One full rotation of the LiDAR is simulated and shown using blue points.

Laboratory dataset

The laboratory dataset is utilized in publications **C** and **D**. Data collection was performed using a Husky UGV (Clearpath Robotics, 2021), which was equipped with a sensor platform. This platform

Table 2.1: *Sensor uncertainties of LiDAR and IMU for simulation. Figure from publication **B**.*

LiDAR standard deviations		
distance measurement σ_d	8.5	mm
horizontal angle σ_α	0.85	mrad
vertical angle σ_e	0.52	mrad
IMU uncertainties for (x,y,z)-axes		
acc. noise density $\sigma_{\tilde{a}_n}$	6.16, 4.61, 3.62	$\mathrm{mm}\sqrt{\mathrm{Hz}}/\mathrm{s}^2$
acc. bias instability σ_{a_η}	22.26, 20.35, 36.36	mm/s^2
gyro. noise density $\sigma_{\tilde{\omega}_n}$	0.26, 0.34, 0.24	$\mathrm{mrad}\sqrt{\mathrm{Hz}}/\mathrm{s}$
gyro. bias instability σ_{ω_η}	1.23, 1.14, 1.19	mrad/s

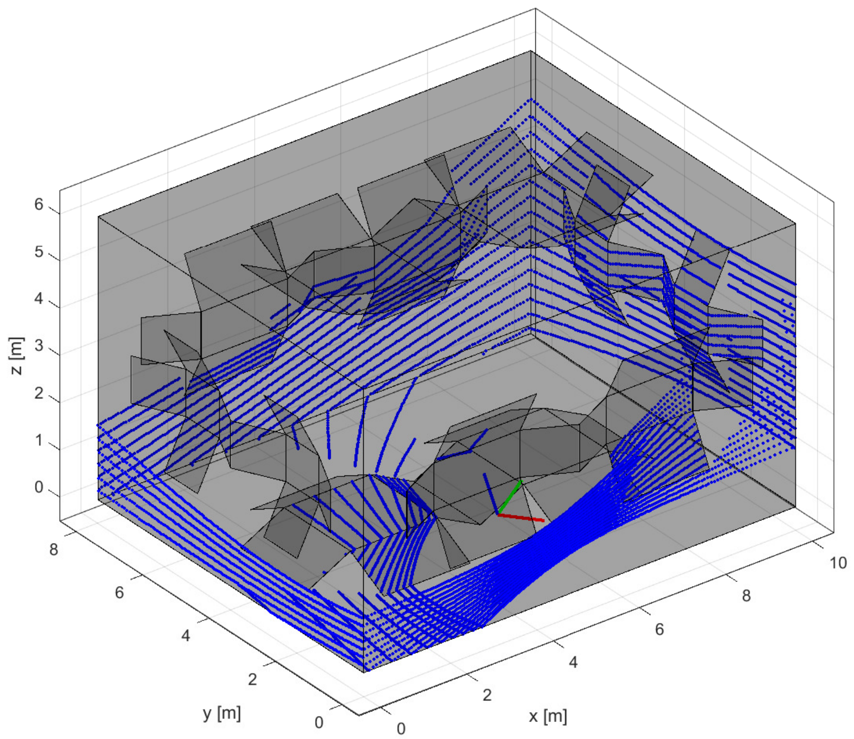

Figure 2.1: *Frame of the simulated environment with planes in grey and simulated LiDAR frame in blue. Position and orientation of the system shown as red, green, and blue trihedron. Figure from publication **B**.*

carried a Velodyne Puck (Velodyne, 2019a), a Microstrain IMU, and a Leica T-Probe (Hexagon Metrology, 2023). While additional sensors were mounted on the platform, they are not relevant to this thesis. The system configuration is depicted in Figure 2.2.

The dataset includes the measurements of the calibration, enabling the use of full information and calibration parameters with their corresponding VCMs, for further sensor data fusion. The experiment was conducted in a laboratory equipped with eleven reference panels, which were measured using an LT to establish a ground truth for further processing. The robot's movement was constrained by the T-Probe's field of view, as maintaining a constant line of sight to the LT was necessary for synchronizing the sensors in time. The individual runs, lasting between 30 and 60 seconds, comprised a variety of movement patterns, from stationary measurements to movement

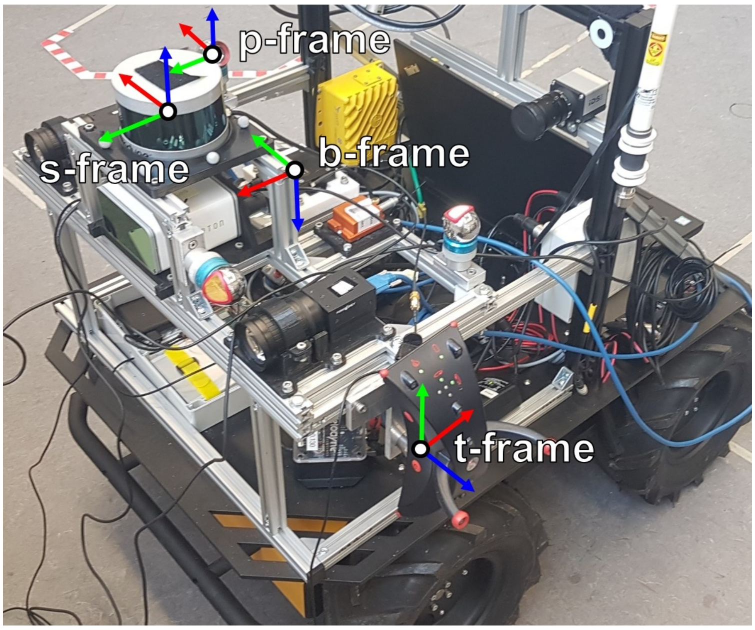

Figure 2.2: *Husky UGV with sensor platform. Frames shown with x-axis in red, y-axis in green, and z-axis in blue. Frames are: s for LiDAR, b for IMU, p for platform, and t for T-Probe. Figure from publication **C**.*

while turning. The uncertainty of the IMU corresponds to the values assumed for the simulation in Table 2.1. The investigation of the Velodyne Puck in publication **A** yields the uncertainties of the LiDAR. The setup, including the robot's movement paths, is illustrated in Figure 2.3.

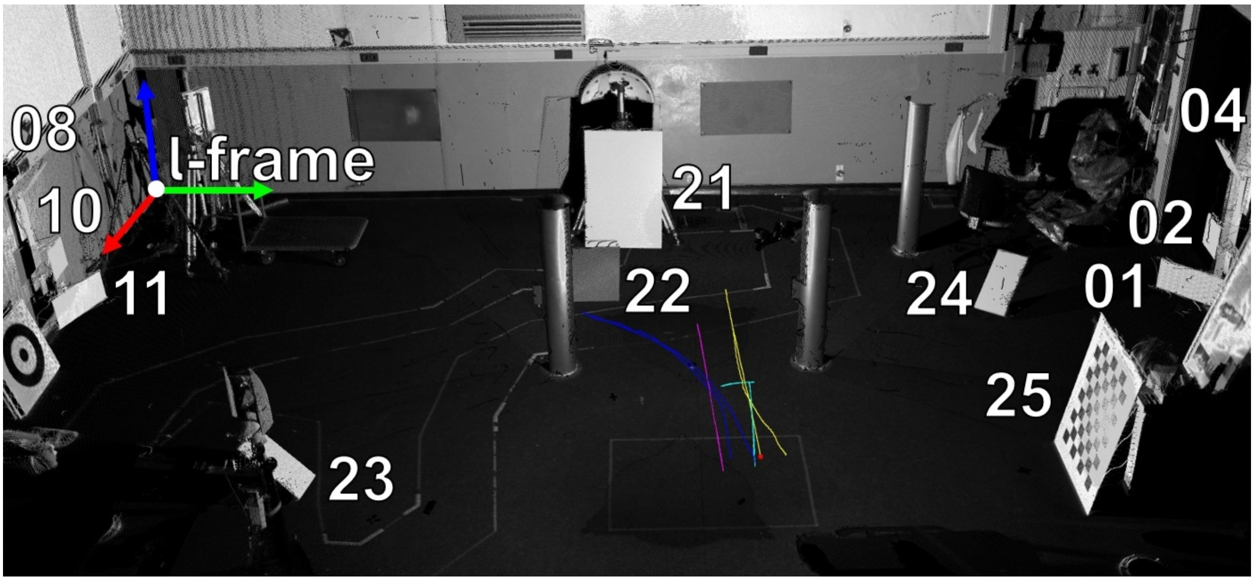

Figure 2.3: *Overview of the experiment setup with numbers denoting panels, including the path of robot movement (in different colors for the runs). Image produced from scan by TLS and colored based on intensities. Figure from publication **C**.*

The dataset is valuable due to the real nature of the data, which is captured in a controlled environment. The laboratory offers a stable atmosphere with low distances to well-known reference geometries. In the investigations conducted for publication **D**, the dataset is additionally simulated using the trajectory provided by the T-Probe, which is transformed into the IMU sensor frame. The main advantage over a simple reflector is the availability of the full 6-DoF pose. Point measurements of a CCR or other reflectors with larger field of views (FoVs) are more flexible, but only provide the time series of a coordinate for a single point on the system. Combined with the reference measurements of the panels, this trajectory enables the generation of the corresponding point cloud.

LUCOOP

The LUCOOP dataset (Axmann et al., 2023) features multi-modal data collected from three different vehicles navigating through an urban environment in proximity to each other. Each vehicle is equipped with at least one IMU, GNSS antenna and receiver, and a LiDAR. Additionally, for further research, the vehicles are outfitted with UWB antennas, complemented by additional stationary stations located along the route. For a segment of the trajectory, tracking via an RTS was available, providing reference measurements for one of the vehicles that was equipped with a 360° prism. Each vehicle's data includes a post-processed INS trajectory and a trajectory computed using ICP of the LiDAR observations in relation to a TLS point cloud of the environment. These elements make the dataset a valuable resource for evaluating the algorithms developed in this thesis. The urban environment presents typical real-world challenges, such as the occlusion of GNSS signals, while simultaneously providing a highly structured environment conducive to localization.

The final application of the aforementioned simulation framework is detailed in Section 2.3. The trajectory and extrinsic calibration utilized are provided within the dataset. Specifically, the trajectory serves as the ground truth and is derived from a post-processed MMS trajectory obtained through commercial software. The map is constructed from a TLS point cloud of the environment. The detailed procedure for generating the localization map is described in Section 4.2.1.

3 Summary of Relevant Publications

In this chapter, the four peer-reviewed publications relevant to this cumulative dissertation are summarized. The structure of this chapter is motivated by the processing chain shown in Figure 1.1. The contributions aim to improve the processing by properly introducing and propagating uncertainties of the used LiDARs. Publication **A** introduces a multi-position calibration approach that reduces parameter correlations and estimates a distance offset for a Velodyne Puck. This work combines intrinsic and extrinsic calibration with a VCE. The evaluation of the calibration parameters, including their uncertainties provided by the estimated VCM is tested by application in sensor data fusion. Publication **B** continues with the topic of sensor data fusion, showing an ESKF with implicit measurement equations. The formulation with an error state enables the use of quaternions, thus providing full rotational freedom with flexibility concerning the functional model in the update step. Additionally, a time interpolation scheme for the georeferencing of the LiDAR points is introduced to compensate for the movement between epochs. The algorithm is only evaluated based on simulated data. Publication **C** provides a different method for the fusion of sensor data by presenting a full processing chain to georeference LiDAR point clouds. The uncertainty propagation is performed for each step of the chain to achieve a realistic uncertainty estimation for the final points. The propagation is performed using a Monte Carlo (MC) variance propagation, which is based on information from manuals and calibrations as defined by Guide to the Expression of Uncertainty in Measurement (GUM) (JCGM, 2008a). The uncertainties are finally evaluated against reference measurements. Publication **D** investigates the ESKF introduced in publication **B**. In a first step, the data set is simulated to quantify the best case with known values for all parameters. Next, the real data set is processed in the filter, and the estimated trajectory is compared against a ground truth. The results show remaining deviations due to biases and missing corrections in the process.

This leads to the following structure of the remaining chapter. First, the intrinsic and extrinsic calibration of an MSS platform is addressed by summarizing publication **A**. Afterwards, the joint processing of sensor data is discussed on the examples shown in publications **B** and **C**. In both cases, KFs are used to process kinematically captured LiDAR point clouds, each with a different focus concerning the fusion process. Finally, the results of the data fusion are investigated concerning the provided uncertainties. Aspects of publications **C** and **D** are used to provide different techniques to check the consistency of the results.

3.1 Calibration

The calibration of a sensor platform is an essential step for processing the captured data. The goal of the extrinsic calibration is to determine the transformation parameters between the frames of different sensors mounted to it. Publication **A** deals with this issue in several aspects. The work in the relevant publication builds upon the concepts for an object-space information-based calibration developed in Strübing & Neumann (2013) using the calibration field implemented based on (Hartmann et al., 2019). The basic concept of the calibration approach is based on the minimization of the distances between observed LiDAR points and reference panels. Given a known pose for the system's platform, this optimization leads to a set of transformation parameters, which can be applied to fuse the LiDAR observations with measurements of other sensors. The contributions of the relevant publication encompass combining the **intrinsic calibration** with the extrinsic calibration, performing **VCE** while using **multiple positions**. This enables a better calibration by correcting and reweighting the sensor measurements. The use of multiple positions reduces correlations between the estimated parameters. Additionally, **subsampling** is utilized to prevent an indirect weighting between the calibration panels.

The results of the chosen extrinsic calibration rely on the sensor observations; thus, the sensor itself should be calibrated. Especially in the case of low-cost sensors, **intrinsic calibration** should be performed to estimate corrections for the measurement elements. Otherwise, uncorrected systematic deviations in the measurements lead to systematic deviations in the estimated parameters, which are not properly described by the estimated VCM. A LiDAR observes distances d_i and horizontal angles α_i at fixed vertical angles e_i in its own frame s. In the first step, the functional model for the calibration is extended to include a distance offset d_0 for the LiDAR observations. For each point $\mathbf{p}_{\,\mathrm{s},i}$, the three measurement elements are transformed into Cartesian coordinates

$$\mathbf{p}_{\,\mathrm{s},i} = \begin{bmatrix} x_{\,\mathrm{s}} \\ y_{\,\mathrm{s}} \\ z_{\,\mathrm{s}} \end{bmatrix}_i = \begin{bmatrix} (d_i + d_0) \cdot \cos e_i \cdot \cos \alpha_i \\ (d_i + d_0) \cdot \cos e_i \cdot \sin \alpha_i \\ (d_i + d_0) \cdot \sin e_i + z_c \end{bmatrix}. \tag{3.1}$$

The vertical correction z_c is specific to the Velodyne Puck used in publication **A** (Velodyne, 2019b). A scale factor is neglected because the calibration field is too small for its determination. It is important to note that the value of the distance offset is specific to the measured material. Darker material generally leads to a higher offset and higher noise in the distance observation. While this limits the transferability of the offset itself, estimating the correction is still important, as it benefits the estimation of the extrinsic calibration (Khami et al., 2025).

Intrinsic calibration is not limited to estimating functional parameters, but can also be used to estimate the covariance structure of the LiDAR observations through performing a **VCE** for the adjustment. The LiDAR observations are assumed to be affected by additive random noise. Here, the idea is to update initial values for the assumed stochastic model of the sensor measurements based on the residuals of the calibration adjustment. This is especially important for low-cost sensors, as manufacturers often do not provide adequate uncertainty descriptions, and variations between sensors of the same type can be significant. In total, three VCs are estimated: one for each measurement element, assuming uncorrelated measurements between elements and between subsequent measurements of the same type. The different measurement elements are assumed to be uncorrelated due to the setup of the sensor, which measures each element differently. Correlations between subsequent measurements are not modeled due to missing information about the magnitude. This leads to too optimistic estimates for the uncertainties of the estimated parameters. The initial VCM of the observations is set up as

$$\mathbf{\Sigma}_{\boldsymbol{\ell\ell}} = \operatorname{diag}\left(\gamma_d\sigma_d^2 \quad \gamma_e\sigma_e^2 \quad \gamma_\alpha\sigma_\alpha^2 \quad \cdots \quad \gamma_\alpha\sigma_\alpha^2\right) \tag{3.2}$$

repeating for each LiDAR point observed. The VCs γ for the observation groups are initialized as 1 and updated after the adjustment converges. While the assumption of no correlations between measurement elements can be justified by the sensor setup, neglecting correlations between subsequent measurements stems from a lack of information. Similar to the distance offset, the estimation of the stochastic model for the distance offset is again limited to the material used in the calibration field, limiting further use in other environments. In contrast, the uncertainties for angle measurements (horizontal and vertical) are generally applicable for the sensor, at least for some time after the calibration.

Including both extensions in the adjustment leads to a more complex adjustment, which, in the case of the VCE, can lead to a failure to estimate a solution at all. To support the VCE and reduce the correlations between the estimated parameters, **multiple positions** for the system are introduced. The goal of the additional positions is to increase the variety of geometric constellations between the sensor and the reference panels. This has multiple benefits. Increasing geometric variety enables the VCE to properly update the stochastic model. Additionally, the adjustment's constellation is improved, leading to a smaller VCM of the estimated parameters and reduced correlations between the estimated parameters. Furthermore, the calibration can be performed with fewer and smaller panels, as different positions can observe panels multiple times. The last benefit is important for two reasons. First, the different parameters of the pose are sensitive to different panels; closer panels are better for estimating the translations, while panels farther away improve the estimation of the orientations. Second, depending on the LiDAR used, the FoV of the sensor can be too large to cover with reference panels in just one position. Multiple positions help capture the reference geometries with multiple scan lines, thereby improving the results.

Finally, **subsampling** is applied to the LiDAR observations. The measured points are randomly subsampled to the same number of points per panel and position. This has three benefits. The correlations between subsequent observations affect the adjustment less. While the ray still travels nearly the same path and hits the same panel, the autocorrelations of subsequent measurements can be nearly eliminated. Moreover, no implicit weighting is applied. Otherwise, panels that are measured from closer distances would provide more observations to the adjustment. Finally, the computational complexity is reduced due to fewer observations.

These contributions are combined in the **experiment**. A LiDAR on a platform for use in kinematic applications should be calibrated. The transformation parameters between the sensor frame s and the body frame b are of interest. The pose of the platform in the superordinate l-frame is known. The calibration field, based on Hartmann et al. (2019), is used with an additional panel on the ground. The setup, with an overview of the frames, is shown in Figure 3.1.

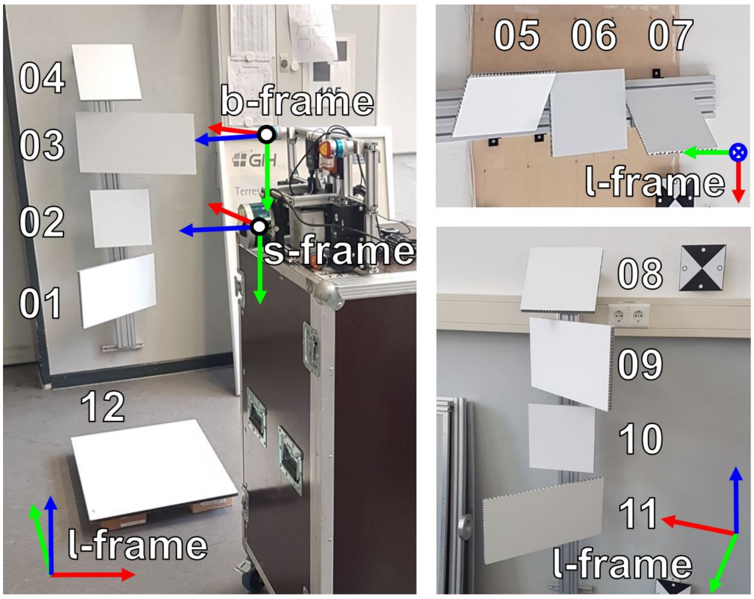

Figure 3.1: *Overview of the calibration field with panel numbers. Left: platform (in Pos4) with frames and panels on left wall. Top right: panels mounted at the ceiling. Bottom right: panels on opposing wall. Figure from publication* ***A****.*

It can be seen that the panels are comparatively small for the FoV of the Velodyne Puck, which is $360° \times 30°$. The use of five different positions within the extents of the calibration field allows for different constellations and capturing the panels with different scan lines. The chosen positions are shown in Figure 3.2. The choice of positions aims at maximizing the variety in the geometric constellation while still observing the calibration panels with as many scan lines as possible. **Pos2** and **Pos3** provide observations of different scan lines, while **Pos4** and **Pos5** provide longer observation distances to improve the estimation of the orientations. This is limited by the extent of the laboratory, providing a maximum distance of about 8 m for the observations.

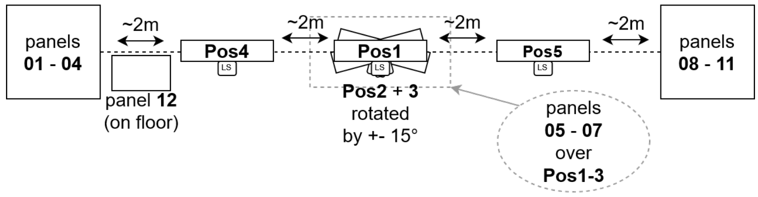

Figure 3.2: *Overview of the five platform positions wrt. to the calibration field. Figure from publication **A**.*

At the beginning, the reference panels are measured once using an LT. For each position of the platform, the pose of the body frame is determined by measuring the CCRs in the nests, which define the frame. Additionally, the LiDAR observations are captured for 10 s for later processing. After the measurements, the LT observations are processed to determine the plane parameters for the panels and the poses for the platform at each position. The LiDAR point clouds are cropped to extract only the relevant points for each panel. This step is done manually to exclude outliers and other disturbances, such as mixed pixels.

The adjustment is performed using a GHM with a VCE. The functional model for points i assigned to panels j with the plane parameters $\mathbf{a}$ with the normal vector $\mathbf{n}$ and the distance parameter $\overline{d}$ is

$$\mathbf{p}_{\mathrm{l},i} = \mathbf{t}_{\mathrm{b}}^{\mathrm{l}} + \mathbf{R}_{\mathrm{b}}^{\mathrm{l}} \left(\mathbf{t}_{\mathrm{s}}^{\mathrm{b}} + \mathbf{R}_{\mathrm{s}}^{\mathrm{b}} \mathbf{p}_{\mathrm{s},i} \right) \tag{3.3}$$

$$w_i = \mathbf{p}_{\mathrm{l},i}^T \mathbf{n}_j - \overline{d}_j = h_i \left(\boldsymbol{\ell}_i, \boldsymbol{\theta}_{\mathrm{s}}^{\mathrm{b}}, d_0, \boldsymbol{\theta}_{\mathrm{b},k}^{\mathrm{l}}, \mathbf{a}_j \right) \overset{!}{=} 0 \tag{3.4}$$

using the Cartesian points computed in (3.1) with the pose of the platform $\boldsymbol{\theta}_b^l$. The pose of the sensor on the platform $\boldsymbol{\theta}_s^b$ should be estimated in conjunction with the distance offset d_0. The standard GHM is applied in an EM scheme to estimate the VCs. The formulas are omitted here for brevity but can be found in Lenzmann & Lenzmann (2004), Heiker (2013) and Koch (2002).

The **results** are split into the functional parameters and the estimated stochastic model. First, the functional parameters are shown. Table 3.1 provides an overview of the estimated parameters. While the values themselves are system-specific, the standard deviations of the parameters provide information about the constellation in the estimation. Generally, the standard deviations show a significant determination of the parameters with the exception of the orientations φ and κ. A small value is expected for these parameters, as the sensor is mounted horizontally on the platform, leading to small values for the roll and pitch. The translations along x and z show similar standard deviations. In comparison, the translation along y seems to be determined worse. A similar effect can be seen in the orientation, where the angle ω is determined worse than the other two angles. This can be explained by the setup, as the platform always points downward during the measurements, leading to a shorter distance and fewer observations for the determination of t_y and ω. The distance offset is estimated to be 14 mm, which confirms the information about the magnitude of uncertainties provided by the manufacturer (Velodyne, 2019b) and is also supported by other studies (Sánchez & Pany, 2019; Glennie et al., 2016). Estimating only one distance offset simplifies the sensor design with 16 independent distance measurement units. An independent calibration of all scan lines might provide better results for applications, but is not feasible with the current calibration field design. Further investigations are found in the literature (Glennie et al., 2016; Bergelt et al., 2017) and supervised theses by the author (Niehsen, 2022; Jüngling, 2023).

While the standard deviations only provide insight into the diagonal of the VCM of the estimated parameters, the correlation matrix also indicates if the parameters can be determined independently. Figure 3.3 compares a calibration using only one position to a calibration using all five positions. The stochastic model is the same for both calibrations, as the VCE does not work when only one position is available.

Table 3.1: *Result for the estimated parameters using five positions with the corresponding standard deviations*

parameter	value	std dev.
t_x	221.1 mm	1.80 mm
t_y	188.7 mm	5.98 mm
t_z	89.2 mm	1.44 mm
ω	-1.471°	0.060°
φ	0.013°	0.017°
κ	-0.007°	0.014°
d_0	-14.0 mm	0.20 mm

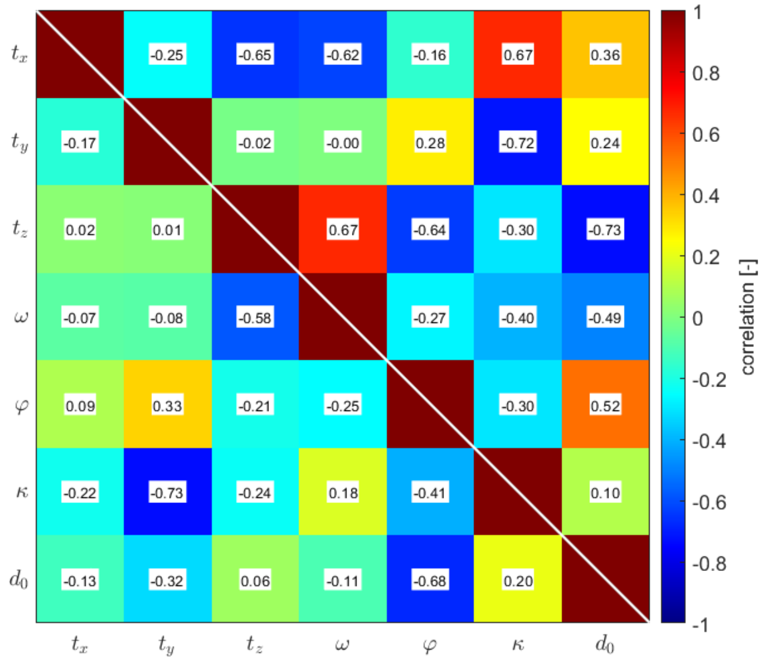

Figure 3.3: *Comparison of correlations of estimated parameters using one position (upper triangular matrix) and using all five positions (lower triangular matrix). Figure from publication **A**.*

It can be seen that the use of multiple positions improves the estimation by reducing most correlations. Especially notable is the correlation between the translation t_z and the distance offset d_0, which is nearly eliminated. The remaining notable correlations are:

- for ω and t_z with -0.58, caused by the same short distance to the ceiling panels,
- for κ and t_y with -0.73, due to the importance of panel **04** for both parameters,
- and φ and d_0, as both parameters are determined by the panels **01**, **03**, **09** and **11**. This is the only correlation increasing in magnitude with multiple positions.

The results of the VCE are shown in Table 3.2. The VCs are multiplied by the initial stochastic model for clarity. The standard deviation of the distance measurement matches the noise of a plane fit and is confirmed by others (Glennie et al., 2016). The magnitude also fits the manufacturer's information with a total accuracy of ±3 cm (Velodyne, 2019b). It has to be noted that the uncertainties for the distance measurements are material dependent, which limits the transferability of the results to other applications. This is shown extensively for TLSs (Wujanz, 2016; Schill et al., 2025) and also for low-cost LiDARs (Kellner, 2024). For TLS, first learned models are available to account for different materials and shapes (Hartmann et al., 2024). The uncertainties for the angle measurements are proportional to the beam divergence of the sensor, with a higher uncertainty for the azimuth angle compared to the elevation.

Table 3.2: *Result of the variance component estimation*

measurement element	std dev.
distance	8.5 mm
azimuth angle	0.0485°
elevation angle	0.0296°

The contributions of this section, aimed at answering research question **1**, can be summarized as follows. Publication **A** presents an enhanced calibration scheme that employs object space information to determine both the extrinsic and intrinsic calibration parameters for a LiDAR-based MSS. This innovative procedure not only encompasses the calibration parameters but also estimates the stochastic model of the LiDAR, which is important for further processing and often unknown for the widely used low-cost LiDARs. The use of multiple positions substantially reduces the correlations between the estimated parameters, improving the determination of the parameters. Publications **C** and **D** utilize the determined transformation parameters and their uncertainties, in the form of a VCM, to accomplish the data fusion of the LiDAR observations. Based on these results, the quality of the results can be assessed.

3.2 Sensor Data Fusion

The fusion of data from different sensors requires **spatial referencing** and **time synchronization.** This section deals with the handling of these issues based on the methodology and results of publications **B** and **C**. Publication **B** presents a novel ESKF to fuse IMU observations with an implicit measurement model for the update step. The update is performed by minimizing distances between assigned LiDAR points and a known map. The position of the LiDAR points is improved by linear interpolation of the system pose, which is evaluated based on a simulated environment. In contrast, publication **C** uses a real dataset to demonstrate uncertainty modeling over multiple steps of a processing chain, extending the concepts of GUM (JCGM, 2008b). The setup involves a robot fitted with a sensor platform including a LiDAR, a T-Probe, and an LT. The observed pose of the T-Probe is used to georeference the system, which in turn is used to georeference the points. In combination with a selection of known reference panels, the accuracy and consistency of the points are evaluated. Table 3.3 provides an overview of the contents for this section organized by the publications and issues. Finally, some results are shown to demonstrate the efficacy of the methods.

Spatial referencing is needed to relate the observations of different sensors to each other. In the case of publication **B**, both the IMU and the LiDAR are simulated to be positioned in the same pose, so the transformation between the sensors can be neglected. In contrast, publication **C** not only applies the transformations between the different sensors, LiDAR and T-Probe, but also models the uncertainties of the individual transformations. The determination of the transformation parameters between the LiDAR and the platform frame and the corresponding uncertainties is described in Section 3.1. The reference panels used for the evaluation are the same as in the calibration; thus the uncertainties determined for the measurements can be used for the uncertainty modeling.

The determination of the pose of the T-Probe t with respect to the platform frame b can be easily computed using measurements of the LT with the frame l. The pose of the T-Probe is determined by the LT using the polar measurement of the reflector and the backprojection of the sensor's LEDs. This provides the 6-DoF pose of the T-Probe in the l-frame $\mathbf{H}_{\mathrm{t}}^{\mathrm{l}}$. The pose of the

Table 3.3: *Overview of methods used in publications **B** and **C** with respect to the spatial referencing and time synchronization.*

	B: ESKF	**C**: linear KF
Spatial referencing	dead reckoning of pose by integrating IMU measurements, update by minimizing distances between assigned LiDAR points to map	measured pose of T-Probe, applied transformations to georeference LiDAR points
Time synchronization	definition of epochs by IMU measurements, linear interpolation of pose for each LiDAR point	definition of epochs by T-Probe measurements, prediction of pose for time stamp of LiDAR points
Uncertainty modeling	process noise based on IMU uncertainties, uncertainties of LiDAR determined in prior calibration	process noise based on design parameters with constant velocity model, uncertainties of T-Probe based on manual, uncertainties of LiDAR determined in prior calibration

platform frame $\mathbf{H}_{\mathrm{b}}^{\mathrm{l}}$ is computed by measuring the three CCRs distributed on the platform. Based on the measured coordinates, the defined coordinate system can be reconstructed, and its pose can be estimated based on the closed-loop solution of rigid body transformations (Horn, 1987). This leads to the transformation between the two frames

$$\mathbf{H}_{\mathrm{b}}^{\mathrm{t}} = \left(\mathbf{H}_{\mathrm{t}}^{\mathrm{l}}\right)^{-1} \mathbf{H}_{\mathrm{b}}^{\mathrm{l}}. \tag{3.5}$$

The determination of the uncertainties for these transformations is performed using a MC variance propagation to account for the nonlinear nature of the transformation. To accomplish this, the observations are perturbed by uncertainties in the order of the accuracy of the sensor for $M = 100,000$ realizations. The number of realizations is chosen based on the recommendation of Möser et al. (2013). For each realization, noise in the magnitude of the measurement uncertainty of the LT is added to the observations. The uncertainty is specified based on the manual of the instrument at 15 μm + 6 ppm for point measurements and 50 μm + 6 ppm for the T-Probe translations as maximum permissible errors (MPEs). The T-Probe orientations are assumed to have a standard deviation of 0.01° (based on the uncertainty of the T-Mac, which is constructed similarly) (Hexagon Metrology, 2023). The random noise is assumed to be uncorrelated due to missing information. The perturbed observations are used to compute the transformation matrices. Afterwards, the transformation matrices $\mathbf{H}_{\mathrm{b}}^{\mathrm{t},(m)}$ are decomposed into the poses $\boldsymbol{\theta}^{(m)}$ with the mean $\bar{\boldsymbol{\theta}}$ and used to compute the VCM of the transformations using (Koch, 2007)

$$\mathbf{E} = \begin{bmatrix} \boldsymbol{\theta}^{(1)} - \bar{\boldsymbol{\theta}} \\ \boldsymbol{\theta}^{(2)} - \bar{\boldsymbol{\theta}} \\ \vdots \\ \boldsymbol{\theta}^{(M)} - \bar{\boldsymbol{\theta}} \end{bmatrix}, \tag{3.6}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{\theta}} = \frac{1}{M-1} \mathbf{E}^T \mathbf{E}. \tag{3.7}$$

As stated before, the pose of the T-Probe is directly provided by the LT, and the processing of the kinematic movement is explained with respect to the time synchronization in the following part. In the last step, the pose of the LT in the room is computed based on a benchmark, for which the uncertainties are also included. An overview of the process chain, including the uncertainty information, is given in Figure 3.4.

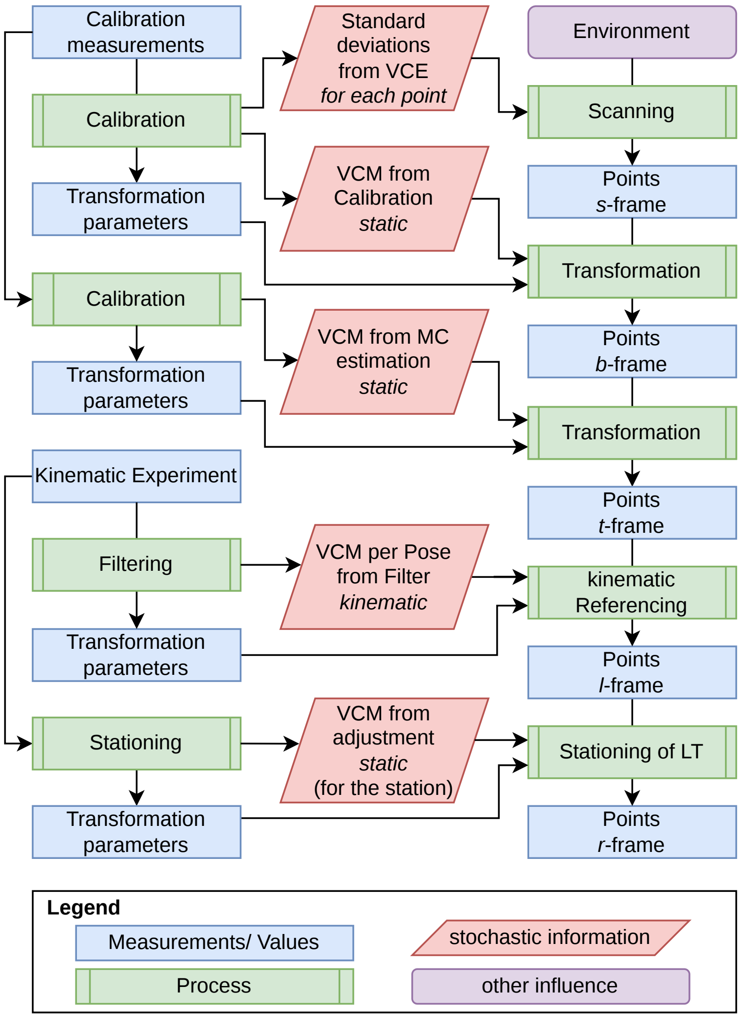

Figure 3.4: *Overview of the processing chain in publication **C**. The left side indicates the experiment, the uncertainties are listed in the middle, and on the right side, the processing chain can be found.*

Time synchronization involves the correct processing of observations considering the time of measurement. This issue is twofold. On the one hand, all observations need to be provided with correct timestamps. This is given for the relevant publications, as the dataset in publication **B** is simulated, and the movement speed of the system in publication **C** is slow enough to assume no influences of possible time delays on the results (cf. Section 2.1). On the other hand, the processing needs to properly include the time information in the computation. LiDARs measure at very high frequencies, providing many observations in short time frames. When fusing these point clouds with other sensors, the different measurement rates need to be accounted for. The relevant publications provide two different ways to accomplish this.

In publication **C**, a **linear KF** is used to model the movement of the system. The final goal is the georeferencing of the point clouds measured by a LiDAR mounted on a moving platform. The pose of the system is determined by a T-Probe also mounted to the system's platform. The use of a linear KF is motivated by the directly available pose of the T-Probe and the low dynamics of the movement, enabling direct georeferencing of the platform and thus, the ability to georeference the point clouds using the estimated poses afterwards. The state vector $\boldsymbol{x}^{(k)}$ consists of the 6-DoF of the platform and their time derivatives (both the velocities and their accelerations). The prediction between the epochs (k) is performed by

$$\boldsymbol{x}_{-}^{(k)} = \boldsymbol{\Phi}^{(k-1)}\boldsymbol{x}_{+}^{(k-1)} + \boldsymbol{w}^{(k-1)} \tag{3.8}$$

and includes no control inputs, as no further information is available. The epochs are defined by the observation times of the T-Probe, measuring at 50 Hz. Each component of the state vector (marked by $*$) is modeled as follows. The transition matrix $\boldsymbol{\Phi}^{(k-1)}$ integrates the velocities and accelerations based on the time difference since the last epoch Δt

$$\boldsymbol{\Phi}^{*(k-1)} = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.9}$$

The system noise $\boldsymbol{\Sigma}_{\boldsymbol{\eta\eta}}$ is modeled as a constant acceleration model by

$$\boldsymbol{\Sigma}_{\boldsymbol{\eta\eta}}^{*} = \begin{bmatrix} \frac{1}{4}\Delta t^4 & \frac{1}{2}\Delta t^3 & \frac{1}{2}\Delta t^2 \\ \frac{1}{2}\Delta t^3 & \Delta t^3 & \Delta t \\ \frac{1}{2}\Delta t^2 & \Delta t & 1 \end{bmatrix} \sigma_\eta^2 \tag{3.10}$$

with one value for σ_η for the acceleration and one for the jerk (Bar-Shalom et al., 2004). Additionally, the VCM of the state is predicted

$$\boldsymbol{\Sigma}_{\boldsymbol{xx},-}^{(k)} = \boldsymbol{\Phi}^{(k-1)}\boldsymbol{\Sigma}_{\boldsymbol{xx},+}^{(k-1)}\left(\boldsymbol{\Phi}^{(k-1)}\right)^T + \boldsymbol{\Sigma}_{\boldsymbol{\eta\eta}} \tag{3.11}$$

with a constant $\boldsymbol{\Sigma}_{\boldsymbol{\eta\eta}}$ as only a time dependency exists and the time differences are constant over the whole experiment. The prediction has two purposes in the processing chain. First, it is used to process all observed poses of the T-Probe in the linear KF. Additionally, and similar to Hartmann et al. (2018), the prediction is used to georeference the point cloud observed by the LiDAR. Each point in the point cloud is provided with a unique time stamp, which is used to predict the pose of the system at that time. The predicted pose is then used to compute the position of the point in the room frame. The predicted uncertainties for the pose are used to perform the MC variance propagation to the uncertainties of the points. The update step follows a standard linear KF, during which the pose of the T-Probe is directly observed, so the remaining formulas are omitted here.

The **MC variance propagation** is performed by computing the processing chain multiple times per point. For each realization, the original measurement elements of the LiDAR points are perturbed by noise in accordance with the order of the uncertainty of the sensor, as determined in Section 3.1. Figure 3.4 provides insight about how the random numbers are generated. The red blocks show the uncertainties included in the simulation. While the random numbers for the *static* steps are only computed once per realization and applied for all points, the noise of the observation and the kinematic pose are affected by noise generated per point. This is repeated $100,000$ times, leading to a point set for each point in the original point cloud. The high number of realizations ensures a good approximation of the results (Möser et al., 2013). Similar to the computation of the pose between the T-Probe and the platform, equations (3.6) and (3.7) can be adapted to compute the VCM for each point.

The experiment to evaluate the results is performed in a laboratory with a robot carrying a sensor platform calibrated using the methods described above. The laboratory is equipped with eleven panels that are used for the evaluation of the results. The robot starts stationary, then moves forward with a slight left turn. After standing for a few seconds, the robot moves back the same way and stops approximately where it started. An overview of the experiment is shown in Figure 2.3. The georeferencing of the LiDAR point cloud is computed in two ways. For the first way, the transformations of the processing chain are applied directly. This corresponds to the classical approach. For the second way, the mean values of the point sets from the MC variance propagation are used as the georeferenced points. This is called ensemble referencing from here on. Figure 3.5 shows the results for four chosen panels. Each plot shows the distribution of distances between points and panels for each frame in the form of the median (blue), and the 2.5% and 97.5% quantiles (green). The number of points per panel used for the computation in each frame is shown in red.

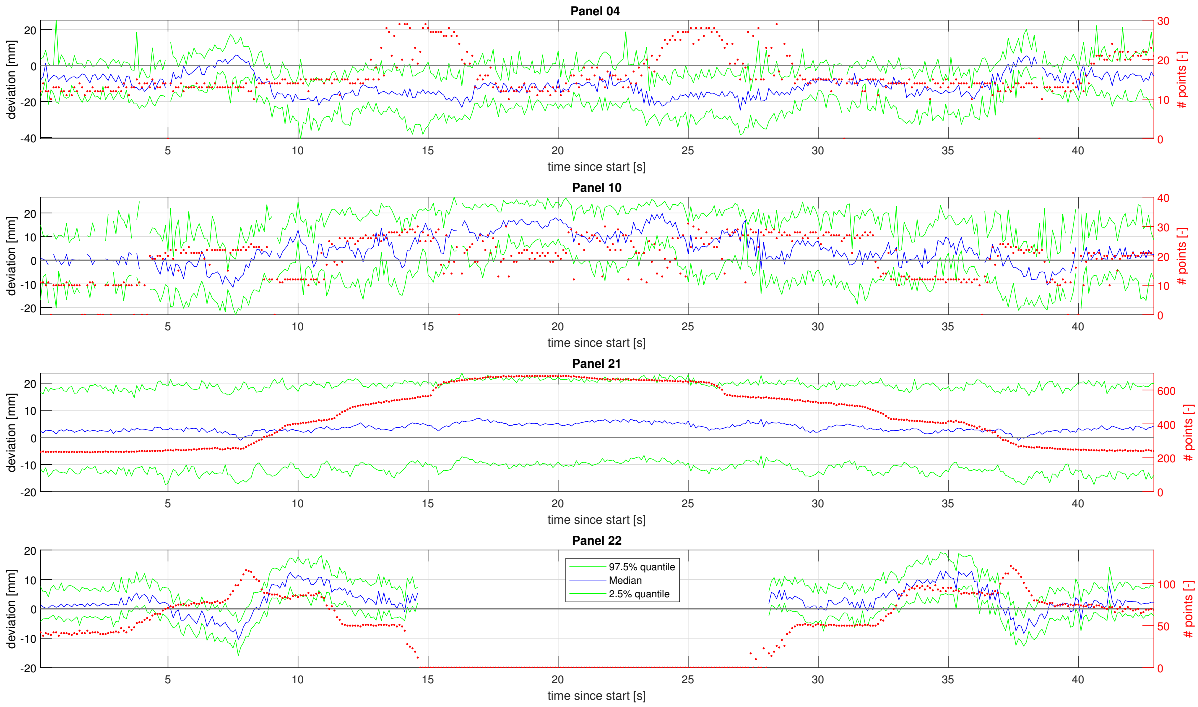

Figure 3.5: *Median and quantiles of distances between point clouds and reference panels over time, panel number from Figure 2.3, numbers of points assigned to panel in red. Figure from publication **C**.*

For all panels, the deviations are constant in the first and last five seconds, as the system is stationary in these time frames. While the deviations for panel **21** are constant, the deviations for panels **04**, **10** and **22** change over time during the movement. Both panels **04** and **10** are located to the side of the robot, but show differences in the behavior of the deviations. While the deviations of panel **04** are nearly constant, panel **10** shows an oscillation in the deviations caused by the different tilt. The time series of panel **22** shows a gap, as the LiDAR is mounted horizontally and measured above the panel when the system has moved furthest to the front. Especially the beginning of the forward movement and the end of the backward movement lead to higher deviations. Remaining deviations of the transformation parameters cause this effect due to the change in the constellation between the system and the panels. Even with correctly estimated uncertainties for the transformation parameters, some remaining systematic deviations remain. Without including the uncertainties in the evaluation, the results cannot be fully interpreted.

Figure 3.6 shows the results for the ensemble referencing. Notably, the deviations decrease across all shown panels. For panel **21**, the slight positive offset is reduced. For panels **04**, **10** and **22**, the changes in the deviations during movement are removed, and the magnitude of the deviations is generally lower. Over all panels, the RMSE of the distances is reduced from 10.0 to 8.1 mm. This demonstrates the positive impact of including the uncertainty information in the georeferencing process. While Gaussian variance propagation would achieve similar values for the VCM, the MC variance propagation can compensate for non-linear effects and thus improve the accuracy of the point cloud itself.

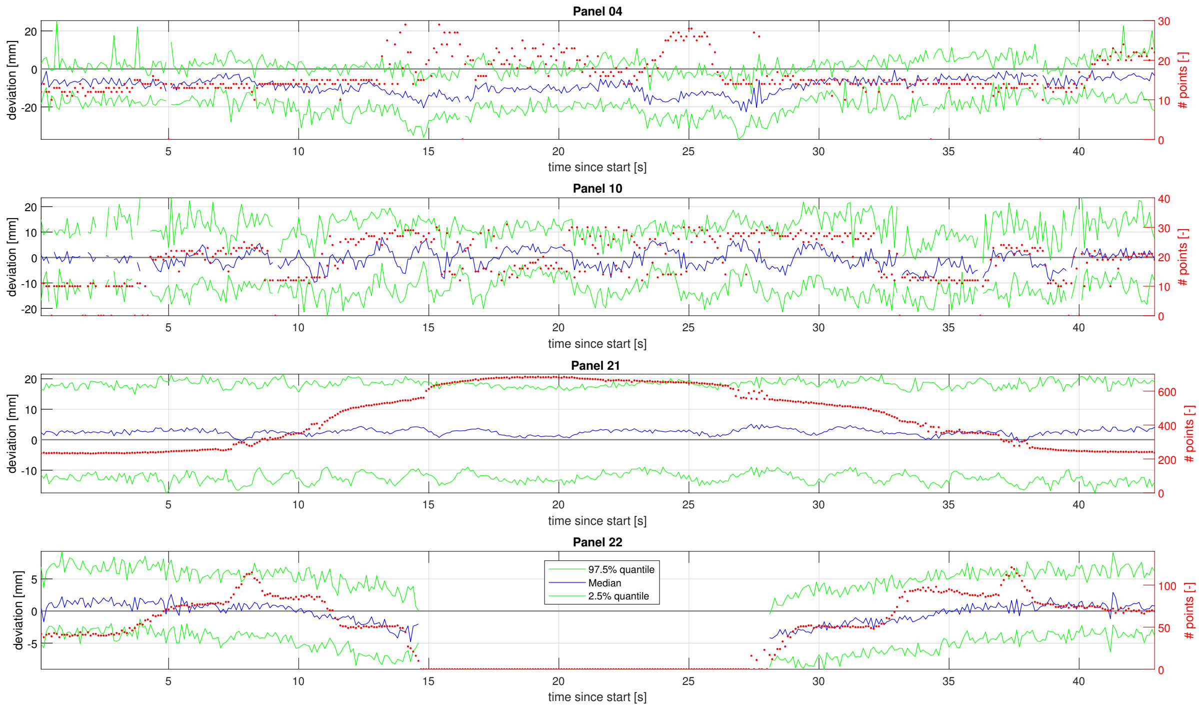

Figure 3.6: *Median and quantiles of distances between point clouds and reference panels over time, points computed using ensemble referencing, panel number from Figure 2.3, numbers of points assigned to panel in red. Figure from publication* ***C****.*

The evaluation of the results in the context of the estimated uncertainties follows in Section 3.3.

In publication **B**, an **ESKF** is used to process the observations of the sensors. The prediction of the filter follows the strap-down paradigm, which integrates IMU observations to propagate the pose to the next epoch. The update step allows for a functional model with implicit measurement equations. Specifically, the functional model for publication **B** uses distances between assigned LiDAR points to a known map to update the states. Additionally, the interpolation scheme is shown. The basic formulation of the ESKF is based on Solà (2017) with changes to the update step initially proposed for an iterative extended KF by Vogel et al. (2018). Here, all formulas are shown, as the further developments shown in Section 4.1 expand this algorithm.

The utilization of the IMU requires the modeling of the IMU biases $\boldsymbol{a}$ for the accelerometer and $\boldsymbol{\omega}$ for the gyroscope, as the bias instability of the sensors leads to deviations in the results otherwise. Here, effects such the effect of the earth rotation and Coriolis force are omitted, as the contribution lies in the update step and the development in this work is aimed at systems with low-cost IMUs with higher bias instability. For other systems or applications, further corrections can be applied. This leads to the state vector

$$\boldsymbol{x} = \begin{bmatrix} {\boldsymbol{t}_{\mathrm{b}}^{\mathrm{l}}}^T & {\boldsymbol{v}_{\mathrm{b}}^{\mathrm{l}}}^T & {\boldsymbol{q}_{\mathrm{b}}^{\mathrm{l}}}^T & \boldsymbol{a}_b^T & \boldsymbol{\omega}_b^T \end{bmatrix}^T . \tag{3.12}$$

While the vector $\boldsymbol{t}_{\mathrm{b}}^{\mathrm{l}}$ and $\boldsymbol{v}_{\mathrm{b}}^{\mathrm{l}}$ contain the position and corresponding velocities between the body frame b and the superordinate frame l, the vector $\boldsymbol{q}_{\mathrm{b}}^{\mathrm{l}}$ is a quaternion describing the orientation of the system. A quaternion is used to provide full flexibility concerning the possible rotations while avoiding a possible gimbal lock. The formulation of the problem as an ESKF is beneficial when using quaternions. The error state is

$$\delta\boldsymbol{x} = \begin{bmatrix} \delta{\boldsymbol{t}_{\mathrm{b}}^{\mathrm{l}}}^T & \delta{\boldsymbol{v}_{\mathrm{b}}^{\mathrm{l}}}^T & \delta{\boldsymbol{\theta}_{\mathrm{b}}^{\mathrm{l}}}^T & \delta\boldsymbol{a}_b^T & \delta\boldsymbol{\omega}_b^T \end{bmatrix}^T \tag{3.13}$$

with $\delta\boldsymbol{\theta}_{\mathrm{b}}^{\mathrm{l}}$ as the 3×1 rotation vector denoting the error in the orientation. The prediction between the epochs defined by the IMU observations is performed by

$$\boldsymbol{x}^{(k)} = \mathbf{f}\left(\boldsymbol{x}^{(k-1)}, \boldsymbol{u}_o^{(k)}\right). \tag{3.14}$$

Here, the accelerations $\boldsymbol{a}_o$ and rotation rates $\boldsymbol{\omega}_o$ observed by the IMU are included as control inputs $\boldsymbol{u}_o$. The IMU observations are composed of the actual accelerations and rotation rates, biases (denoted by b), noise (denoted by n), and gravity $\boldsymbol{g}$ in the case of the acceleration

$$\boldsymbol{\omega}_o = \boldsymbol{\omega} + \boldsymbol{\omega}_b + \boldsymbol{\omega}_n, \tag{3.15}$$

$$\boldsymbol{a}_o = \boldsymbol{a} + \boldsymbol{a}_b + \boldsymbol{a}_n + \left(\mathbf{R}_{\mathrm{b}}^{\mathrm{l}}\right)^T \boldsymbol{g}_{\mathrm{l}}. \tag{3.16}$$

After correcting the IMU observations for the biases

$$\hat{\boldsymbol{a}} = \boldsymbol{a}_o - \boldsymbol{a}_b \tag{3.17}$$

$$\hat{\boldsymbol{\omega}} = \boldsymbol{\omega}_o - \boldsymbol{\omega}_b, \tag{3.18}$$

the prediction

$$\hat{\boldsymbol{x}}_{-}^{(k)} = \boldsymbol{F}_{\boldsymbol{x}}^{(k)} \hat{\boldsymbol{x}}_{+}^{(k-1)} \tag{3.19}$$

is performed using the transition matrix

$$\boldsymbol{F}_{\boldsymbol{x}} = \left.\frac{\partial \mathbf{f}}{\partial \delta \boldsymbol{x}}\right|_{\boldsymbol{x},\boldsymbol{u}} \tag{3.20}$$

$$= \begin{bmatrix} \mathbf{I} & \mathbf{I}\Delta t & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & -\left[\mathbf{R}_{\mathrm{b}}^{\mathrm{l}}\hat{\boldsymbol{a}}\right]_{\times} \Delta t & -\mathbf{R}_{\mathrm{b}}^{\mathrm{l}}\Delta t & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & -\mathbf{R}_{\mathrm{b}}^{\mathrm{l}}\Delta t \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{3.21}$$

where $[\cdot]_{\times}$ denotes the skew operator. The process noise is approximated by using the uncertainty information of the IMU used to provide the control inputs. First, the uncertainties are collected in

$$\mathbf{Q}_{\mathbf{I}} = \mathrm{blkdiag}\left(\boldsymbol{\sigma}_{\tilde{a}_n}^2 \Delta t^2\,\mathbf{I}, \boldsymbol{\sigma}_{\tilde{\omega}_n}^2 \Delta t^2\,\mathbf{I}, \boldsymbol{\sigma}_{a_\eta}^2 \Delta t\,\mathbf{I}, \boldsymbol{\sigma}_{\omega_\eta}^2 \Delta t\,\mathbf{I}\right) \tag{3.22}$$

with the identity matrix $\mathbf{I}$ and

- $\boldsymbol{\sigma}_{\tilde{a}_n}$: noise density of accelerometer [3×1],
- $\boldsymbol{\sigma}_{\tilde{\omega}_n}$: noise density of gyroscope [3×1],
- $\boldsymbol{\sigma}_{a_\eta}$: bias instability of accelerometer [3×1],
- $\boldsymbol{\sigma}_{\omega_\eta}$: bias instability of gyroscope [3×1].

Finally, the uncertainties can be transformed and applied to the VCM of the state

$$\boldsymbol{F}_{\mathbf{I}}^{(k)} = \begin{bmatrix} \mathbf{0}_{[3\times 12]} \\ \mathrm{blkdiag}(-\mathbf{R}_{\mathrm{b}}^{\mathrm{l}} \quad -\mathbf{I} \quad \mathbf{I} \quad \mathbf{I}) \end{bmatrix}, \tag{3.23}$$

$$\mathbf{Q}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}},-}^{(k)} = \boldsymbol{F}_{\boldsymbol{x}}^{(k)} \mathbf{Q}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}}^{(k-1)} \left(\boldsymbol{F}_{\boldsymbol{x}}^{(k)}\right)^T + \boldsymbol{F}_{\mathbf{I}}^{(k)} \mathbf{Q}_{\mathbf{I}} \left(\boldsymbol{F}_{\mathbf{I}}^{(k)}\right)^T. \tag{3.24}$$

The update step of the ESKF applies an implicit observation model to update the state vector. The update procedure is iterative, updating the development point for the computation multiple times until convergence. The initial values for the necessary vectors are

$$\hat{\boldsymbol{x}}_+^{(k,m=1)} = \hat{\boldsymbol{x}}_-^{(k)}, \quad \delta\boldsymbol{x}^{(k,m=0)} = \mathbf{0} \quad \text{and} \quad \hat{\boldsymbol{\ell}}^{(k,m=1)} = \boldsymbol{\ell}^{(k)}$$

with the observations $\boldsymbol{\ell}$. Similar to the GHM, the misclosure $\boldsymbol{h}$, the design matrix $\mathbf{A}$ and the restriction matrix $\mathbf{B}$ are computed first. The VCM of the observations $\mathbf{Q}_{\boldsymbol{\ell\ell}}^{(k)}$ contains the uncertainties of the observations and is used for the weighting of the observations. The algorithm to compute the error state using the measurement model h is

$$\boldsymbol{h}^{(k,m)} = \boldsymbol{h}\left(\hat{\boldsymbol{x}}_+^{(k,m)}, \hat{\boldsymbol{\ell}}^{(k,m)}\right) \tag{3.25}$$

$$\mathbf{A}^{(k,m)} = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_+^{(k,m)}, \hat{\boldsymbol{\ell}}^{(k,m)}} \cdot \left.\frac{\partial \boldsymbol{x}}{\partial \delta\boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_+^{(k,m)}} \tag{3.26}$$

$$\mathbf{B}^{(k,m)} = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\ell}}\right|_{\hat{\boldsymbol{x}}_+^{(k,m)}, \hat{\boldsymbol{\ell}}^{(k,m)}} \tag{3.27}$$

$$\mathbf{O}^{(k,m)} = \mathbf{A}^{(k,m)}\, \mathbf{Q}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}},-}^{(k)} \left(\mathbf{A}^{(k,m)}\right)^T \tag{3.28}$$

$$\mathbf{Q}_{\mathbf{hh}}^{(k,m)} = \mathbf{B}^{(k,m)}\, \mathbf{Q}_{\boldsymbol{\ell\ell}}^{(k)} \left(\mathbf{B}^{(k,m)}\right)^T \tag{3.29}$$

$$\mathbf{Q}_{\boldsymbol{rr}}^{(k,m)} = \mathbf{O}^{(k,m)} + \mathbf{Q}_{\mathbf{hh}}^{(k,m)} \tag{3.30}$$

$$\mathbf{K}^{(k,m)} = \mathbf{Q}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}},-}^{(k)} \left(\mathbf{A}^{(k,m)}\right)^T \left(\mathbf{Q}_{\boldsymbol{rr}}^{(k,m)}\right)^{-1} \tag{3.31}$$

$$\boldsymbol{r}^{(k,m)} = \mathbf{B}^{(k,m)} \left(\boldsymbol{\ell}^{(k,m)} - \boldsymbol{\ell}^{(k)}\right) + \mathbf{A}^{(k,m)} \delta\boldsymbol{x}^{(k,m-1)} \tag{3.32}$$

$$\boldsymbol{w}^{(k,m)} = \boldsymbol{h}^{(k,m)} + \boldsymbol{r}^{(k,m)} \tag{3.33}$$

$$\delta\boldsymbol{x}^{(k,m)} = -\mathbf{K}^{(k,m)} \boldsymbol{w}^{(k,m)} \tag{3.34}$$

$$\hat{\boldsymbol{v}}^{(k,m)} = -\mathbf{Q}_{\boldsymbol{\ell\ell}}^{(k)} \left(\mathbf{B}^{(k,m)}\right)^T \left(\mathbf{Q}_{\boldsymbol{rr}}^{(k,m)}\right)^{-1} \boldsymbol{w}^{(k,m)} \tag{3.35}$$

$$\hat{\boldsymbol{\ell}}^{(k,m+1)} = \boldsymbol{\ell}^{(k)} + \hat{\boldsymbol{v}}^{(k,m)}. \tag{3.36}$$

As intermediate values, the VCMs of the misclosures $\mathbf{Q}_{\mathbf{hh}}$ and of the combined misclosures $\mathbf{Q}_{\boldsymbol{rr}}$ are computed. The Kalman gain $\mathbf{K}$ is used to compute the error state of the iteration. Finally, the residuals $\hat{\boldsymbol{v}}$ can be computed. At the end, the state vector can be updated by injecting the error state

$$\hat{\boldsymbol{x}}_+^{(k,m+1)} = \hat{\boldsymbol{x}}_-^{(k)} \boxplus \delta\boldsymbol{x}^{(k,m)} = \begin{bmatrix} \boldsymbol{t}_{\mathrm{b}-}^{\mathrm{l}} + \delta\boldsymbol{t}_{\mathrm{b}}^{\mathrm{l}} \\ \boldsymbol{v}_{\mathrm{b}-}^{\mathrm{l}} + \delta\boldsymbol{v}_{\mathrm{b}}^{\mathrm{l}} \\ \boldsymbol{q}\{\delta\boldsymbol{\theta}\} \otimes \boldsymbol{q}_{\mathrm{b}-}^{\mathrm{l}} \\ \boldsymbol{b}_{a-} + \delta\boldsymbol{b}_a \\ \boldsymbol{b}_{\omega-} + \delta\boldsymbol{b}_\omega \end{bmatrix}. \tag{3.37}$$

The iteration continues until the change of the error state is smaller than a defined threshold

$$\|\delta\mathbf{x}^{(k,m+1)} \boxminus \delta\mathbf{x}^{(k,m)}\| < \epsilon. \tag{3.38}$$

At last, the epoch can be finalized by updating the VCM of the state vector:

$$\boldsymbol{L}^{(k)} = \boldsymbol{I} - \mathbf{K}^{(k,m)}\mathbf{A}^{(k,m)}, \tag{3.39}$$

$$\boldsymbol{Q}^{(k)}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}},+} = \boldsymbol{L}^{(k)}\boldsymbol{Q}^{(k)}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}},-}\left(\boldsymbol{L}^{(k)}\right)^T + \mathbf{K}^{(k,m)}\mathbf{Q}^{(k,m)}_{\mathbf{hh}}\left(\mathbf{K}^{(k,m)}\right)^T. \tag{3.40}$$

Depending on the application, epochs without observations might occur. In this case, the predicted state and the corresponding VCM are assumed to be the result of the epoch

$$\hat{\boldsymbol{x}}^{(k)}, \mathbf{Q}^{(k)}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}} = \begin{cases} \hat{\boldsymbol{x}}^{(k,m)}_{+}, \mathbf{Q}^{(k)}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}},+} & \text{with update in } (k), \\ \hat{\boldsymbol{x}}^{(k)}_{-}, \mathbf{Q}^{(k)}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}},-} & \text{else.} \end{cases} \tag{3.41}$$

The functional model in publication **B** integrates LiDAR points assigned to a known map. The goal is the minimization of the distances between the observed points and planes to update the system pose. The planes of the map are assumed to be deterministic. This leads to one restriction per measured point, which in turn is composed of three observations. Similar to publication **C**, many LiDAR observations are available per epoch. In contrast to the linear KF shown previously, the LiDAR observations are used to determine the pose of the system. To deal with this issue, the time stamps of the individual points t_i are used to compute the interpolation factor

$$\tau_i = \frac{t_i - t^{(k-1)}}{\Delta t} \tag{3.42}$$

using the time of the last epoch $t^{(k-1)}$. Afterwards, the pose of the system is linearly interpolated between the pose of the last epoch and the current pose

$$\boldsymbol{t}^{\mathrm{l}}_{\mathrm{b}\tau_i} = (1-\tau_i)\,\hat{\boldsymbol{t}}^{\mathrm{l},(k-1)}_{\mathrm{b}} + \tau_i\,\hat{\boldsymbol{t}}^{\mathrm{l},(k,m)}_{\mathrm{b}}, \tag{3.43}$$

$$\Delta\psi = \arccos\left(\left(\hat{\boldsymbol{q}}^{\mathrm{l},(k-1)}_{\mathrm{b}}\right)^T \cdot \hat{\boldsymbol{q}}^{\mathrm{l},(k,m)}_{\mathrm{b}}\right), \tag{3.44}$$

$$\boldsymbol{q}^{\mathrm{l}}_{\mathrm{b}\tau_i} = \hat{\boldsymbol{q}}^{\mathrm{l},(k-1)}_{\mathrm{b}}\frac{\sin\left((1-\tau_i)\Delta\psi\right)}{\sin(\Delta\psi)} + \hat{\boldsymbol{q}}^{\mathrm{l},(k,m)}_{\mathrm{b}}\frac{\sin(\tau_i\Delta\psi)}{\sin(\Delta\psi)}. \tag{3.45}$$

with spherical linear interpolation (SLERP) for the quaternions. The limitation of linear interpolation is done to keep the computation of the partial derivatives for (3.26) and (3.27) tractable. Finally, the interpolated pose can be used to compute the position of the LiDAR points in the superordinate frame and compute the distance to the assigned plane

$$\boldsymbol{p}_{\mathrm{l},i} = \begin{bmatrix} x_{\mathrm{l}} \\ y_{\mathrm{l}} \\ z_{\mathrm{l}} \end{bmatrix}_i = \boldsymbol{t}^{\mathrm{l}}_{\mathrm{b}\tau_i} + \mathbf{R}\left\{\boldsymbol{q}^{\mathrm{l}}_{\mathrm{b}\tau_i}\right\}\begin{bmatrix} x_{\mathrm{b}} \\ y_{\mathrm{b}} \\ z_{\mathrm{b}} \end{bmatrix}_i, \tag{3.46}$$

$$\mathrm{h}_i = \mathrm{h}\left(\boldsymbol{x}, \boldsymbol{\ell}_i, \boldsymbol{a}_{\mathrm{l},j}\right) = \left[n_{x,\mathrm{l}}\, n_{y,\mathrm{l}}\, n_{z,\mathrm{l}}\, d_{\mathrm{l}}\right]_j \cdot \begin{bmatrix} \boldsymbol{p}_{\mathrm{l},i} \\ 1 \end{bmatrix}. \tag{3.47}$$

To investigate the influence of pose interpolation in the functional model, a simulated environment is used to generate random trajectories with the corresponding IMU and LiDAR observations. The LiDAR observations are computed using ray-plane-intersections, and the IMU observations are derived from time differencing of the poses. The simulated environment with a single simulated LiDAR frame is shown in Figure 2.1. The results for 100 randomly generated trajectories of 60 s length, with 10 MC runs each, are shown in Figure 3.7. The first 5 s of the trajectory are stationary for the filter initialization. For the remaining time, the system freely moves in the room, rotating along all axes. The RMSE is computed using the true values (denoted by $\tilde{\cdot}$) for each epoch across the whole pose and all $M = 1000$ runs

$$\mathrm{RMSE}^{(k)} = \sqrt{\frac{1}{M}\sum_{m=1}^{M}\frac{1}{6}\left(\Delta\boldsymbol{\theta}^{(k,m)}\right)^T\Delta\boldsymbol{\theta}^{(k,m)}} \tag{3.48}$$

using the deviations computed for each realization m

$$\Delta\boldsymbol{t}^{(k)} = \tilde{\boldsymbol{t}}_{\mathrm{b}}^{1,(k)} - \hat{\boldsymbol{t}}_{\mathrm{b}}^{1,(k)}, \tag{3.49}$$

$$\Delta\boldsymbol{q}^{(k)} = \theta\left\{\tilde{\boldsymbol{q}}_{\mathrm{b}}^{1,(k)} \boxminus \hat{\boldsymbol{q}}_{\mathrm{b}}^{1,(k)}\right\}, \tag{3.50}$$

$$\Delta\boldsymbol{\theta}^{(k)} = \begin{bmatrix}\Delta\boldsymbol{t}^{(k)}\\ \Delta\boldsymbol{q}^{(k)}\end{bmatrix}. \tag{3.51}$$

Comparing the values, it can be seen that pose interpolation (**base**) leads to a constant deviation along the whole trajectory, indicating that the interpolation works as expected. For faster velocities or rotation rates, these deviations might increase during movement. The limiting factor is the measurement rate of the IMU. It should be noted that the deviations of the variant without interpolation (**nolerp**) lead to smaller deviations in the stationary period compared to the interpolated variant. This is caused by implicit weighting included in the time interpolation. The factor τ_i acts as a damping factor, reducing the influence on the estimation by processing the observation under the assumption of dynamic states.

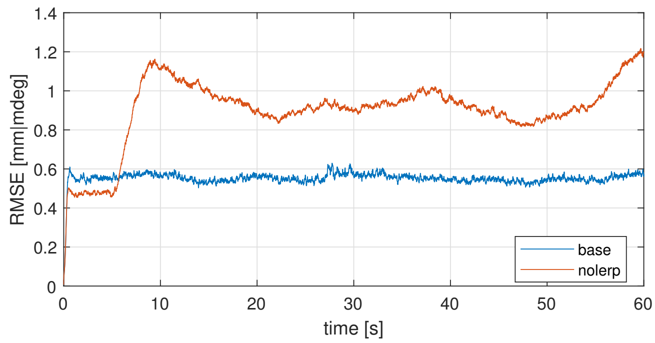

Figure 3.7: *Comparison of RMSE with and without using pose interpolation based on 100 generated trajectories with 10 MC runs each. The system is stationary for the first 5 seconds. Figure from publication **B**.*

The contributions of publications **B** and **C** focus on consistent data fusion, as formulated in research question **2**. The developed ESKF provides a possibility to fuse IMU and LiDAR observations, taking the time stamps of the individual observations into account. This allows for a consistent fusion of observations with an implicit measurement model and quaternions. The results of the prior calibration are utilized to estimate the trajectory under consideration of all known uncertainties. Especially, the novel ensemble georeferencing demonstrates the influence of the uncertainties of prior steps by improving the georeferenced point clouds using a MC propagation of uncertainties. The uncertainties of the LiDAR observations and calibration are the main influences in the shown application. The additional investigations in Chapter 4 develop the ESKF further to estimate the uncertainties for different surfaces during operation.

3.3 Uncertainty Evaluation

The last section showed aspects of sensor data fusion to achieve improved accuracy. The time-dependent processing of point clouds and the inclusion of uncertainties into the georeferencing process both show improvements. In this section, the estimated uncertainties of both methods are investigated. This is done for the **georeferenced point clouds**, as shown in publication **C**, and for the **poses** estimated by the ESKF, as shown in publication **D**. The evaluation of the georeferenced point cloud uses the VCMs estimated by the MC variance propagation in combination with the reference panels to evaluate the uncertainties. The eigenvector matrix of the plane estimation is used to transform the VCMs, yielding the correct uncertainty for the previously computed distance. The evaluation of the estimated poses is based on the combination of publications **B** and **C**. The ESKF from publication **B** is applied to the dataset presented in publication **C** to estimate the pose of the system. For the evaluation, the poses provided by the T-Probe measurements are used. For a thorough investigation, two steps are performed. First, the dataset is simulated and processed with two different sets of uncertainties to investigate the influence of the processing chain. Then, the real dataset is processed to compare the results.

The evaluation of the **point clouds** focuses on the observation space by computing the distances to known reference panels and scaling these distances by their corresponding uncertainties. These corresponding uncertainties are taken from the VCMs for each point provided by the MC variance propagation. These VCMs include all uncertainties accumulated over the georeferencing process. Simply computing Helmert's point error (Stenz et al., 2020) for each point would not properly represent the uncertainties, as the computed distances are one-dimensional. To compute the uncertainties corresponding to the distances, the eigenvector matrix for the reference panels is used to rotate the VCMs of all assigned points. Then, the variance of the component corresponding to the normal vector of the panel $\sigma_{i,\lambda_{j,1}}$ can be used to scale the distance $d_{i,j}$

$$\tilde{d}_{i,j} = \frac{d_{i,j}}{\sigma_{i,\lambda_{j,1}}}. \tag{3.52}$$

The principle for the evaluation is shown in Figure 3.8. Both the frame of the reference panel $(\lambda_{j,1}, \lambda_{j,2})$ and the superordinate frame (x, y) are known.

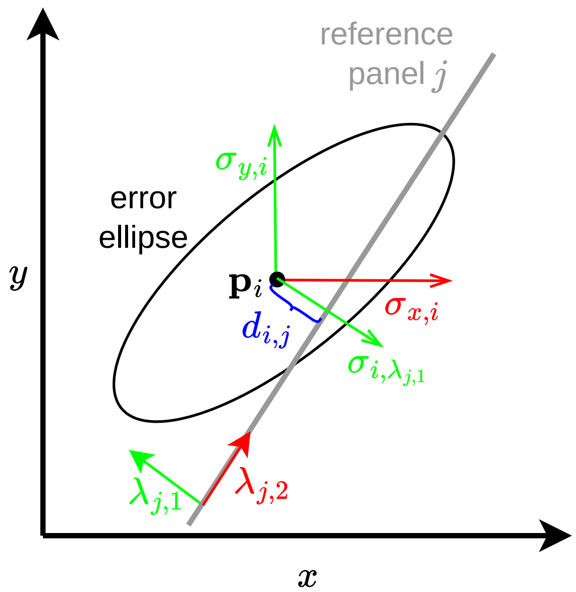

Figure 3.8: *Principle of the evaluation for the predicted point uncertainties in comparison to the reference panels. Relevant frames are shown for the superordinate frame and reference panel. Figure from publication* ***C****.*

Figure 3.9 shows the standardized distances of the points, computed by ensemble referencing, for the same panels as shown in Figures 3.5 and 3.6. The procedure rescales the distances, which are then compared to a t-distribution $t_{[2.5\%,97.5\%],10} = [-2.23, 2.23]$. The t-distribution is used, as the empirical results are to be tested. The usage of 10 degrees of freedom is motivated by the minimum number of points on the panels at any given time. For panels **21** and **22**, the quantiles are within the interval for the whole time frame. Since not all influences are visible on a single panel, panels like **22** can show comparatively small dispersion, while the other two panels demonstrate higher deviations with values up to ± 4. The small dispersion is caused by low measurement noise compared to systematic effects. The high deviations might be caused by overly optimistic estimation of the orientation from the calibration or the georeferencing. At the same time, these panels are small, leading to few points being assigned for the evaluation, which limits the significance of the results.

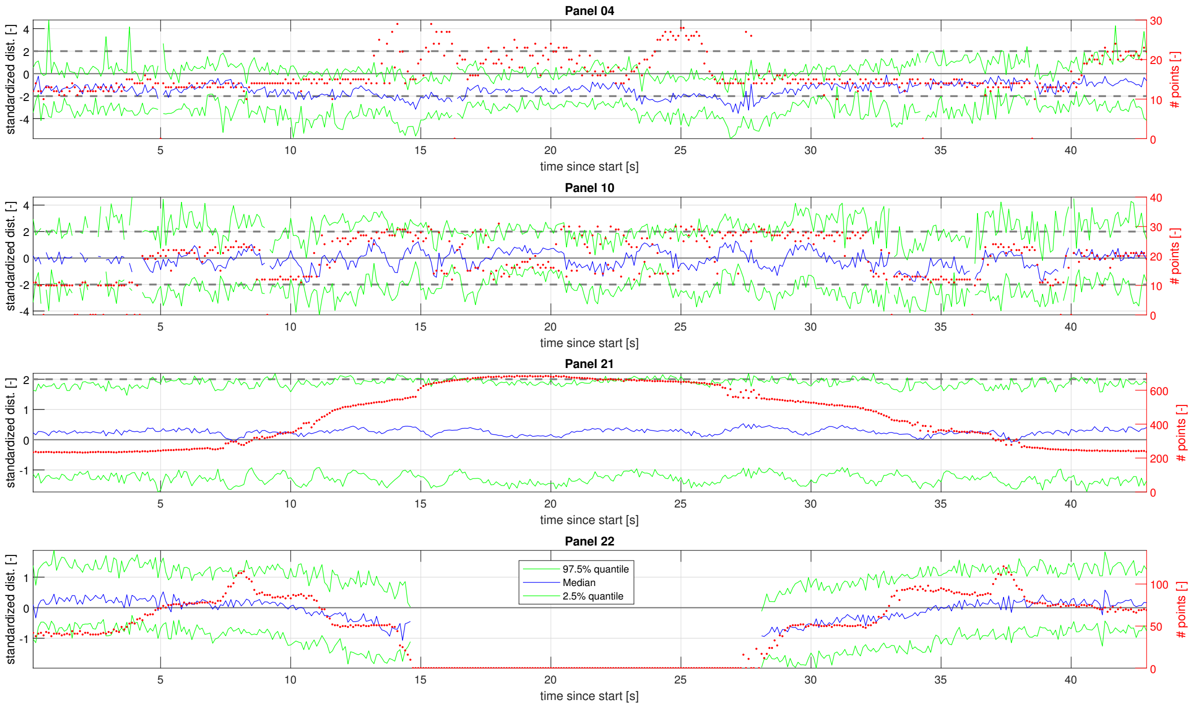

Figure 3.9: *Median and quantiles of standardized distances between point clouds and reference panels over time, ensemble referencing using MC samples, panel numbers from Figure 2.3, number of points assigned to panel in red. Figure from publication **C**.*

Alternatively, the **estimated pose** can be evaluated directly. In this case, a ground truth for the trajectory has to exist, making this type of evaluation more difficult compared to the point cloud due to the requirement for additional sensor fusion for tracking the system during operation. Here, the dataset from publication **C** is used by applying the ESKF to the LiDAR point clouds with the reference panels to compute the georeferencing while using the T-Probe observations as the ground truth. All necessary transformation parameters are known, as shown before. The investigation of the uncertainties is structured in two parts. First, the dataset is simulated to show the differences between an ideal situation with perfectly estimated parameters and a more realistic scenario with remaining deviations in the estimated parameters. Second, the real data is processed to show the actual influence.

The simulations are based on the measured T-Probe trajectory. The sensor data for the LiDAR and IMU are produced similarly to the method described in publication **B**. For 100 MC simulations, the generated sensor data is perturbed using white noise with the assumed uncertainty of the sensors. For the dataset **simu 1**, this serves as the sensor data for the processing. For **simu 2**, the remaining systematic deviations from the calibration are simulated by introducing deviations into the transformation parameters. These values are generated only once per realization and kept constant for all observations. This simulates the imperfect determination of the calibration parameters and their influence on the processing in the KF. Opposed to the processing in publication **B**, variance propagation is applied to define a block diagonal VCM of the observations to include the uncertainties of the transformations. Afterward, these two datasets are processed using the ESKF introduced in section 3.2.

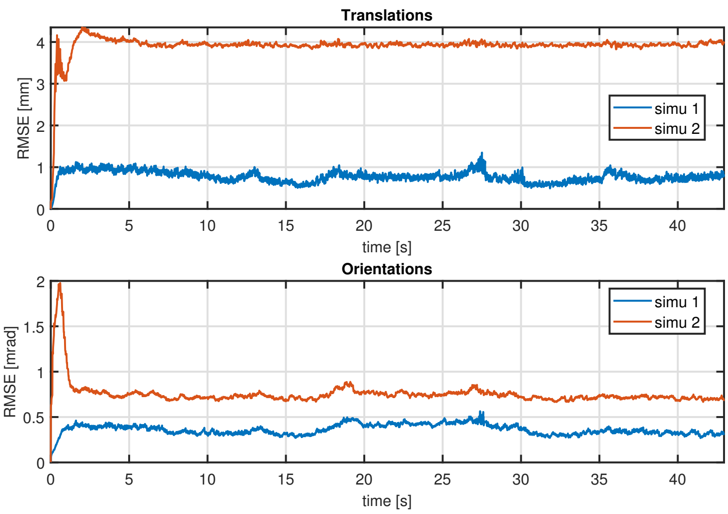

Figure 3.10: *RMSE values for 100 simulated MC runs with different uncertainties applied. Figure from publication **D**.*

After processing the sensor data, the RMSE is computed for translations and orientations (similar to equation (3.48)). The results are shown in Figure 3.10. Both datasets show a nearly constant RMSE over the duration of the experiment, except for the initialization in the first few seconds. The dataset with systematic deviation shows approximately four times higher deviations in the translations and two times higher deviations in the orientations. This is expected, as additional deviations in the calibrations also affect the results. In the next step, the Mahalanobis distance is computed to investigate whether the additional pose deviations are properly represented in the estimated uncertainty. The Mahalanobis distance over all realizations

$$\sqrt{\bar{d}_m}^{(k)} = \sqrt{\frac{1}{M}\sum_{m=1}^{M} \left(\Delta\boldsymbol{\theta}^{(k,m)}\right)^T \left(\boldsymbol{\Sigma}^{(k,m)}_{pose,\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}}\right)^{-1} \Delta\boldsymbol{\theta}^{(k,m)}} \tag{3.53}$$

weights the RMSE by the inverse VCM of the evaluated quantities, in this case, the VCM of the pose of the system $\boldsymbol{\Sigma}^{(k)}_{pose,\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}}$.

The results are shown in Figure 3.11. The computed distances are compared against a χ^2-interval with $6 \cdot 100$ degrees of freedom (Bar-Shalom et al., 2004). The hypothesis test is performed as a two-sided test. Values below the interval are overly pessimistic, while values above are considered overly optimistic. The dataset **simu 1** leads to overly pessimistic results, due to the nonlinearity of the functional model. Dataset **simu 2** leads to higher values, indicating an overly optimistic estimation. Three effects are visible. In the beginning, the initialization of the filter leads to a fast increase in the Mahalanobis distance. For the remainder of the duration, a trend is visible, caused by the difference in the constellation between the system and the panels, leading to a change in both the number of points and the geometry of the estimation. Lastly, a high-frequency oscillation is visible, caused by the variation in deviations depending on the part of the room scanned in the corresponding epoch. The filter processes the data at 100 Hz, while the LiDAR is scanning with a rotation rate of 10 Hz. This mismatch leads to an oscillation of 10 Hz. The simulation demonstrates that remaining systematic deviations severely affect the results, even when the uncertainties are integrated into the processing. The modeling in the KF works best for zero-mean white noise. Deviations from this assumption lead to inconsistent estimations and can bias the results. For this application, the assumption is violated due to the same influence of the calibration parameters on all observations and missing corrections for the sensor.

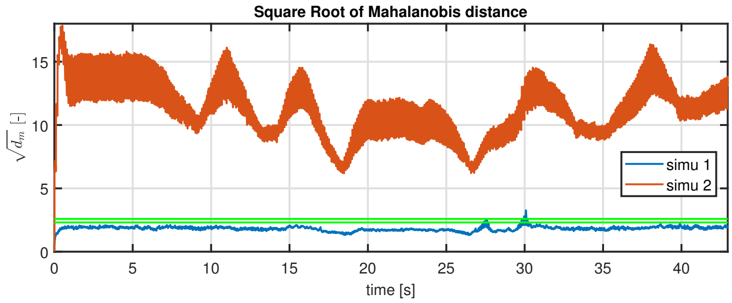

Figure 3.11: *Mahalanobis distances for 100 simulated MC runs with different uncertainties applied. Figure from publication **D**.*

The results of the simulation are compared to the **real data**. The data recorded by the LiDAR and IMU are processed, and the trajectory is compared to the T-Probe trajectory. In contrast to the results in publication **B**, this provides a more realistic impression of the algorithm's performance. The deviations are shown in Figure 3.12. As expected, a higher magnitude of deviations compared to the simulations is observed. The highest deviations are observed for the translation of the z-axis, showing a negative offset along the whole trajectory, and in the rotation around the y-axis of the system. Both deviations are most likely caused by tilted panels affecting the height component.

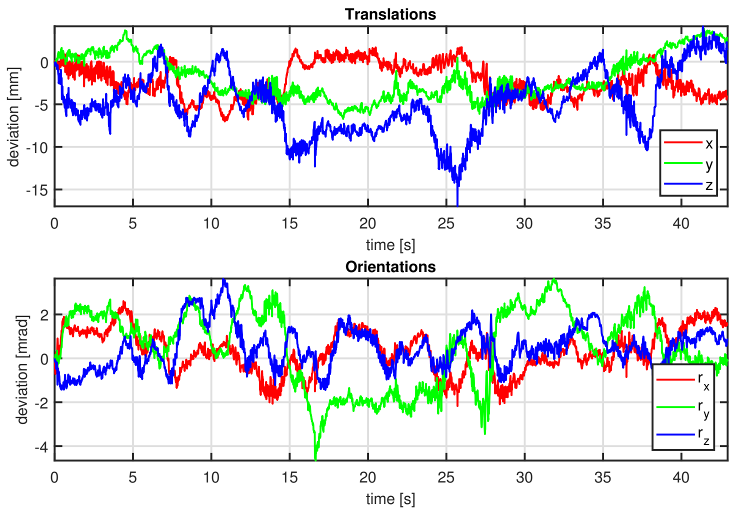

Figure 3.12: *Deviations of the pose of a MSS referenced by the ESKF using real data. Figure from publication **D**.*

When comparing the deviations to the Mahalanobis distance shown in Figure 3.13, the values again exceed the χ^2-interval. Here, the interval is wider because only one run is processed. Similar to the dataset **simu 2**, a systematic pattern in the values depending on the system's position can be seen, with similar values for the beginning and end of the trajectory.

In the final step, the individual elements of the pose are investigated. To achieve this, the deviations of each element are divided by the estimated standard deviation of the corresponding epoch. The results are shown in Figure 3.14. The purple lines are shown at -2 and 2 to indicate a confidence level of about 95%. While the general structure of the deviations is similar, the magnitudes are scaled differently. Specifically, the z-translation and the pitch angle are smaller compared to the other deviations, indicating a higher standard deviation relative to the other elements of the pose (compared to the results in Figure 3.12). This is most likely caused by the

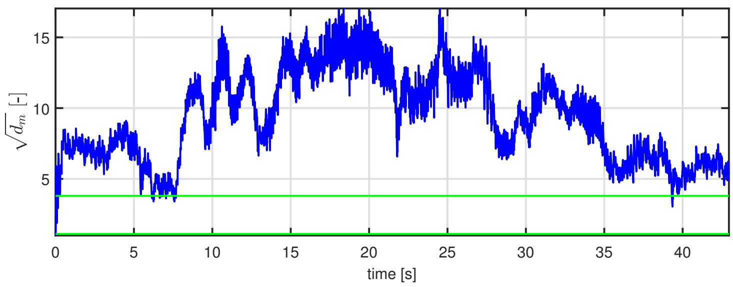

Figure 3.13: *Mahalanobis distance of the pose of an MSS referenced by the ESKF using real data. Figure from publication **D**.*

point distribution on the reference panels. Generally, no clear cause for the inconsistent estimation can be identified, as all elements exceed the interval at different times. This is caused by the complex interaction between the system's pose and the geometry of the panels in the room, additionally affected by remaining deviations in the intrinsic and extrinsic calibration of the LiDAR.

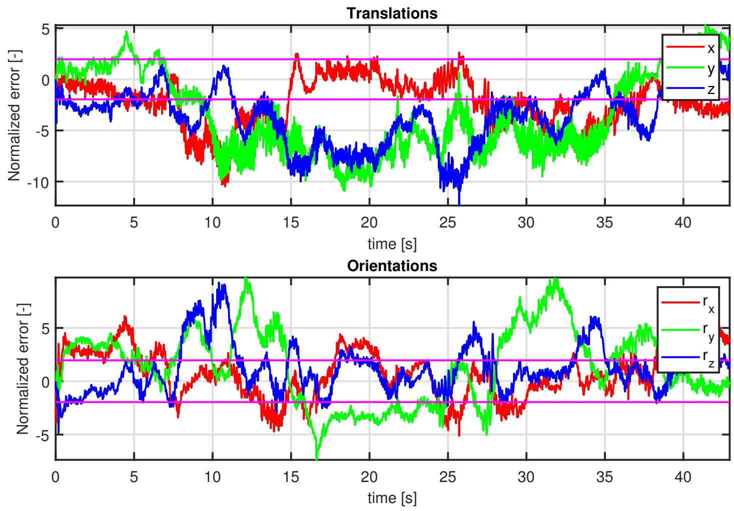

Figure 3.14: *Standardized deviations of the pose of a MSS referenced by the ESKF using real data. Purple lines show an interval of a normal distribution with $\alpha = 5\%$. Figure from publication **D**.*

The evaluation of the results demonstrates two approaches for providing quality metrics for data processed by LiDAR-based MSS. On one hand, as illustrated in publication **D**, the trajectory can be assessed since georeferencing plays a critical role in determining the quality of the final point cloud. Comparisons with reference trajectories can be conducted here. A novel aspect is the weighting by uncertainties, either for the entire pose using the Mahalanobis distance or by weighting individual pose components based on their respective standard deviations. On the other hand, as shown in publication **C**, the georeferenced point cloud captured by the LiDAR can be evaluated. Again, the uncertainties of individual points are used to weight the distance to reference point clouds or geometries. Both methods provide quality metrics to address research question **2**.

To summarize the contributions of publications **A** to **D**, the following progress is made concerning the research questions. Research question **1** is addressed by publication **A**. The presented calibration scheme improves the results by integrating an intrinsic calibration of the used LiDAR and introducing multiple positions to decorrelate the estimated parameters. The validation of the estimated calibration parameters is performed in publications **C** and **D** , showing some remaining deviations. Further limitations include the dependency on the surfaces used for calibration. The goal of achieving consistent predictions for kinematically acquired point clouds of research question **2** is approached from several directions. Publication **B** presents an ESKF to improve the georeferencing of systems in a map matching framework to reduce systematic deviations due to motion during the measuring process. Investigations based on simulations show the improvement due to the interpolation. Publication **C** demonstrate improvements using ensemble georeferencing for kinematically acquired point clouds. Answering research question **3** and improving on the results for the first two research questions is the focus of Chapter 4.

4 Additional Investigations

Continuing from the previous chapter, this chapter addresses the unresolved issues from the relevant publications and provides answers to the research questions. A methodology for recursive VCE within KFs with implicit measurement equations is introduced to adapt the stochastic model during operation. Subsequently, this methodology is applied, and the estimation of plane-specific distance offsets is demonstrated for the position tracking described in Section 3.2. The methodology is then tested using the LUCOOP data set (Axmann et al., 2023). This testing is conducted in two stages: initially with simulated data to validate the methodology, followed by applying the algorithm to real data to assess its effectiveness under real conditions.

4.1 Dual Kalman Filtering

In many applications, information about the stochastic properties of observations is limited. This limitation arises from factors such as insufficient sensor information or complex measurement setups with numerous influences. Following up on the issues shown in Section 2.2, this thesis introduces an estimation scheme designed to provide stable estimation while tracking stochastic information over multiple epochs. To accomplish this, a second state vector is employed to track the VCs. This separation is possible because the states are assumed to be orthogonal to the observations, eliminating the possibility of correlations. The setup of the estimation uses the filter presented in Section 3.2 to estimate the pose of the system and the biases of the IMU. The second filter is used to process the stochastic information needed to properly weight the LiDAR observations. It keeps track of the VCs, which are estimated after every epoch of the first filter. The filter is initialized in the usual manner, setting the initial VCs to 1. A significant advantage of this second filter is the ability to introduce uncertainties for the initial VCs. These initial uncertainties are a design parameter with limited impact on the remaining estimation. At the beginning of the estimation, the initial VCs with uncertainties serve as regularization in the VCE, preventing the aforementioned estimation failures. For the recursive VCE, the uncertainties of the VCs are updated to track the information used to adjust the individual VCs. It is assumed that the behavior of the LiDAR capturing the material is constant, so no process noise is applied. This results in the situation in which observations of earlier epochs contribute more to the VCs compared to observations from later epochs, as the uncertainties decrease quickly. The introduction of process noise would change this, but since plane groups are observed infrequently (often only for one time interval), finding a suitable magnitude would be intractable.

The implementation of recursive VCE for implicit measurement equations involves two steps. First, the required values are computed using an ESKF with an implicit observation model. Subsequently, the VCE is performed, incorporating the prior stochastic information.

The following formulas use results of the algorithm for the ESKF from (3.25) to (3.36). First, the transformed residual vector $\hat{\boldsymbol{v}}_t^{(k)}$ for the current epoch k is computed by

$$\hat{\boldsymbol{v}}_t^{(k)} = \left(\mathbf{Q}_{\boldsymbol{rr}}^{(k)}\right)^{-1} \mathbf{B}^{(k)} \hat{\boldsymbol{v}}^{(k)}, \tag{4.1}$$

using the restriction matrix $\mathbf{B}$, the VCM of the misclosures $\mathbf{Q}_{\boldsymbol{rr}}$ and the residuals $\hat{\boldsymbol{v}}$ of the update step. Next, the vector $\hat{\boldsymbol{q}}$ and the matrix $\hat{\boldsymbol{S}}$ are each computed element-wise, with f for the elements of $\hat{\boldsymbol{q}}$, and r and c for rows and columns of $\hat{\boldsymbol{S}}$, both iterating over the individual observation groups

1 to z

$$\mathbf{Q}^{(k)}_{\boldsymbol{hh},f} = \mathbf{B}^{(k)} \left(\boldsymbol{\Delta}\hat{\boldsymbol{\sigma}}_f \, \boldsymbol{V}^{(k)}_{\mathrm{f}}\right) \left(\mathbf{B}^{(k)}\right)^T, \tag{4.2}$$

$$\hat{q}^{(k)}_f = \left(\hat{\boldsymbol{v}}^{(k)}_t\right)^T \mathbf{Q}^{(k)}_{\boldsymbol{hh},f} \hat{\boldsymbol{v}}^{(k)}_t, \tag{4.3}$$

$$\hat{s}^{(k)}_{(r,c)} = \operatorname{trace}\left(\left(\mathbf{Q}^{(k)}_{\boldsymbol{rr}}\right)^{-1} \mathbf{Q}^{(k)}_{\boldsymbol{hh},c} \left(\mathbf{Q}^{(k)}_{\boldsymbol{rr}}\right)^{-1} \mathbf{Q}^{(k)}_{\boldsymbol{hh},r}\right). \tag{4.4}$$

Based on these computed values, the change of the VCs $\boldsymbol{\Delta}\hat{\boldsymbol{\sigma}}$ can be computed for the current epoch k

$$\boldsymbol{\gamma}^{(k)} = \left(\hat{\boldsymbol{S}}^{(k)}\right)^{-1} \hat{\boldsymbol{q}}^{(k)}. \tag{4.5}$$

This approach is analogous to VCE in batch adjustments. Especially in recursive estimations, the normal matrix $\hat{\boldsymbol{S}}$ can become singular and thus not invertible. Alternatively, the estimated VCs may turn negative, resulting in unusable outcomes. To mitigate this, the least squares VCE method from Amiri-Simkooei (2007) is employed to incorporate prior information into the estimation process. This is achieved by treating VCE as a least squares estimation task. Consequently, the second state vector $\hat{\boldsymbol{x}}_s$ with the VCM $\boldsymbol{Q}_{\hat{\boldsymbol{x}}_s\hat{\boldsymbol{x}}_s}$ is introduced to track the VCs. The initial state vector $\hat{\boldsymbol{x}}^{(0)}_s$ is set to $\mathbf{1}$ because the initial stochastic model is used at the beginning. The initial VCM $\boldsymbol{Q}^{(0)}_{\hat{\boldsymbol{x}}_s\hat{\boldsymbol{x}}_s}$ can be configured in various ways as a design parameter. A practical approach involves using a diagonal matrix with a chosen uncertainty for the VC. As the estimation progresses, mathematical correlations between the VC occur. The computation of the filter update is

$$\boldsymbol{Q}^{(k)}_{\hat{\boldsymbol{x}}_s\hat{\boldsymbol{x}}_s} = \left(\hat{\boldsymbol{S}}^{(k)} + \left(\boldsymbol{Q}^{(k-1)}_{\hat{\boldsymbol{x}}_s\hat{\boldsymbol{x}}_s}\right)^{-1}\right)^{-1}, \tag{4.6}$$

$$\hat{\boldsymbol{x}}^{(k)}_s = \boldsymbol{Q}^{(k)}_{\hat{\boldsymbol{x}}_s\hat{\boldsymbol{x}}_s} \left(\hat{\boldsymbol{q}}^{(k)} + \left(\boldsymbol{Q}^{(k-1)}_{\hat{\boldsymbol{x}}_s\hat{\boldsymbol{x}}_s}\right)^{-1} \hat{\boldsymbol{x}}^{(k-1)}_s\right). \tag{4.7}$$

As mentioned before, the VCs are assumed constant, so neither a prediction nor process noise is applied.

4.2 Dual Filtering for Position Tracking

This section addresses the application of the developed methodology. The principle of recursive VCE is broadly applicable to a variety of tasks, particularly when dealing with a large volume of observations. Typical applications include the estimation of Earth orientation parameters (Heiker, 2013), of gravity fields (Koch & Kusche, 2002; Alkhatib, 2008), of calibration parameters (Vogel et al., 2022), and surface displacement monitoring (Hu et al., 2012). In this thesis, the focus is on enhancing the georeferencing of kinematic LiDAR-based MSSs. The LUCOOP dataset from Axmann et al. (2023) is used for the investigations. This dataset offers diverse sensor data of kinematic systems in an urban setting, including LiDAR data, which is used here to explore the impact of uncertainty modeling within a ESKF framework. In this context, the availability of a TLS point cloud of the entire area of the experiment is of special importance.

This subsection is divided into three parts. It begins with a detailed presentation of the extension to the ESKF introduced in publication **B** by estimating uncertainties and corrections for the distance measurements during the position tracking. This includes modeling various influences using the dual filter and explaining the creation of the localization map. This map is used for both following parts. A simulation of the dataset is conducted to demonstrate the methodology with known values. Finally, the actual dataset is processed, and the results are examined under real-world conditions with a special focus on the data used for evaluation.

4.2.1 Application of Dual Filtering for Georeferencing Using LiDAR

The dual filter's primary objective is to address the challenges in georeferencing outlined in Chapter 3. The uncertainty modeling of the LiDAR is restricted due to the temporal stability of the sensors used and the material-dependent uncertainty of observations. Materials may be different due to color or surface structure, e.g., differences between plastered walls and bricks. This results in limited transferability of calibration results. Different materials can cause varying levels of random noise in the estimated VCs (Wujanz et al., 2017), as well as different systematic deviations. To address the random noise, VCE is applied to adjust the stochastic model of distance measurements effectively. Plane groups are introduced to assemble surfaces with similar properties for improved estimation. Unlike robust adjustment, where each observation is individually reweighted, this approach facilitates the transfer of weighting information among observations belonging to the same group. Besides the aforementioned dual filter, distance corrections are also estimated for two key reasons. First, distance offsets can vary between calibration and georeferencing, or even during georeferencing itself, due to temporal sensor instability. Second, different materials and surface structures introduce varying systematic uncertainties, similar to random uncertainties. Online estimation of these corrections is therefore essential. This approach requires incorporating the extrinsic calibration parameters between the LiDAR and IMU into the state vector. Unlike the Cartesian coordinate approach in publication **B**, observations are represented as spherical coordinates in the observation vector. By including extrinsic calibration parameters, this method enables online calibration updates, addressing issues encountered in publication **D**.

In this section, the modifications made to the filter formulation in publication **B** are outlined and discussed. The observation vector is changed to include the spherical observations

$$\boldsymbol{\ell}^{(k)} = \begin{bmatrix} d_1 & e_1 & \alpha_1 & d_2 & \dots & d_n & e_n & \alpha_n \end{bmatrix}^T . \tag{4.8}$$

This leads to the VCM of the observations

$$\boldsymbol{\Sigma}_{\boldsymbol{\ell\ell}}^{(k)} = \operatorname{diag}\begin{pmatrix} \gamma_j\sigma_d^2 & \gamma_e\sigma_e^2 & \gamma_\alpha\sigma_\alpha^2 & \gamma_j\sigma_d^2 & \dots & \gamma_j\sigma_d^2 & \gamma_e\sigma_e^2 & \gamma_\alpha\sigma_\alpha^2 \end{pmatrix} \tag{4.9}$$

with the VCs γ and the plane j. Similar to the approach in Section 3.1, the observations are assumed to be uncorrelated due to the setup of the LiDAR. Modeling of the correlation of subsequent observations would be useful, but computationally intractable considering the number of observations. The initial variances σ_d^2, σ_e^2, and σ_α^2 are derived from the VCE during calibration. The second state vector consists of the VCs for both the angle measurements and the distance measurements, categorized per plane group

$$\boldsymbol{x}_s = \begin{bmatrix} \gamma_e & \gamma_\alpha & \gamma_1 & \dots & \gamma_J \end{bmatrix}^T \tag{4.10}$$

for a total of $[2 + J]$ states. During operation, the state vector gets extended with each new plane group observed. The plane groups are provided as additional information in the localization map. The angle uncertainties are grouped over all planes, as their uncertainty is determined by the sensor calibration for the elevation and the encoder accuracy for the azimuth, both of which are unrelated to specific surfaces. The VCM of the second state vector is initialized as a diagonal matrix, and additional plane groups are also initialized uncorrelated to the other VCs.

Two additional steps are required for the computation of the misclosures in the update step. First, the spherical coordinates for each point i belonging to a plane group j have to be transformed in

Cartesian coordinates with

$$\mathbf{p}_{\,\mathrm{s},i} = \begin{bmatrix} x_{\,\mathrm{s}} \\ y_{\,\mathrm{s}} \\ z_{\,\mathrm{s}} \end{bmatrix}_i = \begin{bmatrix} (d_i + d_{0,j}) \cdot \cos e_i \cdot \cos \alpha_i \\ (d_i + d_{0,j}) \cdot \cos e_i \cdot \sin \alpha_i \\ (d_i + d_{0,j}) \cdot \sin e_i \end{bmatrix}. \tag{4.11}$$

Compared to equation (3.1) in publication **A**, the distance offset is now dependent on the specific plane to which the point is assigned. The transformation from the sensor frame of the LiDARs to the platform frame is performed using

$$\mathbf{p}_{\,\mathrm{l},i} = \left(\mathbf{t}_{\,\mathrm{s}}^{\,\mathrm{b}}\right)^{(k)} + \left(\mathbf{R}_{\,\mathrm{s}}^{\,\mathrm{b}}\right)^{(k)} \mathbf{p}_{\,\mathrm{s},i}. \tag{4.12}$$

The parameters of the pose are marked for the current epoch, given that the transformation parameters are part of the state vector. The remaining computation of the misclosure follows the procedure listed in equations (3.46) and (3.47). The full state vector is set up as follows

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{t}_{\mathrm{b}}^{\mathrm{l}\,T} & \boldsymbol{v}_{\mathrm{b}}^{\mathrm{l}\,T} & \mathbf{q}_{\mathrm{b}}^{\mathrm{l}\,T} & \boldsymbol{a}_b^T & \boldsymbol{\omega}_b^T & \boldsymbol{\theta}_{\,\mathrm{s}}^{\,\mathrm{b}T} & d_{0,1} & \cdots & d_{0,J} \end{bmatrix}^T, \tag{4.13}$$

totaling a $[22 + J]$ states. The distance offsets $d_{0,j}$ for each plane group $j \in (1, J)$ are treated like the VCs in the second state vector. They are initialized using the distance offset determined during calibration and are subsequently updated with each epoch. New distance offsets are initialized as uncorrelated during operation. For the transformation parameters between LiDAR and the platform, Euler angles are used, following the calibration convention. Since the pose is fixed and only minor changes are anticipated, the risk of Gimbal lock is negligible. The entire algorithm, along with all equations, is detailed in the appendix in Chapter 8.

Subsampling

To enhance the computational efficiency of the filter, the point clouds are subsampled, which is a common practice for many LiDAR data applications (Liu & Zhang, 2021; Mounier et al., 2023; Qin et al., 2020). This approach addresses the challenge of processing large amounts of data in real-time, as the observations are generally assumed to be correlated (Jost, 2023). In this study, voxel-based subsampling is utilized. This method is favored due to its capacity to retain points from the entire scanned volume while balancing their distribution across the FoV of the sensor. Unlike other applications (Hillemann et al., 2019; Vizzo et al., 2023), this study does not retain the centroid of all points in a voxel or the point closest to the center. Instead, one or more random points within the voxel are selected, allowing for a variable rate of subsampled points, determined either by a ratio of the assigned points or a fixed number of observations. The process involves associating all assigned LiDAR points to their respective voxels, then randomizing the order of the voxels. Points are subsequently picked from the voxels in this random sequence until the required number of points is obtained.

Generation of the Localization Map

The localization map aims to provide an absolute reference for the position tracking. A variety of data sources can be utilized for this purpose. However, due to the issues previously discussed concerning LOD2 city models, it is necessary to seek an alternative data source. In this research, a TLS point cloud of the environment is used to create the localization map. TLSs are capable of capturing environmental details with millimeter-level accuracy, encompassing even small features. Through the process of registration, large areas can be documented with accuracy levels of 1 cm or better (Janßen, 2024). Given the comparatively small area of the dataset, referencing is carried out using reduced Universal Transverse Mercator (UTM) coordinates with a scale factor of 1. The coordinates are reduced to enable efficient computation using float accuracy. Considering the precision of the LiDARs deployed, the potential deviations caused by non-parallel facades or the Earth's curvature are negligible. Both factors introduce errors of less than a millimeter over a distance of 100 meters, which is the maximum range for the LiDAR. The height system is determined by the system in the localization map. In this case, the heights are in the Deutsches Haupthöhennetz (DHHN)2016 system.

The functional model of the filter necessitates plane parameters that minimize the distances to the acquired LiDAR point cloud. For representing arbitrary geometries, a triangulated mesh provides substantial flexibility regarding the geometries represented. The localization map is generated following these steps. Initially, the TLS point cloud undergoes semantic segmentation to exclude non-static points. In this process, only points associated with facades or the ground are retained for further processing. The point cloud is then voxelized using a voxel size of 25 cm to provide a regular structure for further processing. Within each voxel, planarity of the points is assessed through principal component analysis (PCA) (Drixler, 1993; Hillemann et al., 2019). Planarity is essential because the functional model requires the calculation of distances to a plane. Areas with numerous scatter points pose a risk of incorrect assignment and are therefore unsuitable for localization. If the points in a voxel demonstrate planarity with a standard deviation of less than 1 cm, the plane parameters are calculated, and the voxel's centroid is added to a KD-tree (Bentley, 1975). The choice of a 1 cm threshold is motivated by the structure of the environment and the requirements on the positioning. Lower thresholds produce fewer map elements and thus a sparser map, as the ground and facades are either slightly curved or rough. A higher threshold is detrimental to the positioning accuracy as higher deviations might affect the state estimates. This threshold excludes facade elements that are not appropriate for localization, such as windows, doors, or balconies. There are possibilities for other observation models, e.g., cylindrical objects. The only requirement is the ability to compute a distance and to partially differentiate. The KD-tree serves as an index structure to ensure quick access to the map. Each voxel is restricted to containing only one plane. This approach balances map complexity, which impacts access times during assignment, with the level of detail the map can provide. The voxel size is a crucial parameter in this trade-off. Smaller voxels can capture more detailed structures but increase the map's data volume. The chosen voxel size is a compromise based on the system's target accuracy, where smaller voxels could represent finer structures, potentially leading to more misassignments and degraded localization results. Similar sizes are found in the literature, either based on experiments (Mounier et al., 2025), as a percentage of the measurement range (Vizzo et al., 2023), or adapted by dividing voxels, which arrive at a similar scale (Liu & Zhang, 2021).

Additionally, it is necessary to establish a grouping for the planes within the map. The filter operates on the assumption that similar surfaces present comparable uncertainties in LiDAR measurements. The assumptions for the utilized localization map are outlined as follows. The ground is categorized as a single group. In urban settings, most of the ground is composed of asphalt roads. Some measurements may include the paving next to the roads, although parked cars along these roads tend to obstruct many observations. Facades are grouped by building. Despite the fact that some buildings may feature facades with diverse materials or colors, this level of simplification is sufficient to provide a basic grouping of the environment, suitable for validating the developed algorithms. Information regarding building outlines is sourced from Amtliche Liegenschaftskatasterinformationssystem (ALKIS), which is publicly accessible cadastral information that includes property boundaries and building outlines. The accuracy of ALKIS data is about 2 cm, leading to the same accuracy for the building outlines used for the grouping.

Figure 4.2 provides an overview of the trajectory, displaying building outlines. Compared to the original dataset, the initial part of the trajectory is cropped to provide a stationary period at the beginning. Random colors are assigned to buildings to clearly distinguish between them. The numbers indicate specific times along the trajectory.

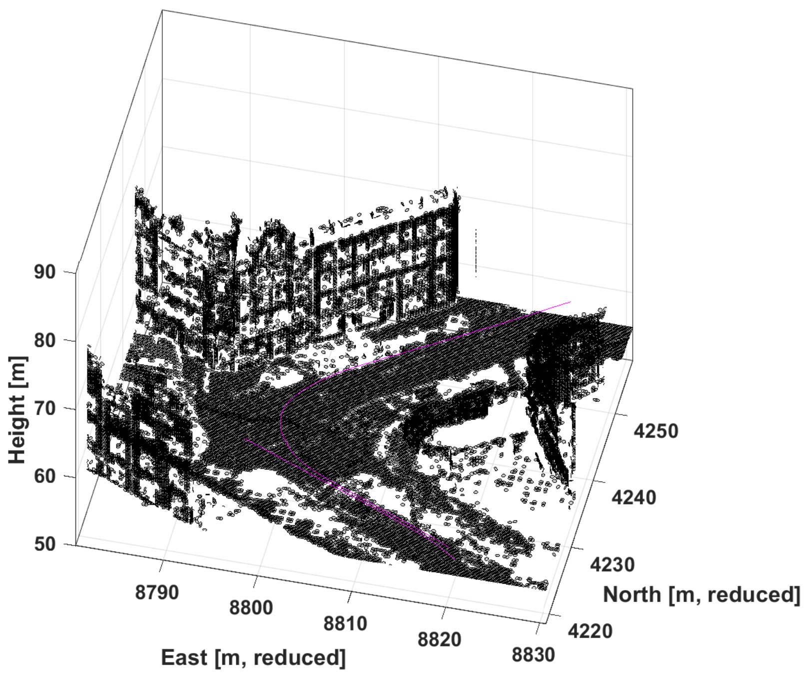

Figure 4.1: *Oblique view of generated map from TLS point cloud with trajectory in purple.*

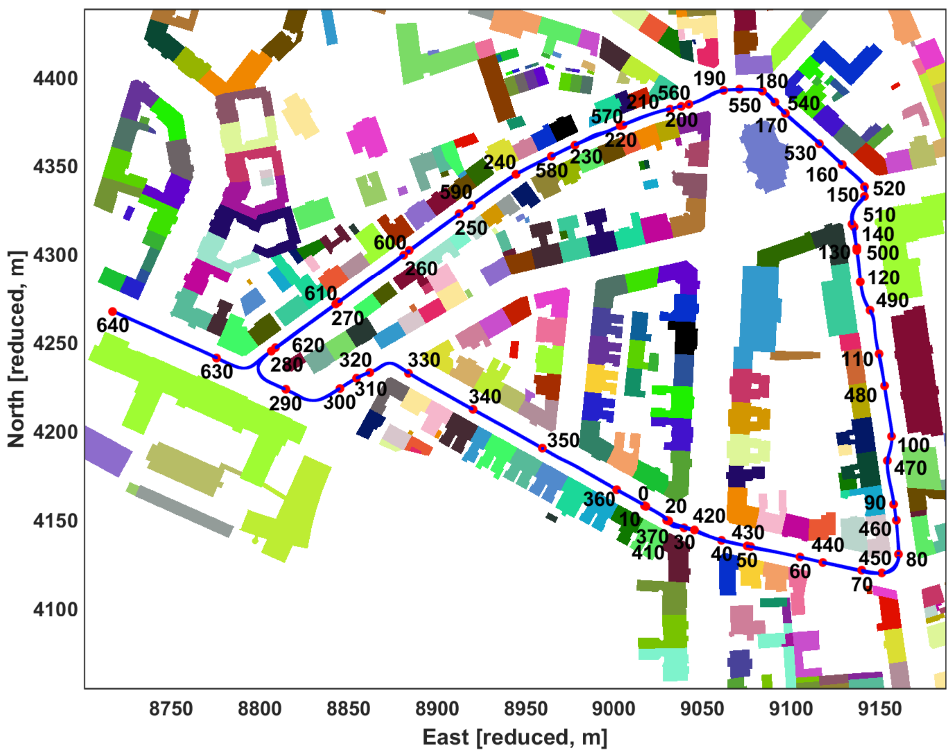

Figure 4.2: *Overview of environment with trajectory (numbers stand for times in s) and randomly colored building outlines from ALKIS.*

Overview of Entire Algorithm

In summary, Figure 4.3 illustrates the developed algorithm. The full formulas can be found in Chapter 8. The necessary inputs include sensor observations (from the IMU and LiDAR), the initial state vector for the system and sensor uncertainties, and the localization map. The IMU observations are utilized to propagate the system's pose to the next epoch, while the other elements of the state vector are assumed to remain constant. Since IMU biases are subject to bias instability, process noise is applied. The magnitude of the noise can be seen in Table 4.1. For the map matching step, the predicted state is then used to transform the LiDAR points, which were collected since the last epoch, to the superordinate frame in which the map is available. These points are subsequently assigned to the map and may optionally be subsampled. The assigned points are employed to compute the update for the state vector. If recursive VCE is applied, the residuals and other intermediate values from the update step are used to compute the VC. The VCE includes a second state vector, which tracks the VCs across epochs, to stabilize the estimation process and appropriately weight the information from different epochs. The estimated VCs are incorporated into the stochastic model of the update, thereby weighting the LiDAR observations. The outcome of the filter is an estimated trajectory, which can be used to generate the georeferenced point cloud.

4.2.2 Simulation Studies

The simulated data is generated using the framework described in Subsection 2.3. The trajectory used for data generation is the MMS trajectory provided in the dataset (Axmann et al., 2023). The point cloud is produced in combination with the previously mentioned localization map. Each generated point is annotated with the ID of the corresponding plane, allowing for the consideration of random noise and a distance offset specific to that plane. Only the first round of the trajectory shown in Figure 4.2 is simulated, as these initial 365 seconds encompass the entire environment once. The simulation is carried out as a MC simulation, allowing for multiple runs based on the original dataset with known uncertainties. In total, 100 runs are conducted to get statistically significant results. The LiDAR point clouds are subsampled during the processing, utilizing 10% of the assigned points for the update. To evaluate the effectiveness of the algorithms, the data is processed using nine different variants. These variants consist of three settings for the VCE and the estimation of the distance offset for the LiDAR. The first setting is *correct*, which uses the correct value from the generation of the data. This represents the best case, where all uncertainties are correctly known. The second setting is *wrong*, which employs incorrect values, initialized randomly within an interval around the true value. For real-world applications, this is often the default scenario. Typically, the approximate range of sensor uncertainties is known, but deviations due to drift, environmental factors, or varying materials are not. The last setting, *estimated*, begins with the *wrong* value and improves upon it through the estimation process previously described. The assumed uncertainties are based on actual sensor characteristics and are detailed in Table 4.1.

Results

As the main objective of the filter is position tracking, the results are evaluated by calculating the deviations of the pose from the true trajectory. A georeferenced point cloud can be produced based on this trajectory, if necessary. The deviations are computed using equations (3.49) through (3.51). These deviations are averaged across all 100 MC runs. The results are presented in several formats. Initially, the RMSEs for all nine variants are displayed. For this comparison, the values are averaged over the entire trajectory to provide an overview of the general performance. Furthermore, the RMSEs of the estimated distance offsets and standard deviations (calculated using the estimated VCs) are presented. Following this, selected variants are compared to gain insights into the benefits and potential challenges of the algorithms. Finally, the estimation of the extrinsic calibration parameters and the observability of the states are examined.

Table 4.2 presents the RMSEs calculated over all epochs of all MC runs, expressed in millimeters and milliradians, respectively. The colors for numbers in the text correspond to the colors in the

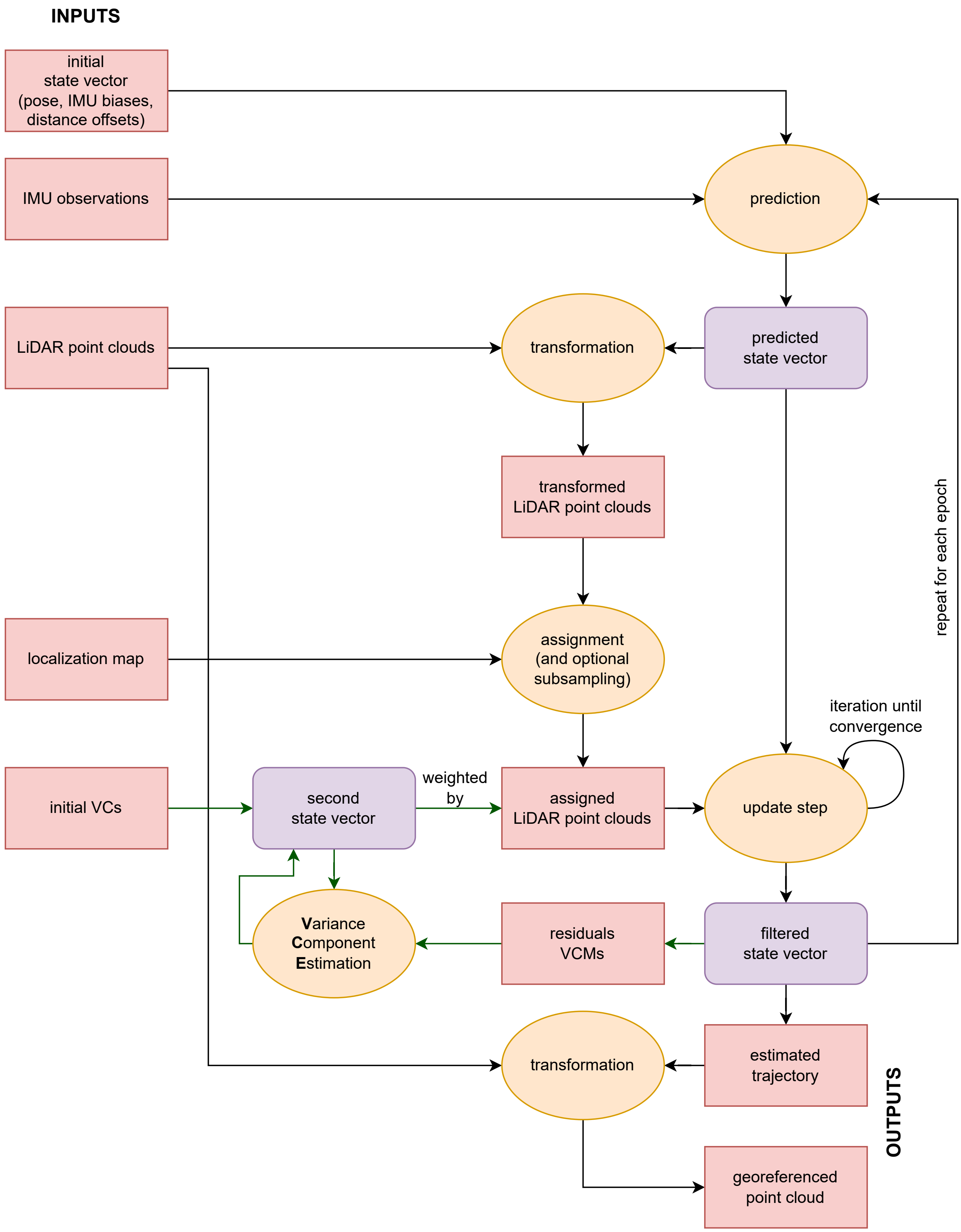

Figure 4.3: *Flowchart of the position tracking algorithm. Data in red, processes in yellow, and state vectors in purple. Steps marked by green arrows are part of the second filter.*

table. The table encompasses the results for all 100 MC runs over all 9 variants for the combinations of the settings for the stochastic and functional model. The rows represent the settings for the stochastic model in the following order: *correct*, *estimated*, and *wrong*. The columns follow the

Table 4.1: *Sensor uncertainties used in the simulation. Random and systematic uncertainties for distance measurements of LiDAR as intervals, as they are randomly sampled per plane.*

LiDAR		
distance measurement σ_d	[5 – 30]	mm
horizontal angle σ_α	0.85	mrad
vertical angle σ_e	0.52	mdeg
distance offset d_0	[-20 – +20]	mm
IMU for (x,y,z)-axes		
acc. noise density $\sigma_{\tilde{a}_n}$	(6.16, 4.61, 3.62) $\cdot 10^{-3}$	$^{\mathrm{m}\sqrt{\mathrm{Hz}}}/_{\mathrm{s}^2}$
acc. bias instability σ_{a_w}	(22.26, 20.35, 36.36) $\cdot 10^{-3}$	$^{\mathrm{m}}/_{\mathrm{s}^2}$
gyro. noise density $\sigma_{\tilde{\omega}_n}$	(0.26, 0.34, 0.24) $\cdot 10^{-3}$	$^{\mathrm{rad}\sqrt{\mathrm{Hz}}}/_{\mathrm{s}}$
gyro. bias instability σ_{ω_w}	(1.23, 1.14, 1.19) $\cdot 10^{-3}$	$^{\mathrm{rad}}/_{\mathrm{s}}$

same format for the distance offset. According to this layout, it is anticipated that the smallest RMSEs will appear in the upper left, as the correct values are used for the estimation, which should result in the lowest deviations compared to the true values in the simulation. The remaining results should show larger deviations, as wrong assumptions for the functional and stochastic model should be detrimental to the filter performance. However, upon examining the results, this assumption does not hold true.

Across all results, translations and orientation behave similarly, even though the differences between the orientations are smaller. Here, the first row shows a large range of results with 2.00 mm for the translation using the *correct* settings compared to 3.92 mm for *wrong* variant. Utilizing the *correct* value of the distance offset results in better estimations of the pose deviations compared to the *estimated* variant, which in turn improves upon the *wrong* values. When employing a more accurate distance offset, either correct or estimated, translations are halved, and orientations improve by approximately 30% (comparing blue and red values). The effect of the distance measurements is greater on the translations. For further improvements of the orientations, corrections for the angle measurements could be beneficial. The estimation of the distance offsets leads to results insignificantly different from using the *correct* distance offset. This leads to the conclusion that for the circumstances in this work, the distance offset can be estimated well, leading to a near parity in the performance for the corresponding variants. For the stochastic model, the aforementioned assumption does not hold. The lowest RMSE values of 1.74 mm for the translation and 0.104 mrad for the orientations occur in the *estimated* setting for the stochastic model when using the correct values for the distance offsets. The impact of the stochastic model is smaller, with a maximum improvement of about 10% between the *wrong* and *estimated* variant. Interestingly, the variant using the *correct* stochastic model with a *wrong* distance offset (in red) yields the worst results, possibly due to overly optimistic weighting of biased distance measurements. To further explore the superior performance of the *estimated* stochastic model, the other results are examined first.

The next step involves examining the estimated distance offsets and VCs, presented as standard deviations. Table 4.3 displays the RMSE for all estimated uncertainties across all MC runs. Here, the settings are alternated. For the estimation of the distance offset, the setting for the stochastic model is varied, and the other way around. For the correct setting, the RMSE is always 0, and for the *wrong* setting, it is 15.76 mm for the distance offsets and 55.34 mm for the standard deviations. The RMSE is calculated solely for the final estimate.

The resulting RMSEs demonstrate that the estimation process effectively recovers the correct value for the distance offsets across all settings for the stochastic model. The *estimated* setting

Table 4.2: *RMSE of the translations and orientations of the processed simulated dataset across all epochs of all 100 MC runs. The rows correspond to the setting of the stochastic model and the columns to the corrections.*

		Distance Offset Setting		
Stochastic Model	**Component**	*correct*	*estimated*	*wrong*
correct	Translation [mm]	2.00	2.01	3.92
	Orientation [mrad]	0.107	0.107	0.134
estimated	Translation [mm]	1.74	1.75	3.00
	Orientation [mrad]	0.104	0.104	0.120
wrong	Translation [mm]	2.84	2.85	3.43
	Orientation [mrad]	0.112	0.112	0.120

yields the best performance, similar to the trajectory deviations. Using a *wrong* stochastic model results in only a slightly worse accuracy. The maximum deviation across all runs when estimating the distance offset is approximately 3 mm. The estimated standard deviations present a different behavior, as the values are only corrected by up to 40%. The settings for the distance offset show a significant impact, with the *wrong* setting yielding the best results. The remaining high deviation is evident in the maximum remaining deviation, which is approximately 23 mm. This is nearly equivalent to the entire range of the initialized measurement noise.

Table 4.3: *RMSE of estimated distance offsets and standard deviations over all MC runs. Values are given in mm. Initial RMSE listed per row.*

	Setting		
Estimated Parameter	*correct*	*estimated*	*wrong*
Distance offset (init. RMSE: 15.26 mm)	0.26	0.22	0.31
Standard Deviations (init. RMSE: 55.15 mm)	42.16	42.72	32.48

These results highlight challenges in estimating the stochastic model, although there is an observed improvement in trajectory estimation seen in Table 4.2. To further explore this issue, the average RMSEs for three variants is visualized over time. The selected settings are *wrong/wrong*, *estimated/estimated*, and *correct/correct*, chosen to identify differences in results along the trajectory. Figure 4.4 illustrates the RMSEs for these selected runs as they progress over time. The overall behavior of all three time series is similar, with the *wrong* setting performing notably worse. The line showing the result for *estimated* is covered by *correct* for the majority of the time. The almost constant deviations exhibit peaks at times corresponding to turns, e.g., around 100 s, 120 s, 280 s, and 325 s. Even with all effects accounted for in the simulation, deviations still increase in these areas. This effect is visible in both the translations and orientations, each varying in magnitude depending on the specific turn. These phenomena are likely influenced by the dynamics of the trajectory, which is derived directly from real data. The underlying issue seems to be an incorrect weighting between IMU and LiDAR observations. Any jerks or abrupt changes in the trajectory might lead to greater deviations in the prediction, which are not adequately modeled by higher uncertainties. Consequently, the update introduces the predicted state with a comparatively high weighting for the actual deviations, resulting in a biased estimate.

A closer examination of these turns explains why the *estimated* stochastic model performs better. In these scenarios, the *estimated* variants exhibit lower deviations as a result of overfitting. The VCE compensates for the overconfidence in the prediction during turns. This compensation leads to an overfit to the trajectory, while yielding worse results compared to the true values of the

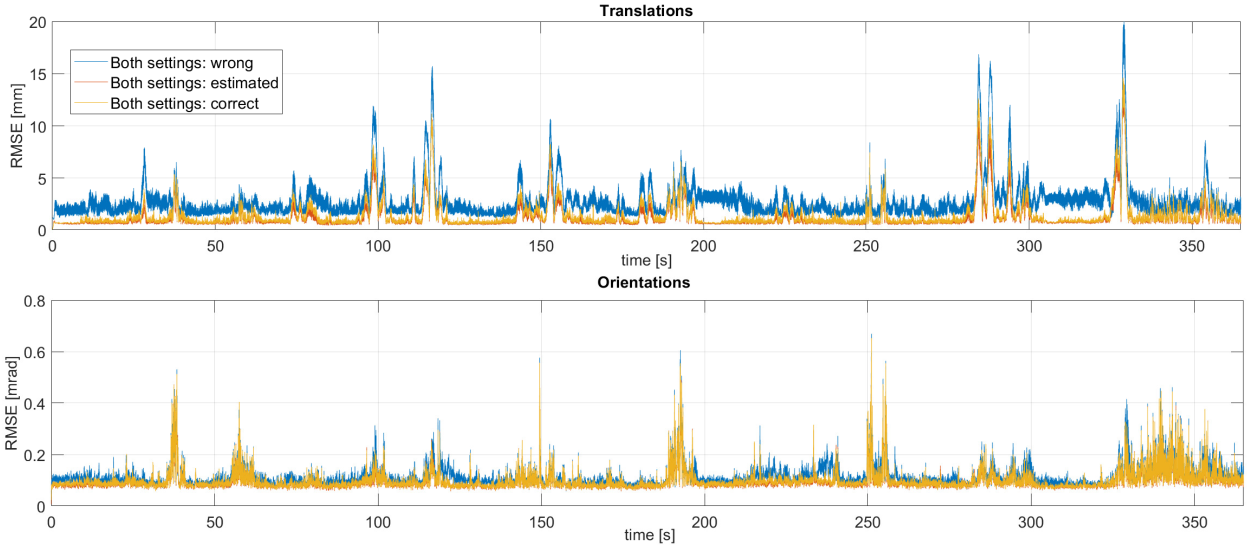

Figure 4.4: *Comparison of epoch-wise RMSEs (translation and orientation) for variants wrong, estimated, and correct using the simulated dataset.*

stochastic model. Across all plane groups, the standard deviation is estimated to be, on average, 8 mm under the correct value. This observation is further supported by consistency measures. The state estimation result is deemed consistent if the Mahalanobis distance falls within the interval of χ^2-distribution with the degrees of freedom $6 \cdot 100$ for the significance levels 2.5% and 97.5% (two-sided test). Figure 4.5 depicts the results for the three selected variants with green lines denoting the χ^2-interval. Once more, the results for the *correct* and *estimated* settings are very similar. Both variants remain within the consistency interval most of the time, except during the mentioned turns. The *wrong* settings exceed the interval of the hypothesis test along the entire time series. The similar structure of deviations and Mahalanobis distance suggests a nearly constant uncertainty estimate by the filter, which does not properly represent the times with higher deviations.

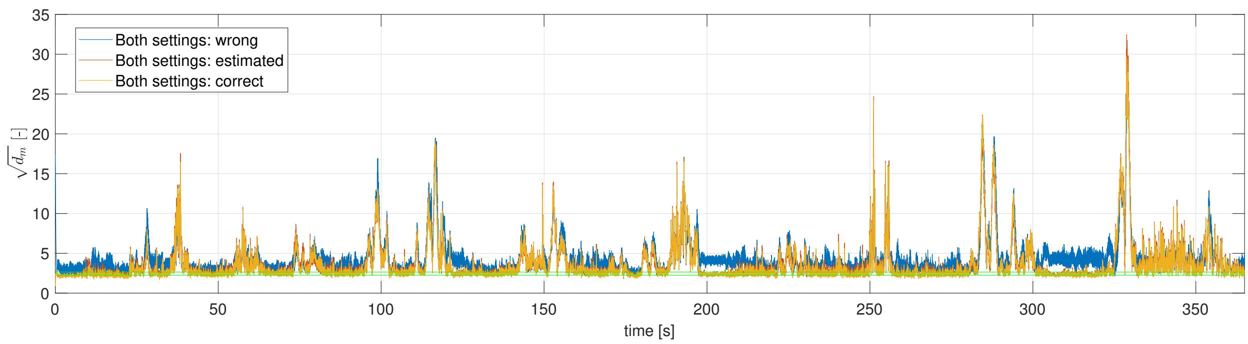

Figure 4.5: *Comparison of epoch-wise Mahalanobis distance for variants wrong, estimated, and correct using the simulated dataset.*

Online Estimation of Extrinsic Calibration

So far, the investigation has primarily concentrated on intrinsic sensor calibration. However, the extrinsic calibration parameters are also a component of the state vector and are estimated during operation. For the initial simulations, extrinsic calibration was set to the correct values and exhibited no significant changes throughout estimation, due to low uncertainty at the start. Here, the initial values are intentionally altered and initialized with higher uncertainties to evaluate the filter's ability to recover the correct values.

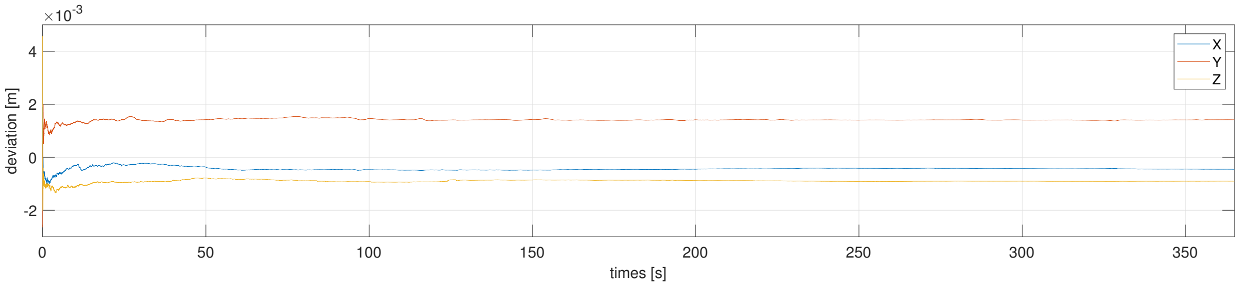

Figure 4.6: *Epoch-wise deviations of the translations of the extrinsic calibration parameters in the b-frame.*

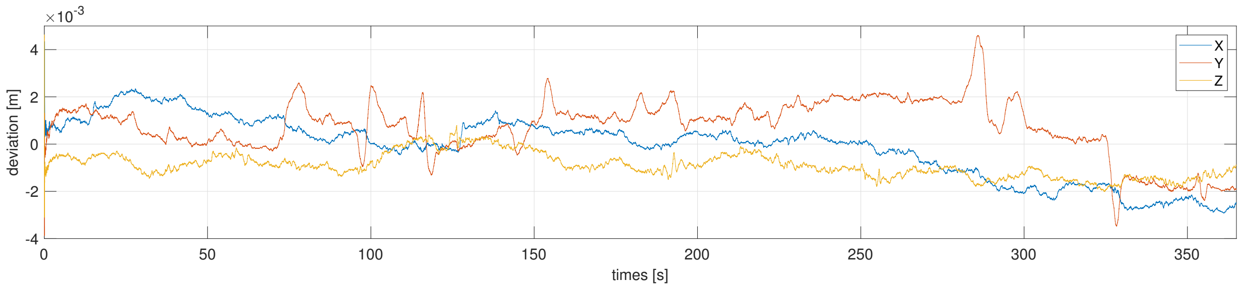

Figure 4.7: *Epoch-wise deviations of the translations of the extrinsic calibration parameters in the b-frame. Additional process noise is added in each prediction.*

Across various runs, the correct extrinsic calibration cannot be recovered due to an inappropriate configuration for estimation. Although the filter does not diverge, the deviations in trajectory estimation exceed previous results. Figure 4.6 displays an exemplary run that visualizes the translations. While the orientations are omitted here, they show a similar behavior. After the first 50 seconds, the extrinsic calibration parameters stabilize and exhibit only minor changes. The estimation leads to no clear improvements. Alternatively, introducing process noise can leverage other parts of the trajectory to influence the estimation. Figure 4.7 shows the results for the same initialization as before, with 10^{-10} mm added process noise per epoch. Introducing process noise to these states results in changing deviations of the calibration parameters along the entire trajectory. The x-component, in particular, appears sensitive to turns, overfitting to the constellation between system and map. With larger process noise, this effect increases until the filter diverges, whereas with smaller values, the results more closely resemble those shown in Figure 4.6. To effectively estimate the extrinsic calibration parameters, a specific movement pattern would be necessary to accurately determine these parameters. Alternatively, a batch adjustment over the entire trajectory (or selected parts (Lv et al., 2022)) can be performed.

Observability

For further investigation of the issue, the observability of the state is regarded, as it is a critical topic in filtering (Bar-Shalom et al., 2004; Simon, 2006). It defines whether the state can be estimated based on the observations and dynamics of the system. The observability is defined as

$$\boldsymbol{\mathcal{O}} = \begin{bmatrix} \mathbf{A}^{(1)} \\ \mathbf{A}^{(2)}\mathbf{F}^{(2)} \\ \mathbf{A}^{(3)}\left(\mathbf{F}^{(3)}\right)^2 \\ \vdots \\ \mathbf{A}^{(K)}\left(\mathbf{F}^{(K)}\right)^{K-1} \end{bmatrix} \qquad \text{with} \qquad \operatorname{rank}\left(\boldsymbol{\mathcal{O}}\right) = u. \tag{4.14}$$

For $K = u$, a linear system is observable. In the case of error states, as the one used here, the maximum observability corresponds to the dimension of the error state. By setting K to span a specific time interval, the system's observability for that interval can be evaluated. However, for the filter used in this study, definitive conclusions are challenging, as observations are highly dependent on the environment and the map. Depending on the orientation of the planes in the environment, a situation can occur in which no information along one direction is available, and the state is not fully observable. The choice of 25 cm voxels for the map generation prevents this by introducing smaller areas, e.g., inside windows/doors, to provide information along the urban canyons. An additional factor is the estimated distance offsets, which are part of the state vector but not continuously observable, as facades are not always visible. Therefore, the observability analysis should be conducted based on the number of actually observed plane groups.

The observability analysis for a 1-second time interval is illustrated in Figure 4.8. A 1-second interval is selected, as even low-cost IMUs are generally stable over this time frame. The figure shows the ratio of the computed observability divided by the number of states relevant in the time interval, which are always the pose and all observed plane groups. The analysis demonstrates that the state is observable with a value of 1.0 along the entire trajectory, indicating that the structure of the environment is diverse enough to fully observe the state in any 1-second-interval along the trajectory. This suggests that the challenges in estimating the extrinsic calibration parameters arise from issues related to weighting and specific movements, rather than a lack of general observability. Important to note is that the environment has a major impact on the observability for the chosen measurement model. The urban environment of the LUCOOP data set provides map information to multiple sides. In areas with fewer static features, the estimation of the distance offset could lead to problems, as the separation of the position and the distance offset could become ambiguous.

Findings and Implications from Simulations

Upon completing the simulations, several conclusions can be drawn. The novel dual filter approach enables the estimation of the distance offsets and standard deviations for the distance measurements during operation. Both of these contributions positively affect the trajectory estimation. The new principle based on plane groups tracks information, as opposed to robust estimations, which work on individual observations without transfer to later epochs. The developed recursive VCE allows for updating the stochastic model during the filtering of the sensor data with an implicit measurement model. Regarding research question **2**, which focuses on the consistent prediction of point uncertainties, the use of VCE demonstrates an improvement of the stochastic model. The improved trajectory estimation positively affects point uncertainties. In the performed simulations, the estimation of the stochastic model leads to an overfit to the dynamics of the trajectory. While this reduces the deviations of the estimated trajectory, the estimated point uncertainty gets biased. For research question **3**, the material-based estimation of distance offsets shows a clear improvement in the trajectory estimation. The distance offsets are accurately recovered across all variants. The estimation of extrinsic calibration parameters proves ineffective due to the minimal configuration of the system, involving only two sensors. Addressing the deviations in transformation parameters

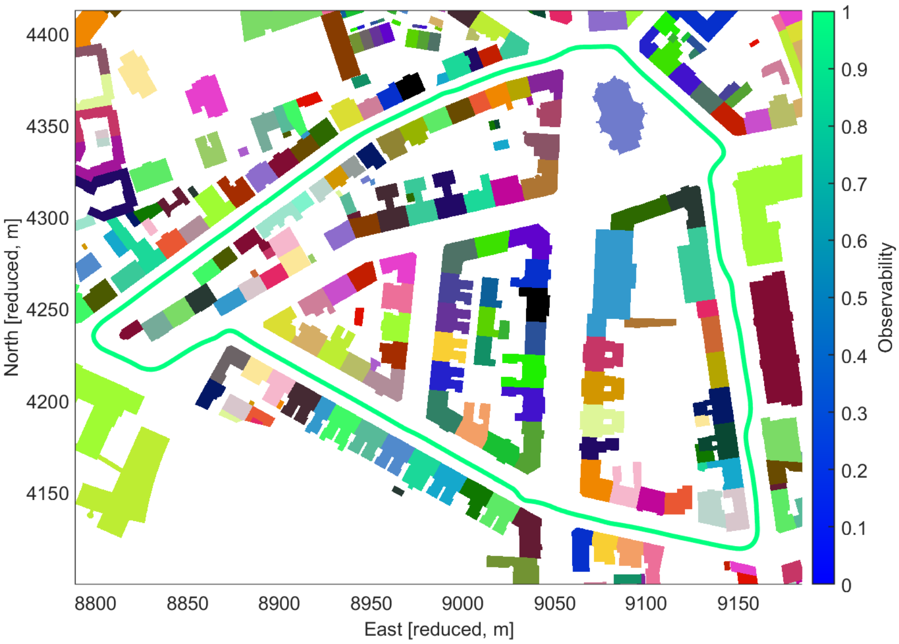

Figure 4.8: *Overview of observability of the state with distance offsets along the trajectory. Colors indicate observability.*

would necessitate either additional sensors, a selection of suitable parts of the trajectory (Qin et al., 2022), or a specially designed motion that excites all pose components at the start of the run (Mishra et al., 2021), which cannot be ensured in typical traffic scenarios.

4.2.3 Real Data Processing

After examining the developed methodology on the simulated dataset, the approach is now applied to the real dataset. The differences to the simulations are the use of the actual sensor observations instead of the simulated values, and the comparison to the reference trajectories used in the dataset. Compared to the controlled conditions of the simulation, the real dataset presents additional challenges. Sensor data deviations are influenced by factors beyond the assumed normally distributed noise. The environment includes elements not covered by the aforementioned localization map, such as other objects and traffic participants, leading to occlusions that can negatively impact localization. Scatter points necessitate an increase in the rate of retained LiDAR points from 10% in the simulation to 25% here. This increase is necessary due to scatter points in the environment in the real measurement. It leads to slightly longer processing times, as the computed matrices are larger. The map also assumes that all surfaces of a building are of the same material, which is not the case for all buildings. Additionally, the observations are affected by outliers, unmodeled correlations, and systematics, such as missing intrinsic calibration parameters.

The primary objective is to evaluate the effect of the estimation of the distance offset and stochastic model on the accuracy and consistency of the position tracking. For assessing the distance offsets and stochastic model themselves, only the plausibility of the results can be verified. For the trajectory, a comparison is made between fixed initial values for sensor uncertainties and their estimated values. This comparison is analogous to the cases *wrong/wrong* (referred to as *assumed* here) and *estimated/estimated* in the simulations, since true values for the poses of the trajectory or the LiDAR uncertainties are unavailable. Instead, the estimated poses of the trajectory are evaluated by comparing to reference data. Two different reference trajectories provided in the LUCOOP dataset and the TLS point cloud used for the map generation are used. Initially, the trajectory offered by the commercial MMS is used to calculate deviations. The MMS uses a navigation-grade Applanix IMU and thus generates a very smooth trajectory. The second reference trajectory is produced by using point clouds acquired by a LiDAR on the system in an ICP to register the system in the TLS point cloud. As only LiDAR data is used, the resulting trajectory is noisier compared to the MMS solution. Finally, the TLS point cloud is used directly by computing the cloud to cloud (C2C) distances between the georeferenced LiDAR point cloud to this reference point cloud. The TLS point cloud is produced with the highest accuracy compared to the trajectories, but the evaluation offers only a general insight into the quality without component-specific deviations. Another notable aspect is the datum definition. The MMS trajectory is produced in a tightly coupled solution by fusing IMU, GNSS and odometry observations. All other solutions (filter, ICP and TLS point cloud) depend on the datum definition of the TLS point cloud, which depends on ground control points (GCPs) distributed in the area.

Evaluation with MMS Trajectory

The result comparison begins with the RMSEs for translations and orientations, as depicted in Figure 4.9 and listed in Table 4.4. Both variants exhibit very similar deviations. The two variants yield practically indistinguishable translational RMSEs, with 43.20 mm for the *estimated* variant and 42.99 mm for the *assumed* variant. Different segments of the trajectory favor different variants concerning translation accuracy. In terms of orientation, the *estimated* variant performs better, achieving an RMSE of 1.16 mrad compared to 1.28 mrad for the *assumed* variant. Aside from a few peaks, such as those occurring around 190 s or 550 s, the *estimated* variant generally provides better results at most times. These are caused by the dynamics of the vehicle driving over a raised part of the street. Here, the estimated variant estimated lower uncertainties for the LiDAR observations, which yield higher deviations during this short period. The improvement over the remaining time is characterized by fewer variations in the deviations, e.g., for the translations at around 350 seconds into the trajectory, and lower deviations at 160 and 530 seconds (also for translations). Table 4.4 shows that for the orientations the 68% and 95% quantiles confirm the overall RMSE. For the translations, the quantiles are lower for the *estimated* variant, indicating reduced systematics.

Table 4.4: *Descriptive statistics of RMSEs of the translations and orientations of the processed real dataset using both settings.*

Component	Metric	*assumed*	*estimated*
Translation [mm]	overall	42.99	43.20
	68% quantile	48.07	45.72
	95% quantile	69.78	66.78
Orientation [mrad]	overall	1.28	1.16
	68% quantile	1.27	1.15
	95% quantile	2.27	2.07

The remarkable similarity over the entire trajectory prompts further investigation to assess whether the estimated uncertainties of the trajectory accurately reflect the deviations in either solution. Again, the Mahalanobis distance is computed for this purpose. Figure 4.10 shows the square root of this value across the entire trajectory for both variants. Notably, the maximum value of the χ^2-interval for the hypothesis test is exceeded in all epochs, by up to a factor of 100. The use of the square root of the Mahalanobis distance scales this factor to the actual mismatch between uncertainties and deviations. This means the deviations are a factor of 100 too large for the estimated uncertainties, largely due to systematic effects affecting the comparison, either in the

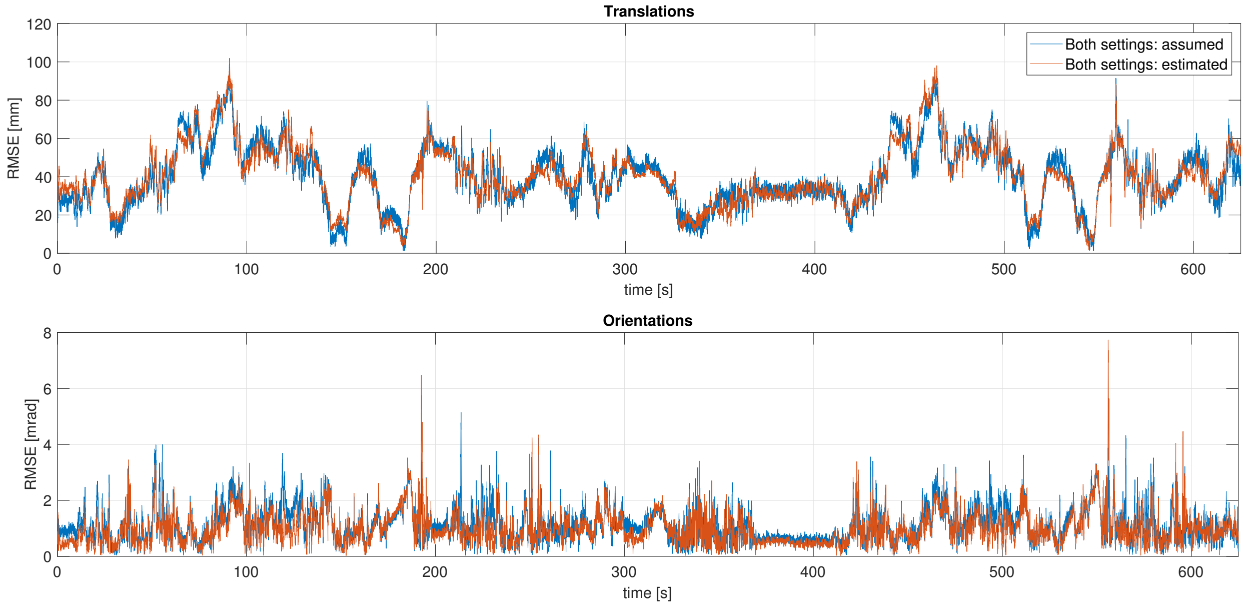

Figure 4.9: *Comparison of epoch-wise RMSE values for translation and orientation for the real dataset for variants assumed and estimated. Deviations are computed from the MMS trajectory.*

estimation or in the reference data. The estimation improves the values by up to 30%, depending on the environment and constellation. This can be attributed to the effects of the distance offset and assumptions about the stochastic model, which are corrected in the estimation. Again, a repeated structure of the values is observed. To investigate the cause, the individual deviations are shown next.

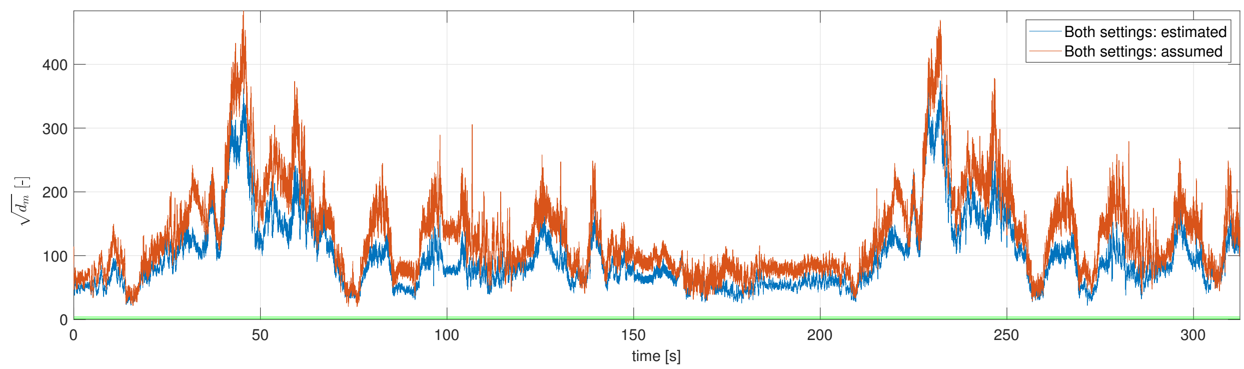

Figure 4.10: *Comparison of epoch-wise Mahalanobis distances of real dataset for estimated variant. Deviations are computed from the MMS trajectory. Green lines show χ^2-interval for the hypothesis test at [1.1, 3.8]. Values computed with the MMS trajectory.*

Finally, Figure 4.11 presents the individual pose deviations along all coordinate axes for the *estimated* run compared to the MMS trajectory. The deviations are transformed into the body frame, which follows the forward-left-up (FLU)-convention. This separation into coordinate components highlights the recurring pattern seen before. In the forward and up directions, deviations surpass 15 cm at various points along the trajectory. The left component shows slightly better performance, likely due to the environmental structure consistently providing information from the side. Even during stationary phases, deviations of up to 5 cm are observed, suggesting discrepancies between the data sources. As the deviations already suggest systematic effects as a dominating influence, standardized deviations, as shown in Figure 3.14, are omitted here. To evaluate the possible reasons for these deviations, the manufacturer information for the MMS is used. The manufacturer specifies the absolute accuracy at 20 - 50 mm (Riegl, 2011) without information about the provided measure (standard deviation or interval). Considering these uncertainties, the majority of the translational deviations can be explained based on the reference trajectory. As the filter will estimate the state variances similar to the simulated case, this also explains the exceeded χ^2 quantiles. For the orientations, the MMS is rated to an accuracy about one order of magnitude more accurate compared to the shown results. In this case, the majority of the rotational uncertainties stem from the filter solution. This is especially evident for the roll and pitch angles, for which the filter delivers unstable results.

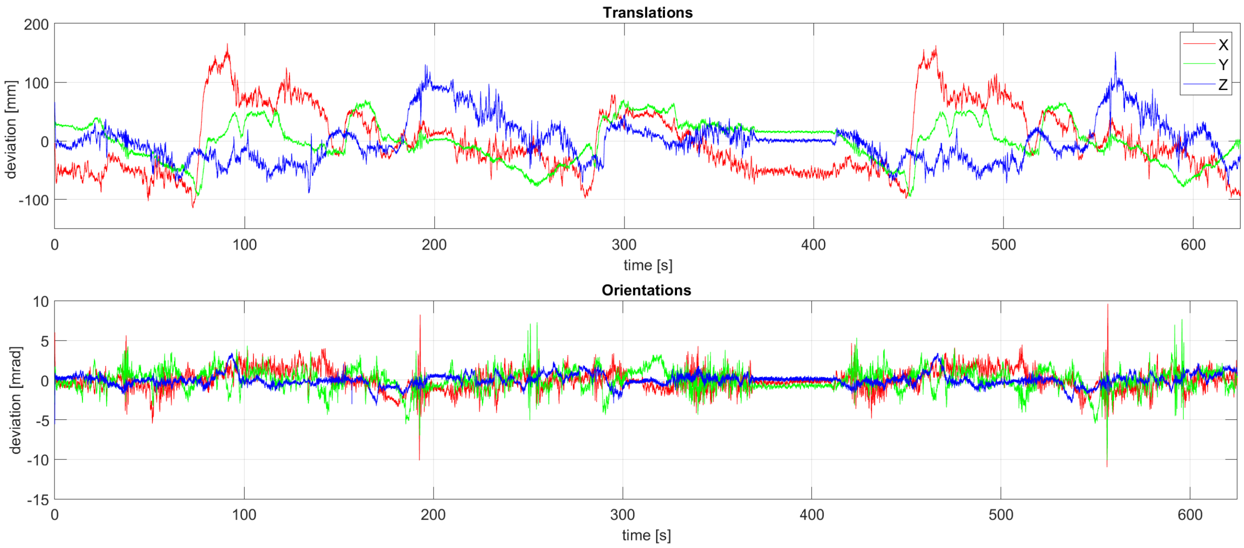

Figure 4.11: *Epoch-wise deviations of real dataset for estimated variant. Deviations are computed with respect to the MMS trajectory and rotated into the body frame of the system. Color for the axes: red for X (forward), green for Y (left), and blue for Z (up).*

Since true values for the estimated distance offsets and standard deviations are unavailable, their plausibility is evaluated. Figures 4.12 and 4.13 display the histograms for the estimated values. For the histogram of the distance offsets, a bin width of 2 mm is chosen, as this corresponds to the quantization of the used Velodyne Puck (Velodyne, 2019b). The majority of distance offsets fall between -2 cm and 0 cm, aligning with findings regarding the measurement accuracy of the Velodyne Puck from both own research and the literature (Glennie et al., 2016; Sánchez & Pany, 2019). The mean value of about -15 mm coincidentally matches very well the intrinsic calibration in publication **A**, even though different sensors of the same type are used. The most extreme value, at -3.1 cm, belongs to a facade that is barely visible in the scans due to occlusion by a smaller building in the foreground and only measured at high AOIs.

The estimated standard deviations align well with the expected behavior of the sensor. The bin width is set to 1 mm to clearly show the single values at the tail ends of the distribution. The lowest values fall within the typical range for calibration panels, while the highest value is just slightly above 2 cm. This higher value relates to measurements of the ground, which can be attributed to the sensor's orientation relative to the ground. All measurements of the ground have a high AOI, resulting in larger measurement uncertainty (Soudarissanane et al., 2009). Since AOI effects are not part of the model yet, the higher uncertainty is assigned to the ground surface as a whole.

For some surfaces, the estimated standard deviations can be compared to LiDAR measurements. This requires the system to be stationary and the planes of the facade to be clearly visible at a low AOI. For each facade, at least three planes of about 1 m^2 are cropped. Each of these cropped planes should consist of at least two scan lines. The areas are chosen to be observed at a low AOI, so the standard deviation of an estimated plane corresponds approximately to the standard deviation of the distance measurements of the LiDAR. Only three facades fulfill these requirements, as the system only stops once per round. The ground is excluded, as the observations are always at a high

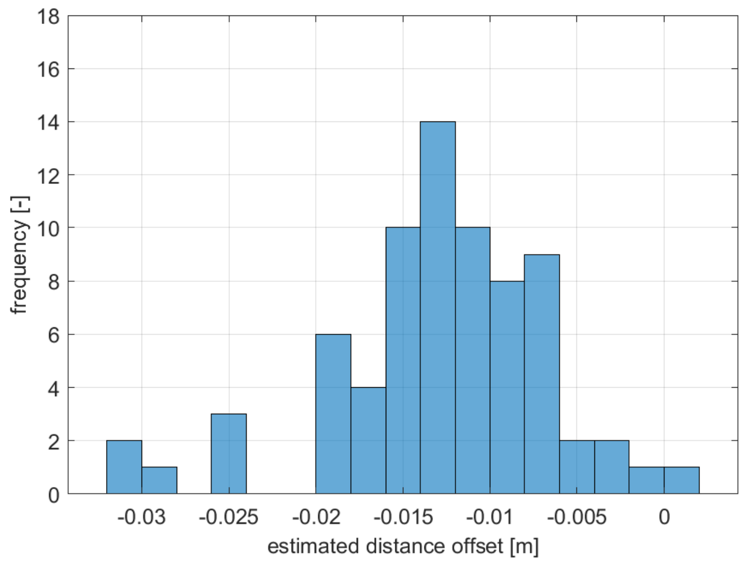

Figure 4.12: *Histogram of estimated LiDAR distance offsets for processed real dataset.*

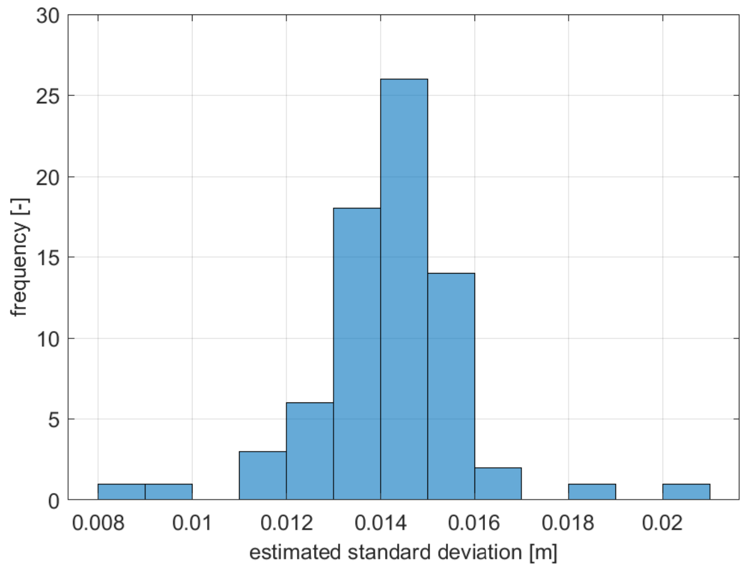

Figure 4.13: *Histogram of estimated LiDAR standard deviations for the distance measurements for the processed real dataset.*

AOI due to the mounting of the LiDAR. Table 4.5 shows the estimated mean standard deviations for the facades and the standard deviations estimated for the plane group by the dual filter. The filter generally estimates the standard deviations are few millimeters higher than the result of the plane estimation. This effect has multiple reasons. First, the simulation already demonstrated that issues in the modeling of the dynamics lead to overfitting of the estimated standard deviations of the distance measurements. Next, the modeling currently omits the effect of the AOI of the distance measurement. Higher AOIs lead to higher random noise in the distance measurements, including observations with higher AOI, thus increasing the estimated standard deviation. Finally, deviations in the estimation of the distance offset also lead to higher values for the standard deviations, as the residuals are computed compared to the map, which should be free of systematics. While the estimation of the distance offset works well, assumptions about the grouping of the planes in the map can lead to increased deviations in the estimated offsets.

Table 4.5: *Comparison of standard deviations of estimated planes in the stationary phase and standard deviations estimated by the dual filter.*

Plane ID	Plane [mm]	Filter [mm]
6	12.1	14.7
18	9.5	13.5
19	10.7	12.9

Evaluation with ICP Trajectory

In the simulations, deviations were predominantly observed during turns, whereas the real data reveals more widespread deviations, making evaluation the next focus. For the real dataset, true values are not known, so reference trajectories are utilized. Specifically, the trajectory from a commercial MMS is employed. The estimated trajectory of the system is noted to have an absolute accuracy of 20 to 50 mm in the absence of GNSS outages (Riegl, 2011). These outages or other problems for GNSS, such as multipath effects, are common in urban environments (Hsu, 2018). The locality of these effects causes deviations that vary along the trajectory but repeat across multiple rounds. As an alternative, the dataset provides a trajectory calculated by registering the LiDAR point clouds to the TLS using ICP (Axmann et al., 2023). The original TLS point cloud is registered using GCPs, which are independent of GNSS measurements. For this ICP trajectory, only the *estimated* variant is compared, as the suitability of the solution has to be investigated. The pose deviations are displayed in Figure 4.14 and the RMSEs are shown in Table 4.6.

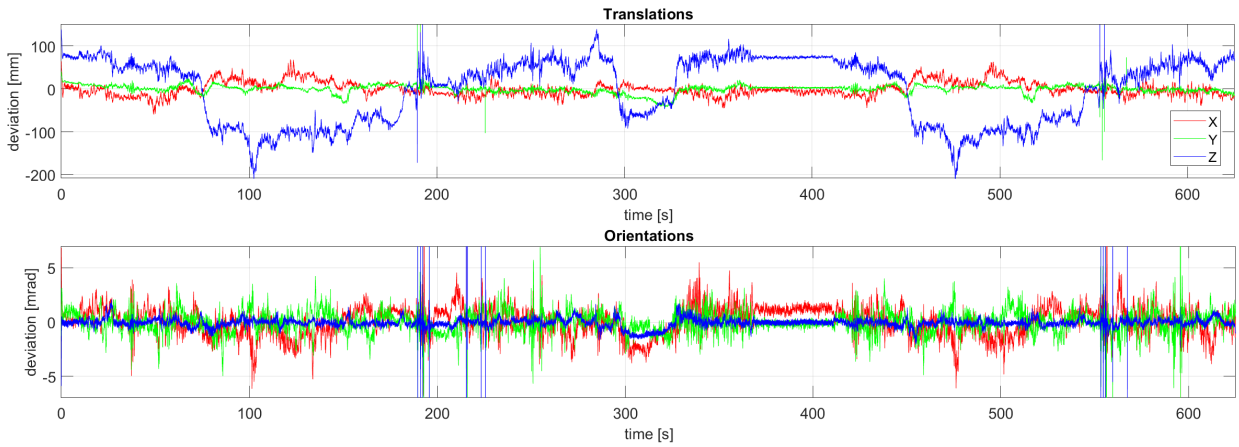

Figure 4.14: *Epoch-wise deviations of the real dataset for estimated variant computed with respect to the ICP trajectory. Deviations are rotated into the body-frame of the system. Color for the axes: red for X (forward), green for Y (left), and blue for Z (up).*

The ICP trajectory exhibits slightly more noise than the MMS trajectory, resulting in some brief outliers within the time series. As the evaluation using the MMS trajectory already showed the stability of the filter result, only the relevant deviations are presented by limiting the y-axis. Again, the deviations display recurring patterns, which are different compared to the MMS trajectory. This difference confirms the assumption that deviations of the reference trajectory strongly affect the computed deviations. The horizontal translations exhibit lower deviations, with the left component aligning very well, showing deviations of less than 20 mm. The height component displays higher deviations than previously observed, which can be attributed to the LiDAR's orientation, measuring the environment at a higher AOI. In this case, both solutions are affected by this effect compared to the comparison to the MMS solution, which used a smoothed INS trajectory.

Table 4.6: *Descriptive statistics of RMSEs of the translations and orientations of the processed real dataset with the ICP trajectory (after removal of peaks).*

Component	Metric	*assumed*
Translation [mm]	overall	43.88
	68% quantile	44.98
	95% quantile	70.14
Orientation [mrad]	overall	0.88
	68% quantile	0.92
	95% quantile	1.57

Regarding the overview in Table 4.6, the translational deviations are similar to the deviations computed using the MMS trajectory. The higher deviations of the height component mask the better performance in the horizontal components. The evaluation of the orientations delivers smaller deviations when excluding the peaks. Without excluding 48 peaks in the time series caused by problems in the ICP trajectory, the RMSE of the orientations is 11.29 mrad. The LiDAR data utilized for producing this trajectory is similar to that used for the ESKF in this thesis. Since the IMU data is excluded and the original TLS point cloud is used, the ICP point cloud can be regarded as a worse solution, although it serves as an adequate evaluation benchmark for GNSS solutions.

Evaluation based on Point Clouds

An alternative method for evaluating the estimated trajectory involves using the poses to georeference the LiDAR point clouds. To begin with, the georeferenced point cloud is computed and checked visually. Figure 4.15 provides a qualitative view of the georeferenced point cloud. The selected area shows the crossing scanned at 75 and 450 seconds, each time for about 15 seconds. An RTS for tracking one of the vehicles is set up on an industrial tripod with a street sign in the background. The structure can be clearly seen without noticeable shifts, with only the persons moving during the data acquisition. The point cloud is colored by the intensities of the point cloud, showing the highest values for retro-reflective surfaces like the high visibility vests and signs. This indicates generally good results, as both rounds fit well and show the scene clearly.

For a quantitative evaluation, the C2C distances between the georeferenced point cloud and the TLS reference point cloud are computed. Lower deviations in trajectory estimation should result in lower deviations in the point clouds. Although neither the reference point cloud nor the georeferenced LiDAR scans are cleaned to remove scatter points, this method still offers valuable insights into the georeferencing accuracy. The results of this computation for the *estimated* setting are presented as a 2D histogram in Figure 4.16. The bins are grouped per second for the time and per 2 mm for the distances. Scatter points lead to a few longer distances (up to meters), which are barely visible even with a logarithmic scaling for the colors. While the results do not directly relate deviations to specific pose components, they offer a general perspective on the accuracy within the georeferenced point cloud. Across the entire time series, the majority of distances fall within the 1-3 cm range. The distances are constant along the trajectory without peaks seen in the trajectory comparison, indicating a stable result. However, during certain time intervals, fewer distances fall within this range. These time intervals often correspond to turns, so these changes in the distances might indicate similar deviations seen in the simulations. Alternatively, as these areas are crossing, blocking objects in the environment may occlude the environment for the comparison. The result for the *assumed* setting is visually indistinguishable and thus omitted.

Table 4.7 summarizes the results over the entire trajectory for both settings. The median of the C2C distances is lower for the *estimated* setting 20.4 mm, compared to 20.6 mm for the *assumed*

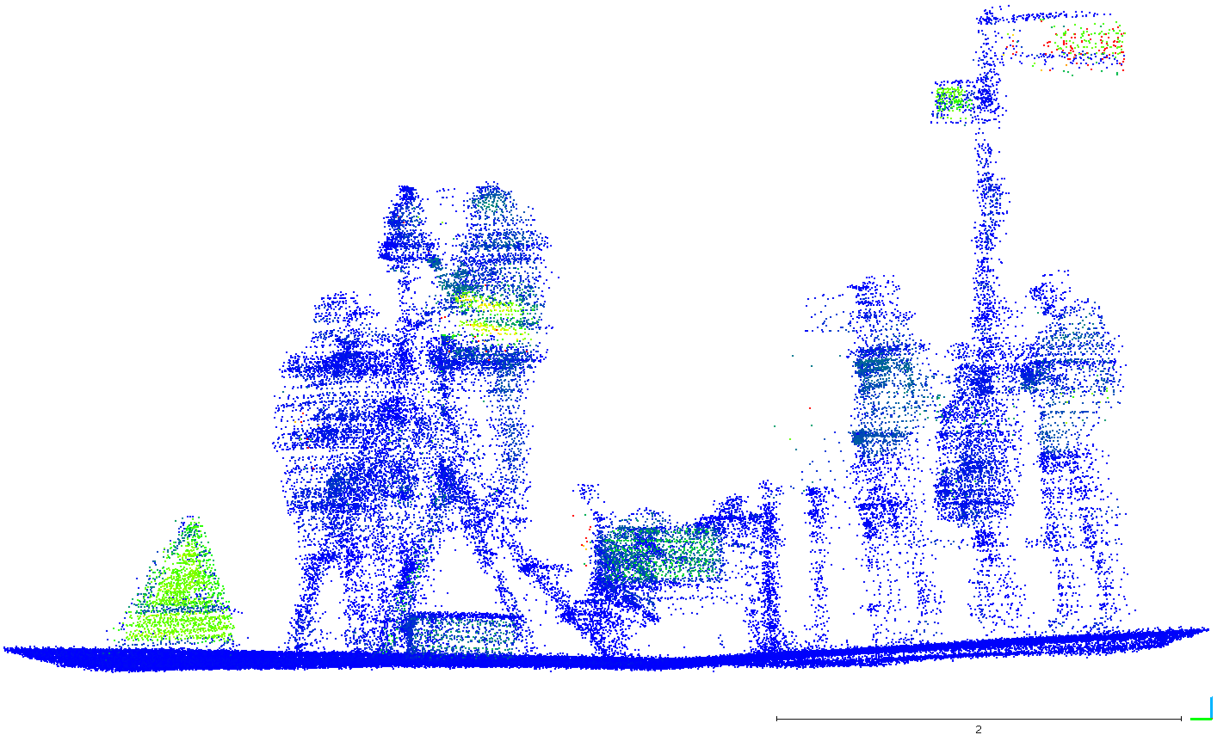

Figure 4.15: *Selected part of georeferenced LiDAR point cloud, showing tracking setup with RTS on industrial tripod next to street sign. Points colored by intensity (low to high: blue-green-red).*

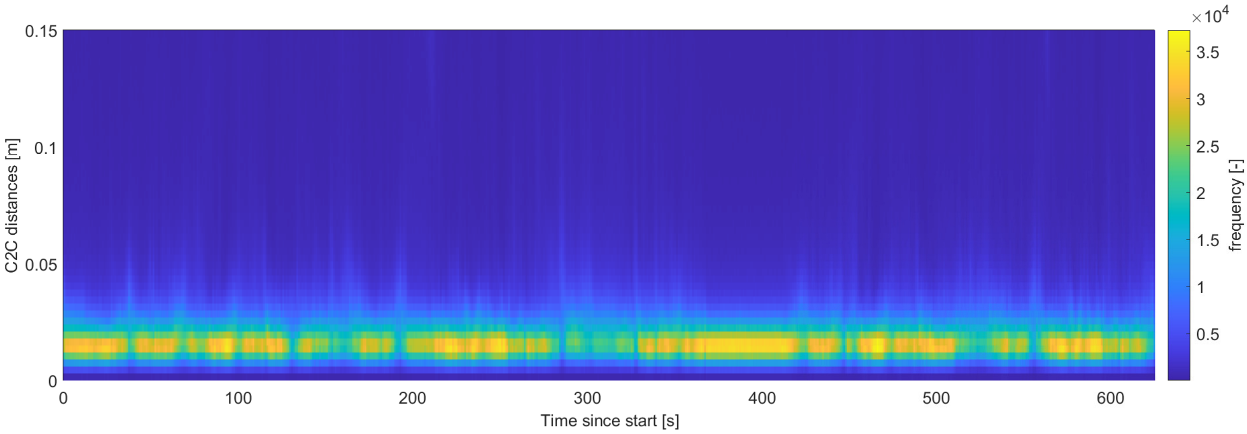

Figure 4.16: *2D histogram of C2C distances between georeferenced LiDAR point clouds using the estimated variant and TLS reference point cloud.*

setting. For higher quantiles of the distances, the results become more similar, with the same value of 69% for the 95% quantile. This is similar for the ratio of distances within a certain distance threshold, with more shorter distances for the *estimated* setting and with 80.3% the same number of distances for the 10 cm threshold. The difference between the results is small for multiple reasons. First, the actual difference in the trajectory estimation is small, as seen in the evaluation using the MMS trajectory. Second, the estimated distance offset used for the filter is not transferable here, as the distance computation is applied to any captured point, including points from objects not used in the position tracking. Thus, the estimated distance offsets are not applied for this evaluation and only a general offset, i.e., from the calibration, can be applied.

Table 4.7: *Descriptive statistics of C2C distances computed between the georeferenced LiDAR point cloud and the TLS point cloud for the assumed and estimated variants.*

distance [mm]	*assumed*	*estimated*
Median	20.6	20.4
68% quantile	33.9	34.1
95% quantile	69.1	69.0
distances within [%]	*assumed*	*estimated*
3 cm	64.9	65.0
5 cm	74.2	75.2
10 cm	80.3	80.3

Findings and Implications from Real-World Evaluation

The real data processing demonstrates that the developed algorithm is capable of producing high-accuracy trajectories and georeferenced point clouds for a LiDAR-based MSS. Estimating distance offsets and standard deviations for distance measurements results in improved position tracking. Regarding research question **2**, the Mahalanobis distance is decreased by up to 30% by the extended modeling. The remaining inconsistency can be attributed to the insufficient accuracy of the available reference trajectories. Concerning research question **3**, the estimation of uncertainties notably reduces the deviations of the orientations. Quantitatively, when compared to a TLS point cloud, approximately 80% of all georeferenced LiDAR points have a C2C distance of less than 10 cm, including those from moving objects and scatter points. The estimated values for the distance offsets and standard deviations are plausible.

5 Conclusion and Outlook

5.1 Conclusion

Autonomous systems are used in various applications for their benefits in terms of cost and safety. Their automated operation increases the requirements of the data and algorithms, as no human intervention is possible. For this, LiDARs, fused with data from other sensors, are a crucial tool for autonomous systems to reliably detect the environment and position themselves within it. To gain the best possible results, this fusion requires a proper calibration and processing strategy. Opposed to MMSs, which acquire data for post-processing, autonomous systems need to work reliably at any time to ensure safe operation. Hence, bundle adjustments or other batch approaches are not suitable. Instead, online processing is necessary, which needs to be able to adapt to changing conditions. The need to adapt during operation requires new methods, which were introduced in this thesis. Finally, the results need to be evaluated to provide quality measures for the system.

This thesis aims to investigate the uncertainties of LiDARs along the processing chain to georeference an MSS. The processing chain includes the calibration, data fusion, and evaluation of the results. Uncertainties from each step affect the remaining processing and thus need to be properly included in the data processing to achieve the best possible results for the system. The processing should be methodologically able to process the data in real-time. The results need to be investigated based on their deviations and in the context of their estimated uncertainties to properly assess them. The research questions from Section 1.2 are answered in this thesis, which leads to the contributions from Chapters 3 and 4, that can be summarized as follows:

- **Consistent estimation of calibration parameters**
  Research question **1** is addressed by employing an extrinsic calibration process, which utilizes reference panels to determine the transformation parameters to a defined platform frame, relying on observations of the LiDAR. To enhance the outcomes of this calibration, the intrinsic calibration of the LiDAR is simultaneously conducted. Besides adjusting for a distance offset, the sensor's stochastic model is estimated via VCE. Estimating all parameters necessitates multiple positions within the calibration field to provide sufficient geometric diversity. This has the additional benefit of reducing all but three correlations between the estimated parameters to below 50%. Each parameter is accompanied by its respective VCMs, offering information on the associated uncertainties. Since the estimation leaves a remaining deviation from the true values, the parameters serve as initial values for the state estimation in the data fusion, allowing for further refinement.

- **Consistent prediction of point uncertainties**
  For research question **2**, it is essential to propagate all uncertainties throughout the processing chain to ensure a consistent prediction of point uncertainties. The primary data fusion is conducted using an ESKF, where IMU and LiDAR observations are combined to localize the system across different environments. IMU observations define the epochs for the filter and are responsible for propagating the system's pose, while LiDAR observations are matched to a localization map. To enhance accuracy, the system's pose is linearly interpolated to the timestamp of each LiDAR point. The random uncertainties of the distance measurements are iteratively estimated during operation by a recursive VCE. This procedure is applied to account for the changing uncertainties due to the environment and observed materials. The map provides information about groups of observed surface planes with varying properties. In

comparison to true trajectories in a simulated environment, the recursive VCE improves the estimated translations by up to 40% and the orientations by about 10%. For the processed real data, no true values for the uncertainties are available, but all uncertainties are in a plausible range. Here, specifically the accuracy of the map and reference data are more important in the evaluation than the system's modeling itself. Acquiring reference trajectories with adequate accuracy can be particularly challenging when working with real datasets. Alternatively, the georeferenced point cloud's quality can be assessed directly by comparison against a reference measurement, which could be another point cloud or predefined reference geometries. Distance computations to the reference are used to compute measures of the point cloud's quality. In publication **C**, the accuracy improvement of the georeferenced point cloud by integrating uncertainties can be seen, reducing the RMSE of the distance from 10.0 to 8.1 mm. Again, the VCM of the point cloud can be utilized to weight the computed distances, offering insight into the result's consistency. Here, the importance lies in rotating the VCM to properly apply the weighting.

- **Determination of systematic deviations**
  Research question **3** is handled by estimating corrections, as systematic deviations are detrimental to estimation processes. In the georeferencing procedure presented, the extrinsic calibration of the LiDAR and its distance measurements are major sources of such deviations within the system. To address this, both are incorporated into the state vector, allowing for their updates during position tracking. Distance offsets are estimated per plane group, similar to the random uncertainties. This approach accommodates variations arising from material and surface-dependent properties. In the simulations, these offsets can be estimated very well, reducing the RMSE of the distance offset to below 1 mm (from 15 mm). For the real dataset, the estimated offsets correspond to the range of values expected for the used sensor. This extended estimation benefits the trajectory estimation as a whole, improving the orientation estimation by 30% in the simulations and 15% in the real data. The estimation of the extrinsic calibration parameters of the LiDAR requires either additional sensors to differentiate between the information from the sensors or specifically designed motion to determine the transformation parameters. The latter option is not feasible in normal traffic scenarios.

This dissertation introduces a recursive VCE for the adaptation of the stochastic model during recursive estimation procedures. It is demonstrated on the position tracking of a LiDAR-based MSS with an implicit measurement model for map matching, taking uncertainties of the calibration into account. For further adaptation of this methodology in applications, some limitations have to be accounted for. The current methodology is limited by the high dependency on the constellation between the system and the localization map, which leads to situations in which the extrinsic calibration parameters can not be estimated. Further extensions to the modeling of the VC are necessary to account for effects, like the AOI. The evaluation is currently limited by the availability of sufficiently accurate reference data.

5.2 Outlook

Although this thesis has contributed significantly to various aspects of multi-sensor data fusion, numerous challenges remain.

Related to the developments discussed here, the modeling of **sensor uncertainties** could benefit from further enhancements. Currently, only the material-dependent effects on measurement uncertainty have been modeled. Since geometric factors such as distance and AOI also impact measurements (Soudarissanane et al., 2009; Zámečníková & Neuner, 2018), these effects should be integrated into the uncertainty propagation framework. While more detailed intrinsic sensor models are available, their estimation depends on the specific sensor and setup, which may not always allow for significant estimation (Glennie & Lichti, 2010; Glennie et al., 2016). Correlations within point cloud data also influence estimation accuracy, as neglected correlations lead to biased estimates (Jost, 2023; Kerekes, 2023). Currently, subsampling techniques are used to mitigate computational demands and reduce the impact of unaddressed correlations. Further investigation into the correlation structure of low-cost LiDARs and kinematic scanning could enhance the consistency of estimations. To achieve this, synthetic VCMs could be generated by using autocorrelation functions or similar techniques. The primary challenge here is the movement of the system, which results in dynamic changes to the geometry and the path of the rays during measurement.

Another major consideration for the methodology's applicability is **robustness**, which presents challenges in two areas. First, outliers, common in LiDAR point clouds due to scattering, affect estimation results. While robust assignment methods like RANSAC can minimize their occurrence, the estimation process should itself be robust. This is crucial since the residuals of the state estimation are also used for the VCE. Implementing semantic segmentation of LiDAR point clouds can assist in eliminating points that capture the relevant infrastructure (Milioto et al., 2019; Zhao et al., 2024). Additionally, scatter points might be identifiable by features such as intensity, and points classified as other traffic participants could be used to track these entities. This information could be fused with image or Radar observations. Second, anomalies, like environmental changes, can deteriorate georeferencing performance. As the online uncertainty estimation relies on plane grouping, substantial deviations could indicate changes and trigger map updates. Literature indicates worse performance with generalized maps such as LOD2 city models compared to more detailed alternatives (Moftizadeh, 2024; Wahbah et al., 2025). New data sources could enhance map creation, as MMSs are well-suited for rapidly capturing large areas with high accuracy, provided suitable reference data is available. Exploring existing sources of such information could be key to generating accurate, large-scale maps for guiding MSSs localization.

Regarding the broader aspects of multi-sensor fusion, **uncertainty modeling** in multi-step processes is crucial. Here, only a minimal MSS with two sensors is investigated. Additional sensors introduce more information, but also additional uncertainty sources. Assuming a normal distribution for uncertainties from previous steps, such as calibration parameters or corrections, can fall short of accurately representing their constant influence on the current estimation. Alternative models, such as interval-based estimation techniques (Su, 2025), might better describe this influence. Consequently, other processing frameworks, like PF, set-membership approaches (Andrew, 2002) or similar methodologies, are needed for the estimation. However, these techniques currently face significant challenges in computational efficiency, which is especially important considering the use cases for the methodology. Hardware used for machine learning (ML) applications provides the possibility for parallelization to enable faster computation.

On the topic of ML, uncertainty modeling for LiDAR observations can leverage models that incorporate **prior knowledge**. While this thesis utilizes no ML approaches for the uncertainty modeling, it can serve as a baseline for these techniques. The exploration of ML methods for quantifying LiDAR uncertainties is still in its early stages but shows considerable promise. Initial studies suggest the feasibility of predicting systematic corrections for various surfaces (Hartmann et al., 2024). This could replace the estimation of distance offsets by correcting them before the update. Additionally, extensions to model random uncertainties are achievable, especially with advanced neural networks capable of predicting distributions (Li et al., 2024). Also, studies are being done on how to model the overall accuracy of point clouds acquired by kinematic MSS (Xu et al., 2025). The main limitation of these models is the necessity of comprehensive training data, as they must encompass a diverse range of surface types to adequately capture the variability of different materials. Furthermore, integrating image data can enhance the prediction of uncertainties by providing additional context (Wei et al., 2018; Zhu et al., 2024). Images can aid in identifying surface types or plane groups, thereby facilitating more accurate grouping.

The **evaluation** of kinematically acquired data is an important issue for many applications. In classical geodetic measurements, quality assessment and assurance are paramount. For kinematic measurements, evaluations often rely on similar principles but involve extended chains of processes or fail to thoroughly assess all outcomes. Methods such as tracking via RTS can provide trajectories, but typically only for a single position on the system. While approaches that track multiple points are feasible, they require greater effort for time synchronization (Lerke & Schwieger, 2021; Thalmann & Neuner, 2021; Dammert et al., 2025). The method presented for point cloud comparison can only evaluate the entire pose if the environment is structured properly. Another option is tracking via camera, where the vehicle's pose can be identified in images, either by using markers affixed to the vehicle (Trusheim et al., 2024) or by employing models that represent the vehicle's shape (Lou et al., 2005; Coenen & Rottensteiner, 2021). For the latter, the accuracy of pose determination remains an area of investigation. Combined polar measurement systems with cameras, like the T-Probe of the LT, or similar systems, are constrained by the necessity of a line of sight between the evaluating sensor and the kinematic system. Depending on the type of system, repeated trajectories can be used for evaluation. Possibilities from robot arms to rail tracks (Tombrink et al., 2025) offer evaluation for repeated movements to distinguish precision and accuracy.

Despite these ongoing challenges, multi-sensor fusion remains a crucial tool for the efficient data acquisition processes of the future. Within this context, this dissertation offers significant contributions to the modeling of essential processes aimed at achieving optimal results, including the modeling of uncertainties.

6 List of Further Publications

This chapter lists further publications of the author. These publications differ from the publications listed in Section 1.3 in that their content is either included in the relevant publications or that they are not directly connected to the goals of the dissertation.

- Axmann, J., Moftizadeh, R., Su, J., Tennstedt, B., Zou, Q., Yuan, Y., **Ernst, D.**, Alkhatib, H., Brenner, C. & Schön, S. (2023). LUCOOP: Leibniz University Cooperative Perception and Urban Navigation Dataset. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, Anchorage, AK, USA. IEEE.
  `https://doi.org/10.1109/iv55152.2023.10186693`
- **Ernst, D.**, Jüngerink, J., Kindervater, L., Moftizadeh, R., Alkhatib, H. & Vogel, S. (2021). Data fusion for georeferencing a laser scanner based multi-sensor system in a city environment. In *2021 IEEE 24th International Conference on Information Fusion (FUSION)*, 1 - 8.
  `https://doi.org/10.23919/FUSION49465.2021.9627026`
- **Ernst, D.**, Vogel, S., Neumann, I. & Alkhatib, H. (2022). Analyse unterschiedlicher Positionskombinationen zur intrinsischen und extrinsischen Kalibrierung eines Velodyne VLP-16. *allgemeine vermessungs-nachrichten (avn)*, 129(6), 244–252.
- **Ernst, D.**, Vogel, S., Alkhatib, H. & Neumann, I. (2022). Intrinsische und extrinsische Kalibrierung eines Velodyne VLP-16. In T. Luhmann & C. Schumacher (Hrsg.), *Beiträge der Oldenburger 3D-Tage 2022* 20, 186–193). Wichmann Verlag.
- Hartmann, J., **Ernst, D.**, Neumann, I. & Alkhatib, H. (2024). PointNet-based modeling of systematic distance deviations for improved TLS accuracy. *Journal of Applied Geodesy*, 18(4), 613–628.
  `https://doi.org/10.1515/jag-2023-0097/html`
- Vogel, S., **Ernst, D.**, Neumann, I. & Alkhatib, H. (2022). Recursive Gauss-Helmert model with equality constraints applied to the efficient system calibration of a 3D laser scanner. *Journal of Applied Geodesy*, 16(1), 37–57.
  `https://doi.org/10.1515/jag-2021-0026`
- Wahbah, M., Ramme, L., **Ernst, D.**, Vogel, S., Neumann, I. & Alkhatib, H. (2025). Georeferencing Autonomous Vehicles Using LoD2 and HD Maps: Performance Assessment in Simulated Urban Environments. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, X-G-2025, 915–922.
  `https://doi.org/10.5194/isprs-annals-X-G-2025-915-2025`

7 Publications

1. **Publication A:** © 2022 IEEE. Reprinted, with permission, from Dominik Ernst, Hamza Alkhatib, Ingo Neumann, Sören Vogel, Analysis of Multiple Positions for the Intrinsic and Extrinsic Calibration of a Multi-Beam LiDAR, 2022 25th International Conference on Information Fusion (FUSION), 08/2022
2. **Publication B:** © 2023 IEEE. Reprinted, with permission, from Dominik Ernst, Sören Vogel, Ingo Neumann, Hamza Alkhatib, Error State Kalman Filter with Implicit Measurement Equations for Position Tracking of a Multi-Sensor System with IMU and LiDAR, 2022 25th International Conference on Information Fusion (FUSION), 12/2023

Analysis of Multiple Positions for the Intrinsic and Extrinsic Calibration of a Multi-Beam LiDAR

1st Dominik Ernst	2nd Hamza Alkhatib	3rd Ingo Neumann	4th Sören Vogel
Geodetic Institute	*Geodetic Institute*	*Geodetic Institute*	*Geodetic Institute*
Leibniz University Hannover	*Leibniz University Hannover*	*Leibniz University Hannover*	*Leibniz University Hannover*
Hanover, Germany	Hanover, Germany	Hanover, Germany	Hannover, Germany
ernst@gih.uni-hannover.de	alkhatib@gih.uni-hannover.de	neumann@gih.uni-hannover.de	vogel@gih.uni-hannover.de

***Abstract*—The usage of light detection and ranging sensors (LiDARs) has grown rapidly in recent years. The ability to directly capture 3D point clouds is a big advantage compared to other visual systems like cameras. One disadvantage is that uncertainty information is difficult to obtain for these systems, although this information is crucial for the decisions based on the measurements.**

This becomes even more important, when LiDARs are used in conjunction with other sensors in multi-sensor systems (MSS). The sensor data fusion with different sensors requires an extrinsic calibration, which describes the transformation between the LiDAR frame and the body frame of the platform. This can be done utilizing object space information measured by the LiDAR, which is used to infer the origin of the sensor frame based on reference geometries. This process can be used to additionally determine intrinsic parameters of the sensor. Possible intrinsic parameters are corrections for the distance measurements or approximations for the uncertainty of the measurement elements. In this work, the determination of extrinsic and intrinsic parameters is combined for the first time with the approximation of a stochastic model for a multi-beam LiDAR. This is demonstrated on a real-data set of a Velodyne VLP-16, for which the transformation parameters between sensor frame and body frame are determined. Additionally, a distance offset is determined and the variance components are estimated to establish a better approximation for the stochastic model. The impact of the calibration field and choice of positions within this calibration field are shown and discussed. The results are evaluated in a separate experiment using a kinematic MSS.

***Index Terms*—LiDAR, calibration, multi-sensor system, data fusion, parameter estimation, variance component estimation**

I. Introduction

Light detection and ranging sensors (LiDARs) are increasingly popular for a multitude of applications. Their ability to directly capture 3D point clouds can be useful in many areas. The comparatively cheap price and robust construction enable the use of kinematic systems, which can quickly capture large amounts of data. With improving hardware for data processing, it becomes possible to use LiDARs for localization and obstacle detection. Although the uncertainty of the acquired point clouds becomes a safety-critical factor in these applications, the uncertainty budget of LiDARs has not yet been sufficiently modelled. Previous studies have often been limited to investigating the distance measurements in controlled environments [1], [2].

An important aspect of the uncertainty budget of LiDARs is the use case for the sensors. In most cases, LiDARs are used in multi-sensor systems (MSSs), for which observations from multiple sensors are fused and processed together. The quality of the sensor data fusion is dependent on the system calibration of the sensors. Generally, the used sensors can be broadly divided into two groups: referencing and capturing sensors. While referencing sensors, like Global Navigation Satellite System (GNSS) antennas or inertial measurement units (IMUs), are purely used to determine the pose of the system during operation, the capturing sensors, e.g. cameras or LiDARs, are used to capture the environment. In recent years, the data acquired by these capturing sensors are also used for localization. This can be done by integrating information from maps [3]–[5] or by creating own maps using Simultaneous Localization and Mapping (SLAM) [6].

With the increasing use of capturing sensors for pose estimation, the importance of system calibration of cameras and LiDARs is also growing. System calibration describes the determination of transformation parameters between frames of different sensors or between a sensor frame and a body frame, which can be defined to standardize sensor data fusion. In computer vision, the term extrinsic is used for the transformation parameters between different frames, often in the context of cameras.

Depending on the type of sensors and the goal of the application, different techniques can be used to determine the extrinsic parameters. The direct approach is to determine the transformation parameters between the frames based on a CAD model of the platform and construction drawings of the sensors. This approach is dependent on the availability of the CAD model and the construction drawings. Additionally, manufacturing tolerances of the platform and sensors are neglected or at least not sufficiently included in the determined parameters.

Approaches using object space information eliminate the influence of manufacturing tolerances by utilizing measurements of the sensors. For applications aimed at fusing images with depth information from LiDARs, planar checkerboards in multiple orientations are commonly used [7]. MSSs using two

This work was funded by the German Research Foundation (DFG) as part of the Research Training Group i.c.sens [RTG 2159] and NE 1453/5-1.

LiDARs with an overlapping field of view can be calibrated using suitable planar surfaces without the need for external reference measurements or knowledge of dimensions [8].

Alternatively, the extrinsic parameters can be determined for a defined body frame. Strübing & Neumann [9] present an approach using reference geometries. While the original application was intended for a railway measurement system, the method was also used in industrial quality assurance [10] and for mobile mapping systems referenced by GNSS [11].

The calibration procedures can be performed in different modes. The calibrations can be either done before the measurement or during the measurement (in-situ). The prior calibration is either static using external referencing sensors ([9], [10]), or during movement using the referencing sensors of the MSS [11], [12]. Since MSSs are typically used for kinematic measurements, the in-situ calibrations are performed based on the measurements during the movement to improve the extrinsic calibration parameters based on the comparison of multiple runs [13].

The calibration of intrinsic models for LiDARs often focuses on the distance measurement. Studies by various authors analyse stability and accuracy of distance measurements [1], [2], [14]. Complete models for distance and angle measurements are used in [14], [15]. The authors in [15] combine the instrinsic and extrinsic calibration for an MSS with a camera, LiDAR and IMU using a variety of geometries. The method relies on the camera to derive the transformation to the superordinate frame. Heinz [11] et al. show an extension of the system calibration based on [9], which is the simultaneous determination of intrinsic and extrinsic parameters for a profile laser scanner. Additionally, the stochastic model of the adjustment is validated using a variance component estimation (VCE).

The contribution of this work is the adaption of the extrinsic and intrinsic calibration in combination with the variance component estimation to a multi-beam LiDAR. The procedure is applied to a Velodyne VLP-16 on a platform and the results are discussed in the context of the used calibration field with regards to the reached standard deviations and correlations for the parameters using multiple positions. Additionally, the determined parameters are evaluated in an independent experiment.

The paper is structured as follows. Section II describes the setup used for the system calibration using object space information including the used sensors. Afterwards, the methodology for the estimation of the parameters is presented in section III. The focus lies on the model extension of the adjustment and the VCE. The results are shown in section IV. First, the results of the adjustment itself are presented and discussed, before the determined parameters are evaluated in an independent experiment. The paper concludes in a summary and an outlook in section V.

II. Experimental Setup

In this section, the setup for the calibration is shown. This is done before the introducing the algorithms used for the parameter estimation, since some decisions for the adjustment are motivated by the available instruments and the calibration field.

The calibration is performed using object space information. For this procedure, artificial reference geometries are measured with the LiDAR. Here, a Velodyne VLP-16 [16] is used. It has 16 rings, which cover a field of view of $360° \times 30°$. As reference geometries, panels or other planar objects are commonly used [9]–[11]. These reference panels and the platform itself are referenced by a sensor of superior accuracy. In this case, a Leica Absolute Tracker AT960-LR (LT) is used. This sensor is commonly used for surveying tasks in industrial applications and can track different targets. Either corner cube reflectors (CCRs) can be used, which are tracked with an accuracy of about 50 μm, or a T-Probe, which is a 6-DoF sensor with an accuracy of about 100 μm, is available [17]. Using approximate parameters for the transformation between sensor frame and body frame of the platform, the point clouds captured by the LiDAR are transformed and the euclidean distances to the reference panels are computed. These distances act as misclosures for the adjustment described later, in which they are eliminated by minimization of the square sum of the residuals.

Three frames are involved in the calibration. The sensor frame of the LiDAR s is defined by the origin and orientation of the point clouds. The body frame of the platform b is uniquely defined by three CCRs placed in magnetic nests, which are fixed to the platform. The superordinate frame l is the laboratory coordinate system of the 3D laboratory of the Geodetic Institute Hannover. The LT is stationed in the laboratory to directly deliver measurements in the l-frame. An overview of the calibration setup including the frames is shown in Fig. 1. A black ring marks frames, which are shown at their origin. The colors are red-green-blue for the X-, Y-, and Z-axis.

The constellation of the reference panels is a crucial factor for the quality of the resulting transformation parameters. Depending on the orientation of the panels and the constellation concerning the platform, the panels are sensitive towards different transformation parameters. For example, panels 02 and 10 are sensitive for the translation in the X-axis of the b-frame. In its depicted configuration, the calibration field is based on [10]. It was designed for an MSS with a profile laser scanner. This leads to the fact, that only some of the rings of the Velodyne are capturing the panels for a position of the platform.

To further enhance the determination of the calibration parameters, multiple positions for the platform are used. Depending on the choice of positions, different rings of the LiDAR are contributing to the parameter estimation. Furthermore, multiple positions enable a higher variety in the geometry of the constellations between the sensor and reference panels. This is beneficial for the determination of the parameters, since the influence of the rotations increases with the distance between the sensor and the reference panels. This leads to high importance of varying distances in the calibration.

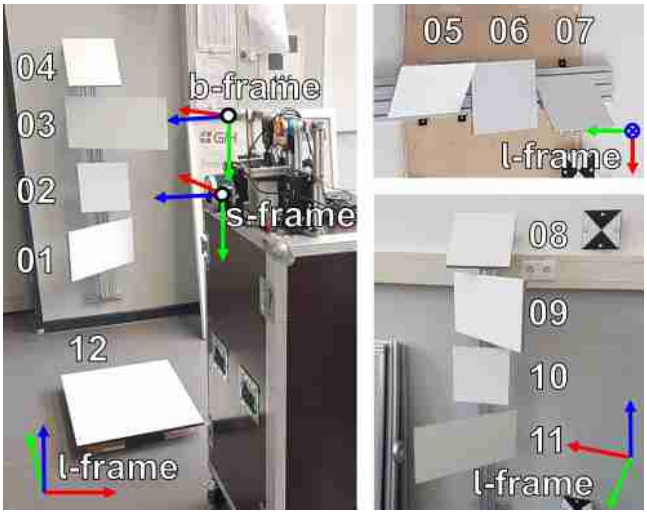

Fig. 1. Overview of the calibration field with panel numbers: involved frames with panels (left), panels mounted on ceiling (top right), panels on opposing wall (bottom right)

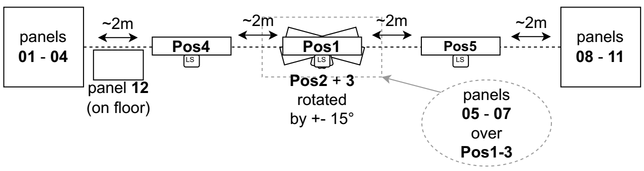

Fig. 2. Overview of the five positions in total of the platform for calibration

Additionally, the correlations of the covariance matrix of the determined parameters can be reduced. This principle is similar to the idea behind the reduction of correlations for camera calibrations [18]. Fig. 2 shows an overview of the used positions for the calibration field. For this procedure, the exact distances between the positions are less important than the general constellation.

III. Methodology

After showing the used setup, the algorithm used for the parameter estimation can be introduced. The functional model for the adjustment is presented in multiple steps. First, the original spherical observations of the LiDAR are converted to cartesian coordinates. The original observations $\boldsymbol{\ell}_i$ for a point i are the distance d_i, the elevation e_i (determined by the rings), and the azimuth angle α_i. Additionally, this transformation introduces the distance offset d_0 as a model extension compared to earlier applications [10], [19],

$$\mathbf{p}_{s,i} = \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix}_i = \begin{bmatrix} (d_i + d_0) \cdot \cos e_i \cdot \cos \alpha_i \\ (d_i + d_0) \cdot \cos e_i \cdot \sin \alpha_i \\ (d_i + d_0) \cdot \sin e_i + z_c \end{bmatrix}. \quad (1)$$

The transformation includes the vertical correction z_c recommended by the manufacturer [16]. No scale factor is estimated, since the distances in the calibration field are too short. For high-end laser scanners, the scale factor is about 10 ppm [20], which would amount to 60 μm in the current calibration field. For the used VLP-16, this would be hard to determine given the quantization of the measurements with 2 mm [16].

Due to the usage of reference geometries, not all captured points are relevant. Based on the cartesian coordinates, an assignment of the point clouds to the reference panels is done by cropping the point clouds. This is done by hand to exclude scatter points and mixed pixels at the borders of the panels. The cropped point clouds are subsampled before the usage in the calibration adjustment. This has two reasons. Firstly, it reduces the computational cost of the adjustment computation. Earlier investigations [19] have shown, that increasing the number of observations does not necessarily improve the results. Secondly, subsampling is done to standardize the number of points per panel. This avoids an indirect weighting between the panels by the number of assigned points. In this case, about 100 rotations of the VLP-16 are recorded. For the adjustment, 50 points per plane and position are used.

After the transformation from spherical to cartesian coordinates, two transformations between frames are concatenated. The first transformation is based on the required transformation parameters between the s-frame and the b-frame. The second transformation describes the pose of the platform in the superordinate frame l. Both transformations are rigid body motions, which are described with a 3D translation vector $\mathbf{t} = [t_x \; t_y \; t_z]^T$ and a 3D rotation matrix

$$\mathbf{R}(\omega, \varphi, \kappa) = \mathbf{R}_z(\kappa) \mathbf{R}_y(\varphi) \mathbf{R}_x(\omega) \quad (2)$$

with the elementary rotation matrices:

$$\mathbf{R}_x(\omega) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\omega & -\sin\omega \\ 0 & \sin\omega & \cos\omega \end{bmatrix}, \quad (3)$$

$$\mathbf{R}_y(\varphi) = \begin{bmatrix} \cos\varphi & 0 & \sin\varphi \\ 0 & 1 & 0 \\ -\sin\varphi & 0 & \cos\varphi \end{bmatrix} \text{ and} \quad (4)$$

$$\mathbf{R}_z(\kappa) = \begin{bmatrix} \cos\kappa & -\sin\kappa & 0 \\ \sin\kappa & \cos\kappa & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (5)$$

The transformation parameters are grouped as

$$\boldsymbol{\theta} = \begin{bmatrix} t_x & t_y & t_z & \omega & \varphi & \kappa \end{bmatrix}. \quad (6)$$

No scale factor is applied, since the LiDAR and LT measure in metric units. The concatenation of the two transformations leads to

$$\mathbf{p}_{l,i} = \mathbf{t}_b^l + \mathbf{R}_b^l \left(\mathbf{t}_s^b + \mathbf{R}_s^b \mathbf{p}_{s,i} \right). \quad (7)$$

The original frame is noted in the subscript, while the target frame is noted in the superscript. The last step for the functional model is the computation of the euclidean distances to the reference panels. Each point i of a position k is assigned to a reference panel j. The reference panels are parameterized using Hesse form with $\mathbf{a} = \begin{bmatrix} \mathbf{n}^T \; d \end{bmatrix}^T$ and $\mathbf{n} = [n_x \; n_y \; n_z]^T$. This leads to the computation of the misclosures w_i with

$$w_i = \mathbf{p}_{l,i}^T \mathbf{n}_j - d_j = h_i \left(\boldsymbol{\ell}_i, \boldsymbol{\theta}_s^b, d_0, \boldsymbol{\theta}_{b,k}^l, \mathbf{a}_j \right) \stackrel{!}{=} 0. \quad (8)$$

The parameter vector $\mathbf{x}$ of the adjustment can be defined in two different ways. Usually, the parameters of the transformation between the body frame and the superordinate frame, and the plane parameters are assumed to be deterministic, due to the higher measurement accuracy of the sensor used for the reference measurements. This leads to the parameter vector

$$\mathbf{x} = \left[\boldsymbol{\theta}_s^b, d_0\right]^T. \tag{9}$$

Alternatively, the parameters derived from the LT measurements can be introduced into the adjustment as stochastic values, too. This requires the introduction of restrictions for the estimation of the plane parameters in the Hesse form [19] and is not further addressed here.

The parameters and the observations cannot be separated in the functional model in (8), so the optimization has to be done in a Gauß-Helmert model (GHM) [21], [22]. The parameters and observations are estimated iteratively and require the computation of the Jacobian matrices of the functional model with respect to the parameters for $\mathbf{A}$ and to the observations for $\mathbf{B}$ using the approximated parameter vector $\mathbf{x}_0$ and observations vector $\boldsymbol{\ell}_0$,

$$\mathbf{A} = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\boldsymbol{\ell}_0,\mathbf{x}_0} \quad \text{and} \quad \mathbf{B} = \left.\frac{\partial \mathbf{h}}{\partial \boldsymbol{\ell}}\right|_{\boldsymbol{\ell}_0,\mathbf{x}_0}. \tag{10}$$

Additionally, the misclosure vector is computed with

$$\mathbf{w} = \mathbf{h}\left(\boldsymbol{\ell}_0, \mathbf{x}_0\right) + \mathbf{B}\left(\boldsymbol{\ell} - \boldsymbol{\ell}_0\right). \tag{11}$$

Using these quantities, the update to the parameters and the residuals $\hat{\mathbf{v}}$ can be computed using the least squares method with the covariance matrix of the misclosures $\mathbf{Q}_{\mathbf{ww}}$ and the covariance matrix of the parameters $\mathbf{Q}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}$

$$\mathbf{Q}_{\mathbf{ww}} = \mathbf{B}\mathbf{Q}_{\boldsymbol{\ell}\boldsymbol{\ell}}\mathbf{B}^T, \tag{12}$$

$$\mathbf{Q}_{\hat{\mathbf{x}}\hat{\mathbf{x}}} = \left(\mathbf{A}^T\mathbf{Q}_{\mathbf{ww}}^{-1}\mathbf{A}\right)^{-1}, \tag{13}$$

$$\hat{\mathbf{x}} = \mathbf{x}_0 + \mathbf{Q}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}\mathbf{A}^T\mathbf{Q}_{\mathbf{ww}}^{-1}(-\mathbf{w}), \tag{14}$$

$$\hat{\mathbf{v}} = \mathbf{Q}_{\boldsymbol{\ell}\boldsymbol{\ell}}\mathbf{B}^T\mathbf{Q}_{\mathbf{ww}}^{-1}\left(\mathbf{I} - \mathbf{A}\mathbf{Q}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}\mathbf{A}^T\right)\mathbf{Q}_{\mathbf{ww}}^{-1}(-\mathbf{w}). \tag{15}$$

For the iteration, the approximated vectors are updated using

$$\mathbf{x}_0 = \hat{\mathbf{x}} \quad \text{and} \quad \boldsymbol{\ell}_0 = \boldsymbol{\ell} + \hat{\mathbf{v}}. \tag{16}$$

The convergence criterion for the iteration is set to 10^{-10} as the threshold for the non-linear functional model in (8) computed with $\mathbf{x}_0$ and $\boldsymbol{\ell}_0$.

An important influence on the results of the adjustment is the stochastic model, which is defined by the covariance matrix of the observations $\mathbf{Q}_{\boldsymbol{\ell}\boldsymbol{\ell}}$ (setup shown below). The problem with the used sensor here, and with many other LiDARs, is, that the uncertainty budget of the measurements are not well known. The manufacturer provides a value of $\pm$ 3 cm for the distance measurement [16]. It is not clear if this value is a standard deviation or an interval. Furthermore, there is no information given for the accuracy of angle measurements.

A VCE is employed to estimate a better approximation for the stochastic model of the LiDAR. In this procedure, the residuals of the adjustment are used to update the stochastic model. Afterwards, the adjustment is repeated. This leads to another iteration, which ends when the update for the stochastic model is smaller than a threshold (here: 10^{-4}).

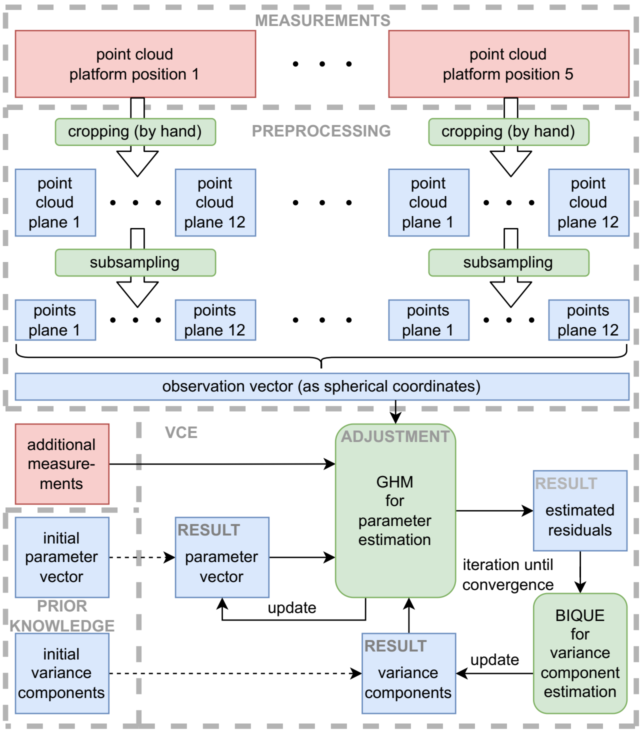

Fig. 3. Flow chart of the processing of measurement data for the system calibration including preprocessing and the adjustment with GHM and VCE

The chosen algorithm for the VCE is the Best Invariant Quadratic Unbiased Estimator (BIQUE) [23], [24]. The update of the covariance matrix of the observations is carried out using the z disjunct variance components $\mathbf{V}_i$ (here: three, one per measurement element) and the update vector $\Delta\boldsymbol{\sigma}$

$$\mathbf{Q}_{\boldsymbol{\ell}\boldsymbol{\ell}} = \sum_{i=1}^{z} \Delta\sigma_i \mathbf{V}_i \quad \text{with} \quad \Delta\boldsymbol{\sigma} = \mathbf{S}^{-1}\mathbf{q}. \tag{17}$$

The quantities $\mathbf{S}$ and $\mathbf{q}$ are computed element-wise. For the matrix $\mathbf{S}$, the indices describe the rows r and columns c, and for the vector $\mathbf{q}$, the indice i iterate over the three measurement elements,

$$q_i = \hat{\mathbf{v}}^T\mathbf{B}^T\mathbf{Q}_{\mathbf{ww}}^{-1}\mathbf{B}\mathbf{V}_i\mathbf{B}^T\mathbf{Q}_{\mathbf{ww}}^{-1}\mathbf{B}\hat{\mathbf{v}}, \tag{18}$$

$$\mathbf{M} = \mathbf{Q}_{\mathbf{ww}}^{-1} - \mathbf{Q}_{\mathbf{ww}}^{-1}\mathbf{A}\mathbf{Q}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}\mathbf{A}^T\mathbf{Q}_{\mathbf{ww}}^{-1}, \tag{19}$$

$$s_{r,c} = \operatorname{trace}\left(\mathbf{M}\mathbf{B}\mathbf{V}_c\mathbf{B}^T\mathbf{M}\mathbf{B}\mathbf{V}_r\mathbf{B}^T\right). \tag{20}$$

A flow chart showing the steps for the processing and computation is shown in Fig. 3. First, the acquired point clouds are cropped by hand to assign the points to the corresponding reference panels. The cropped point clouds are then subsampled for the aforementioned reasons. For the setup of the observation vector, all points are transformed back to spherical coordinates. The estimation itself consists of two previously described nested iterations, which require initial

TABLE I

RESULT FOR THE ESTIMATED PARAMETERS WITH THE CORRESPONDING STANDARD DEVIATIONS

parameter	value	std dev.
t_x	221.1 mm	1.80 mm
t_y	188.7 mm	5.98 mm
t_z	89.2 mm	1.44 mm
ω	-1.471°	0.060°
φ	0.013°	0.017°
κ	-0.007°	0.014°
d_0	-14.0 mm	0.20 mm

start values for the parameter vector and the stochastic model. The additional measurements for the positions of the platform, the parameters of the reference panels, and possibly the initial parameters for the transformation are derived from the LT measurements.

IV. RESULTS

The results are shown in two aspects. First, the results of the adjustments are shown and discussed. Afterward, the determined parameters are evaluated in an independent experiment using a robot with the calibrated MSS platform. The parameters from the proposed method are compared to parameters determined by different variants for the system calibration.

A. *Results of the Adjustment*

The determined parameters with their estimated standard deviations are shown in Table I.

While the actual values of the transformation parameters are specific to the platform, the standard deviations can be used to analyze the chosen setup. The standard deviations for the translation along the Y-axis and the rotation ω around the X-axis are noteworthy. Both of these standard deviations are three to four times larger compared to the standard deviations for the other translations and rotations. This is caused by the orientation of the platform during the measurement, which is always pointing downwards during the calibration, as shown in Fig. 1. This leads to a constant distance to the panels mounted on the ceiling with about 3 m. As mentioned before, longer distances are better for the determination of the orientation, so this causes the higher standard deviation for ω. The opposite reasoning is the cause for the higher standard deviations for t_y. The distance to the ceiling is longer than the minimum distance to the panels mounted on the side walls leading to higher standard deviations.

More findings can be obtained by investigating the sensitivities of the different parameters with regard to the reference panels. The orientations and positions of the panels are the main influence. Since the platform is always oriented similarly during the calibration, the sensitivities are also similar for all positions. An overview of the sensitivities is given in Table II. The distance offset d_0 is not listed, since every observation contributes to its estimation.

It is obvious that different panels contribute differently to the estimation of the transformation parameters. Generally,

TABLE II

SENSITIVITY OF THE TRANSFORMATION PARAMETERS TOWARDS THE REFERENCE PANELS, +: HIGH SENSITIVITY, O: SOME SENSITIVITY

panel	$\mathbf{t_x}$	$\mathbf{t_y}$	$\mathbf{t_z}$	$\boldsymbol{\omega}$	$\boldsymbol{\varphi}$	$\boldsymbol{\kappa}$
01			o		+	
02	+					
03			o		+	
04	o	o				+
05			o	+		
06		+				
07			o	+		
08	o	o				+
09			o		+	
10	+					
11			o		+	
12		+				

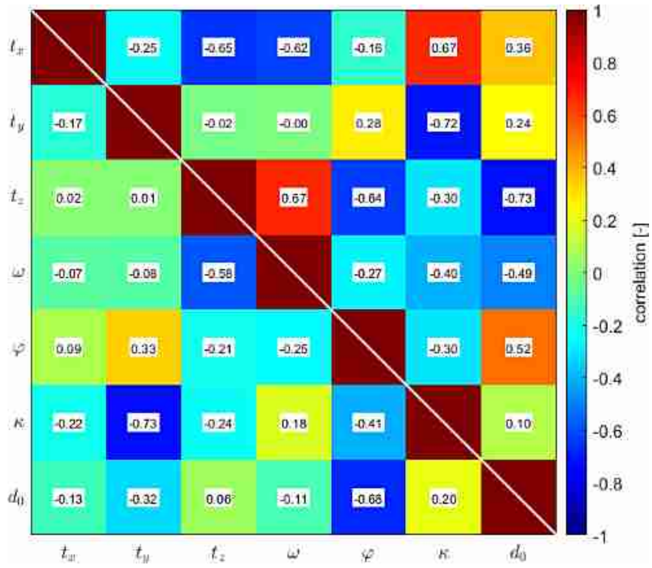

Fig. 4. Comparison of correlations of estimated parameters using one position (upper triangular matrix) and using all five positions (lower triangular matrix)

panels facing the sensors are useful for determining the translations, whereas tilted panels are useful to determine the rotations. The translation along the Z-axis is an exception, because the VLP-16 scans in the X-Y plane $\pm$ 15°, so the translation t_z is also dependent on tilted panels. The maximum tilt is set to 45°, since higher tilts lead to higher incidence angles, which decrease the measurement quality [10].

With the context of the sensitivities, the correlations of the estimated parameters can be discussed. Low correlations between the estimated parameter indicate good constellations for the adjustment. Fig. 4 shows the comparison of the correlations for the calibration with one position (only Pos1 from Fig. 2) in the upper triangular matrix and for all five positions in the lower triangular matrix. This illustration enables a compact form without losing information due to the symmetry of correlation matrices derived from the covariance matrix of the estimated parameters.

The comparison shows an improvement for most of the correlations by using multiple positions. The remaining notable correlations are:

- -0.58 between ω and t_z, which is caused by the short distances towards the ceiling influencing both parameters,
- -0.73 between κ and t_y, due to the reliance on the panels 04 and 08 for both parameters, and
- -0.68 between φ and d_0, which is the only correlation, that notably increases in magnitude. The rotation φ is determined by the panels 01, 03, 09, and 11, which are also critical for the determination of d_0 due to the changing distances from different positions.

While not possible for the platform used in this paper, the platform should be rotated around its Z-axis to provide longer distances along the Y-axis of the body frame for further calibrations. The rotation would reduce correlations between the parameters noted above and the longer distances enable a better determination of the parameters t_y and ω. With more degrees of freedom in the positioning of the platform, sensitivity charts like in Table II would have to be analyzed on a position-by-position basis.

To demonstrate the impact of the additional estimation of the distance offset, the adjustment can be computed with and without this model extension. The comparison is shown by the distances of the original point cloud transformed by the estimated parameters. This provides an impression of the difference in the achieved precision for the point cloud. Fig. 5 shows the misclosures without (top) and with (bottom) the model extension by the distance offset. The plot is divided into the different panels in blue and different positions in red. For reasons of readability, only the misclosures for positions Pos1 to Pos3 are shown, even though the parameters are determined using all five positions. The remaining two positions show similar results.

The normal vectors of the planes are oriented in such a way, that positive distances are caused by a positive bias in the distance measurements, i.e. the LiDAR measures too long distances. This can be seen quite frequently for the results without the model extension. The mean value of the distances significantly deviates from zero. Furthermore, systematic effects between the different panels can be seen, resulting in even higher deviations of the mean of distances. This can be seen especially for panels 10 and 11.

The comparison to the lower plot shows the positive impact of the model extension. The distance offset is estimated at -14.0 mm, which is in a similar magnitude to the results of other studies [2], [14]. The mean of the distances shows no clear deviation from zero anymore. The magnitude of the systematic deviations is significantly reduced. Even though the VCE optimizes the stochastic model based on the functional model, the dispersion of the distances is still, at least visually, no white noise, which indicates the necessity to further extend the intrinsic model of the VLP-16.

Finally, the results from the VCE are discussed. Similar to the correlations, the VCE benefits from the usage of multiple positions. Processing of only one position leads to negative values for the stochastic model. This indicates problems in the functional or stochastic model [24]. Joint processing of multiple positions stabilizes the estimation, leading to the valid values for the stochastic model, as shown in Table III with the standard deviations for the original observations. These values can be checked for plausibility by comparing them with other available information. For the distance measurement, the precision given by the standard deviation can be seen in the distance plots in Fig. 5. Furthermore, other investigations determine similar values for the precision of the distance measurement [14]. It should be emphasized, that the determined standard deviation for the distance measurements is only correct for measurements over a similar distance, i.e. up to 6 m, and only on white surfaces. For longer distances or darker surfaces, the standard deviation will increase.

The standard deviations for the angle measurements can be compared to the beam divergence for the VLP-16. The manufacturer provides a value of 3.0 mrad for the horizontal angle and 1.5 mrad for the vertical angle [16]. The tendency towards a larger value for the horizontal angle can be observed for both the beam divergence and the estimated standard deviation.

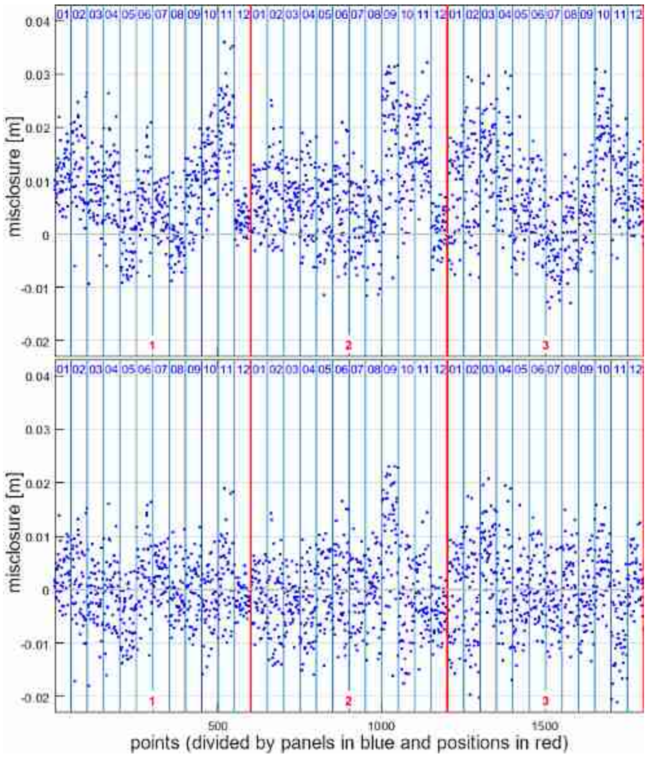

Fig. 5. Comparison of misclosures computed with the estimated parameters and the original observations without distance offset (top) and with distance offset (bottom), for positions Pos1 to Pos3

TABLE III
RESULT OF THE VARIANCE COMPONENT ESTIMATION

observation type	standard deviation
distance	8.5 mm
azimuth angle	0.0485°
elevation angle	0.0296°

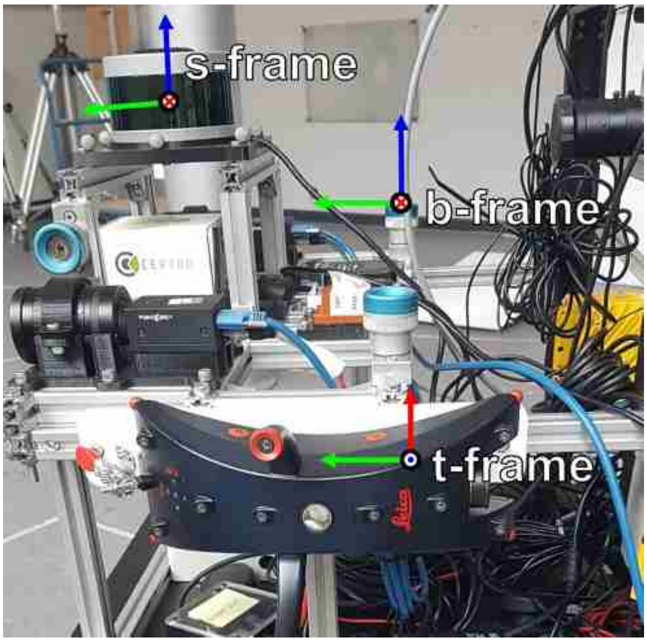

Fig. 6. Platform mounted on Husky UGV with T-Probe mounted to the side, only frames of relevant sensors are marked

B. Evaluation in an independent experiment

To evaluate the results, the parameters obtained with the model extension are compared to parameters obtained with different variants. The evaluation is based on a data set of a kinematic experiment performed in the same laboratory as the calibration to ensure similar conditions. For the experiment, a Husky UGV [25] is used to carry the calibrated platform. The robot is shown in Fig. 6 with the frames of the relevant sensors marked. The trajectory is determined by the LT, which tracks a T-Probe (t-frame) fixed to the platform to determine the 6-DoF pose at a measurement rate of 50 Hz.

The time synchronization between the LiDAR and LT is realized by separate GNSS receivers. The VLP-16 is connected to a Garmin 18x, while the LT receives a trigger signal by a Javad GNSS receiver. Hence, timestamps in GPS time are available for measurements of both sensors to synchronize them.

Fig. 7 show the setup for the experiment. The movement of the robot is limited by the viewing angle for the T-Probe. The shown trajectory takes about 40 s. The overview in Fig. 7 is based on the full-dome scan of a terrestrial laser scanner for illustration purposes.

Similar to the calibration, the evaluation is based on reference panels. In this case, the computed distances are used to evaluate the parameters. Smaller deviations indicate better solutions. The positions and orientations of the eleven reference panels in the room enable us to check all six degrees of freedom of the transformation. The comparison includes three variants for the determination of the parameters. The first variant is the measurement of the LiDAR housing with the reconstruction of the sensor frame based on the values provided by the manufacturer. The measurement is performed using the T-Probe stylus, so the resulting parameters are already quite accurate. The second variant is the system calibration estimating only the extrinsic parameters. The last variant is the proposed method estimating intrinsic and extrinsic parameters simultaneously.

TABLE IV
DEVIATIONS OF THE REFERENCED POINT CLOUDS OF THE KINEMATIC MEASUREMENTS COMPUTED USING DIFFERENT VARIANTS FOR THE CALIBRATION COMPARED TO THE REFERENCE PANELS

variant	RMSE [mm]
measurement housing	15.8
only extrinsic	14.7
intrinsic and extrinsic	10.3

The results of the evaluation are shown in Table IV. The values are listed as root mean square errors (RMSE) of the distances to the eleven reference panels. It can be seen, that the calibration solely for the extrinsic parameters is only slightly better than measuring the housing. This is caused by the influence of the uncorrected distance measurements during the calibration, compromising the results. Significantly better results are obtained when the intrinsic and extrinsic parameters are estimated jointly. The correction of the distance offset leads to improved results for calibration and to a more accurate point cloud during the kinematic experiment.

V. Summary and Outlook

In this paper, a model extension for a system calibration approach was shown to additionally estimate intrinsic parameters of the calibrated LiDAR. The estimated distance offset improves the results for the adjustment and for the application of the parameters in an independent measurement. In this case, the evaluation using a comparison to a reference point cloud shows a reduction of the RMSE of distances from 14.7 mm to 10.3 mm. Furthermore, the performed VCE delivers an improved stochastic model, which needs to be further validated but can be seen as plausible.

The principle of calibration offers multiple aspects for further optimizations. The choice of positions can be further improved by expert knowledge like discussed in section IV or using optimization algorithms similar to [10]. The functional model used in this paper is applicable to any profile laser scanner or multi-beam LiDAR. Further improvements are possible by adapting the intrinsic model to the sensor. For multi-beam LiDARs, like the VLP-16, estimating a distance offset per ring might improve the results further. Depending on the sensor, even more sophisticated models for the intrinsic calibration exist [14]. In general, it is to be investigated if more complex intrinsic models can still be estimated simultaneously to the extrinsic parameters, or if a two-step process would be necessary.

Acknowledgment

The authors would like to thank Aaronkumar Ehambram and Raphael Voges of the Real Time Systems Group (RTS) of

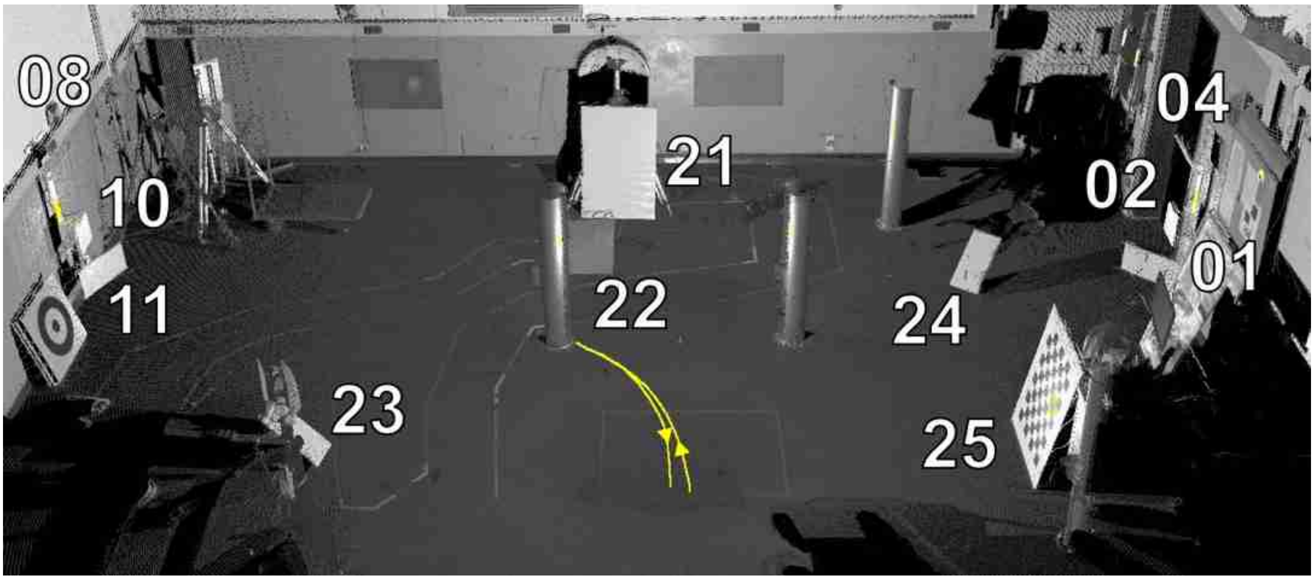

Fig. 7. Overview scan of the kinematic measurement with numbers of the reference panels and the movement of the robot in yellow

the Leibniz University Hannover for providing access to the robot and the support with the measurements.

References

[1] T. O. Chan, D. D. Lichti, G. Roesler, D. Cosandier, and K. Al-Durgham, "Range scale-factor calibration of the velodyne vlp-16 lidar system for position tracking applications," in *2019 11th International Conference on Mobile Mapping*, 2019, pp. 70–77.

[2] D. E. Sánchez and T. Pany, "Analysis and calibration of the vlp-16 lidar for automotive applications," in *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*, 2019, pp. 2611–2621.

[3] 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*, vol. 87, no. 5-6, pp. 229–247, 2019.

[4] A. Schlichting, *Fahrzeuglokalisierung durch automotive Laserscanner unter Verwendung statischer Merkmale: Ph.D. Thesis*, ser. Reihe C. München: DGK, 2018, vol. 826.

[5] 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*, vol. 4, pp. 35–42, 2019.

[6] P. Geneva, K. Eckenhoff, Y. Yang, and G. Huang, "Lips: Lidar-inertial 3d plane slam," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018.

[7] L. Zhou, Z. Li, and M. Kaess, "Automatic extrinsic calibration of a camera and a 3d lidar using line and plane correspondences," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 5562–5569.

[8] B. Shi, P. Yu, M. Yang, C. Wang, Y. Bai, and F. Yang, "Extrinsic calibration of dual lidars based on plane features and uncertainty analysis," *IEEE Sensors Journal*, vol. 21, no. 9, pp. 11 117–11 130, 2021.

[9] T. Strübing and I. Neumann, "Positions-und orientierungsschätzung von lidar-sensoren auf multisensorplattformen," *ZfV*, vol. 3, pp. 210–221, 2013.

[10] J. Hartmann, I. von Gösseln, N. Schild, A. Dorndorf, J.-A. Paffenholz, and I. Neumann, "Optimisation of the calibration process of a k-tls based multi-sensor-system by genetic algorithms," in *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Copernicus GmbH, 2019, pp. 1655–1662.

[11] E. Heinz, C. Eling, M. Wieland, L. Klingbeil, and H. Kuhlmann, "Development, calibration and evaluation of a portable and direct georeferenced laser scanning system for kinematic 3d mapping," *Journal of Applied Geodesy*, vol. 9, no. 4, pp. 227–243, 2015.

[12] M. H. Shahraji and C. Larouche, "Case study: Rigorous boresight alignment of a marine mobile lidar system addressing the specific demands of port infrastructure monitoring," *Marine Geodesy*, pp. 1–18, 2022.

[13] S. Kalenjuk and W. Lienhart, "A method for efficient quality control and enhancement of mobile laser scanning data," *Remote Sensing*, vol. 14, no. 4, p. 857, 2022.

[14] C. L. Glennie, A. Kusari, and A. Facchin, "Calibration and stability analysis of the vlp-16 laser scanner," *2194-9034*, vol. XL-3/W4, pp. 55–60, 2016.

[15] C. Cortes, M. Shahbazi, and P. Ménard, "Uav-licam system development: Calibration and geo-referencing," *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 42, no. 1, 2018.

[16] Velodyne, "Vlp-16 manual: User's manual and programming guide," San Jose, CA, USA, 2019.

[17] Hexagon Metrology, "Leica absolute tracker at960," 2021. [Online]. Available: https://www.hexagonmi.com/en/products/laser-tracker-systems/leica-absolute-tracker-at960

[18] T. Luhmann, *Nahbereichsphotogrammetrie*. Wichmann Verlag, 2000.

[19] S. Vogel, D. Ernst, I. Neumann, and H. Alkhatib, "Recursive gauss-helmert model with equality constraints applied to the efficient system calibration of a 3d laser scanner," *Journal of Applied Geodesy*, pp. 1–21, 2021.

[20] Zoller+Fröhlich, "Z+f imager® 5016: Zoller+fröhlich," 2022. [Online]. Available: https://www.zofre.de/laserscanner/3d-laserscanner/z-f-imagerr-5016

[21] L. Lenzmann and E. Lenzmann, "Strenge auswertung des nichtlinearen gauß-helmert-modells," *allgemeine vermessungs-nachrichten (avn)*, no. 2, pp. 68–73, 2004.

[22] K.-R. Koch, "Räumliche helmert-transformation variabler koordinaten im gauß-helmert- und im gauß-markoff-modell," *ZfV*, vol. 127, no. 3, pp. 147–152, 2002.

[23] Z. Yu, "A generalization theory of estimation of variance-covariance components," *Manuscripta geodaetica*, vol. 17, no. 5, p. 295, 1992.

[24] A. Heiker, *Mutual validation of Earth orientation parameters, geophysical excitation functions and second degree gravity field coefficients*, ser. Reihe C. München: DGK, 2013, vol. 697.

[25] Clearpath Robotics, "Husky unmanned ground vehicle robot," 2021. [Online]. Available: https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/

Error State Kalman Filter with Implicit Measurement Equations for Position Tracking of a Multi-Sensor System with IMU and LiDAR

Dominik Ernst
Geodetic Institute
Leibniz University Hannover
Hannover, Germany
0000-0001-9826-5610

Sören Vogel
Geodetic Institute
Leibniz University Hannover
Hannover, Germany
0000-0003-1288-6139

Ingo Neumann
Geodetic Institute
Leibniz University Hannover
Hannover, Germany
0000-0001-9110-7345

Hamza Alkhatib
Geodetic Institute
Leibniz University Hannover
Hannover, Germany
0000-0002-4480-1067

***Abstract*—With many applications requiring individuals to share spaces with autonomous systems, not only do accurate positioning solutions become crucial, but also understanding of the uncertainty associated with these systems. For applications like public transportation or logistics, position tracking algorithms need to be evaluated on accuracy and consistency. The selection of the appropriate algorithms heavily relies on the specific requirements of the applications, thus demanding novel algorithms for these new use cases.**

This paper introduces a novel error state Kalman filter with implicit measurement equations, presenting a solution for new applications. The filter facilitates the fusion of inertial measurement unit (IMU) sensor data with other sensors using arbitrary measurement models. To showcase its effectiveness, the filter is demonstrated by fusing simulated IMU and LiDAR observations for position tracking (complete code on Github). Specifically, the LiDAR points are employed in the update step by minimizing the distances to known planes. Furthermore, the performance of the filter is enhanced by motion compensation through pose interpolation, utilizing the available timestamps. The results are thoroughly discussed in terms of accuracy and consistency, revealing significant improvements by employing pose interpolation. However, the consistency analysis indicates slightly pessimistic results, suggesting the need for further optimizations.

***Index Terms*—Error state Kalman filter, multi-sensor system, position tracking, IMU, LiDAR, simulation.**

I. Introduction

Accurate determination for the pose of a system over time is essential for numerous applications. When the initial pose is known, it is referred to as position tracking. Especially if persons are sharing spaces with autonomous systems, such as warehouses or construction sites, the position tracking of these systems needs to not only be accurate but also aware of the uncertainty associated with their estimated pose. This concept of integrity is already implemented in aviation and is in progress of being expanded to other fields, including automotive, maritime or rail [1]. This expands the evaluation of algorithms for position tracking from just comparing the deviations between estimated and true trajectory to additionally investigating the estimated pose uncertainty of the system. Unfortunately, this aspect is often overlooked.

This work was funded by the German Research Foundation (DFG) as part of the Research Training Group i.c.sens [RTG 2159]. Furthermore, this work was supported by the LUH compute cluster, which is funded by the Leibniz Universität Hannover, the Lower Saxony Ministry of Science and Culture (MWK) and the German Research Association (DFG).

A. Related Works

Position tracking can be accomplished using different techniques, each with its own advantages and drawbacks. Some possibilities include filters [2], [3], sliding-window adjustments [4], [5] or factor graph optimizations (FGOs) [6], [7]. This paper focuses on filters, specifically, Kalman filters, which are widely used and computationally efficient [8]. Although the assumptions of normal distribution restrict the Kalman filter in terms of the distributions it can handle, new algorithms enable different state representation, such as error states [9], or more versatile functional models for novel use cases or sensor combinations [10], [11].

Additional map information can be integrated into position tracking. Various approaches are found in the literature, including the use of landmarks [12], [13], map information from city models [14], [15] or active sensors like ultra wide band beacons as anchors [16]. In this context, the required measurement models might not be representable as an explicit function, which is required by most Kalman filter formulations. This may be circumvented using loosely coupled integration of sensor data by preprocessing observations [17], [18], which might instead lead to a loss of information. Depending on the application, the system's orientations with respect to the superordinate frame might also be arbitrary. This can be the case for hand-held systems or global positioning in an Earth Centered Earth Fixed (ECEF) frame.

B. Contributions

This paper introduces a novel and versatile filter that combines these aspects. The fundamental basis of the contribution is a error state Kalman filter (ESKF) with implicit measurement equations for the update step. This formulation enables a tightly-coupled fusion of the inertial measurement unit (IMU) observations in the state prediction and light detection and ranging (LiDAR) observations in the update step. The update is computed by assigning the points to known planes in the environment and minimizing the distance to update the pose. To improve the transformation of the point clouds for the assignment, linear interpolation of the pose is performed to compensate for the rolling shutter effect caused by LiDAR rotation during motion. This significantly enhances the filtering performance compared to neglecting the motion between the IMU measurements. The results are evaluated based on data from a simulated environment and discussed in terms of accuracy and consistency. Consistency in the Kalman filter refers to the property of the filter's estimation being in agreement with the true system behavior and the statistical properties of the measurements.

C. Outline

The structure of the paper is as follows: Section II describes the methodology, including the filter formulation and the novel update step for position tracking. Section III presents the simulation environment used for the experiments. The results are then presented and discussed in Section IV, followed by the conclusion in Section V.

II. Methodology

This section presents the formulation of ESKF using quaternions, based on [19]. It begins by introducing the states and the prediction step, followed by the new update step for implicit measurement equations with error states and the principle of position tracking for the system.

A. States

The state vector for the filter is defined as:

$$\boldsymbol{x} = \begin{bmatrix} {\boldsymbol{t}_\mathrm{b}^\mathrm{l}}^T & {\boldsymbol{v}_\mathrm{b}^\mathrm{l}}^T & {\boldsymbol{q}_\mathrm{b}^\mathrm{l}}^T & \boldsymbol{a}_b^T & \boldsymbol{\omega}_b^T \end{bmatrix}^T . \tag{1}$$

This state vector consists of the following elements with their sizes indicted in brackets and relating from the body frame b to the local frame l

- $\boldsymbol{t}_\mathrm{b}^\mathrm{l}$: position, translation for transformations [3×1],
- $\boldsymbol{v}_\mathrm{b}^\mathrm{l}$: velocity [3×1],
- $\boldsymbol{q}_\mathrm{b}^\mathrm{l}$: orientation, rotation for transformations [4×1],
- $\boldsymbol{a}_b$: bias of accelerometer [3×1],
- $\boldsymbol{\omega}_b$: bias of gyroscope [3×1].

To avoid gimbal lock [20], the orientation is represented using quaternions $\boldsymbol{q}$. Quaternions multiplication is shown by $\otimes$, vector additions and subtraction including quaternions are represented by $\boxplus$ and $\boxminus$, respectively. Conversions between different representations of rotations (rotation matrix $\boldsymbol{R}$ or rotation vector $\boldsymbol{\theta} = [\phi\ \boldsymbol{r}]$ with $\boldsymbol{r} = [r_x\ r_y\ r_z]$) are indicated by $\boldsymbol{q}\{\cdot\}$. The error state is denoted by δ and is defined as:

$$\delta\boldsymbol{x} = \begin{bmatrix} \delta{\boldsymbol{t}_\mathrm{b}^\mathrm{l}}^T & \delta{\boldsymbol{v}_\mathrm{b}^\mathrm{l}}^T & \delta{\boldsymbol{\theta}_\mathrm{b}^\mathrm{l}}^T & \delta\boldsymbol{a}_b^T & \delta\boldsymbol{\omega}_b^T \end{bmatrix}^T . \tag{2}$$

The error in the orientation is described by the rotation vector $\delta\boldsymbol{\theta}_\mathrm{b}^\mathrm{l}$ with a size of [3×1].

B. Prediction

The prediction step integrates the measurements of an IMU $\boldsymbol{u}_o^{(k)}$ to propagate the pose of the system to the next epoch (k)

$$\boldsymbol{x}^{(k)} = f\left(\boldsymbol{x}^{(k-1)}, \boldsymbol{u}_o^{(k)}\right). \tag{3}$$

The available measurements o are usually obtained by a three-axis accelerometer $\boldsymbol{a}$ and a three-axis gyroscope $\boldsymbol{\omega}$:

$$\boldsymbol{\omega}_o = \boldsymbol{\omega} + \boldsymbol{\omega}_b + \boldsymbol{\omega}_n, \tag{4}$$

$$\boldsymbol{a}_o = \boldsymbol{a} + \boldsymbol{a}_b + \boldsymbol{a}_n + \left(\boldsymbol{R}_\mathrm{b}^\mathrm{l}\right)^T \boldsymbol{g}_\mathrm{l} \tag{5}$$

with $\boldsymbol{R}_\mathrm{b}^\mathrm{l} = \boldsymbol{R}\left\{\boldsymbol{q}_\mathrm{b}^\mathrm{l}\right\}$. The subscript n denotes the noise afflicting the measurements. The measurement of the accelerometer is additionally corrected for the gravity vector $\boldsymbol{g}_\mathrm{l}$. The corrected IMU measurements are $\hat{\boldsymbol{a}} = \boldsymbol{a}_o - \boldsymbol{a}_b$ and $\hat{\boldsymbol{\omega}} = \boldsymbol{\omega}_o - \boldsymbol{\omega}_b$.

The prediction is given as differential equations by:

$$\delta\dot{\boldsymbol{t}} = \delta\boldsymbol{v}, \tag{6}$$

$$\delta\dot{\boldsymbol{v}} = -\boldsymbol{R}_\mathrm{b}^\mathrm{l}\left[\hat{\boldsymbol{a}}\right]_\times \delta\boldsymbol{\theta}_\mathrm{b}^\mathrm{l} - \boldsymbol{R}_\mathrm{b}^\mathrm{l}\delta\boldsymbol{a}_b + \boldsymbol{g}_\mathrm{l} - \boldsymbol{R}_\mathrm{b}^\mathrm{l}\boldsymbol{a}_n, \tag{7}$$

$$\delta\dot{\boldsymbol{\theta}} = -\boldsymbol{R}_\mathrm{b}^\mathrm{l}\delta\omega_b - \boldsymbol{R}_\mathrm{b}^\mathrm{l}\boldsymbol{\omega}_n, \tag{8}$$

$$\delta\dot{\boldsymbol{a}}_b = \boldsymbol{a}_w, \tag{9}$$

$$\delta\dot{\boldsymbol{\omega}}_b = \boldsymbol{\omega}_w, \tag{10}$$

with $[\cdot]_\times$ as the skew operator and subscript w for the bias instability. For the discrete time with the time difference to the previous epoch Δt, the prediction can be expressed as a matrix

$$\boldsymbol{F}_{\boldsymbol{x}} = \left.\frac{\partial f}{\partial \delta\boldsymbol{x}}\right|_{\boldsymbol{x},\boldsymbol{u}_m} \tag{11}$$

$$= \begin{bmatrix} \boldsymbol{I} & \boldsymbol{I}\Delta\mathrm{t} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & -\left[\boldsymbol{R}_\mathrm{b}^\mathrm{l}\hat{\boldsymbol{a}}\right]_\times\Delta\mathrm{t} & -\boldsymbol{R}_\mathrm{b}^\mathrm{l}\Delta\mathrm{t} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & -\boldsymbol{R}_\mathrm{b}^\mathrm{l}\Delta\mathrm{t} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}.$$

The prediction of the variance covariance matrix (VCM) of the states requires information about the properties of the IMU denoted as

$$\boldsymbol{Q_i} = \operatorname{diag}\left(\boldsymbol{\sigma}_{\tilde{a}_n}^2\Delta\mathrm{t}^2\boldsymbol{I}, \boldsymbol{\sigma}_{\tilde{\omega}_n}^2\Delta\mathrm{t}^2\boldsymbol{I}, \boldsymbol{\sigma}_{a_w}^2\Delta\mathrm{t}\boldsymbol{I}, \boldsymbol{\sigma}_{\omega_w}^2\Delta\mathrm{t}\boldsymbol{I}\right) \tag{12}$$

using the values

- $\boldsymbol{\sigma}_{\tilde{a}_n}$: noise density of accelerometer [3×1],
- $\boldsymbol{\sigma}_{\tilde{\omega}_n}$: noise density of gyroscope [3×1],
- $\boldsymbol{\sigma}_{a_w}$: bias instability of accelerometer [3×1],
- $\boldsymbol{\sigma}_{\omega_w}$: bias instability of gyroscope [3×1].

Additionally, the matrix $\boldsymbol{F}_{\boldsymbol{i}}^{(k)}$ is necessary to transform the perturbations accordingly:

$$\boldsymbol{F}_{\boldsymbol{i}}^{(k)} = \begin{bmatrix} & \boldsymbol{0}_{[3\times12]} & & \\ \operatorname{diag}(-\boldsymbol{R}_\mathrm{b}^\mathrm{l} & -\boldsymbol{I} & \boldsymbol{I} & \boldsymbol{I}) \end{bmatrix}. \tag{13}$$

This yields the prediction equations for discrete time between the epochs $(k-1)$ and (k)

$$\hat{\boldsymbol{x}}_-^{(k)} = \boldsymbol{F}_{\boldsymbol{x}}^{(k)}\hat{\boldsymbol{x}}^{(k-1)}, \tag{14}$$

$$\boldsymbol{Q}_{\hat{x}\hat{x},-}^{(k)} = \boldsymbol{F}_{\boldsymbol{x}}^{(k)}\boldsymbol{Q}_{\hat{x}\hat{x}}^{(k-1)}\left(\boldsymbol{F}_{\boldsymbol{x}}^{(k)}\right)^T + \boldsymbol{F}_{\boldsymbol{i}}^{(k)}\boldsymbol{Q}_{\boldsymbol{i}}\left(\boldsymbol{F}_{\boldsymbol{i}}^{(k)}\right)^T. \tag{15}$$

C. Update

The update computation for the state vector is performed using the iterated extended Kalman Filter (iEKF) with implicit measurement equations, as described in [10], [21]. This algorithm is adapted for error states. The iterative update process begins by initializing the current as states as the predictions, the error states and the original observations as the current values:

$$\hat{\boldsymbol{x}}_{+}^{(k,m=1)} = \hat{\boldsymbol{x}}_{-}^{(k)}, \quad \delta\boldsymbol{x}^{(k,m=0)} = \mathbf{0} \quad \text{and} \quad \hat{\boldsymbol{\ell}}^{(k,m=1)} = \boldsymbol{\ell}^{(k)}.$$

Here m represents the iteration number of the update. Subsequently, the following algorithm is computed using the VCM of the observations from the current epoch $\boldsymbol{Q}_{\boldsymbol{\ell\ell}}^{(k)}$:

$$\boldsymbol{h}^{(k,m)} = \boldsymbol{h}\left(\hat{\boldsymbol{x}}_{+}^{(k,m)}, \hat{\boldsymbol{\ell}}^{(k,m)}\right) \tag{16}$$

$$\boldsymbol{A}^{(k,m)} = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_{+}^{(k,m)}, \hat{\boldsymbol{\ell}}^{(k,m)}} \cdot \left.\frac{\partial \boldsymbol{x}}{\partial \delta\boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_{+}^{(k,m)}} \tag{17}$$

$$\boldsymbol{B}^{(k,m)} = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\ell}}\right|_{\hat{\boldsymbol{x}}_{+}^{(k,m)}, \hat{\boldsymbol{\ell}}^{(k,m)}} \tag{18}$$

$$\boldsymbol{O}^{(k,m)} = \boldsymbol{A}^{(k,m)}\, \boldsymbol{Q}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}},-}^{(k)} \left(\boldsymbol{A}^{(k,m)}\right)^{T} \tag{19}$$

$$\boldsymbol{S}^{(k,m)} = \boldsymbol{B}^{(k,m)}\, \boldsymbol{Q}_{\boldsymbol{\ell\ell}}^{(k)} \left(\boldsymbol{B}^{(k,m)}\right)^{T} \tag{20}$$

$$\boldsymbol{Q}_{\boldsymbol{rr}}^{(k,m)} = \boldsymbol{O}^{(k,m)} + \boldsymbol{S}^{(k,m)} \tag{21}$$

$$\boldsymbol{K}^{(k,m)} = \boldsymbol{Q}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}},-}^{(k)} \left(\boldsymbol{A}^{(k,m)}\right)^{T} \left(\boldsymbol{Q}_{\boldsymbol{rr}}^{(k,m)}\right)^{-1} \tag{22}$$

$$\boldsymbol{r}^{(k,m)} = \boldsymbol{B}^{(k,m)} \left(\boldsymbol{\ell}^{(k,m)} - \boldsymbol{\ell}^{(k)}\right) + \boldsymbol{A}^{(k,m)} \delta\boldsymbol{x}^{(k,m-1)} \tag{23}$$

$$\boldsymbol{w}^{(k,m)} = \boldsymbol{h}^{(k,m)} + \boldsymbol{r}^{(k,m)} \tag{24}$$

$$\delta\boldsymbol{x}^{(k,m)} = -\boldsymbol{K}^{(k,m)} \boldsymbol{w}^{(k,m)}. \tag{25}$$

After the error state is computed, the state can be updated by injecting the error state into the predicted state

$$\hat{\boldsymbol{x}}_{+}^{(k,m+1)} = \hat{\boldsymbol{x}}_{-}^{(k)} \boxplus \delta\boldsymbol{x}^{(k,m)} = \begin{bmatrix} \boldsymbol{t}_{\mathrm{b}-}^{\mathrm{l}} + \delta\boldsymbol{t}_{\mathrm{b}}^{\mathrm{l}} \\ \boldsymbol{v}_{\mathrm{b}-}^{\mathrm{l}} + \delta\boldsymbol{v}_{\mathrm{b}}^{\mathrm{l}} \\ \boldsymbol{q}\left\{\delta\boldsymbol{\theta}\right\} \otimes \boldsymbol{q}_{\mathrm{b}-}^{\mathrm{l}} \\ \boldsymbol{b}_{a-} + \delta\boldsymbol{b}_{a} \\ \boldsymbol{b}_{\omega-} + \delta\boldsymbol{b}_{\omega} \end{bmatrix}. \tag{26}$$

The iteration continues until the change of the error state between iterations is smaller than a chosen threshold $\|\delta\mathbf{x}^{(k,m+1)} \boxminus \delta\mathbf{x}^{(k,m)}\| < \epsilon$. If the iteration continues, the updated observations are computed

$$\hat{\boldsymbol{\ell}}^{(k,m+1)} = \boldsymbol{\ell}^{(k)} - \boldsymbol{Q}_{\boldsymbol{\ell\ell}}^{(k)} \left(\boldsymbol{B}^{(k,m)}\right)^{T} \left(\boldsymbol{Q}_{\boldsymbol{rr}}^{(k,m)}\right)^{-1} \boldsymbol{w}^{(k,m)}. \tag{27}$$

After the end of the iteration, the VCM of the states can be updated with

$$\boldsymbol{L}^{(k)} = \boldsymbol{I} - \boldsymbol{K}^{(k,m)} \boldsymbol{A}^{(k,m)}, \tag{28}$$

$$\boldsymbol{Q}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}},+}^{(k)} = \boldsymbol{L}^{(k)} \boldsymbol{Q}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}},-}^{(k)} \left(\boldsymbol{L}^{(k)}\right)^{T} + \boldsymbol{K}^{(k,m)} \boldsymbol{S}^{(k,m)} \left(\boldsymbol{K}^{(k,m)}\right)^{T}. \tag{29}$$

The resulting state vector with the VCM is depending on the availability of observations in the epoch. If no observations are available in the epoch, the state vector for the epoch is the predicted state. Otherwise, the result of the update is used

$$\hat{\boldsymbol{x}}^{(k)}, \boldsymbol{Q}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}}^{(k)} = \begin{cases} \hat{\boldsymbol{x}}_{+}^{(k,m)}, \boldsymbol{Q}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}},+}^{(k)} & \text{with update in } (k), \\ \hat{\boldsymbol{x}}_{-}^{(k)}, \boldsymbol{Q}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}},-}^{(k)} & \text{else.} \end{cases} \tag{30}$$

D. Principle of Position Tracking

The simulated application in this paper utilizes an IMU for the prediction and a LiDAR for the update step. While the IMU is used in a common strapdown fashion, as explained in Section II-B, the update step using the LiDAR point clouds is discussed in more detail here.

The localization principle employed in this work was first introduced in [14] and has also been applied to different LiDARs in [22]. The main idea is to minimize the distance between the transformed point cloud in the local-frame (l) and the known planes $\boldsymbol{a}$ in the environment to update the pose of the system $\left[\boldsymbol{t}_{\mathrm{b}}^{\mathrm{l,\,(k,m)}}, \boldsymbol{q}_{\mathrm{b}}^{\mathrm{l,\,(k,m)}}\right]$, which is used for the transformation of the point clouds. To compensate for the system's motion during the acquisition of the points for an epoch, the transformation parameters are first interpolated based on the timestamps t_i of the individual points $\boldsymbol{p}_i$ converted to the interpolation factor

$$\tau_i = \frac{\mathrm{t}_i - \mathrm{t}^{(k-1)}}{\Delta \mathrm{t}} \tag{31}$$

following a similar approach to [2]. For the SLERP of the quaternions, the method described in [23] is used:

$$\boldsymbol{t}_{\mathrm{b}\tau_i}^{\mathrm{l}} = (1 - \tau_i)\, \hat{\boldsymbol{t}}_{\mathrm{b}}^{\mathrm{l,\,(k\text{-}1)}} + \tau_i\, \hat{\boldsymbol{t}}_{\mathrm{b}}^{\mathrm{l,\,(k,m)}}, \tag{32}$$

$$\Delta\theta = \arccos\left(\left(\hat{\boldsymbol{q}}_{\mathrm{b}}^{\mathrm{l,\,(k\text{-}1)}}\right)^{T} \cdot \hat{\boldsymbol{q}}_{\mathrm{b}}^{\mathrm{l,\,(k,m)}}\right), \tag{33}$$

$$\boldsymbol{q}_{\mathrm{b}\tau_i}^{\mathrm{l}} = \hat{\boldsymbol{q}}_{\mathrm{b}}^{\mathrm{l,\,(k\text{-}1)}} \frac{\sin\left((1-\tau_i)\Delta\theta\right)}{\sin(\Delta\theta)} + \hat{\boldsymbol{q}}_{\mathrm{b}}^{\mathrm{l,\,(k,m)}} \frac{\sin(\tau_i \Delta\theta)}{\sin(\Delta\theta)}. \tag{34}$$

Once, the interpolation is completed, the points can be transformed from the body-frame b to the l-frame. In the local frame, the distance computation to the planes $\boldsymbol{a}_{\mathrm{l},j}$ can be performed using the implicit measurement equation h_i

$$\boldsymbol{p}_{\mathrm{l},i} = \begin{bmatrix} x_{\mathrm{l}} \\ y_{\mathrm{l}} \\ z_{\mathrm{l}} \end{bmatrix}_i = \boldsymbol{t}_{\mathrm{b}\tau_i}^{\mathrm{l}} + \boldsymbol{R}\left\{\boldsymbol{q}_{\mathrm{b}\tau_i}^{\mathrm{l}}\right\} \begin{bmatrix} x_{\mathrm{b}} \\ y_{\mathrm{b}} \\ z_{\mathrm{b}} \end{bmatrix}_i, \tag{35}$$

$$h_i = h\left(\boldsymbol{x}, \boldsymbol{\ell}_i, \boldsymbol{a}_{\mathrm{l},j}\right) = \left[n_{x,\mathrm{l}}\, n_{y,\mathrm{l}}\, n_{z,\mathrm{l}}\, d_{\mathrm{l}}\right]_j \cdot \begin{bmatrix} \boldsymbol{p}_{\mathrm{l},i} \\ 1 \end{bmatrix}. \tag{36}$$

III. Simulation

In this paper, a simulation is employed to validate the proposed filter for the position tracking of a multi-sensor system (MSS), which includes an IMU and a LiDAR[1]. The availability of the true poses and the control over the assumptions enables definitive statements to be made regarding the performance of the filter.

[1]The code for the filter and simulation is available under github.com/DoErnst/ErrorStateKalmanFilter_implicitMeas. Additionally, the processing of a real data set is included (not shown here).

A. Environment

The evaluation of the filter is conducted in a simulated environment, which consists of a $10 \times 8 \times 6\ \mathrm{m}^3$ room. In order to enhance the estimation of the pose, smaller panels are distributed along the walls. This is necessary, since the LiDAR can only contribute to the pose information along the normal direction of the walls. If the LiDAR would only detect four walls in a room without auxilliary panels (similar to a long hallway), the filter might diverge over time due to the IMU drift.

For the investigations, random trajectories are generated, and sensor data is produced accordingly. The random trajectories consist of multiple six degrees of freedom (6-DoF) poses, and individual poses for specific times are determined through interpolation. The translations are generated by sampling from a uniform distribution scaled to the room dimensions, e.g. for the x-translation $t_x \sim U(0,1) \cdot dim_x$. For the rotations, three random values are sampled uniformly and converted to angles by $r_1 \sim \arccos(2 \cdot U(0,1) - 1)$ and $r_2, r_3 \sim 2\pi \cdot U(0,1)$. Afterwards, these rotation angles are converted to quaternions. The time between the individual poses is determined based on the Euclidean distances and rotation angles between consecutive pose. This is done to limit the maximum velocities and rotation rates, which are set to values of $1.0\ ^{\mathrm{m}}/_{\mathrm{s}}$ and $0.4\ ^{\mathrm{rad}}/_{\mathrm{s}}$ respectively, in order to simulate a carried system for the presented results. The trajectories always begin with $5\,\mathrm{s}$ of stationary time to initialize the filter and to compare stationary and kinematic phases.

B. Sensors

The sensor data is generated based on the trajectory and environment. In the first step, the sensor data for the IMU and LiDAR are generated without sensor noise, allowing for the addition of noise in multiple Monte Carlo (MC) realizations. The sensor noise is based on real sensors. For the IMU, the specifications of a calibration of a Microstrain 3DM-GQ4-45 [24] are used, while the LiDAR simulation is based on a Velodyne Puck [25], specifically regarding the laser ray configuration. The noise of the LiDAR is generated based on empirical values from [26]. To properly weight the observations, the transformation from spherical coordinates in the sensor-frame s to Cartesian coordinates in the b-frame is accompanied by a variance-propagation. Fig. 1 shows an overview of the simulated environment, displaying the LiDAR point cloud and the pose of the simulated system for one individual epoch.

The values for the sensor uncertainties are shown in Tab. I. All uncertainties are assumed to be zero-mean Gaussian noise.

IV. Results

The results of the simulations are shown in two steps. First, the results for a single run are shown to demonstrate the evaluation procedure. Afterwards, the influence of the linear interpolation for the pose parameters is investigated. This is done using MC runs to reduce the impact of individual realizations due to the random number generation.

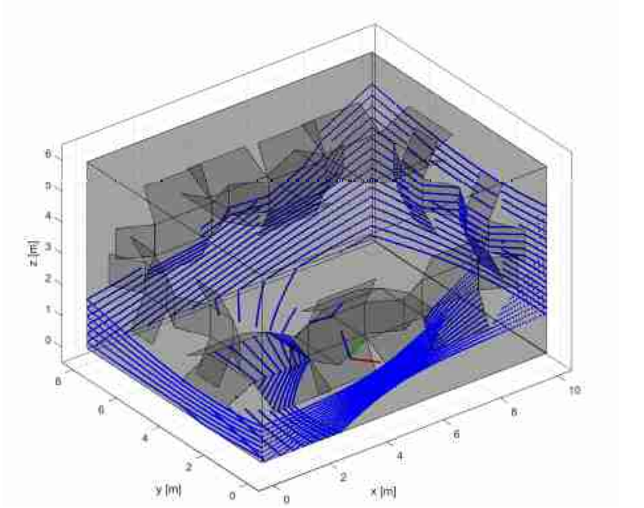

Fig. 1. Overview of the simulated environment: Wall and panels of the room (grey), LiDAR point cloud (blue points) and current pose of system shown by the axes (red-X, green-Y, blue-Z)

TABLE I
SENSOR UNCERTAINTIES FOR SIMULATION

LiDAR (standard deviations)		
distance measurement σ_d	8.5	mm
horizontal angle σ_α	48.7	mdeg
vertical angle σ_e	29.8	mdeg
IMU for (x,y,z)-axes		
acc. noise density $\sigma_{\tilde{a}_n}$	$(6.16, 4.61, 3.62) \cdot 10^{-3}$	$\mathrm{m}\sqrt{\mathrm{Hz}}/\mathrm{s}^2$
acc. bias instability σ_{a_w}	$(22.26, 20.35, 36.36) \cdot 10^{-3}$	m/s^2
gyro. noise density $\sigma_{\tilde{\omega}_n}$	$(0.26, 0.34, 0.24) \cdot 10^{-3}$	$\mathrm{rad}\sqrt{\mathrm{Hz}}/\mathrm{s}$
gyro. bias instability σ_{ω_w}	$(1.23, 1.14, 1.19) \cdot 10^{-3}$	rad/s

The evaluation of the results is based on three different metrics. The assessment is primarily focused on the deviation $\boldsymbol{\Delta}$ between the true pose, denoted by $\tilde{\ }$ and esimated pose obtained from the filter, which are used for the root mean square error (RMSE) and Mahalanobis distance $\sqrt{d_m}$

$$\boldsymbol{\Delta}^{(k)} = \begin{bmatrix} \tilde{\boldsymbol{t}}_{\mathrm{b}}^{\mathrm{l,\,(k)}} \\ \tilde{\boldsymbol{q}}_{\mathrm{b}}^{\mathrm{l,\,(k)}} \end{bmatrix} \boxminus \begin{bmatrix} \hat{\boldsymbol{t}}_{\mathrm{b}}^{\mathrm{l,\,(k)}} \\ \hat{\boldsymbol{q}}_{\mathrm{b}}^{\mathrm{l,\,(k)}} \end{bmatrix}, \tag{37}$$

$$\mathrm{RMSE}^{(k)} = \left(\boldsymbol{\Delta}^{(k)}\right)^T \boldsymbol{\Delta}^{(k)}, \tag{38}$$

$$\sqrt{d_m}^{(k)} = \sqrt{\left(\boldsymbol{\Delta}^{(k)}\right)^T \boldsymbol{Q}_{pose,\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}}^{(k)} \boldsymbol{\Delta}^{(k)}}, \tag{39}$$

where $\boldsymbol{Q}_{\mathrm{pose},\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}}^{(k)}$ represents the VCM estimated by the filter for the pose-related states. This square root value provides a weighted RMSE using the estimated VCM for assessing the consistency of the filter's estimation. The consistency of the result can be assessed by testing the Mahalanobis distance against a χ^2-distribution, assuming a normal distribution for the estimated states [27]. The square root is presented for easier interpretation.

In the results presented, not all points of the point cloud are used in the update step. Various methods, such as voxel grid filters [28] or line-based methods [29], are possible alterna-

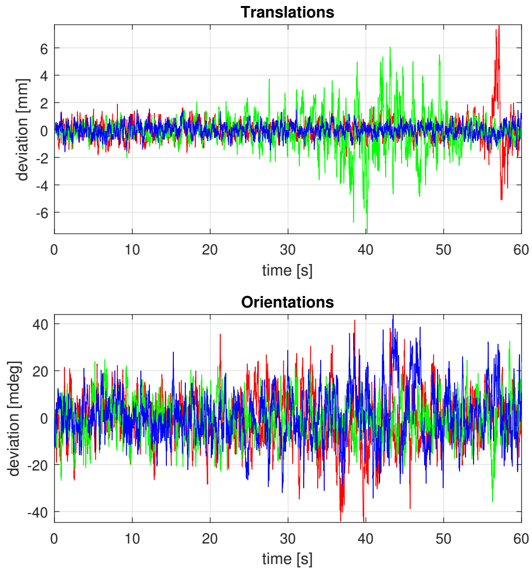

Fig. 2. Deviations of one trajectory in the three translations (top) and three orientation (bottom), translation along and orientation angles around the axes (red-X, green-Y, blue-Z)

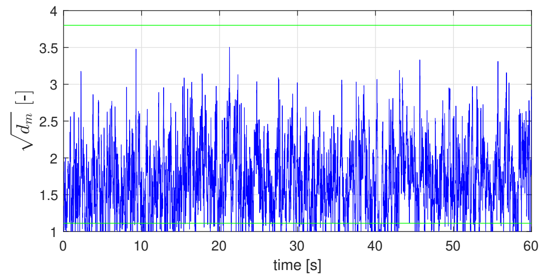

Fig. 3. Square root of the Mahalanobis distances of one trajectory with the limits of the χ^2-interval in green.

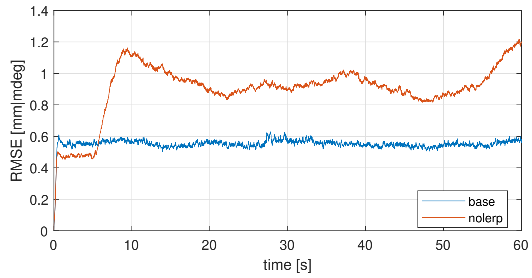

Fig. 4. RMSE of the trajectories of the results of the base and nolerp variants

tives. However, in this paper, a random subsampling approach is employed to reduce the size of the fully populated matrix $\boldsymbol{Q}_{rr}$, which needs to be inverted. The subsampling retains the original observations keeping the uncertainty information of the variance propagation and timestamps. In this paper, 2 % of the points are used in the update step.

A. Single Run

The deviations between the true pose and the estimated pose for a single run of a random trajectory are depicted in Fig. 2. The upper plot illustrates the individual elements of the translation while the lower plot represents the elements of the rotation vector corresponding to orientation. The translation is shown along and the orientation angles around the axes (red-X, green-Y, blue-Z).

The deviations show differences between the true pose and the estimated pose in the low millimeter range for translation and up to 40 mdeg in orientation. The absolute values are highly dependent on the settings of the simulation, which are not the focus here. Instead, the comparison to the Mahalanobis distance in Fig. 3 offers further insights.

The Mahalanobis distance is within and below the interval of the χ^2-distribution for the entire trajectory. This indicates that the estimation is consistent for about 80 % of the epochs and slightly pessimistic for the remaining portion. The higher deviations in the y-translation between 30 and 50 s and the deviations in the x-translation after 55 s are not visible in the Mahalanobis distance showing that the increased deviations are properly represented in the covariances.

B. Influence of the Pose Interpolation

To evaluate the influence of the pose interpolation in the update step of the filter, 100 trajectories with 10 MC runs are each performed with (*base*) and without the pose interpolation (*nolerp*) in (31 - 34). To disable the interpolation, τ_i is set to 1 for all points. The results are shown in Fig. 4 and 5.

In both figures, a clear systematic can be observed for the results without interpolation. During the stationary phase, the results are highly accurate, with a low RMSE and consistent estimation. However, once the motion begins, both the RMSE and the Mahalanobis distance increase. On the other hand, the results with interpolation exhibit relatively stable performance throughout the entire duration, albeit with slightly higher RMSE and slightly lower Mahalanobis distance.

The higher RMSE can be explained by the functional model. The interpolation factor, obtained by τ_i, acts as a damping factor for the functional model. Since the states of the prior epoch

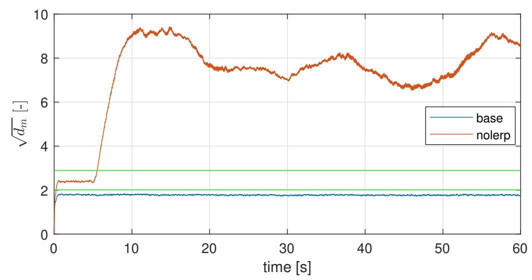

Fig. 5. Square root of Mahalanobis distance for results of the base and nolerp variants with the limits of the χ^2-interval in green.

are fixed and only the current states are estimated, a point with $\tau_i \approx 0.5$ only has about half the impact on the estimation compared to a point with $\tau_i \approx 1$. This discrepancy contributes to the higher RMSE values. The behaviour of the Mahalanobis distance is more challenging to explain. The results without interpolation demonstrate consistent performance during the stationary phase but exhibit inconsistent and overly optimistic results during motion, primarily due to the absence of motion compensation. This pattern is reproducible for all subsampling rates. On the other hand, the interpolation approach yields a constant Mahalanobis distance regardless of the motion, but it results in a somewhat pessimistic uncertainty estimation. The underlying cause of this mismatch between accuracy and uncertainty estimation is currently unclear and requires further investigation.

V. Conclusion

This paper presents an ESKF with implicit measurement equations applied to the position tracking of a MSS consisting of an IMU and a LiDAR in a simulated environment. The use of implicit measurement equations in the error state filter update allows for innovative concepts like the such as point-plane distance minimization, as demonstrated in this study. The results of the simulations show that the filter achieves satisfactory performance due to the utilization of linear interpolation of the pose for motion compensation during the update step. However, further investigation is required to address the slightly pessimistic uncertainty estimation indicated by the Mahalanobis distance. Meanwhile, the constant nature of the Mahalanobis distance can be used to determined a correction factor.

Future works should focus on using the filter with real-world data, which presents up several areas for improvements. Due to the nature of LiDAR data, outliers and different characteristics of the observations lead to challenges, so robust estimation and adaptive stochastic models would improve the filter greatly for the usage with real data.

References

[1] S. Schön, K.-N. Baasch, L. Icking, A. KarimiDoona, Q. Lin, F. Ruwisch, A. Schaper, and J. Su, "Towards integrity for gnss-based urban navigation – challenges and lessons learned," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, 2022, pp. 1774–1781.

[2] W. Lee, P. Geneva, Y. Yang, and G. Huang, "Tightly-coupled gnss-aided visual-inertial localization," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 9484–9491.

[3] C. Qin, H. Ye, C. E. Pranata, J. Han, S. Zhang, and M. Liu, "Lins: A lidar-inertial state estimator for robust and efficient navigation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.

[4] Z. Liu and F. Zhang, "Balm: Bundle adjustment for lidar mapping," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3184–3191, 2021.

[5] P. Trusheim, M. Mehltretter, F. Rottensteiner, and C. Heipke, "Cooperative image orientation considering dynamic objects," in *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Copernicus GmbH, 2022, pp. 169–177.

[6] F. Dellaert, "Factor graphs: Exploiting structure in robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. 1, pp. 141–166, 2021.

[7] P. Geneva, K. Eckenhoff, Y. Yang, and G. Huang, "Lips: Lidar-inertial 3d plane slam," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018.

[8] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[9] S. I. Roumeliotis, G. S. Sukhatme, and G. A. Bekey, "Circumventing dynamic modeling: evaluation of the error-state kalman filter applied to mobile robot localization," in *1999 IEEE International Conference on Robotics and Automation*. IEEE, 1999.

[10] T. Dang, "An iterative parameter estimation method for observation models with nonlinear constraints," *Metrology and Measurement Systems*, vol. 15, no. 4, pp. 421–432, 2008.

[11] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007.

[12] A. Schlichting, *Fahrzeuglokalisierung durch automotive Laserscanner unter Verwendung statischer Merkmale: Ph.D. Thesis*, ser. Reihe C. München: DGK, 2018, vol. 826.

[13] F. Esser, J. A. Moraga, L. Klingbeil, and H. Kuhlmann, "Accuracy improvement of mobile laser scanning point clouds using graph-based trajectory optimization," in *5th Joint lnternational Symposium on Deformation Monitoring*, L. Garcia-Asenjo and J. L. Lerma, Eds. València: Editorial Universitat Politècnica de València, 2023.

[14] 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*, vol. 87, no. 5-6, pp. 229–247, 2019.

[15] L. Lucks, L. Klingbeil, L. Plümer, and Y. Dehbi, "Improving trajectory estimation using 3d city models and kinematic point clouds," *Transactions in GIS*, vol. 25, no. 1, pp. 238–260, 2021.

[16] M.-G. Li, H. Zhu, S.-Z. You, and C.-Q. Tang, "Uwb-based localization system aided with inertial sensor for underground coal mine applications," *IEEE Sensors Journal*, vol. 20, no. 12, pp. 6652–6669, 2020.

[17] F. Aghili and C.-Y. Su, "Robust relative navigation by integration of icp and adaptive kalman filter using laser scanner and imu," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 4, pp. 2015–2026, 2016.

[18] A. Schütz, D. E. Sanchez-Morales, and T. Pany, "Precise positioning through a loosely-coupled sensor fusion of gnss-rtk, ins and lidar for autonomous driving," in *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. IEEE, 2020.

[19] J. Solà, *Quaternion kinematics for the error-state Kalman filter*. arXiv, 2017. [Online]. Available: https://arxiv.org/abs/1711.02508

[20] T. Luhmann, *Nahbereichsphotogrammetrie*. Wichmann Verlag, 2000.

[21] 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)*, 2018, pp. 1209–1216.

[22] S. Vogel, *Kalman Filtering with State Constraints Applied to Multi-sensor Systems and Georeferencing: Ph.D. Thesis*. München: DGK, 2020, vol. 856.

[23] K. Shoemake, "Animating rotation with quaternion curves," in *Proceedings of the 12th annual conference on Computer graphics and interactive techniques - SIGGRAPH '85*, ACM, Ed. New York, New York, USA: ACM Press, 1985.

[24] Parker, "Inertial sensors," 2023. [Online]. Available: https://www.microstrain.com/inertial-sensors/all-sensors

[25] Velodyne, "Vlp-16 datasheet," San Jose, CA, USA, 2019.

[26] D. Ernst, H. Alkhatib, I. Neumann, and S. Vogel, "Analysis of multiple positions for the intrinsic and extrinsic calibration of a multi-beam lidar," in *2022 25th International Conference on Information Fusion (FUSION)*, 2022, pp. 01–08.

[27] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[28] M. Hillemann, M. Weinmann, M. S. Mueller, and B. Jutzi, "Automatic extrinsic self-calibration of mobile mapping systems based on geometric 3d features," *Remote Sensing*, vol. 11, no. 16, pp. 1–29, 2019.

[29] W. Błaszczak-Bak, Z. Koppanyi, and C. Toth, "Reduction method for mobile laser scanning data," *ISPRS International Journal of Geo-Information*, vol. 7, no. 7, 2018.

Dominik Ernst*, Sören Vogel, Hamza Alkhatib and Ingo Neumann

Monte Carlo variance propagation for the uncertainty modeling of a kinematic LiDAR-based multi-sensor system

https://doi.org/10.1515/jag-2022-0033
Received September 8, 2022; accepted August 4, 2023;
published online September 7, 2023

Abstract: Kinematic multi-sensor systems (MSS) are widely used for various applications, like mobile mapping or for autonomous systems. Depending on the application, insufficient knowledge of a system, like wrong assumptions about the accuracy of calibrations, might lead to inaccurate maps for mapping tasks or it might endanger humans in the context of autonomous driving. Uncertainty modeling can help to gain knowledge about the data captured by a system. Usually, uncertainty estimations for MSSs are done as backward modeling based on a comparison to reference datasets. In this paper, a forward modeling approach for the uncertainty modeling of a LiDAR-based kinematic MSS is chosen to estimate the uncertainty of an acquired point cloud. The MSS consists of a Leica Absolute Tracker and a platform with a 6-DoF sensor and Velodyne VLP-16 LiDAR. Results of multiple calibrations are used as the source for the uncertainty information for a Monte Carlo (MC) variance propagation of the point uncertainties. The deviations of the acquired point clouds in comparison to a ground truth can be decreased by an ensemble referencing process using the MC samples. Furthermore, the predicted uncertainties for the point clouds are well representing the actual deviations for reference panels closer to the system. Panels farther away indicate remaining distance depending effects.

Keywords: kinematic laser scanning; variance propagation; uncertainty modeling; Monte Carlo method; GUM; multi-sensor system

***Corresponding author: Dominik Ernst**, Geodetic Institute, Leibniz University Hannover, Hannover, Germany,
E-mail: ernst@gih.uni-hannover.de. https://orcid.org/0000-0001-9826-5610
Sören Vogel, **Hamza Alkhatib and Ingo Neumann**, Geodetic Institute, Leibniz University Hannover, Hannover, Germany,
E-mail: vogel@gih.uni-hannover.de (S. Vogel),
alkhatib@gih.uni-hannover.de (H. Alkhatib),
neumann@gih.uni-hannover.de (I. Neumann)

1 Introduction

Modern kinematic multi-sensor systems (MSSs) are used in various areas. Possible applications may include the usage as mobile mapping systems to quickly map large areas or as autonomous systems, that must locate themselves and avoid obstacles. In any case, knowledge about the trustworthiness of the captured data is essential to further processing steps. While wrong assumptions about the uncertainty of data may “only” lead to inaccurate maps, the same misinformation might lead to danger for human life in the context of autonomous driving. This need for in-depth knowledge about a system has to be balanced with the choice of the sensors used in an MSS. Low-cost sensors may be economically efficient, but the information about their uncertainty budget is also often limited.

In general, the sensors used for MSSs can be broadly divided into two groups. The referencing sensors, such as inertial measurement units (IMU) and Global Navigation Satellite System (GNSS) units, are used to estimate the trajectory of the system. As these sensors directly determine the position or orientation of the system, they can also be called direct referencing sensors [1]. Capturing sensors are mainly used to perceive the environment. These sensors can be cameras, light detection and ranging (LiDAR)/laser scanners (LS), radars or other active or passive sensors. In this paper, the focus is on low-cost LiDARs as they are useful for their ability to directly gather 3D data for a relatively low sensor price. Specifically, the Velodyne VLP-16 [2] is investigated as this sensor or similar multi-beam LiDARs are commonly used in different applications, e.g. autonomous cars, robotics or kinematic scanning systems for surveying.

The line between the direct referencing and capturing sensor groups blurs with the usage of data-driven trajectory estimation methods. These can either be done using available information of the environment [3, 4] or in unknown environments to build new maps using simultaneous localization and mapping (SLAM) [5]. This is called indirect referencing due to the intermediate step of the interaction with the environment. These methods lead to a redundancy in the trajectory estimation for situations, in which the referencing sensors cannot contribute, e.g. not enough visible satellites for GNSS. Using capturing sensors for referencing increases the importance of trustworthy observations from these sensors. In addition, the system becomes more complex, since not only do the sensors interact with each other, but also the environment plays a more important role in the trajectory estimation.

This complexity becomes clear in the technical aspects of sensor data fusion. The measurements of all sensors should refer to a common body frame and all measurements should refer to the same time frame. Both of these aspects have to be dealt with for each sensor, e.g. by a system calibration and affect the quality of the data fusion, which feeds into the later processing steps.

To acquire knowledge about the uncertainty of the data captured by a kinematic MSS, two different approaches to quality evaluation can be performed [6]. Either backward modeling can be performed, for which the data captured by the kinematic MSS is compared to a ground truth to derive the quality by some metric. Alternatively, a forward modeling approach can be chosen. This requires in-depth knowledge about the full system including the sensors and the processing steps.

1.1 Related works

Since this paper focuses on kinematic MSSs with LiDARs, the scope of this section is also limited to the quality evaluation of point clouds, which are captured by LiDARs or profile laser scanners.

Quality evaluation can be performed with two different approaches. Backward modeling describes the approach of acquiring a point cloud with the investigated system and comparing it to a ground truth. This ground truth should consist of measurements, which are at least one order of magnitude more accurate (ten times smaller standard deviation) than the assumed uncertainty of the measurements captured by the system [7]. The comparison of the datasets can be used to quantify the measurement uncertainty under the conditions during the measurement (e.g. temperature or illumination). This approach is widely used for different MSSs, either vehicle-bound [8–11] or airborne [12]. The ground truth can consist of point targets [12], small areas in the environment (mostly planar) [8, 9] or complete point clouds of the scene [10, 11]. The main effort for these analyses is the capturing of datasets for different situations or conditions. The main drawback of backward modeling is the missing ability to transfer the results to different conditions.

The second approach is forward modeling, which attempts to model all influences of the measurement process to achieve a full uncertainty model of the system. This requires knowledge about the measurement uncertainties of the used sensors and about the assumptions made in the algorithms used to fuse and process the data. Due to this required in-depth knowledge, this process is more linked to the MSS and, therefore, rarely possible for commercial systems.

Stenz et al. [6] present the results of forward modeling for a kinematic system used in industrial quality assurance. The presented system consists of two sensors for an indoor application limiting the number of influences. The system is similar to the one used in this paper. A Leica Absolute Tracker 960-LR (LT) [13] tracks a T-Probe to derive the 6-DoF pose of a platform with a terrestrial LS (TLS). Information about the sensors is provided by the manufacturers and the algorithms for the data fusion are self-implemented so that uncertainties can be modeled throughout the processing.

Variance propagation is essential to forward modeling, as it is used to propagate the uncertainties of the original measurements from the sensor frame to the target frame. Depending on the linearity of the functions in the processing chain for the referencing and the magnitude of the uncertainties, this can lead to linearization biases caused by the Taylor series approximation in the Gaussian variance propagation. These biases can be reduced through various methods. One approach is to employ Unscented Transformations (UT), which provide better approximations for nonlinear functions by utilizing Sigma points. However, it should be noted that UT is constrained to normal distributions [14]. Alternatively, the Guide to the Expression of Uncertainty in Measurements (GUM) [15] suggests the use of Monte Carlo (MC) simulation to perform the variance propagation. The usage of MC samples to approximate the distributions also avoids possible linearization biases.

Niemeier and Tengen [16] present the adjustment of a geodetic network following the GUM procedure for MC variance propagation with various types of uncertainty influences. For applications involving LSs, Koch [17] presents the variance propagation for a system to detect deformations of road surfaces. The modeling is limited to the measurement uncertainties of the LS. Alkhatib and Kutterer [18] use the MC variance propagation for the monitoring of a bridge using a static LS. Ehrhorn [19] applies the MC variance propagation to the system used by [6] for forward modeling of the uncertainties. To our knowledge, this is the only application of the MC variance propagation for kinematically referenced point clouds so far.

For uncertainty modeling, the GUM [20] proposes the classification of uncertainties in two types: type A, which are random dispersion of measurements determined empirically and type B, which are derived from additional information. Type A uncertainties are usually normally distributed and correspond to the classical modeling of white noise observations. Type B uncertainties can include any information from manuals, calibration reports, expert knowledge and other sources. This classification of the uncertainty components is independent of the classical division of random and systematic deviations, meaning both types can be either random or systematic. As opposed to type A uncertainties, type B uncertainties can be distributed with arbitrary density functions. To apply these distributions, they can either be transformed to normal distributions or directly used in the aforementioned MC simulations to propagate the uncertainties.

1.2 Contributions

In the GUM, the uncertainty classification is described concerning a single measurement process connecting the input quantities to the measurand. To model the measurement process properly, all influencing factors need to be included. This process is expanded to the presented processing chain with multiple steps for the referencing of point clouds. The point clouds are of interest since they can be used for different tasks, like localization or obstacle detection. The results of the preprocessing steps are used for the uncertainty estimation in the processing chain for the referencing. Due to this multi-step process, the types of uncertainties are not discussed for every input and intermediate step of the processing chain. As one simplification in this paper, all distributions of the influencing factors are assumed to be normal distributions. In the end, the accuracy of the referenced points and their predicted uncertainties are evaluated in comparison to a ground truth derived from a measurement of superior accuracy. This investigation is done in a laboratory environment to limit the number of influencing factors.

The main contributions of this paper can be summarized in the following three aspects. The system calibration using object-space information is used to additionally estimate the intrinsic parameters of the LiDAR. In particular, a variance component estimation (VCE) is performed to determine the uncertainties of the measurement elements of the LiDAR. This information is integrated into the processing chain for the referencing of the point clouds.

In addition, the MC variance propagation is used to model the uncertainties of the referenced point cloud under consideration of the uncertainties resulting from the influencing intermediate steps in the processing chain. This is an extension to the modeling of measurement uncertainties described in the GUM for a multi-step processing chain. The impact of the uncertainty modeling is demonstrated with the decrease of deviations of the resulting point cloud produced by ensemble referencing using the MC samples for a real data set.

Finally, a quantitative evaluation strategy based on the predicted variance-covariance matrix (VCM) of the points is presented to evaluate the consistency of the uncertainty in comparison to reference panels. This procedure is used to demonstrate the performance of the uncertainty modeling in its current state.

1.3 Outline

The paper is structured as follows. The established processing chain for the kinematic experiment itself is shown in Section 2. First, the MSS used for the experiment is shown, before all necessary steps are described. The results for the referenced point cloud and the corresponding uncertainties are presented in Section 3. These results are further discussed in Section 4, before Section 5 concludes the paper.

2 Materials and methods

Sensor data fusion consists of two main aspects. The first aspect is the referencing to fuse the data of multiple sensors in a common coordinate frame. The other aspect is the time synchronization of the different sensors. This section shows, how these aspects are dealt with. Given the utilization of a reliable system employing trigger pulses, the primary focus lies in the processing chain for referencing the kinematically acquired point clouds. Subsequently, only a brief overview of the experimental setup for the kinematic experiment is presented.

2.1 Processing chain

The kinematic MSS consists of a Husky Unmanned Ground Vehicle (UGV) with a sensor platform mounted on it [21]. For the usage in this paper, only the Velodyne VLP-16 LiDAR and the Leica T-Probe on the platform are of interest. The VLP-16 is a multi-beam LiDAR with 16 rings covering a $360° \times 30°$ field of view [2]. A Leica AT960-LR laser tracker (LT) is used to track the T-Probe to determine the 6-DoF trajectory of the platform. This yields five frames of interest for the experiment:

- LiDAR sensor frame s,
- platform body frame b,
- T-Probe sensor frame t,
- LT sensor frame l,
- room frame r.

For the referencing of the kinematically acquired point clouds of the VLP-16, four transformations between the above-mentioned frames are necessary. In this section, the determination of the transformation parameters and their uncertainties for each step is shown and discussed. Figure 1 gives an overview of the experiment with the frames involved in the referencing of a point $\mathbf{p}_i$. The coordinate systems are colored as XYZ corresponding to RGB. A black ring around the white origin of the trihedron marks, if the trihedron is shown in the origin of the frame.

The steps in the processing chain are described in the following subsections. Each step includes the determination of the necessary

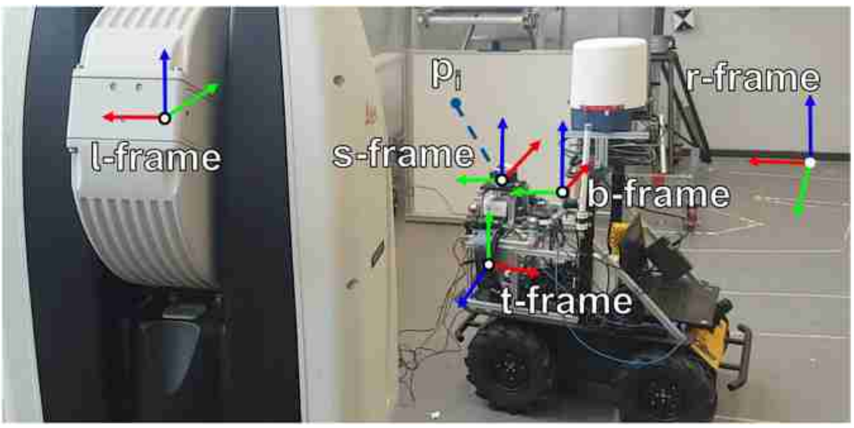

Figure 1: Image of the experimental setup with the frames involved in the processing chain. The frames *s*, *t* and *l* are defined by the sensors, the *b*-frame is defined by the CCRs and the *r*-frame is defined by magnetic nests distributed in the room.

parameters and the corresponding uncertainties for the forward modeling. An overview of all steps with the influencing uncertainties is provided in Figure 2. The figure gives a schematic overview of the processing chain for determining the required parameters and their corresponding uncertainties.

2.1.1 System calibration of the LiDAR: Extrinsic calibration describes the procedure to determine the 3D transformation parameters between two frames, in this case even on the same platform. The transformation usually consists of a 3D translation and 3D rotation. The platform of the system defines a *b*-frame, towards which all sensors have to be referenced. If a CAD model of the platform is available, it can be used to derive the necessary transformation parameters. This requires the information about the sensor construction to be available to compute the pose of the sensor center with regard to the *b*-frame. The uncertainty of these transformation parameters depends on the manufacturing tolerances of the platform and the sensors themselves. When no CAD model is available, the transformation parameters from the *s*-frame to the *b*-frame can also be determined by measuring the housing of the sensor regarding the *b*-frame and reconstructing the frame by the information given by the manufacturer. For applications with high accuracy requirements, the transformation parameters derived from measurements of the housing are only useful as approximate values for further procedures. Alternatively, translations (lever arms) can be derived based solely on the housing, while orientations (boresight angles) can be estimated in subsequent procedures. Especially with low-cost sensors, deviations in the lever arms relatively have a minimal impact compared to the influences of the boresight angles, which increase proportionally with the measurement distance.

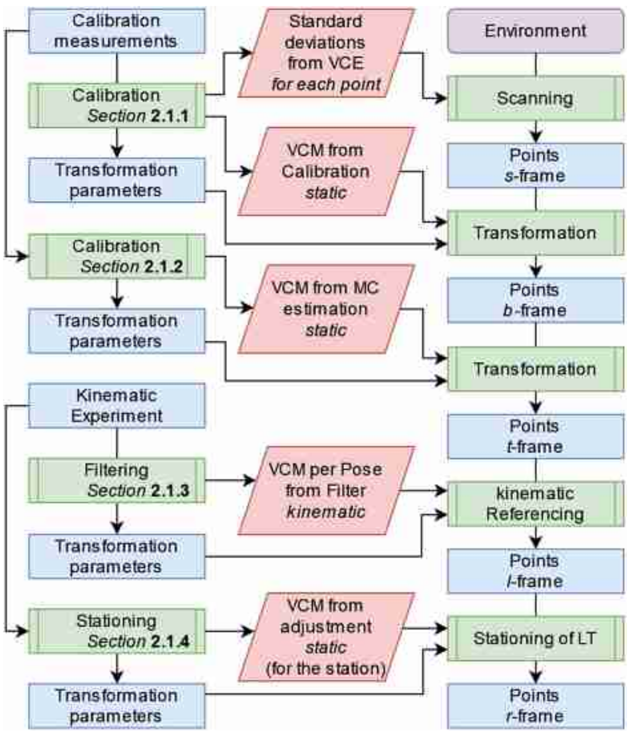

Figure 2: Schematic overview of the processing chain with the different measurements with values (blue), processes (green) and stochastic information (red). The relevant subsections are marked for reference.

In the case of capturing sensors, extrinsic calibration using object space information is a common method to estimate the transformation parameter accurately. The basic idea involves the usage of observations acquired by the sensors to better estimate the origin of the sensors. Depending on the sensor, different setups are used. For the calibration of cameras with respect to a *b*-frame, targets with known coordinates can be used [22]. For the direct fusion of images with point clouds, the relative transformation parameters between a camera and a LiDAR have to be determined. This is commonly done using a planar checkerboard [23], which is also commonly used for intrinsic camera calibrations.

For the extrinsic calibration of LiDARs on a platform, Strübing and Neumann [24] present an approach for using reference geometries. These geometries are scanned by LiDAR. Next, the acquired point clouds are transformed using approximate transformation parameters and the distances between the point cloud and ground truth are minimized to improve the transformation parameters. The procedure is originally presented for a system to check railways, but also found applications in industrial quality assurance [25] and outdoor mobile mapping systems [26]. The calibration in this paper builds upon this idea expanding it from an extrinsic calibration to a system calibration. It achieves this by estimating the distance offset of the LiDAR and determining uncertainties for the first two steps of the processing chain. These first two steps are the measurement of the points by the LiDAR and the transformation of the points from the *s*-frame to the *b*-frame.

The procedure used expands on the ideas described in [27]. The frame of the LiDAR and the platform have to be defined beforehand. While the *s*-frame is defined by the origin of the measurements, the *b*-frame can be defined depending on the application. In this case, it is uniquely defined by three corner cube reflectors (CCR) placed in nests on the platform (cf. Figure 1). The frame defined by the CCRs is stable over time and enables the calibration of additional sensors, like IMUs or cameras with respect to the *b*-frame.

Measurements of the sensor housing are used to derive approximate values to start the estimation process. Furthermore, a calibration field with reference geometries is needed. The reference geometries can be any geometric primitive, but usually, planes are preferred [24, 26, 28]. For the determination of the transformation parameters between *s*-and *b*-frame, the platform is positioned and referenced by a sensor of superior accuracy. In this case, an LT is used to provide the ground truth for the pose of the platform and the reference panels. The measurement accuracy for measurements on a CCR is given as a maximum permissible error ($\mathrm{MPE_{XYZ}}$) of $\pm 15\ \mu\mathrm{m} + 6\ \mathrm{ppm}$ [13]. This makes the LT a suitable reference sensor for the VLP-16 with a range accuracy of ± 3 cm [2]. Then, the reference geometries of the calibration field are measured by LiDAR. Using approximate values for the transformation between the *s*-and the *b*-frame, the acquired point cloud can be transformed to assign them to the reference geometries. This assignment is done by hand to eliminate outliers and remove mixed pixels at the borders of the panels. The adjustment aims at minimizing the distances between the transformed point cloud and the reference geometries by updating the parameters for the transformation between the *s*-and the *b*-frame.

The setup of the calibration field is important for the determination of the parameters. The reference panels $j = 1 \ldots J$ need to be positioned in different orientations to be sensitive to different transformation parameters. J describes the number of reference panels. In this case, the calibration field of the Geodetic Institute Hannover is used [28].

The measurements can also be repeated for P positions of the platform $p = 1 \ldots P$ to acquire more data in different constellations with respect to the calibration field. This improves the geometric constellation of the adjustment to decrease the correlations between the determined transformation parameters. Additionally, it improves the determination of the transformation parameters, since the influence of deviations caused by the orientations increases with longer distances. So, a variety of distances is necessary. Figures 3 and 4 show an overview of the calibration field, the panels and the positions used for the calibration.

In addition to the transformation parameters, intrinsic parameters of the LiDAR can be determined. Here, this is realized by estimating an additional distance offset d_0 for all rings of the LiDAR. Estimating additional parameters is feasible, but typically requires longer distances, e.g. for an estimation of a scale factor [29], and changes to the setup to determine all parameters significantly [30]. Consequently, in this study, only the distance offset is estimated. For each of the n points $\mathbf{p}_i$ with $i = 1 \ldots n$, there are the original observations of the

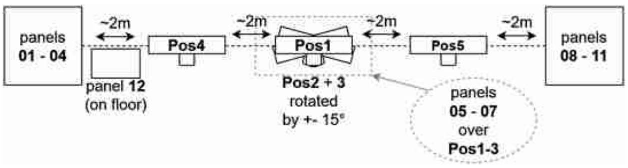

Figure 3: Overview of the calibration field and the positions for the system calibration shown as a top view.

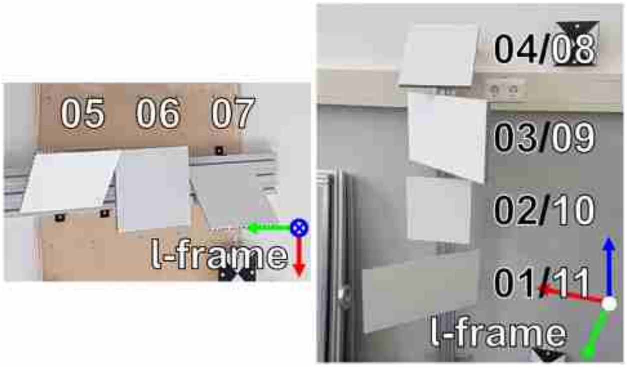

Figure 4: Overview of calibration panels. Left: Panels on ceiling, right: panels on walls (mirrored on both sides, numbers in black/white for different sides in the room), not shown: panel 12 placed horizontally on the floor.

LiDAR $\boldsymbol{\ell}_i = \left[d_i\ \alpha_i\ e_i\right]^T$ with the distances d, the azimuth angles α and the elevations e, which are used to compute the Cartesian points by

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}_s = \begin{bmatrix} (d_i + d_0) \cdot \cos e_i \cdot \cos \alpha_i \\ (d_i + d_0) \cdot \cos e_i \cdot \sin \alpha_i \\ (d_i + d_0) \cdot \sin e_i + z_c \end{bmatrix}. \tag{1}$$

The value z_c corresponds to a ring-dependent vertical correction, as described by the manufacturer [2].

The following transformations have 6-DoF, which means that it is estimated with 3 translations $\mathbf{t} = \left[t_x\ t_y\ t_z\right]^T$ and a 3D rotation matrix $\mathbf{R}$, which can be set up in multiple ways. Here, Euler angles ω, φ and κ are used to compose three rotation matrices around their corresponding axes, which are then multiplied for the full 3D rotation matrix

$$\mathbf{R}(\omega, \varphi, \kappa) = \mathbf{R}_z(\kappa)\mathbf{R}_y(\varphi)\mathbf{R}_x(\omega) \tag{2}$$

$$\text{with } \mathbf{R}_z(\kappa) = \begin{bmatrix} \cos\kappa & -\sin\kappa & 0 \\ \sin\kappa & \cos\kappa & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{R}_y(\varphi) = \begin{bmatrix} \cos\varphi & 0 & \sin\varphi \\ 0 & 1 & 0 \\ -\sin\varphi & 0 & \cos\varphi \end{bmatrix} \text{and}$$

$$\mathbf{R}_x(\omega) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\omega & -\sin\omega \\ 0 & \sin\omega & \cos\omega \end{bmatrix}.$$

Since the order of rotation matters for the described rotation, the sequence is given by the variables. Using this notation and the computed Cartesian coordinates, the transformations from the *s*-frame to the *b*-frame and then to the *l*-frame are concatenated as

$$\mathbf{p}_{b,i} = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}_b = \mathbf{t}_s^b + \mathbf{R}_s^b(\omega, \varphi, \kappa) \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}_s \text{ and} \tag{3}$$

$$\mathbf{p}_{l,i} = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}_l = \mathbf{t}_{b(LT)}^l + \mathbf{R}_{b(LT)}^l(\omega, \varphi, \kappa) \cdot \mathbf{p}_{b,i}. \tag{4}$$

The source frame is noted in subscript and the target frame is noted in superscript. The elements marked by (*LT*) are derived from LT measurements. For a shorter notation, the parameters for a transformation are summarized as

$$\boldsymbol{\theta} = \begin{bmatrix} t_x & t_y & t_z & \omega & \varphi & \kappa \end{bmatrix}. \tag{5}$$

Next, the referenced points $\mathbf{p}_{l,i}$ can be used to compute the distances for the functional model of the adjustment using the plane parameters of the reference panel $\mathbf{a}_j = \begin{bmatrix} n_{x\,j} \; n_{y\,j} \; n_{z\,j} \; d_j \end{bmatrix}^T$ in the *l*-frame

$$h_i = \begin{bmatrix} n_{x\,j} \\ n_{y\,j} \\ n_{z\,j} \\ d_j \end{bmatrix}_l^T \cdot \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}_l, \tag{6}$$

$$h_i\left(\ell_i, \boldsymbol{\theta}_s^b, d_0, \boldsymbol{\theta}_b^l, \mathbf{a}_j\right) \overset{!}{=} 0. \tag{7}$$

The misclosure h_i is the distance between the transformed point *i* and the reference panel *j*. For the estimation, the values derived from the LT measurements are introduced as deterministic values due to the difference in measurement accuracy between the sensors. This leads to the parameter vector:

$$\mathbf{x} = \begin{bmatrix} \boldsymbol{\theta}_s^b \; d_0 \end{bmatrix}^T. \tag{8}$$

The adjustment is performed using least squares. Since the functional model is in the form of implicit measurement equations, the computation has to be realized as a Gauß-Helmert model (GHM) [31]. The computation of the parameters requires the Jacobian matrices $\mathbf{A}$ derived concerning the parameters $\mathbf{x}_0$ and $\mathbf{B}$ derived for the observations $\boldsymbol{\ell}_0$, as well as the misclosure vector $\mathbf{w}$

$$\mathbf{A} = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\ell_0, \mathbf{x}_0}, \qquad \mathbf{B} = \left.\frac{\partial \mathbf{h}}{\partial \boldsymbol{\ell}}\right|_{\ell_0, \mathbf{x}_0}, \tag{9}$$

$$\mathbf{w} = \mathbf{h}(\boldsymbol{\ell}_0, \mathbf{x}_0) + \mathbf{B}(\boldsymbol{\ell} - \boldsymbol{\ell}_0). \tag{10}$$

For the computation of the updated parameters $\hat{\mathbf{x}}$ and the residuals $\hat{\mathbf{v}}$, the following constraint has to be fulfilled

$$\mathbf{B}\mathbf{v} + \mathbf{A}\Delta\mathbf{x} + \mathbf{w} = \mathbf{0}. \tag{11}$$

For the estimation of the parameters, the VCM of the misclosures $\mathbf{Q}_{\mathrm{ww}}$ and the parameters $\mathbf{Q}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}$ are computed first using

$$\mathbf{Q}_{\mathrm{ww}} = \mathbf{B}\mathbf{Q}_{\ell\ell}\mathbf{B}^T, \tag{12}$$

$$\mathbf{Q}_{\hat{\mathbf{x}}\hat{\mathbf{x}}} = \left(\mathbf{A}^T\mathbf{Q}_{\mathrm{ww}}^{-1}\mathbf{A}\right)^{-1}, \tag{13}$$

$$\hat{\mathbf{x}} = \mathbf{x}_0 + \mathbf{Q}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}\mathbf{A}^T\mathbf{Q}_{\mathrm{ww}}^{-1}(-\mathbf{w}), \tag{14}$$

$$\hat{\mathbf{v}} = \mathbf{Q}_{\ell\ell}\mathbf{B}^T\mathbf{Q}_{\mathrm{ww}}^{-1}\left(\mathbf{I} - \mathbf{A}\mathbf{Q}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}\mathbf{A}^T\mathbf{Q}_{\mathrm{ww}}^{-1}\right)(-\mathbf{w}). \tag{15}$$

The computation with equations (9)–(15) iterates with

$$\mathbf{x}_0 = \hat{\mathbf{x}} \qquad \text{and} \qquad \boldsymbol{\ell}_0 = \boldsymbol{\ell} + \hat{\mathbf{v}} \tag{16}$$

until the maximum of the absolute misclosure vector is smaller than a chosen threshold (here: 10^{-10}). The stochastic model $\mathbf{Q}_{\ell\ell}$ is assumed to consist of the three original measurement elements, i.e. the distances, azimuth angles and elevations. The standard deviations are assumed to be the same within a group and all observations are assumed to be uncorrelated:

$$\underset{3n\times 3n}{\mathbf{Q}_{\ell\ell}} = \operatorname{diag}\left(\begin{bmatrix} \sigma_d^2 & \sigma_\alpha^2 & \sigma_e^2 & \dots & \sigma_e^2 \end{bmatrix}\right) \tag{17}$$

The assumption of uncorrelated observations is based on the fact, that not all observations acquired are used for the adjustment. Instead, the observations are subsampled down to 50 points per reference panel and position. This reduces the influence of correlations of subsequent points. Apart from this reduction of correlations, this also avoids implicit weighting of the reference panels by the number of observations assigned to the individual panels.

To determine the weighting of the observation groups, the algorithm is expanded by adding a VCE to determine a stochastic model for the LiDAR. A similar approach is used in [32], where the VCE is used to validate the initial assumptions, which are based on the information given by the manufacturer. Here, the goal is to determine an approximation for the measurement uncertainties under the given circumstances concerning the surfaces, distances and conditions in the laboratory. These conditions are the same for the kinematic experiment, so they can be used for the processing chain to reference the LiDAR point clouds. By keeping these conditions constant, their influence on the measurement process does not have to be modeled.

Similar to the adjustment itself, approximate values are necessary to start the iterative process. The algorithm used for the VCE is the best invariant quadratic unbiased estimator (BIQUE) [33, 34]. The update to the stochastic model of the measurement elements $u = 3$, which are represented by disjunct covariance matrices $\mathbf{V}_i$, is performed using the vector $\Delta\boldsymbol{\sigma}$

$$\mathbf{Q}_{\ell\ell} = \sum_{i=1}^{u} \Delta\sigma_i \mathbf{V}_i \qquad \text{with} \qquad \Delta\boldsymbol{\sigma} = \mathbf{S}^{-1}\mathbf{q}. \tag{18}$$

Both required values, the matrix $\mathbf{S}$ and the transformed residual sum $\mathbf{q}$, for the computation of $\Delta\boldsymbol{\sigma}$ are computed element-wise. The elements of $\mathbf{S}$ are indexed with *r* for rows and *c* for columns, while $\mathbf{q}$ has *u* elements. The matrix $\mathbf{M}$ corresponds to the VCM of the estimated and transformed observations,

$$q_i = \hat{\mathbf{v}}^T\mathbf{B}^T\mathbf{Q}_{\mathrm{ww}}^{-1}\mathbf{B}\mathbf{V}_i\mathbf{B}^T\mathbf{Q}_{\mathrm{ww}}^{-1}\mathbf{B}\hat{\mathbf{v}}, \tag{19}$$

$$\mathbf{M} = \mathbf{Q}_{\mathrm{ww}}^{-1} - \mathbf{Q}_{\mathrm{ww}}^{-1}\mathbf{A}\mathbf{Q}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}\mathbf{A}^T\mathbf{Q}_{\mathrm{ww}}^{-1}, \tag{20}$$

$$s_{r,c} = \operatorname{trace}\left(\mathbf{M}\mathbf{B}\mathbf{V}_c\mathbf{B}^T\mathbf{M}\mathbf{B}\mathbf{V}_r\mathbf{B}^T\right). \tag{21}$$

2.1.2 Extrinsic calibration of the platform: The next step in the processing chain is the transformation from the *b*-frame to the sensor frame of the T-Probe *t*. This step is necessary, since the T-Probe is tracked by the LT, so the poses are initially available for the *t*-frame. The *b*- and *t*-frame are separated, because the T-Probe can be mounted on different parts of the platform. Separating the two frames avoids multiple (extensive) calibrations with the LiDAR.

The transformation parameters are determined by LT measurements of the CCRs mounted on the platform and direct observations of the 6-DoF pose of the T-Probe. Both of these sets of observations can be used respectively to compute the transformation to the l-frame defined by the LT, which results in the homogeneous transformation matrices $\mathbf{H}_b^l$ and $\mathbf{H}_t^l$. The desired transformation can be computed by

$$\mathbf{H}_b^t = \left(\mathbf{H}_t^l\right)^{-1}\mathbf{H}_b^l. \tag{22}$$

The resulting transformation matrix $\mathbf{H}_b^t$ could be used for the referencing, but the uncertainty information is not available. The determination of the uncertainty of the transformation parameters is done using an MC variance propagation under the assumption of the measurement accuracies of the LT as provided by the manufacturer. For measurements on the T-Probe, the accuracy for measurements of CCRs has to be extended by ±35 μm [13]. The conversion from the provided MPE values to standard deviations is done by [15]

$$\sigma = \frac{\mathrm{MPE}}{\sqrt{3}}. \tag{23}$$

For the uncertainty of the orientation delivered by T-Probe, no uncertainty information is provided by the manufacturer. Instead, the value provided for the T-Mac is used (0.01° for all axes) [13]. There is no information on the correlation between the observations, so they are assumed to be independent. For the T-Probe, this assumption simplifies the principle of the pose estimation, leading to too optimistic estimations for the uncertainty.

The variance propagation is performed $M = 100,000$ times as suggested by [17]. For each realization, $m = 1 \ \dots \ M$, the transformation matrix $\mathbf{H}_b^{t,(m)}$ is computed based on observations, which are polluted by generated noise with the assumed measurement accuracy. Afterwards, the transformation matrices

$$\mathbf{H}_b^{t,(m)} = \begin{bmatrix} \mathbf{R}_b^{t,(m)} & \mathbf{t}_b^{t,(m)} \\ \mathbf{0}_{[1\times3]} & 1 \end{bmatrix} \tag{24}$$

are decomposed into the translation vectors $\mathbf{t}_b^{t,(m)}$ and the rotation matrices $\mathbf{R}_b^{t,(m)}$, which can subsequently be decomposed to yield the orientation as Euler angles. The translation vectors and Euler angles are collected in vectors of the transformation parameters $\boldsymbol{\theta}^{(m)}$. Next, the VCM of the transformation parameters can be computed using the error matrix with the mean vector of the transformation parameters $\bar{\boldsymbol{\theta}}$ for all M realizations [35]:

$$\mathbf{E} = \begin{bmatrix} \boldsymbol{\theta}^{(1)} - \bar{\boldsymbol{\theta}} \\ \boldsymbol{\theta}^{(2)} - \bar{\boldsymbol{\theta}} \\ \vdots \\ \boldsymbol{\theta}^{(M)} - \bar{\boldsymbol{\theta}} \end{bmatrix}, \tag{25}$$

$$\boldsymbol{\Sigma}_{\theta\theta} = \frac{1}{M-1}\mathbf{E}^T\mathbf{E}. \tag{26}$$

2.1.3 Filtering of the kinematic movement: The following step in the processing chain involves the kinematic movement of the platform. The kinematic referencing is realized by the LT measurements of the T-Probe attached to the robot, which can deliver a complete 6-DoF pose at 50 Hz.

The processing of these pose measurements is realized by a linear Kalman Filter (KF) [36]. The implementation of the filter serves multiple purposes. It is used to smooth the noise of the LT measurements and to provide VCMs for the poses of the kinematic measurements. Additionally, the state transitions can be used to interpolate poses and their VCMs by a prediction for the referencing of the point cloud on a point-by-point basis. The principle of this procedure is shown in [37]. The pose for the timestamp of each point is predicted separately to avoid deviations from time delays. This procedure is necessary, since the poses are only available at 50 Hz, while the point cloud is recorded with a data rate of 300 kHz.

The state vector $\mathbf{x}$ consists of the pose of the T-Probe $\boldsymbol{\theta}_t^l$ in the l-frame of the LT, which is directly observed by the LT, and the corresponding velocities and accelerations for a total number of 18 states. The initialization of the filter is done by using the first observation of the time series for the pose and zeroes for all velocities and accelerations since the robot starts in a stationary position. The state transitions are defined by the dynamic model [38], which is used to predict the states $\mathbf{x}$ for the next epoch (k)

$$\mathbf{x}^{(k)} = \boldsymbol{\Phi}^{(k-1)}\mathbf{x}^{(k-1)} + \mathbf{w}^{(k-1)} \tag{27}$$

with the transition matrix $\boldsymbol{\Phi}^{(k-1)}$ composed of matrices $\boldsymbol{\Phi}^{*(k-1)}$ for each element of the pose with their temporal derivatives, e.g. the translation t_x with the velocity v_x and the acceleration a_x depending of the time difference Δt

$$\boldsymbol{\Phi}^{*(k-1)} = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \tag{28}$$

with $\Delta t = t^{(k)} - t^{(k-1)}$. The system noise $\mathbf{w}$ has an important influence on the performance of the filter. In this case, the discrete Wiener process acceleration model is used [38]. This model relates the parameters of the poses to their derivatives to set up the VCM of the system noise

$$\boldsymbol{\Sigma}_{\mathbf{ww}}^* = \begin{bmatrix} \frac{1}{4}\Delta t^4 & \frac{1}{2}\Delta t^3 & \frac{1}{2}\Delta t^2 \\ \frac{1}{2}\Delta t^3 & \Delta t^3 & \Delta t \\ \frac{1}{2}\Delta t^2 & \Delta t & 1 \end{bmatrix}\sigma_w^2. \tag{29}$$

Here, the parameters of all translations and all rotations are shared, meaning that only two design parameters σ_w have to be determined. The optimization of the parameters is based on a grid search approach [39], which is evaluated based on the distances of the referenced point clouds to the reference panels shown in Section 2.2.

The VCM of the system noise $\boldsymbol{\Sigma}_{\mathbf{ww}}$ can be used to predict the VCM of the states $\boldsymbol{\Sigma}_{\mathbf{xx},+}$ for the next epoch. Predicted and updated values are marked by minus and plus, respectively

$$\boldsymbol{\Sigma}_{\mathbf{xx},-}^{(k)} = \boldsymbol{\Phi}^{(k-1)}\boldsymbol{\Sigma}_{\mathbf{xx},+}^{(k-1)}\left(\boldsymbol{\Phi}^{(k-1)}\right)^T + \boldsymbol{\Sigma}_{\mathbf{ww}}. \tag{30}$$

The update step of the KF includes the LT measurement of the T-Probe $\boldsymbol{\ell}^{(k)}$. Since the LT directly delivers the 6-DoF poses as observations, the measurement model is

$$\boldsymbol{\ell}^{(k)} = \mathbf{C}^{(k)}\mathbf{x}_-^{(k)} + \boldsymbol{v}^{(k)} \tag{31}$$

$$\text{with } \mathbf{C}^{(k)} = \begin{bmatrix} \mathbf{I}_{[6\times6]} & \mathbf{0}_{[6\times12]} \end{bmatrix} \tag{32}$$

with the residuals $\boldsymbol{v}^{(k)}$. Thus, the updated states $\mathbf{x}_+^{(k)}$ and their VCM $\boldsymbol{\Sigma}_{\mathbf{xx},+}^{(k)}$ can be computed using the Kalman gain matrix

$$\mathbf{K}^{(k)} = \boldsymbol{\Sigma}_{\mathbf{xx},-}^{(k)}\mathbf{C}^{(k)T}\left(\mathbf{C}^{(k)}\boldsymbol{\Sigma}_{\mathbf{xx},-}^{(k)}\mathbf{C}^{(k)T} + \boldsymbol{\Sigma}_{\ell\ell}^{(k)}\right)^{-1} \tag{33}$$

$$\mathbf{x}_+^{(k)} = \mathbf{x}_-^{(k)} + \mathbf{K}^{(k)}\left(\boldsymbol{\ell}^{(k)} - \mathbf{C}^{(k)}\mathbf{x}_-^{(k)}\right) \tag{34}$$

$$\boldsymbol{\Sigma}_{\mathbf{xx},+}^{(k)} = \left(\mathbf{I} - \mathbf{K}^{(k)}\mathbf{C}^{(k)}\right)\boldsymbol{\Sigma}_{\mathbf{xx},-}^{(k)} \tag{35}$$

The observations $\ell^{(k)}$ are assumed to be distributed with the stochastic model of the observations on the T-Probe $\Sigma^{(k)}_{\ell\ell}$, which is based on the uncertainty of the T-Probe mentioned above.

In the processing chain, the poses resulting from the filter are used with the transition matrix from equation (28) to interpolate the poses for the time stamps of the points. Hartmann et al. [37] show, that this method performs better than a linear interpolation of the poses.

2.1.4 Stationing of the laser tracker: The last step of the processing chain is the stationing of the LT. This corresponds to the transformation from the l-frame to the room frame r, which is defined by the laboratory network.

The stationing is done by measuring five control points of the laboratory network, for which the coordinates are known. The stochastic model for the observations of the LT is the same as in the system calibration of the platform in Section 2.1.2. The transformation parameters $\boldsymbol{\theta}^r_l$ are derived based on an adjustment using a GHM, which additionally determines the VCM of the transformation parameters $\Sigma_{\theta^r_l\theta^r_l}$.

2.1.5 Monte Carlo variance propagation through the processing chain: For the processing chain shown in Figure 2, each step results in transformation parameters together with a VCM. The assumption of normally distributed quantities for all values is given by the chosen methods (adjustments by the GHM, the KF and the density estimation for the MC variance propagation). This assumption of the normal distribution is also used for the dispersion of the referenced MC samples.

The determined values for the measurement uncertainty of the LiDAR from the VCE are used to generate the MC samples for each point of the point cloud. These samples are propagated through the complete processing chain to estimate the point uncertainty in the r-frame. The transformations are divided into two categories, static and kinematic. The static transformations are the system calibrations and the stationing. For these transformations, the realizations are generated once in the beginning and then used for all points. This introduces the correlation between points of the same realization due to the same deviations in the calibration. The transformation parameters for the kinematic tracking are interpolated by prediction based on the time stamps of the individual points and thus, the realizations of the transformation parameters are generated individually per point. The generation of random numbers is performed in Matlab [40]. The estimations of the transformation parameter yield fully populated VCMs, so the function “multivariate normal random numbers” (*mvnrnd*) is used to account for the correlations in the random number generation by using the Cholesky factorization [35].

After the complete chain, a mean coordinate $\bar{\mathbf{p}}$ and a VCM $\Sigma_{\bar{\mathbf{p}}\bar{\mathbf{p}}}$ for each point is derived. This is done similarly to the computation in equation (25). Both of these quantities are useful. The VCM provides the predicted uncertainty information and the computed mean coordinates can be used as an alternative method of referencing (ensemble referencing). The improvement in the accuracy using ensemble referencing is shown in Section 3.2.

2.2 Kinematic experiment

The kinematic experiment takes place in the same laboratory as the calibration to ensure similar conditions. Figure 5 shows an overview of the laboratory for the experiment. The movement of the run is shown

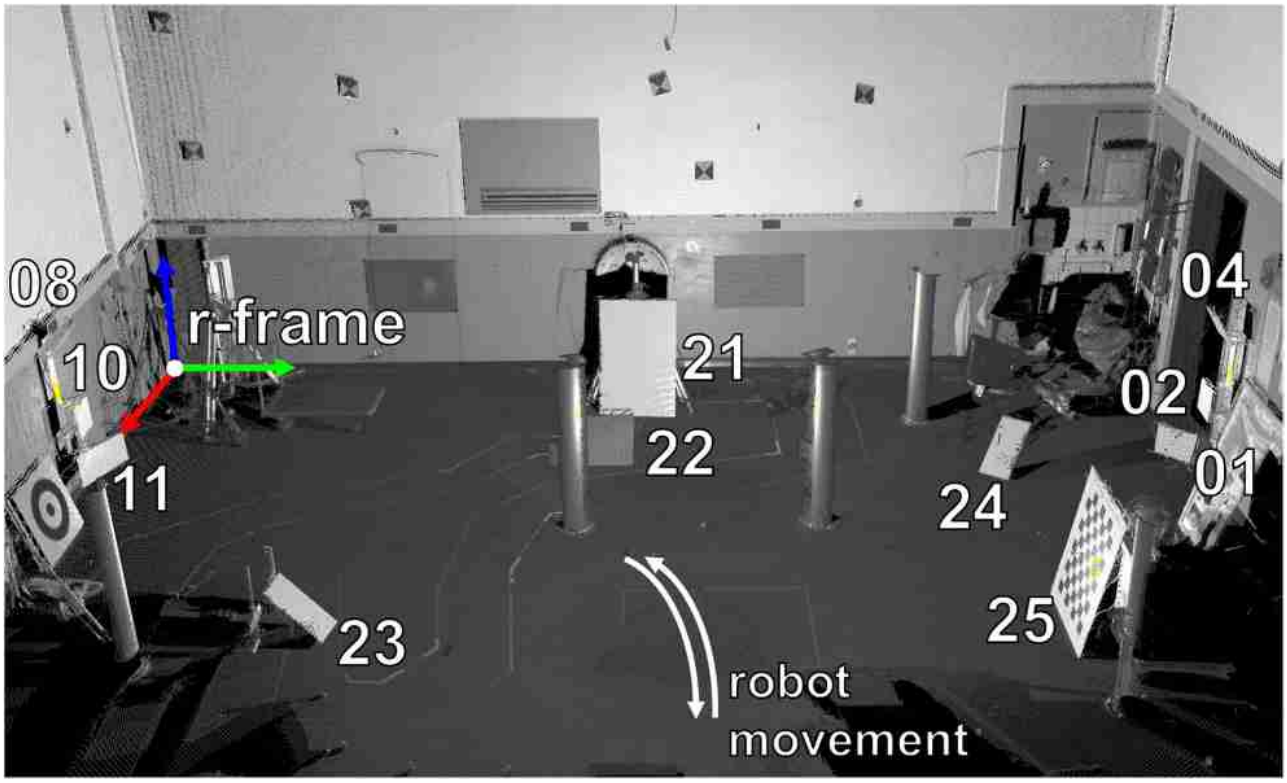

Figure 5: Overview setup for the kinematic experiment with r-frame, numbers of reference panels and the location of the movement of the robot. The image is produced from a scan by a TLS.

by the white arrows. It starts in a stationary position for 5 s, then the forward movement takes 15 s. After 5 s of standing, the backward movement takes 20 s, after which the robot stands for a few more seconds. The comparison of the time frame to the situation in the room shows, that the movement of the robot is quite slow for the chosen scenario. The maximum speed is about 0.2 $\frac{m}{s}$. The trajectory is limited by the viewing angle of the T-Probe.

For the evaluation of the experiment, multiple reference panels with the same surface properties are used. All panels are alucore with a white surface, apart from panel 25, which is a checkerboard used for camera calibrations. Some of the panels are part of the calibration field (01–11) for the system calibration of the LiDAR (cf. Section 2.1.1) and others are specifically set up to evaluate the kinematically acquired point cloud (21–25). Similar to extrinsic calibration, the panels are referenced by the LT. The constellation of the panels enables to detect deviations from all parameters of the 6-DoF pose.

The second aspect of sensor data fusion is time synchronization, which is only presented shortly here, since it is mainly a technical requirement. Two sensors are involved in the kinematic experiment, the Velodyne VLP-16 LiDAR and the Leica AT960-LR. The measurements of both sensors need to have timestamps in the same time system. Usually, GPS time is chosen, since it is widely available and very accurate [41]. For the experiment in this paper, the GPS antennas of the sensors are placed outside the window.

Both, the VLP-16 and the LT are provided with GPS time stamps by separate GPS receivers. The influence on the uncertainty of the referenced point cloud is negligible since a time delay of 1 ms would lead to deviations of up to 0.2 mm in the performed experiment. The used GPS receivers provide time stamps multiple orders of magnitude more accurate.

3 Results

This section is divided into three parts. First, the results of the calibration are shown and discussed. Next, the results of the referencing are presented and the difference between the ordinary referencing is compared to ensemble referencing by using the MC samples. Finally, the evaluation of uncertainty modeling is demonstrated.

3.1 Results of the system calibration

While the results of the uncertainty modeling for the referencing are specific to the presented processing chain, the results of the system calibration for the LiDAR on the platform are easily transferable to other applications. Since the Velodyne VLP-16 is a widely used LiDAR, the results in the estimated intrinsic parameters, the distance offset and the standard deviations for the measurement elements, are discussed here.

The estimation of the distance offset has two advantages. The distance offset affects all measurements of the LiDAR, which means determining this value improves the quality of the acquired point clouds in general. Additionally, the system calibration procedure directly uses the observations of the LiDAR for the determination of the transformation parameters. Hence, the quality improvement for the point cloud also leads to an improvement of the estimated transformation parameters.

Table 1: Results of the VCE from the system calibration of the LiDAR.

Measurement type	Standard deviation
Distance	8.5 mm
Horizontal angle	0.85 mrad
Vertical angle	0.52 mrad

The estimated distance offset is −14.0 mm with a standard deviation of 0.2 mm. The value is determined for all rings, which simplifies the actual sensor due to the actual construction with separate laser diodes for each ring [2]. This could lead to unmodeled deviations later. The distance offset is valid for the experiments presented in this paper. From experience with multiple sensors of the same type, which were calibrated for different experiments and measurement campaigns, the value for the distance offset can vary between −20 to 20 mm between different sensors, but also for the same sensor over time (months to years). Similar magnitudes are also determined in analyses by other authors [30, 42].

The results of the VCE are shown in Table 1. These values influence the uncertainty modeling due to the usage for the generation of the MC samples in the processing chain.

3.2 Evaluation of the referencing

The evaluation of the referenced point clouds is realized by comparison to the reference panels distributed in the room. The usage of the reference panels ensures the assumptions of the VCE are correct. An evaluation using different surfaces, especially darker ones, would lead to higher uncertainties, which are not modeled at the current stage. While the referencing is performed pointwise, so nearly for continuous time, the point clouds are evaluated on a frame-by-frame basis. A frame describes a full rotation of the LiDAR. The rotation rate of the LiDAR is set to 600 RPM, so the distances of the points to the panels are combined in 0.1 s time steps.

The point clouds are automatically segmented to extract points on the reference panels. The borders of the panels are excluded to reduce mixed pixel effects. The orthogonal distances of these points to the reference panels are computed and shown in the following figures as quantiles since single points are strongly influenced by the measurement noise of the LiDAR. Figure 6 shows the results for four of the eleven reference panels.

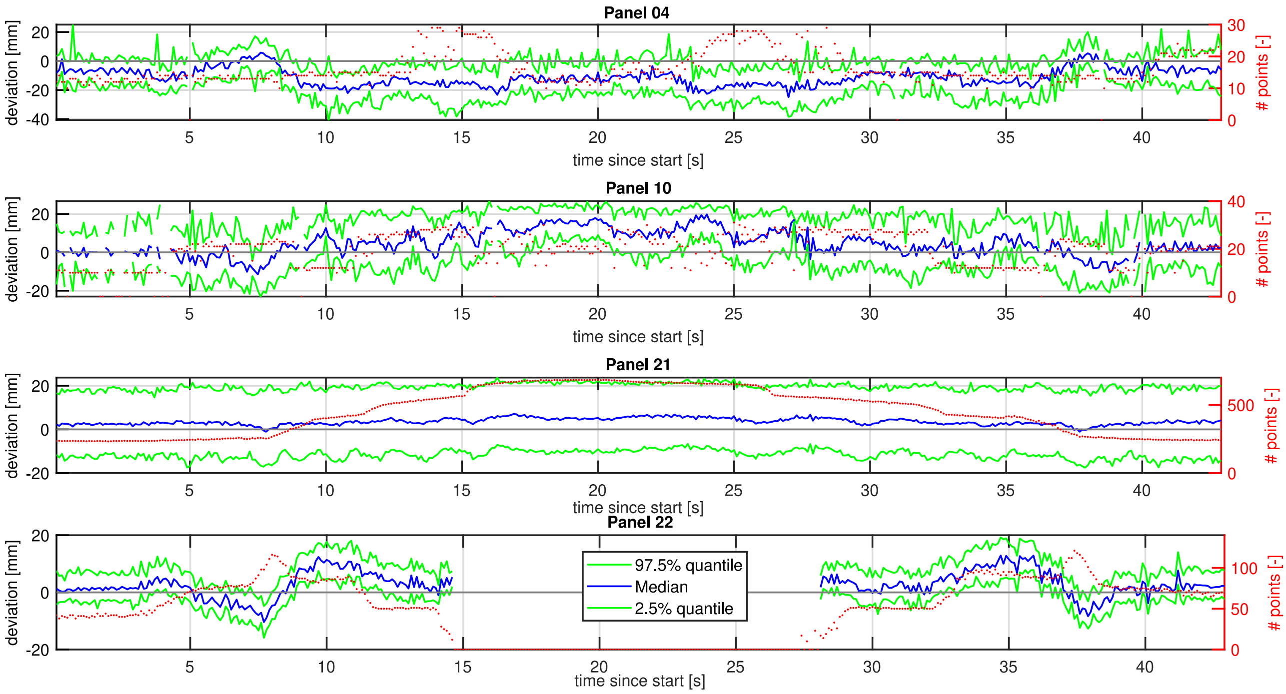

Figure 6: Median and quantiles of distances between point clouds and reference panels over time, panel numbers from Figure 5, number of points assigned to panel in red.

In addition to the quantiles of the distances, the plot shows the number of points in red, which are assigned to the panels. Due to the different sizes of the panels and different distances to the moving robot, the number of points varies between panels and over time. In situations, in which fewer than 10 points are assigned to a panel, the evaluation for the panel is skipped. Due to the low number of points on some panels, the median and quantiles of the deviations can be noisy.

The shown panels are representative to give an impression of the reached deviations for the point clouds. All 6-DoF have an impact on the distances to these four reference panels. In total, the maximum median deviation on a panel is about 2 cm. The RMSE of the deviations for all eleven panels is 10.0 mm.

The deviations of the median are caused by remaining deviations in the processing chain. These deviations can be divided into systematic deviations, which are constant over the whole time series and time-varying deviations. An example of the constant effect is the positive offset for panel 21 of about 5 mm. This could be explained by either a deviation in the translation in the y-axis of the system calibration, a deviation in the referencing by the LT or a combination of multiple factors. Panel 10 shows an increasing offset depending on the position of the robot. The largest offset appears around 20–25 s, when the robot has moved furthest into the room. This effect is most probably caused by the remaining deviations from the system calibration of the robot. In general, the causes for these deviations are most likely to be found in the first steps in the processing chain, as they are influenced by the measurements of the LiDAR, which is about two orders of magnitude less accurate than the LT.

An example of the time-varying deviations are visible for panels 04 and 22, which show peaks around 7 s and 34 s, respectively. At the same time, there is a negative correlation between the number of points and the deviations observed in this panel. The higher deviations can not be attributed to the additional points, as evidenced by the 2.5 % and 97.5 % quantiles which also reflect an increase in deviation. These peaks are probably related to the start of the movement of the robot and caused by unmodeled effects in the movement.

The quantiles are shown to give an impression of the dispersion of values. For panel 21, the values show an interval of about 3 cm. In contrast, the interval is only about 1 cm for panel 22. This is mainly influenced by the constellation between LiDAR and the panel. A smaller angle of incidence leads to a larger dispersion in the values caused by the higher noise for the distance measurements in comparison to the angle measurements for short distances.

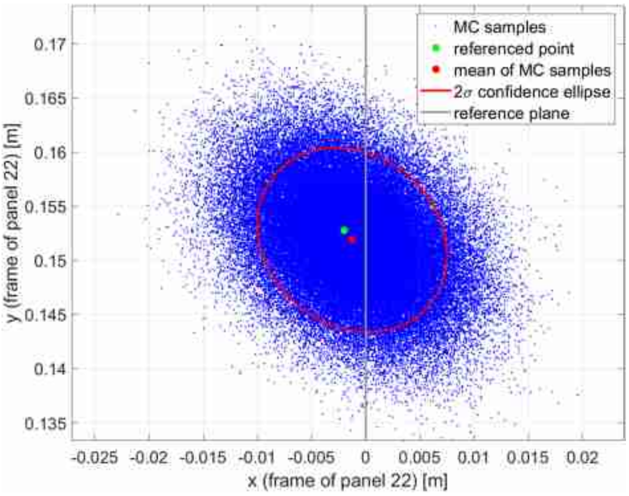

Figure 7: Comparison of a referenced point (green) and the ensemble referenced point (red, with MC samples in blue) transformed in the frame of the corresponding panel 22.

For the prediction of the point uncertainties, the MC variance propagation is used. As an alternative to the ordinarily computed referenced point clouds, the mean of the MC samples can be used to compute the point positions in the *r*-frame. This can be seen as an ensemble referencing approach. The usage of the MC samples includes the uncertainty information in the referencing improving the accuracy of the resulting point cloud. Figure 7 shows an example of the difference between the referenced point and the mean of the MC samples as the result of the ensemble referencing.

For this example, the mean of the MC samples is about 1 mm closer to the panel than the referenced point. The magnitude of the different between the two referencing methods varies for each point. For some points, the ensemble referencing even leads to a higher distance, but for all points, the ensemble referencing decreases the RMSE of the deviations from 10.0 mm to 8.1 mm. The resulting deviation plot is shown in Figure 8.

In comparison to the deviations in Figure 6, a clear improvement can be seen. For all panels, the median deviation is similar or smaller than for the ordinarily referenced points. For panels 04 and 22, the peaks at 7 s and 37 s are no longer apparent. For panel 10, the offset in the middle of the trajectory is reduced by about 10 mm. This improvement is caused by the additional information introduced by the MC sampling in form of the uncertainty information, reducing the linearization bias.

3.3 Evaluation of the uncertainty modeling

For the evaluation of uncertainty modeling, multiple approaches are possible. A possible approach would be to compute a 3D point uncertainty measure like the Helmert point error (trace of the VCM). This can be helpful to

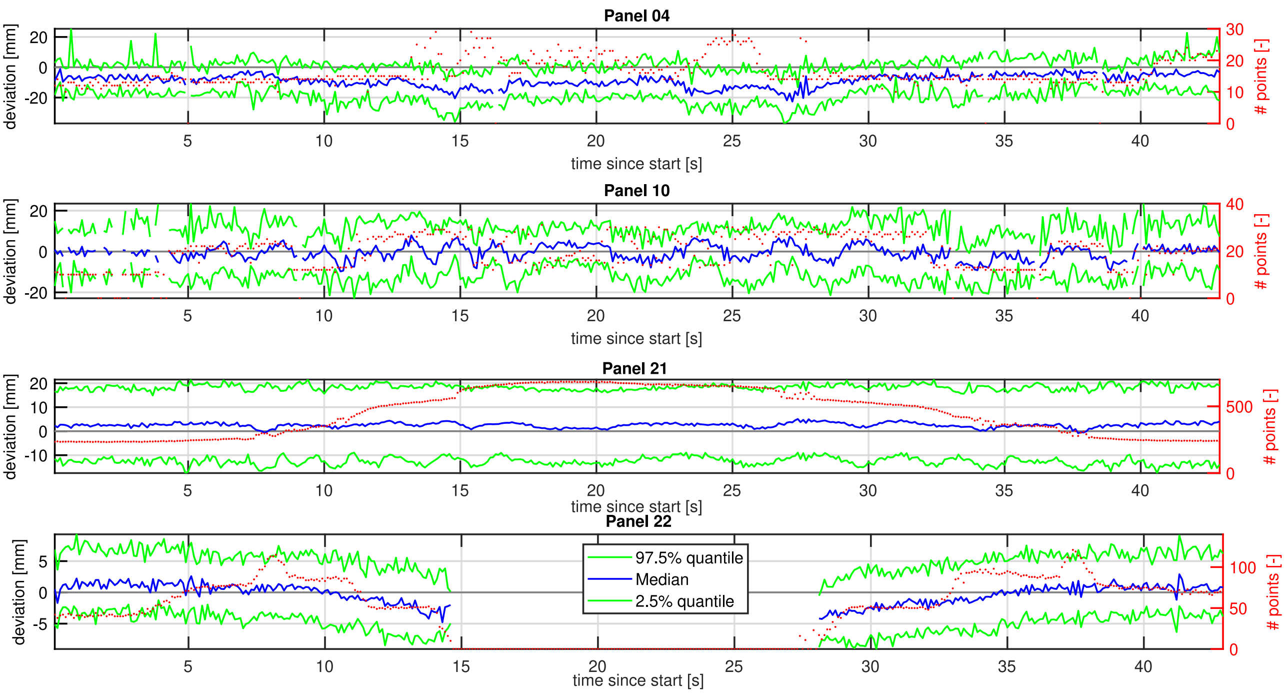

Figure 8: Median and quantiles of distances between point clouds and reference panels over time, ensemble referencing using MC samples, panel numbers from Figure 5, number of points assigned to panel in red.

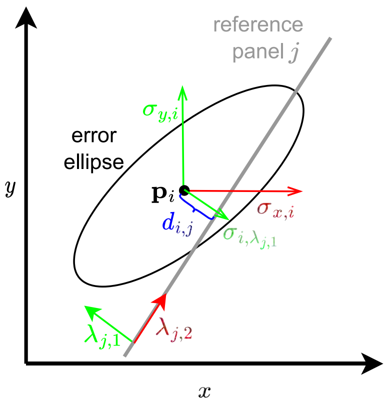

Figure 9: Principle of the evaluation for the predicted point uncertainties in comparison to the reference panels from the ground truth. Relevant frames are shown for superordinate frame and reference panel.

illustrate the point uncertainty [6]. For the quantitative evaluation, these values will always be too large, since all directions contribute to the uncertainty. This is contrary to the evaluation, which is performed by computing the distance along the normal vector of the reference panels.

To achieve a correct evaluation of the predicted uncertainties, the actual distances $d_{i,j}$ are standardized by the predicted standard deviation in the direction of the computed distance to the reference panel j. The principle behind this idea is shown in Figure 9. In the figure, the error ellipse is used as an illustration of the predicted VCM [7]. The goal is to determine the predicted standard deviation $\sigma_{i,\lambda_{j,1}}$ for point i.

This standard deviation is computed by rotating the VCM of the referenced point using the matrix of eigenvectors $\mathbf{M}_j$ of the reference panel j. Additionally, the matrix $\mathbf{\Lambda}_j$ contains the eigenvalues λ_i. These matrices are known since they are used to compute the plane parameters for the reference panels from point measurements of the LT using the error matrix $\mathbf{E}$ [43]

$$\mathbf{E}_j^T\mathbf{E}_j = \mathbf{M}_j\mathbf{\Lambda}_j\mathbf{M}_j^T, \tag{36}$$

$$\mathbf{\Sigma}_{\mathbf{pp},j} = \mathbf{M}_j\mathbf{\Sigma}_{\mathbf{pp},r}\mathbf{M}_j^T. \tag{37}$$

After the rotation, the predicted VCM $\mathbf{\Sigma}_{\mathbf{pp}}$ of the point $\mathbf{p}$ refers to the frame of the reference panel j. Thus, the square root of the diagonal value corresponding to the smallest eigenvalue $\lambda_{j,1}$ is the predicted standard deviation $\sigma_{i,\lambda_{j,1}}$ for the distance. The standardized distances $\tilde{d}_{i,j}$ can be computed by

$$\tilde{d}_{i,j} = \frac{d_{i,j}}{\sigma_{i,\lambda_{j,1}}}. \tag{38}$$

The illustration of the evaluation for the uncertainty modeling is performed similarly to the evaluation of the referencing. Figure 10 shows the results for the selected panels.

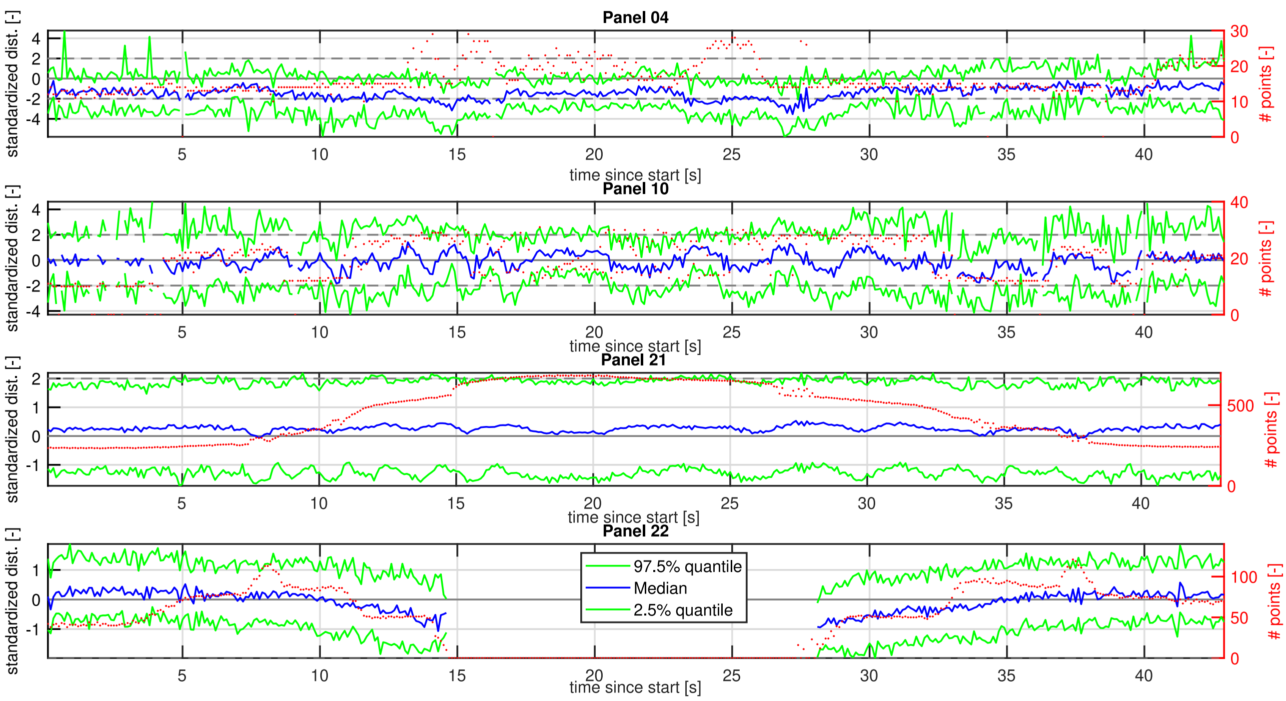

Figure 10: Median and quantiles of standardized distances between point clouds and reference panels over time, ensemble referencing using MC samples, panel numbers from Figure 5, number of points assigned to panel in red.

The standardization of the distances leads to a rescaling of the values, keeping the general impression of the deviations from Figure 8. Using the assumption of normal distribution for the input quantities, the resulting distribution of the distances should follow a *t*-distribution [35] with the number of points as the degrees of freedom. The shown quantiles at 2.5 % and 97.5 % enclosing 95 % of the distances should be in the interval $t_{[2.5\,\%,97.5\,\%],700} = [-1.96, 1.96]$ for the maximum number of points on the panels (shown as dashed lines) and $t_{[2.5\,\%,97.5\,\%],10} = [-2.23, 2.23]$ for the smallest number of points evaluated.

The deviation being enclosed by the intervals can be seen for the points on panels 21 and 22. Especially for panel 22, the dispersion for the standardized distances is smaller than the interval. This does not necessarily imply too pessimistic modeling, since the dispersion of the points is mainly influenced by the measurement noise of the LiDAR. The estimated uncertainties for the referenced points additionally include the uncertainty of the rest of the processing chain. The influence of all these uncertainties is not visible on one panel.

For panels 04 and 10, the quantiles of the standardized distances exceed the aforementioned intervals. Overall panels, a trend can be seen. For smaller panels, which are also farther away, the interval is more likely to be exceeded. This could be caused by too optimistic uncertainty estimations for the orientations since they have a larger impact on longer distances.

4 Discussion

The evaluation of the kinematic experiment shows two aspects of the variance propagation using MC samples. The deviations of the point cloud to the reference panels are decreased when using MC samples for ensemble referencing and the uncertainty modeling shows good results for some and slightly too optimistic results for the other reference panels. These aspects are first discussed for the system used. Afterwards, the transferability to other systems is considered.

Concerning the referencing itself, the ordinary referencing corresponds to the Gaussian variance propagation, for which the variance propagation is performed by using Jacobian matrices, similar to the computation of the cofactor matrix of the misclosures in equation (12). The estimated uncertainties have no influence on the computation of the point coordinates. This leads to a linearization bias in the case of nonlinear functions, which deteriorates the quality of the referenced point cloud. The MC method and UT avoid this problem by applying the functional model of the processing chain to randomly (MC) or deterministically (UT) generated samples. This yields better results by integrating the uncertainty information.

For the shown experiment, the difference between ordinary referencing and ensemble referencing is in the millimeter range and the MC samples are still normally distributed (tested along the coordinate axes using an Anderson-Darling test [44]), even after multiple transformations in the processing chain.

For the uncertainty modeling, the MC variance propagation shows a slight mismatch between the predicted uncertainties and the actual deviations. For the assumed normal distributions in this paper, a Gaussian variance propagation would achieve similar results. The main advantage of the MC method is the flexibility towards the used distributions, in addition to the avoidance of the linearization bias. The presented processing chain can be extended for various influences with arbitrary distributions to properly model the uncertainties. Possible extensions could include the use of the uniform distributions for quantization of the distance measurements for the used LiDAR [2], or incorporating models to represent uncertainties in step widths, as demonstrated in [18].

The exceeded intervals for the smaller panels can have multiple causes. The length of the processing chain complicates the search for possible reasons. The different levels of accuracy of the involved sensors make it more likely, that the mismatch is caused by insufficient knowledge of the uncertainty model of the LiDAR. Due to this large influence on the results, the first steps of the chain are more promising for further optimizations. The uncertainties for the orientations in the system calibration could be too optimistic, which would mainly be visible on the smaller panels because they are farther away. This could be the result of the VCE, which estimates the stochastic parameter over all rings, generalizing possible differences between the scan lines. Another plausible explanation could be the omission of correlations in the processing chain, leading to a too optimistic uncertainty modeling. Particularly, subsequent LiDAR observations exhibit high correlations, but due to limited knowledge regarding their magnitude, these correlations are not introduced for the modeling process

The choice of the MC variance propagation is mainly motivated by the possibility to expand the model by any function or distribution. This flexibility enables the investigation of the various assumptions for the processing chain. But the usage of the MC techniques comes at a large computational cost. In this case, each point of the point cloud is modeled by 100,000 samples causing a huge amount of computations to predict the uncertainty for the point cloud.

On a laptop with an Intel i7-7700HQ with 16 GB of RAM, the Matlab implementation takes 12 min to compute a complete frame with about 30,000 points.

4.1 Transferability to other systems

While the goal of this paper is to demonstrate the pipeline for an experimental system, the transferability to other systems is also relevant. Adapting the presented methodology to other systems requires multiple adaptions. The system used in this study is unique in its design due to significant variations in sensor measurement accuracy. While the VLP-16 is commonly used low-cost LiDAR, the LT with T-Probe is rarely used for georeferencing systems. In most cases, MSSs are georeferenced using GNSS and IMU, optionally supported by LiDARs and/or cameras. Although these setups offer more flexibility, they tend to be less accurate.

Furthermore, methods that utilize LiDAR or camera data for indirect georeferencing are subject to a feedback loop of the uncertainties. Incorrect assumptions about the uncertainties of the georeferenced data can influence the pose estimation, which in turn affect the uncertainties of the georeferenced data, creating a cycle. In the applied methodology, less accurate referencing would lead to larger distortions of the distributions. As a consequence, investigations would be necessary to analyze, whether the normal distribution is a suitable choice for the point positions and uncertainties. If this is the case, using UTs could be a compromise, as they can help avoid linearization biases with relatively few samples [14].

Another challenge when transferring the methodology to other systems is the prevalence of distributions other than the normal distribution. Most common distributions can be transformed into normal distributions [15], but the quality of the results has needs to be investigated. One example of such alternative distributions are multipath effects of GNSS signals. While receivers are able to identify reflected signals delayed by more than a quarter of the wavelength, shorter reflections induce uncertainties that must be modeled by wide uniform distributions (about 5 cm for carrier phase observations, multiple meters for code observations) [45]. Depending on the processing chain, transforming this distribution might not yield satisfactory results, necessitating the use of MC methods.

5 Summary and outlook

This paper presents a processing chain to accomplish two things. On one hand, the point clouds acquired by LiDAR in a kinematic experiment are referenced using ensemble referencing with MC samples. On the other hand, the uncertainty of the referenced points is predicted under consideration of the uncertainties from all steps in the processing chain. The uncertainty modeling is performed by an MC variance propagation as proposed by the GUM.

The processing chain is established by multiple calibrations, each of which yields transformation parameters and VCMs for the uncertainty modeling. The results for the platform calibrations are transferable to different applications. Especially the system calibration of the LiDAR can be used for various applications and the uncertainty estimation for the measurements of the LiDAR is useful for a variety of algorithms.

The results of the ensemble referencing show an improvement by using the mean of the MC samples compared to computing the referenced points using the original measurements only. This can be explained by the reduction of linearization bias due to the knowledge introduced by the modeled uncertainties from the individual steps in the processing chain.

For the uncertainty modeling, further investigations in the sensor model can improve the system calibration and reduce the effect of the generalizations [30], which are probable reasons for the mismatch between prediction and actual deviations. This could help to better calibrate the uncertainty model to match the confidence to the actual accuracy. Furthermore, the parametrization of the processing steps should be further investigated. Currently, all rotations are represented using Euler angles. However, for rotations with higher uncertainties, such as those that my arise from less accurate referencing methods, quaternions might yield better results due to the avoidance of trigonometric functions.

Further steps to improve the model for more widely used systems should include extending the model with additional sensors and different influences, such as different surfaces, longer distances, correlations between observations, etc. In the long term, the model should also be applicable to outdoor measurements, which means a significant increase in influencing factors. These new factors and also worse referencing methods could lead to more distorted point distributions that are no longer normally distributed. Therefore, other methods of estimating point positions and uncertainties need to be investigated. Only after these aspects have been solved or at least thoroughly investigated, attempts at reducing the number of samples should be made, as the analysis of the shape of distribution requires larger sample sizes than just estimating mean and variance [46].

Acknowledgments: The authors would like to thank Aaronkumar Ehambram and Raphael Voges of the Real

Time Systems research group (RTS) of the Institute for Systems Engineering at the Leibniz University Hannover for providing access to the robot and the support with the measurements.

Research ethics: Not applicable.
Author contributions: DE, SV, HA, IN; Experiments, Coding, Processing, Writing: DE; Proof-Reading: SV, HA, IN; Project Supervision: HA, IN. The authors have accepted responsibility for the entire content of this manuscript and approved its submission.
Competing interests: The authors state no conflict of interest.
Research funding: This work was funded by the German Research Foundation (DFG) as part of the Research Training Group i.c.sens [RTG 2159].
Data availability: The raw data can be obtained on request from the corresponding author.

References

1. Vogel S, Alkhatib H, Neumann I. Accurate indoor georeferencing with kinematic multi sensor systems. In: 2016 international conference on indoor positioning and indoor navigation (IPIN); 2016:1−8 pp.
2. Velodyne. Velodyne LiDAR, Inc. VLP-16 datasheet. San Jose, CA, USA: Velodyne; 2019.
3. Bureick J, Vogel S, Neumann I, Unger J, Alkhatib H. Georeferencing of an unmanned aerial system by means of an iterated extended kalman filter using a 3D city model. PFG−J Photogrammetry, Rem Sens Geoinf Sci 2019;87:229−47.
4. Schlichting A. Fahrzeuglokalisierung durch automotive Laserscanner unter Verwendung statischer Merkmale [Ph.D. thesis]. vol. 826 of Reihe C. München: DGK; 2018.
5. Geneva P, Eckenhoff K, Yang Y, Huang G. LIPS: LiDAR-inertial 3D plane SLAM. In: 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS): IEEE; 2018:123−30 pp.
6. Stenz U, Hartmann J, Paffenholz JA, Neumann I. High-precision 3D object capturing with static and kinematic terrestrial laser scanning in industrial applications—approaches of quality assessment. Rem Sens 2020;12:290.
7. Niemeier W. Ausgleichungsrechnung: statistische auswertemethoden. Berlin and New York: Walter de Gruyter; 2008.
8. Hofmann S, Brenner C. Accuracy assessment of mobile mapping point clouds using the existing environment as terrestrial reference. Int Arch Photogram Rem Sens Spatial Inf Sci 2016;41:601−8.
9. Shi B, Bai Y, Zhang S, Zhong R, Yang F, Song S, et al. Reference-plane-based approach for accuracy assessment of mobile mapping point clouds. Measurement 2020;171:108759.
10. Julge K, Vajakas T, Ellmann A. Performance analysis of a compact and low-cost mapping-grade mobile laser scanning system. J Appl Remote Sens 2017;11:1−23.
11. Al-Durgham K, Lichti DD, Kwak E, Dixon R. Automated accuracy assessment of a mobile mapping system with lightweight laser scanning and MEMS sensors. Appl Sci 2021;11:1007.
12. Dreier A, Janßen J, Kuhlmann H, Klingbeil L. Quality analysis of direct georeferencing in aspects of absolute accuracy and precision for a UAV-based laser scanning system. Rem Sens 2021;13:3564.
13. Hexagon Metrology. Leica absolute tracker AT960; 2023. Available from: https://hexagon.com/products/leica-absolute-tracker-at960.
14. Perez DA, Gietler H, Zangl H. Automatic uncertainty propagation based on the unscented transform. In: 2020 IEEE international instrumentation and measurement technology conference (I2MTC): IEEE; 2020:1−8 pp.
15. JCGM. JCGM 100: evaluation of measurement data − guide to the expression of uncertainty in measurement (GUM). Sèvres, Cedex: Joint Commitee for Guides in Metrology (JCGM/WG1); 2008.
16. Niemeier W, Tengen D. Uncertainty assessment in geodetic network adjustment by combining GUM and Monte-Carlo-simulations. J Appl Geodesy 2017;11:67−76.
17. Koch KR. Determining uncertainties of correlated measurements by Monte Carlo simulations applied to laserscanning. J Appl Geodesy 2008;2:139−47.
18. Alkhatib H, Kutterer H. Estimation of measurement uncertainty of kinematic TLS observation process by means of monte-carlo methods. J Appl Geodesy 2013;7:125−34.
19. Ehrhorn A. Unsicherheitsmodellierung eines k-TLS basierten Multi-Sensor-Systems [master thesis (unpublished)]. Hannover: Leibniz Universität Hannover; 2019.
20. JCGM. JCGM 101: evaluation of measurement data − supplement 1 to the "Guide to the expression of uncertainty in measurement" − propagation of distributions using a Monte Carlo method. Sèvres, Cedex: Joint Commitee for Guides in Metrology (JCGM/WG1); 2008.
21. Clearpath Robotics, editors. Husky unmanned ground vehicle robot; 2021. Available from: https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/.
22. Schön S, Brenner C, Alkhatib H, Coenen M, Dbouk H, Garcia-Fernandez N, et al. Integrity and collaboration in dynamic sensor networks. Sensors 2018;18:2400−21.
23. Zhou L, Li Z, Kaess M. Automatic extrinsic calibration of a camera and a 3d lidar using line and plane correspondences. In: 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS); 2018:5562−9 pp.
24. Strübing T, Neumann I. Positions- und Orientierungsschätzung von LIDAR-Sensoren auf Multisensorplattformen. ZFV − Z Geodasie, Geoinf Landmanagement 2013;3:210−21.
25. Hartmann J, Paffenholz JA, Strübing T, Neumann I. Determination of position and orientation of LiDAR sensors on multisensor platforms. J Survey Eng 2017;143:1−11.
26. Heinz E, Holst C, Kuhlmann H, Klingbeil L. Design and evaluation of a permanently installed plane-based calibration field for mobile laser scanning systems. Rem Sens 2020;12:555.
27. Vogel S, Ernst D, Neumann I, Alkhatib H. Recursive Gauss-Helmert model with equality constraints applied to the efficient system calibration of a 3D laser scanner. J Appl Geodesy 2022;16: 37−57.

28. Hartmann J, von Gösseln I, Schild N, Dorndorf A, Paffenholz JA, Neumann I. Optimisation of the calibration process of a k-TLS based multi-sensor-system by genetic algorithms. In: The international archives of the photogrammetry, remote sensing and spatial information sciences. Copernicus GmbH; 2019:1655−62 pp.
29. Chan TO, Lichti DD, Roesler G, Cosandier D, Al-Durgham K. Range scale-factor calibration of the Velodyne VLP-16 LiDAR system for position tracking applications. In: ISPRS, IAG, FIG, editors. 2019 11th international conference on mobile mapping; 2019:70−7 pp.
30. Glennie CL, Kusari A, Facchin A. Calibration and stability analysis of the VLP-16 laser scanner. Int Arch Photogram Rem Sens Spatial Inf Sci 2016;XL-3/W4:55−60.
31. Lenzmann L, Lenzmann E. Strenge Auswertung des nichtlinearen Gauß-Helmert-Modells. AVN Allg Vermessungs-Nachrichten 2004;5:68−73.
32. Heinz E, Eling C, Wieland M, Klingbeil L, Kuhlmann H. Development, calibration and evaluation of a portable and direct georeferenced laser scanning system for kinematic 3D mapping. J Appl Geodesy 2015;9:227−43.
33. Yu Z. A generalization theory of estimation of variance-covariance components. Manuscripta Geod 1992;17:295.
34. Heiker A. Mutual validation of earth orientation parameters of geophysical excitation functions and second degree gravity field coefficients. vol. 697 of Reihe C. München: DGK; 2013.
35. Koch KR. Introduction to bayesian statistics, 2nd ed. Berlin and Heidelberg: Springer; 2007.
36. Simon D. Optimal state estimation: Kalman, H infinity, and nonlinear approaches. John Wiley & Sons; 2006.
37. Hartmann J, Trusheim P, Alkhatib H, Paffenholz JA, Diener D, Neumann I. High accurate pointwise (geo-) referencing of a k-TLS based multi-sensor-system. ISPRS Ann Photogram Rem Sens Spatial Inf Sci 2018;4:81−8.
38. Bar-Shalom Y, Li XR, Kirubarajan T. Estimation with applications to tracking and navigation: theory algorithms and software. New York: John Wiley & Sons; 2004.
39. Lerman PM. Fitting segmented regression models by grid search. Appl Stat 1980;29:77.
40. MATLAB. version R2021a. Natick, Massachusetts: The MathWorks Inc; 2021.
41. Stempfhuber WV. Ein integritätswahrendes Messsystem für kinematische Anwendungen [Ph.D. thesis]. vol. 576 of Reihe C. München: DGK; 2004.
42. Sánchez DE, Pany T. Analysis and calibration of the VLP-16 LiDAR for automotive applications. In: Proceedings of the 32nd international technical meeting of the satellite division of the institute of navigation (ION GNSS+2019); 2019:2611−21 pp.
43. Drixler E. Analyse der Form und Lage von Objekten im Raum [Ph.D. thesis]. vol. 409 of Reihe C. München: DGK; 1993.
44. Anderson TW, Darling DA. Asymptotic theory of certain “goodness of fit” criteria based on stochastic processes. Ann Math Stat 1952;23:193−212.
45. Seeber G. Satellite geodesy: foundations, methods, and applications, 2nd ed. Berlin and New York: de Gruyter; 2003.
46. Neitzel F, Lösler M, Lehmann R. On the consideration of combined measurement uncertainties in relation to GUM concepts in adjustment computations. J Appl Geodesy 2022;16:181−201.

Dominik Ernst*, Sören Vogel, Ingo Neumann and Hamza Alkhatib

Empirical uncertainty evaluation for the pose of a kinematic LiDAR-based multi-sensor system

https://doi.org/10.1515/jag-2023-0098
Received October 31, 2023; accepted May 6, 2024;
published online May 22, 2024

Abstract: Kinematic multi-sensor systems (MSS) describe their movements through six-degree-of-freedom trajectories, which are often evaluated primarily for accuracy. However, understanding their self-reported uncertainty is crucial, especially when operating in diverse environments like urban, industrial, or natural settings. This is important, so the following algorithms can provide correct and safe decisions, i.e. for autonomous driving. In the context of localization, light detection and ranging sensors (LiDARs) are widely applied for tasks such as generating, updating, and integrating information from maps supporting other sensors to estimate trajectories. However, popular low-cost LiDARs deviate from other geodetic sensors in their uncertainty modeling. This paper therefore demonstrates the uncertainty evaluation of a LiDAR-based MSS localizing itself using an inertial measurement unit (IMU) and matching LiDAR observations to a known map. The necessary steps for accomplishing the sensor data fusion in a novel Error State Kalman filter (ESKF) will be presented considering the influences of the sensor uncertainties and their combination. The results provide new insights into the impact of random and systematic deviations resulting from parameters and their uncertainties established in prior calibrations. The evaluation is done using the Mahalanobis distance to consider the deviations of the trajectory from the ground truth weighted by the self-reported uncertainty, and to evaluate the consistency in hypothesis testing. The evaluation is performed using a real data set obtained from an MSS consisting of a tactical grade IMU and a Velodyne Puck in combination with reference data by a Laser Tracker in a laboratory environment. The data set consists of measurements for calibrations and multiple kinematic experiments. In the first step, the data set is simulated based on the Laser Tracker measurements to provide a baseline for the results under assumed perfect corrections. In comparison, the results using a more realistic simulated data set and the real IMU and LiDAR measurements provide deviations about a factor of five higher leading to an inconsistent estimation. The results offer insights into the open challenges related to the assumptions for integrating low-cost LiDARs in MSSs.

Keywords: uncertainty evaluation; uncertainty modeling; multi-sensor system; LiDAR; IMU

***Corresponding author: Dominik Ernst**, Geodetic Institute, Leibniz University Hannover, Hannover, Germany,
E-mail: ernst@gih.uni-hannover.de. https://orcid.org/0000-0001-9826-5610
Sören Vogel, **Ingo Neumann and Hamza Alkhatib**, Geodetic Institute, Leibniz University Hannover, Hannover, Germany,
E-mail: vogel@gih.uni-hannover.de (S. Vogel),
neumann@gih.uni-hannover.de (I. Neumann),
alkhatib@gih.uni-hannover.de (H. Alkhatib)

1 Introduction

Determining the position and orientation of systems is necessary for most of today's applications, i.e. mapping and navigation. As systems become more automated, reliable information on positioning quality is essential, since manual intervention becomes impossible. Properly weighting positioning information helps with further processing and ensures the safety of the environment and other people near the system.

As this weighting is highly dependent on the used sensors, their selection for the positioning greatly depends on the environment. Typically, systems utilize an inertial measurement unit (IMU) for continuous monitoring of the system's motion [1]. In case of outdoor applications, the Global Navigation Satellite System (GNSS) is used to provide absolute positions in the global reference frame. Nevertheless, GNSS observations suffer from various signal influences, including signal blockage and multipath effects [2]. Alternative localization strategies such as WiFi [3] or ultra-wide band (UWB) [4] can be employed for indoor positioning. Additionally, sensor data collected from systems, either by light detection and ranging sensors (LiDARs), cameras or other exteroceptive sensors, can assist in positioning [5]. This paper is focused specifically on matching point clouds measured by LiDARs to known maps.

As with GNSS, the observations gathered by LiDARs are impacted by various uncertainty factors like the atmosphere or surrounding objects, which pose challenges in modeling the process. The impact of these factors on the measurement quality is partially investigated for terrestrial laser scanners (TLSs) [6, 7]. Apart from the measurements of the sensor itself, sensor data fusion, particularly the fusion of the sensor data into a common frame, introduces additional uncertainties. This issue is linked to the measurement quality, as measurements of the LiDAR are often used to determine the necessary transformation parameters [8, 9]. Therefore calibrations are necessary. Intrinsic calibrations are used to correct the sensor measurements itself, while extrinsic calibration determine the transformations between the sensors. The combination of the observations also requires the synchronization of the sensor increasing the number of uncertainty sources [10]. However, even after determining the uncertainties, determining the magnitude to which these uncertainties affect subsequent processing steps, such as positioning, remains a significant challenge. The main difficulty arises from the fusion of data collected from different sensors in the georeferencing of the multisensor system (MSS), which forms a complex system that requires careful analysis [11].

1.1 Related works

Sensor data fusion can be achieved through various methods [5]. This paper will primarily focus on Kalman filters (KFs), though alternative algorithms such as factor graphs [12] or bundle adjustments [13, 14] are possible. By recursively estimating the current pose, KFs allow for efficient position tracking from a known initial pose [15]. Due to its versatile nature, this algorithm has been utilized in various applications, ranging from a basic fusion of IMU and GNSS observations [16, 17] to the incorporation of additional sensor modalities [18, 19]. This can be done with or without information about the environment (simultaneous localization and mapping (SLAM), [1]). When focusing on integrating LiDAR observations, different techniques can be employed. Either known landmarks can be observed and integrated [20, 21], or city models [22, 23] or point clouds [24, 25] of the environment can serve as an absolute position reference.

Different functional models can be utilized for integrating observations in the update step of the KF. The observations can either be incorporated loosely coupled [26, 27] or tightly coupled [19]. Modifications to the classical KF may be necessary in certain situations to facilitate these concepts [28, 29]. One strategy used in this paper involves the application of implicit measurement equations, which were first introduced in [30] and initially used for the trajectory estimation of a system in [22].

Estimating the calibration and correction parameters for the system and its sensors is another crucial aspect. Properly weighting the different observations for a consistent trajectory estimation requires careful consideration of these parameters and their uncertainties. Any incorrect estimates of uncertainties for the parameters can bias the estimation, leading to deviations in the trajectory. The various sensors require different calibration procedures [31–33], which are typically done prior to operation, but can also be repeated afterwards or even updated during operation [34].

The evaluation of kinematic MSSs has been done for different systems, like mobile mapping systems (MMSs) [35–37] or unmanned aerial vehicles (UAVs) [38]. The evaluation is usually based on the comparison of the data from the kinematic system to a ground truth derived by a system or sensor of superior accuracy. Mostly, these ground truths are derived by stationary sensors.

1.2 Contributions

This paper investigates the uncertainty evaluation of a kinematic trajectory of an IMU and LiDAR-based MSS. In the first step, the parameters necessary for the sensor data fusion are estimated through calibration measurements. These parameters include the transformation between the different sensor frames and the sensor noise characteristics parameters of the sensors. In the second step, these parameters and their associated uncertainties are incorporated into an error state Kalman filter (ESKF), which serves as the position-tracking algorithm. Positioning is achieved by predicting the system's motion using the IMU and updating the system by matching the LiDAR scans to a known map. The paper addresses the following key questions:

1. How can uncertainties of sensors and their transformations from the calibrations be modeled and integrated into the position tracking of a LiDAR-based MSS?
2. Do the modeled uncertainties lead to a consistent trajectory estimation using a real data set?

1.3 Outline

These questions are answered as follows: Section 2 details the data set and the methodology. The methodology includes calibrations and the filter algorithm utilized for position tracking. Additionally, as a baseline, the situation of the real setup is simulated. The results are shown in Section 3 with the two different simulations and the real data. After a summary, the paper is outlining future extensions and improvements.

2 Materials and methods

The section begins by introducing the dataset, which comprises multiple trajectories from a kinematic experiment. In the second part of this section, the methodology is explained, starting with the calibration measurements essential for establishing the parameters of the sensor data fusion. In addition, the implemented filter algorithm is described to provide insight into the integration of the uncertainty information into the utilized principle of matching the LiDAR scans to the known map for position tracking. Furthermore, the procedure for the simulations is detailed.

2.1 Materials

This section presents the data used to explore the research questions posed earlier. The data set comprises multiple trajectories of an MSS within a laboratory setting. Additionally, strategically positioned static panels throughout the room act as reference points, offering map information for the system.

The data set used in this paper is originally introduced in [39]. The robot depicted in Figure 1 is equipped with the MSS that contains a variety of sensors. Notably among them are a tactical grade IMU (Microstrain 3DM-GQ4-45 [40]), a Velodyne Puck LiDAR [41] and a Leica T-Probe observed by a laser tracker (LT) [42]. The T-Probe observations are utilized as ground truth to evaluate the filter performance. Spatial referencing and time synchronization are the two crucial factors for performing sensor data fusion and evaluating its outcomes. The high accuracy of the ground truth trajectory and availability of the stochastic information from the calibration are crucial for the evaluation conducted.

Figure 2 presents an overview of the experiment in the laboratory conducted after the calibrations (cf. Section 2.2.1) to ensure similar conditions. The panels, representing the known environment, are referenced using LT measurement with a corner cube reflector (CCR). Therefore, it is assumed that the plane parameters of the panels are deterministic. The trajectories, shown using differently colored lines, are limited by the viewing angle of the T-Probe. Each trajectory begins and ends with 3–5 s of stationary time. Between the stationary periods, the robot moves forward as described in Table 1 (same way backward). The maximum speed of the robot is about 0.4 m/s.

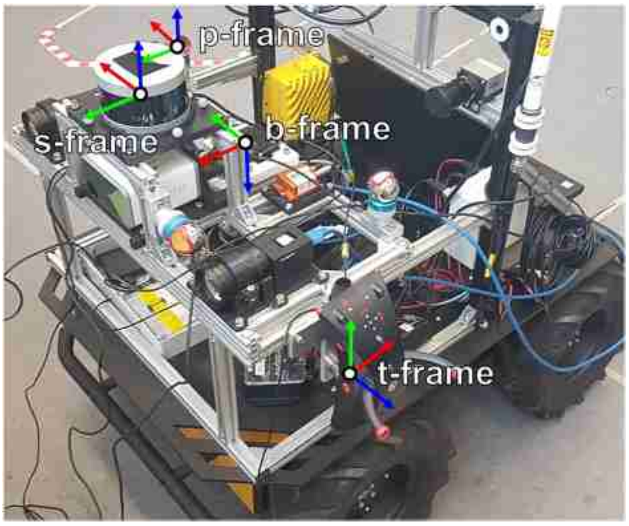

Figure 1: Robot carrying the MSS platform with IMU (b-frame), LiDAR (s-frame), T-Probe (t-frame) and CCRs for platform-frame (p-frame). Frames are shown with red for *x*-axis, green for *y*-axis and blue for *z*-axis.

To establish the time synchronization across all three involved sensors, each is equipped with GNSS antennas setup outside the laboratory to provide Global Positioning System (GPS) time information. While the manufacturer-provided solutions are used for the LiDAR and IMU, the LT receives a trigger signal provided from a GNSS receiver. In post-processing, the timestamps of the trigger signals and pose measurements are combined to determine the ground truth trajectory. The

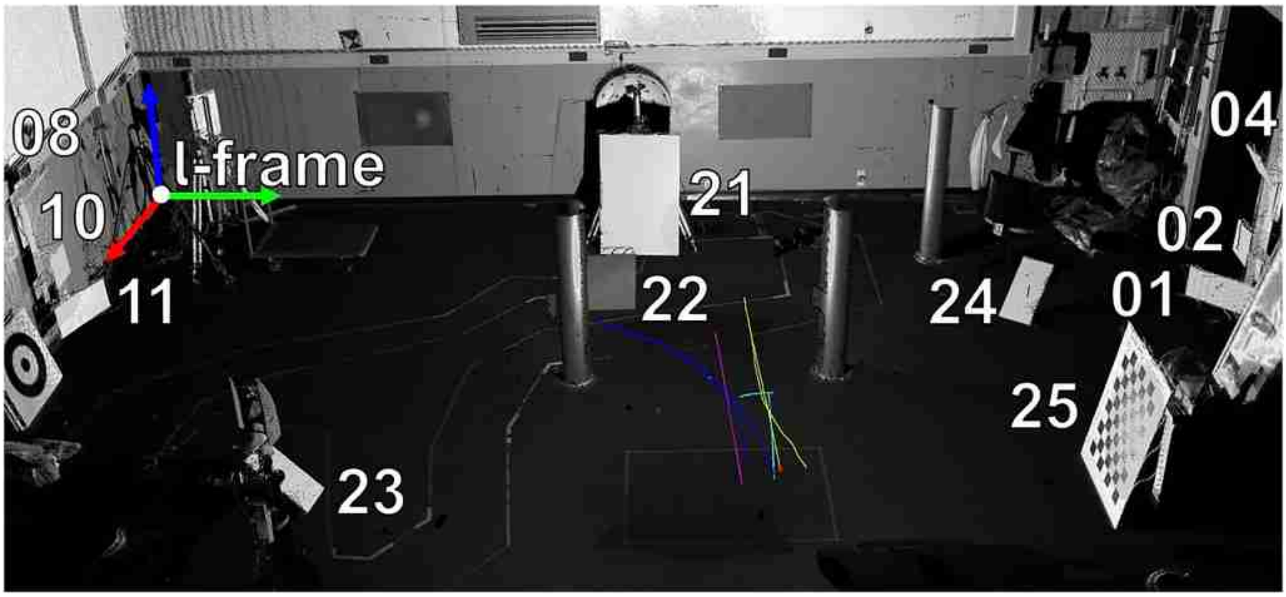

Figure 2: Overview of the laboratory setup for the experiment showing the l-frame and numbers of reference panels. Trajectories of body-frame are colored: red:1, yellow:2, cyan:3, blue:4, purple:6. The image is produced from a scan by a terrestrial laser scanner.

Table 1: Overview of the experiment trajectories with length and description of the movement. Traj. 5 is left out due to problems in the ground truth determination.

Traj. #	Length [s]	Maneuver
1	25	Standing (no movement)
2	36	Straight forward
3	50	Straight forward – turning on spot
4	45	Forward while turning left
6	13	Straight forward (fast)

Garmin 18× used for the LiDAR provides the least accurate time information with a quantified time uncertainty of 1 μs [43], resulting in a deviation of approximately 0.4 μm at the maximum speed of 0.4 m/s in the experiment. The effect of time synchronization on the uncertainties is therefore ignored.

2.2 Methodology

The methodology is divided into three parts. First, the calibration measurements are described to illustrate the origins of the essential parameters for sensor data fusion. Afterwards, the simulation procedure is outlined. Lastly, the employed position tracking algorithm, an ESKF, is introduced, encompassing the state vector, prediction, update (which includes the functional model for matching the LiDAR observations with the reference panels), integration of the uncertainty information, and the initialization of the filter.

2.2.1 Calibration: Multiple calibrations are necessary to determine all required parameters. Extrinsic calibrations are performed to determine the transformation parameters between the various frames of the used sensors. Intrinsic calibrations are necessary for correcting the measurements of specific sensors, such as LiDAR and IMU. Initially, the general extrinsic calibration is described, followed by a description of the setup for the LiDAR calibration. Finally, the section concludes with a summary of the intrinsic IMU calibration. As the calibrations are performed directly before the kinematic experiment, the values are assumed to be constant.

2.2.1.1 *Extrinsic calibration by LT measurements:* Using the extrinsic calibrations, the transformation parameters between the frames of the platform, the T-Probe, and the IMU are determined. The determination relies solely on LT measurements and reconstruction of the sensor frame for the IMU with respect to the sensor housing.

Each frame is defined individually. The T-Probe observation directly yields a complete set of six degrees of freedom (6-DoF), providing the sensor frame. The platform frame is defined by three CCRs: one for the origin and two for the x- and y-axis directions, respectively. The IMU's sensor frame is referenced by taking point measurements across the housing and reconstructing it based on manufacturer specifications. These measurements yield transformation parameters to the l-frame, in which the LT is located, as an intermediate step. The computation of the relative transformation between two frames f_1 and f_2 is calculated as:

$$\boldsymbol{H}_{f_1}^{f_2} = \left(\boldsymbol{H}_{f_2}^{l}\right)^{-1} \boldsymbol{H}_{f_1}^{l}. \tag{1}$$

Frames denoted in subscript and superscript represent the origin and target frames, respectively. The uncertainty determination is carried out using a Monte Carlo (MC) variance propagation, similar to the approach in [39]. The measurement uncertainty of the LT is assumed to be normally distributed with a precision of 7.5 μm + 3 ppm on a CCR. For T-Probe measurements, an extended 17.5 μm is included for the position or translation uncertainty, while the orientation uncertainty of the T-Probe is assumed to be 0.01° according to [42] (all typical uncertainties). Based on these assumed uncertainties, noise sampled from a normal distribution is introduced to the measurements for computing the transformation matrices for each realization (m). These matrices are decomposed as:

$$\boldsymbol{H}_{f_1}^{f_2,(m)} = \begin{bmatrix} \boldsymbol{R}_{f_1}^{f_2,(m)} & \boldsymbol{t}_{f_1}^{f_2,(m)} \\ \boldsymbol{0}_{[1\times3]} & 1 \end{bmatrix}, \tag{2}$$

collected as a vector $\boldsymbol{\gamma}^{(m)} = \left[\boldsymbol{t}_{f_1}^{f_2,(m)}, e\left\{\boldsymbol{R}_{f_1}^{f_2,(m)}\right\}\right]$ with orientations saved as Euler angles (shown by $e\{\cdot\}$) and used to compute the dispersion of the transformation parameters as a variance covariance matrix (VCM) as follows [44]:

$$\mathbf{E} = \begin{bmatrix} \boldsymbol{\gamma}^{(1)} - \bar{\boldsymbol{\gamma}} \\ \boldsymbol{\gamma}^{(2)} - \bar{\boldsymbol{\gamma}} \\ \vdots \\ \boldsymbol{\gamma}^{(M)} - \bar{\boldsymbol{\gamma}} \end{bmatrix}, \tag{3}$$

$$\boldsymbol{\Sigma}_{\gamma\gamma} = \frac{1}{M-1}\mathbf{E}^T\mathbf{E} \tag{4}$$

with the mean of the transformation parameter $\bar{\boldsymbol{\gamma}}$ and M as the total number of realization, here 100,000.

2.2.1.2 *Calibration of the LiDAR:* For the LiDAR, a different procedure is employed. While the sensor housing can be utilized to determine the transformation parameters, numerous studies have demonstrated enhanced calibration by leveraging object-space information captured by the sensor to determine the sensor frame's pose relative to the body frame [8, 10, 45]. In [8, 10], a distance offset for the LiDAR is also determined to refine the measurements. The detailed calibration procedure used is outlined in [46]; this section only offers a general overview.

The calibration principle is based on minimizing the distance of the transformed point clouds acquired by the LiDAR against known reference panels. Figure 3 provides an overview of the panels, while different system positions within the calibration field, as depicted in Figure 4, aim to enhance the results and decorrelate the parameters.

The functional model relies on a series of transformations necessary for establishing correspondences between points and panels. First, the LiDAR observations must be transformed into the Cartesian coordinates. The original measurement elements $\boldsymbol{\ell}_o$ for each point $\boldsymbol{p}_i$ include the distance d_i, the elevation angle e_i, and the azimuth angle α_i. In addition, the distance offset d_0 is determined. These elements are transformed into Cartesian coordinates following [41]:

$$\boldsymbol{g}_I(d_i, \alpha_i, e_i) = \boldsymbol{\ell}_i = \boldsymbol{p}_{s,i} = \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix}_i = \begin{bmatrix} (d_i + d_0)\cdot\cos e_i \cdot \cos\alpha_i \\ (d_i + d_0)\cdot\cos e_i \cdot \sin\alpha_i \\ (d_i + d_0)\cdot \sin e_i \end{bmatrix}. \tag{5}$$

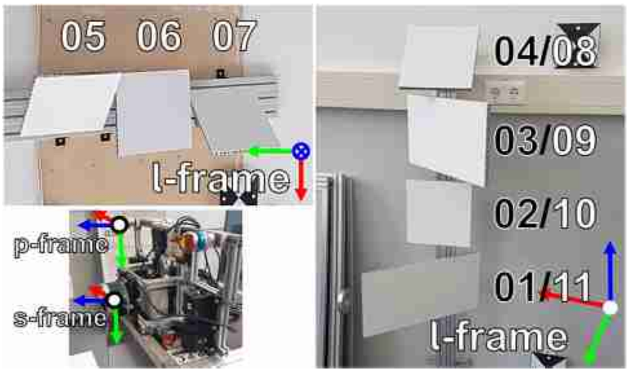

Figure 3: Overview of the calibration field for the LiDAR calibration. Left top: Panels on ceiling. Left bottom: Vertically tilted platform. Right: Panels on walls (mirrored on both sides).

Next, these points are transformed to the platform frame using the approximate (or in later iterations, estimated) transformation parameters $\hat{\boldsymbol{\gamma}}_s^b = \left[\hat{\boldsymbol{t}}_s^b \hat{\boldsymbol{R}}_s^b\right]$

$$\boldsymbol{g}_{II}\left(\boldsymbol{p}_{s,i}, \hat{\boldsymbol{\gamma}}_s^b\right) = \boldsymbol{p}_{b,i} = \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix}_i = \hat{\boldsymbol{t}}_s^b + \hat{\boldsymbol{R}}_s^b \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix}_i . \tag{6}$$

Subsequently to this, the points are further transformed using the platform pose $\boldsymbol{\gamma}_b^l = \left[\boldsymbol{t}_b^l \boldsymbol{R}_b^l\right]$ and assigned to the nearest panel in the calibration field $\boldsymbol{\alpha}_{l,j} = \left[n_{x,l}\, n_{y,l}\, n_{z,l}\, d_l\right]$. The platform's pose is determined from LT measurements, resulting in very low uncertainties. Finally, distances for all assigned points are computed for the implicit measurement equation of the adjustment

$$\boldsymbol{p}_{l,i} = \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix}_i = \boldsymbol{t}_b^l + \boldsymbol{R}_b^l \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix}_i , \tag{7}$$

$$h_i = h\left(\boldsymbol{\gamma}_b^l, \boldsymbol{\ell}_i, \boldsymbol{\alpha}_{l,j}\right) \tag{8}$$

$$= \left[n_{x,l}\, n_{y,l}\, n_{z,l}\, d_l\right]_j \cdot \begin{bmatrix} \boldsymbol{p}_{l,i} \\ 1 \end{bmatrix} . \tag{9}$$

In addition to estimating the transformation parameters and the distance offset, a variance component estimation (VCE) is performed to determine the uncertainties of the observations [47]. All observations of the three different types are grouped. The individual observations are assumed to be uncorrelated resulting in the matrix

$$\boldsymbol{\Sigma}_{\ell\ell,i} = \mathrm{diag}\left(\sigma_d, \sigma_\alpha, \sigma_e\right) \tag{10}$$

for each point i. The matrices of all points are then concatenated diagonally. The assumption of uncorrelated observations neglects actual correlations between subsequent points [6], but further investigation of the magnitude of these correlations exceeds the scope of this article. The resulting uncertainties for the transformation parameters and the observations are presented in Table 2. The correlations of the transformation parameters are omitted here. Detailed adjustment formulas can be found in [48].

Table 2: Sensor and transformation uncertainties determined from calibrations.

Transformation LiDAR-IMU for (x, y, z)-axes (standard dev.)		
Translation σ_t	(6.0, 1.8, 1.5)	mm
Orientation around axes σ_r	(1.1, 0.3, 0.2)	mrad
LiDAR (standard deviations)		
Distance measurement σ_d	8.5	mm
Azimuth angle σ_α	48.7	mdeg
Elevation angle σ_e	29.8	mdeg
Distance offset σ_{d_0}	0.2	mm
IMU for (x, y, z)-axes		
Acc. noise density $\boldsymbol{\sigma}_{\tilde{a}_n}$	$(6.2, 4.6, 3.6) \times 10^{-3}$	$\mathrm{m}\sqrt{\mathrm{Hz}}/\mathrm{s}^2$
Acc. bias instability $\boldsymbol{\sigma}_{a_w}$	$(22.3, 20.4, 36.4) \times 10^{-3}$	$\mathrm{m/s}^2$
Gyro. noise density $\boldsymbol{\sigma}_{\tilde{\omega}_n}$	$(2.6, 3.4, 2.4) \times 10^{-4}$	$\mathrm{rad}\sqrt{\mathrm{Hz}}/\mathrm{s}$
Gyro. bias instability $\boldsymbol{\sigma}_{\omega_w}$	$(12.3, 11.4, 11.9) \times 10^{-4}$	rad/s

2.2.1.3 ***Intrinsic IMU calibration:*** To incorporate the IMU in the sensor data fusion, its noise parameters must be determined. This is achieved through the Allan variance analysis [49, 50]. Only a brief period of approximately 25 s is available to compute the required values. These values are also shown in Table 2.

2.2.2 Simulation: To enrich the discussion of the results, the experiment setup and trajectories are simulated using a simple environment adapted from [39]. This simulation offers two key benefits. Firstly, it allows for the validation of the filter's performance to ensure its functionality. Secondly, it facilitates a comparison between the simulation results and the results of the actual measurements.

The simulated data is derived from the real experimental setup, incorporating stochastic information obtained from the calibrations (refer to Section 2.2.1). Consequently, the simulation results are based on correct assumptions, presenting an optimal scenario. By contrasting these results with those from real data, any differences can be more effectively analyzed.

The environment is reconstructed using the LT measurements of the panels placed in the laboratory. Sensor data for the IMU and LiDAR is generated by combining the T-Probe trajectories with the aforementioned measurements. Rotation rates $\tilde{\boldsymbol{\omega}}$ and accelerations

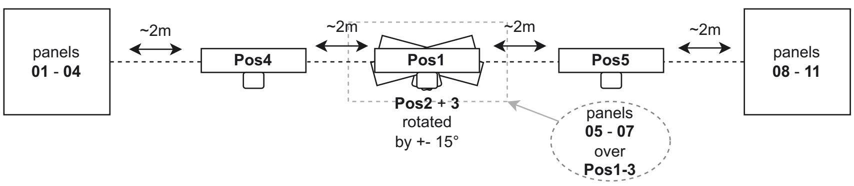

Figure 4: Overview of the poses of the system during the calibration (top view).

$\tilde{\boldsymbol{a}}$ for the IMU are computed as time derivatives of the trajectories. LiDAR observations $\tilde{\boldsymbol{\ell}}_o$ are generated based on the sensor model of the Velodyne Puck. Error-free observations serve as a basis for simulating observations (q) of the epoch (k) sampled from normal distributions with their corresponding VCM determined during calibrations (values shown in Table 2)

$$\boldsymbol{\ell}_{o,i}^{(q,k)} = \mathcal{N}\left(\tilde{\boldsymbol{\ell}}_{o,i}^{(k)}, \boldsymbol{\Sigma}_{\ell\ell,i}\right), \tag{11}$$

$$\boldsymbol{a}^{(q,k)} = \mathcal{N}\left(\tilde{\boldsymbol{a}}^{(k)}, \mathrm{diag}\left(\boldsymbol{\sigma}^2_{\tilde{a}_n} / \sqrt{\Delta t}\right)\right) + \mathcal{N}\left(\boldsymbol{0}, \mathrm{diag}\left(\boldsymbol{\sigma}^2_{a_w} \cdot \sqrt{\Delta t}\right)\right), \tag{12}$$

$$\boldsymbol{\omega}^{(q,k)} = \mathcal{N}\left(\tilde{\boldsymbol{\omega}}^{(k)}, \mathrm{diag}\left(\boldsymbol{\sigma}^2_{\tilde{\omega}_n} / \sqrt{\Delta t}\right)\right) + \mathcal{N}\left(\boldsymbol{0}, \mathrm{diag}\left(\boldsymbol{\sigma}^2_{\omega_w} \cdot \sqrt{\Delta t}\right)\right) \tag{13}$$

where Δt represents the IMU sampling rate. Apart from the bias instability, which is compensated by the corresponding states (cf. Section 2.2.3), this constitutes the random deviations. Similar procedures are applied to transformation parameter between LiDAR and IMU and the LiDAR distance offset, but executed once per realization

$$d_0^{(q)} = \mathcal{N}\left(\tilde{d}_0, \sigma^2_{d_0}\right), \tag{14}$$

$$\boldsymbol{\gamma}_{\mathrm{s}}^{\mathrm{b},(q)} = \mathcal{N}\left(\tilde{\boldsymbol{\gamma}}_{\mathrm{s}}^{\mathrm{b}}, \boldsymbol{\Sigma}_{\gamma\gamma}\right). \tag{15}$$

This results in systematic deviations, as they are randomly sampled only once and thus, affect all epochs of a realization in the same magnitude. These data sets are subsequently processed using the same filter as the real data sets.

2.2.3 Principle of position tracking:

The position tracking is realized by an ESKF. This section offers a concise overview of the method, outlining the state vector definition, prediction, and update steps. The primary emphasis is on the functional model and the incorporation of uncertainty information.

2.2.3.1 *State vector:*

The system of the MSS is parameterized by
- $\boldsymbol{t}_{\mathrm{b}}^{\mathrm{l}}$: position in m $[3 \times 1]$,
- $\boldsymbol{v}_{\mathrm{l}}$: velocity in m/s $[3 \times 1]$,
- $\boldsymbol{q}_{\mathrm{b}}^{\mathrm{l}}$: orientation as a unit quaternion [−] $[4 \times 1]$,
- $\boldsymbol{a}_\beta$: biases of 3D accelerometer in m/s^2 $[3 \times 1]$,
- $\boldsymbol{\omega}_\beta$: biases of gyroscope for three axis in rad/s $[3 \times 1]$.

The dimensions are indicated in the brackets. These vectors are combined into the state vector

$$\boldsymbol{x} = \left[\boldsymbol{t}_{\mathrm{b}}^{\mathrm{l}\,T} \quad \boldsymbol{v}_{\mathrm{l}}^T \quad \boldsymbol{q}_{\mathrm{b}}^{\mathrm{l}\,T} \quad \boldsymbol{a}_\beta^T \quad \boldsymbol{\omega}_\beta^T\right]^T. \tag{16}$$

The orientation is represented using quaternions $\boldsymbol{q}$. Conversions between different rotations representations, i.e. rotation matrix $\boldsymbol{R}$, are indicated by $\boldsymbol{R}\{\cdot\}$. The error state, denoted by δ, is defined as:

$$\delta\boldsymbol{x} = \left[\delta\boldsymbol{t}_{\mathrm{b}}^{\mathrm{l}\,T} \quad \delta\boldsymbol{v}_{\mathrm{l}}^T \quad \delta\boldsymbol{\theta}_{\mathrm{b}}^{\mathrm{l}\,T} \quad \delta\boldsymbol{a}_\beta^T \quad \delta\boldsymbol{\omega}_\beta^T\right]^T. \tag{17}$$

Utilizing an error state allows for quantifying the uncertainty of the orientation through the rotation vector $\delta\boldsymbol{\theta}_{\mathrm{b}}^{\mathrm{l}} = \phi\,\boldsymbol{r}$ where $\mathbf{r}$ is a $[3 \times 1]$ vector with the vector components r_x, r_y, and r_z which is multiplied with the rotation angle ϕ. The error state approach circumvents singular VCM arising from the unit length requirement of quaternions. This leads to a difference between the length of the state vector and the size of the VCM of the states u by 1.

2.2.3.2 *Prediction:*

The system state is propagated into the next epoch (k) by integrating the measurements from an IMU as control inputs $\boldsymbol{u}_o^{(k)}$ in the filter using function $\boldsymbol{f}$

$$\boldsymbol{x}^{(k)} = \boldsymbol{f}\left(\boldsymbol{x}^{(k-1)}, \boldsymbol{u}_o^{(k)}\right). \tag{18}$$

The IMU measurements denoted by o consist of data from a three-axis accelerometer and a three-axis gyroscope, yielding the 3D accelerations $\boldsymbol{a}$ and rotation rates $\boldsymbol{\omega}$ around the three axes:

$$\boldsymbol{\omega}_o = \boldsymbol{\omega} + \boldsymbol{\omega}_\beta + \boldsymbol{\omega}_n, \tag{19}$$

$$\boldsymbol{a}_o = \boldsymbol{a} + \boldsymbol{a}_\beta + \boldsymbol{a}_n + \left(\boldsymbol{R}\{\boldsymbol{q}_{\mathrm{b}}^{\mathrm{l}}\}\right)^T \boldsymbol{g}_{\mathrm{l}}. \tag{20}$$

Here, the subscript n indicates the noise affecting the measurements. The observed accelerations are further corrected for the known gravity vector $\boldsymbol{g}_{\mathrm{l}}$. The prediction computation is performed similarly to common applications, such as [18, 51, 52].

2.2.3.3 *Update:*

The update step of the filter serves to incorporate the measurements of the LiDAR into the estimation process. The approach of updating the system state by minimizing point-to-plane distances for localization using LiDAR measurements was initially proposed by [22] for city model plane localization. This minimization requires the transformation of the points observed by the LiDAR from the sensor frame s to the local frame l, where the plane parameters $\boldsymbol{\alpha}$ are known sufficiently well to be considered deterministic. Thus, the required map information must include the location and plane parameters of nearby panels. This principle is very similar to the calibration of the LiDAR using most of the same reference panels.

For the functional model of the update step, only the changes compared to the calibration are shown. The transformation of the original LiDAR observations to Cartesian coordinates and the transformation to the body frame are performed as shown in equations (5) and (6) based on the results of the calibrations. The transformation from the b to the l-frame is done using the transformation parameters from the state vector

$$\boldsymbol{p}_{\mathrm{l},i} = \begin{bmatrix} x_{\mathrm{l}} \\ y_{\mathrm{l}} \\ z_{\mathrm{l}} \end{bmatrix}_i = \boldsymbol{t}_{\mathrm{b}}^{\mathrm{l}} + \boldsymbol{R}\{\boldsymbol{q}_{\mathrm{b}}^{\mathrm{l}}\} \begin{bmatrix} x_{\mathrm{b}} \\ y_{\mathrm{b}} \\ z_{\mathrm{b}} \end{bmatrix}_i. \tag{21}$$

These transformed points can afterwards be assigned to the reference panels to compute the misclosure h for minimizing the implicit measurement equations:

$$h_i = h\left(\boldsymbol{x}^{(k)}, \boldsymbol{\ell}_i^{(k)}, \boldsymbol{\alpha}_{\mathrm{l},j}\right) \tag{22}$$

$$= \left[n_{x,\mathrm{l}}\; n_{y,\mathrm{l}}\; n_{z,\mathrm{l}}\; d_{\mathrm{l}}\right]_j \cdot \begin{bmatrix} \boldsymbol{p}_{\mathrm{l},i} \\ 1 \end{bmatrix}. \tag{23}$$

The detailed procedure for the computation of the update can be found in [39].

2.2.3.4 *Stochastic model:*

The stochastic modeling necessary to incorporate uncertainties into the estimation process is achieved through the VCM of the observations $\boldsymbol{\Sigma}_{\ell\ell}$. This matrix is established by applying variance-covariance propagation to the uncertainties of the original LiDAR observations determined using the VCE in the calibration. To compute the VCM $\boldsymbol{\Sigma}_{\ell\ell,i}$ for each point i, the values are computed individually and then combined into the complete block diagonal matrix $\boldsymbol{\Sigma}_{\ell\ell}$, treating all observations as uncorrelated:

$$\boldsymbol{\Sigma}_s = \begin{bmatrix} \operatorname{diag}\left(\sigma_d^2, \sigma_e^2, \sigma_\alpha^2\right) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma}_{\gamma\gamma} \end{bmatrix}, \tag{24}$$

$$\boldsymbol{G}_{I,i} = \left.\frac{\partial \boldsymbol{g}_I}{\partial \boldsymbol{\ell}_o}\right|_{\boldsymbol{\ell}_{o,i}}, \tag{25}$$

$$\boldsymbol{G}_{II,i} = \left.\frac{\partial \boldsymbol{g}_{II}}{\partial \left[\boldsymbol{\ell}, \boldsymbol{\gamma}_s^b\right]}\right|_{\boldsymbol{\ell}_i, \boldsymbol{\gamma}_s^b}, \tag{26}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{\ell\ell},i} = \boldsymbol{G}_{II,i} \boldsymbol{G}_{I,i} \boldsymbol{\Sigma}_s \boldsymbol{G}_{I,i}^T \boldsymbol{G}_{II,i}^T \tag{27}$$

with $\boldsymbol{\Sigma}_{\gamma\gamma}$ as the VCM of the 6-DoF transformation parameters from the extrinsic calibration of the LiDAR done prior to the kinematic measurement and $\boldsymbol{g}_I$ and $\boldsymbol{g}_{II}$ from equations (5) and (6).

2.2.3.5 ***Initialization:*** A crucial aspect of the proposed approach is the initialization of the states. The system's pose is determined within a KF, a position tracking algorithm that necessitates accurate initial states for optimal performance and convergence. Incorrect initialization may result in erroneous assignments and filter divergence. To mitigate this issue, for this paper, the filter is consistently initialized with the pose of the reference trajectory (or true trajectory in simulations). Velocities are set to 0, reflecting the robot's stationary starting position, while biases are initialized as the mean value of IMU observations during the initial three seconds (with the acceleration corrected for gravity)

$$\boldsymbol{x}^{(0)} = \left[\tilde{\boldsymbol{t}}_{\mathrm{b}}^{\mathrm{l}\ T} \quad \boldsymbol{0} \quad \tilde{\boldsymbol{q}}_{\mathrm{b}}^{\mathrm{l}\ T} \quad \bar{\boldsymbol{a}}_\beta^T \quad \bar{\boldsymbol{\omega}}_\beta^T\right]^T. \tag{28}$$

For the initialization of the VCM of the states, the choice of uncertainties for the biases and the pose are different. The uncertainties associated with the pose and the velocity have minimal influence on the filter performance, as the filter stabilizes quickly for the setup in this paper, so the values are chosen in a similar magnitude to the results. On the other hand, the uncertainty of the biases is set to the corresponding bias instabilities. Thus, the initial VCM is set up as

$$\boldsymbol{\Sigma}_{\boldsymbol{xx}}^{(0)} = \operatorname{diag}\begin{bmatrix} \boldsymbol{I}_3 \cdot (1\ \mathrm{mm})^2 \\ \boldsymbol{I}_3 \cdot (1\ \mathrm{mm/s})^2 \\ \boldsymbol{I}_3 \cdot (1\ \mathrm{mrad})^2 \\ \operatorname{diag}\left(\boldsymbol{\sigma}_{a_w}^2\right) \\ \operatorname{diag}\left(\boldsymbol{\sigma}_{\omega_w}^2\right) \end{bmatrix}. \tag{29}$$

3 Results

The results are divided into three parts. In the first part, simulated data sets are used to investigate the influence of the uncertainties on the results. In the second part, the results for the real data set are analyzed and compared to the simulations before the results are discussed.

The discussion of the filter results is based on the deviations of the estimated trajectory denoted by $\hat{\ }$ from the true trajectory, marked by $\tilde{\ }$ (or the ground truth for the real data established by the LT). Three different metrics are utilized. The computations are all based on the deviations $\boldsymbol{\Delta}$ of the translation t and orientation o for every epoch (k)

$$\boldsymbol{\Delta}_{\mathrm{t}}^{(k)} = \Delta \boldsymbol{t}_{\mathrm{b}}^{\mathrm{l},(k)} = \tilde{\boldsymbol{t}}_{\mathrm{b}}^{\mathrm{l},(k)} - \hat{\boldsymbol{t}}_{\mathrm{b}}^{\mathrm{l},(k)}, \tag{30}$$

$$\boldsymbol{\Delta}_{\mathrm{o}}^{(k)} = \boldsymbol{\theta}\left\{\tilde{\boldsymbol{q}}_{\mathrm{b}}^{\mathrm{l},(k)} \boxminus \hat{\boldsymbol{q}}_{\mathrm{b}}^{\mathrm{l},(k)}\right\}, \tag{31}$$

$$\boldsymbol{\Delta}^{(k)} = \begin{bmatrix} \boldsymbol{\Delta}_{\mathrm{t}}^{(k)} \\ \boldsymbol{\Delta}_{\mathrm{o}}^{(k)} \end{bmatrix} \tag{32}$$

with $\boxminus$ for quaternion subtraction. Based on the deviations, either the root mean square error (RMSE) for each the translation or orientation or the weighted RMSE in the form of the Mahalanobis distance for all 6-DoF is computed. For a single run of the simulation or the real data sets, this is computed as

$$\mathrm{RMSE}_{\mathrm{t}}^{(k)} = \sqrt{\frac{1}{3}\left(\boldsymbol{\Delta}_{\mathrm{t}}^{(k)}\right)^T \boldsymbol{\Delta}_{\mathrm{t}}^{(k)}}, \tag{33}$$

$$\mathrm{RMSE}_{\mathrm{o}}^{(k)} = \sqrt{\frac{1}{3}\left(\boldsymbol{\Delta}_{\mathrm{o}}^{(k)}\right)^T \boldsymbol{\Delta}_{\mathrm{o}}^{(k)}}, \tag{34}$$

$$\sqrt{d_m}^{(k)} = \sqrt{\left(\boldsymbol{\Delta}^{(k)}\right)^T \left(\boldsymbol{\Sigma}_{pose,\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}}^{(k)}\right)^{-1} \boldsymbol{\Delta}^{(k)}}. \tag{35}$$

The weighting is determined by using the VCM $\boldsymbol{\Sigma}_{pose,\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}}^{(k)}$ for the relevant components of the state vector. This enables the incorporation of the filter's self-reported uncertainty information into the measure. Assuming the states are normally distributed, a χ^2-distribution with 6-DoF can be utilized in a hypothesis test, using the Mahalanobis distance to validate the consistency of the filter [53]. Consistency requires that the filter properly represents the deviations from the true values in the uncertainties.

For the evaluation of M MC runs, the measures are computed as

$$\overline{\mathrm{RMSE}}_{\mathrm{t}}^{(k)} = \sqrt{\frac{1}{M}\sum_{m=1}^{M} \frac{1}{3}\left(\boldsymbol{\Delta}_{\mathrm{t}}^{(k,m)}\right)^T \boldsymbol{\Delta}_{\mathrm{t}}^{(k,m)}}, \tag{36}$$

$$\overline{\mathrm{RMSE}}_{\mathrm{o}}^{(k)} = \sqrt{\frac{1}{M}\sum_{m=1}^{M} \frac{1}{3}\left(\boldsymbol{\Delta}_{\mathrm{o}}^{(k,m)}\right)^T \boldsymbol{\Delta}_{\mathrm{o}}^{(k,m)}}, \tag{37}$$

$$\overline{\sqrt{d_m}}^{(k)} = \sqrt{\frac{1}{M}\sum_{m=1}^{M} \left(\boldsymbol{\Delta}^{(k,m)}\right)^T \left(\boldsymbol{\Sigma}_{pose,\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}}^{(k,m)}\right)^{-1} \boldsymbol{\Delta}^{(k,m)}}. \tag{38}$$

3.1 Simulation

The simulation results are shown for two different simulated data sets. In one data set, the calibration parameters are assumed to be perfectly determined and corrected, so only random deviations affect the data set (equations (14) and (15) in the generation of the simulated observations are

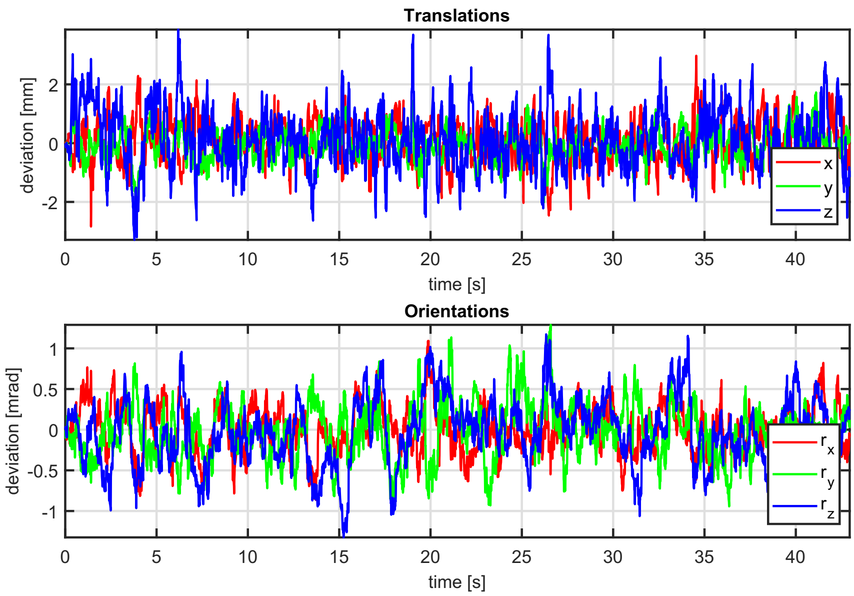

Figure 5: Deviations of the pose in the simulated run 4.

omitted). In the second data set, all known uncertainties are incorporated.

3.1.1 No systematic deviations

The deviations in the trajectory of simulated trajectory 4 are displayed in Figure 5. The translations exhibit deviations in the low millimeter range without noticeable patterns. Similarly, the orientations show comparable behavior across all axes, with deviations reaching up to 1 mrad, occasionally showing oscillations caused by the bias instability of the simulated IMU.

The obtained Mahalanobis distances are depicted in Figure 6. Most of the epochs fall within the intervals of the χ^2-distribution for the 95 % confidence interval (indicated by the green lines), confirming the consistency hypothesis. The tendency indicates that the values are leaning toward the lower boundary of the hypothesis interval, which suggests a slightly pessimistic uncertainty estimate, similar to the results in [39].

3.1.2 All known uncertainties

While the previous results showcased the best possible results due to the perfect correction of the systematic effects, the following results present a more realistic case where the remaining systematic deviations affect the georeferencing. The results for the averaged $\overline{\text{RMSE}}$ and the square root of the averaged Mahalanobis distance $\sqrt{\bar{d}_m}$ are displayed for 100 MC runs, both for the simulation without (**simu 1**) and with systematic deviations (**simu 2**), in Figures 7 and 8.

Even though the magnitude of these possible systematic deviations is known, the deviations are still significantly higher than before. The reason for this is that the systematic

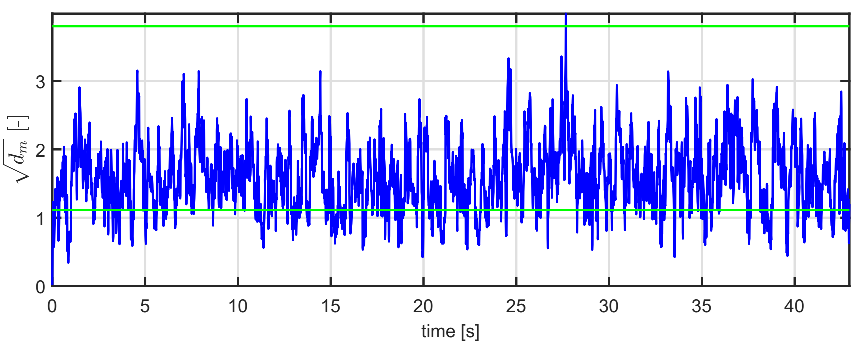

Figure 6: Mahalanobis distance of the pose in the simulated run 4. 95 % confidence interval of χ^2-distribution in green.

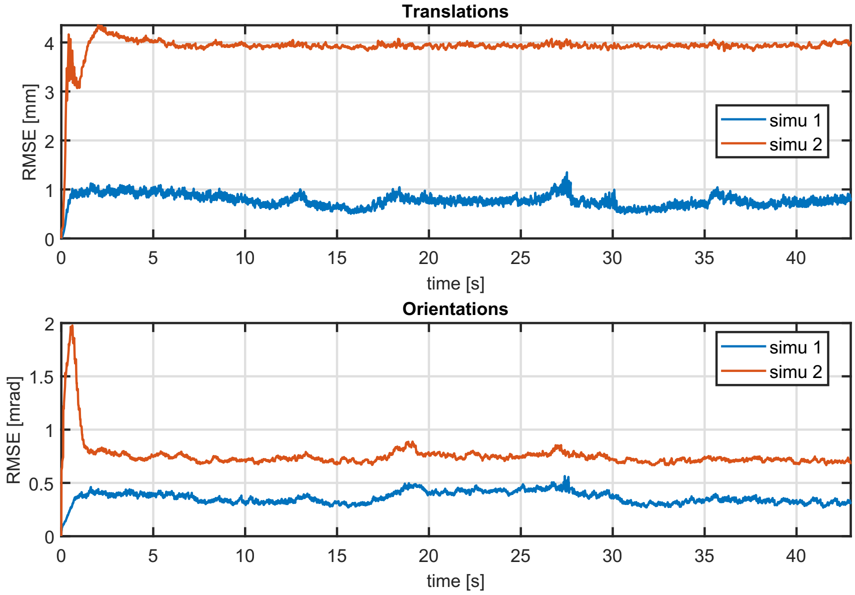

Figure 7: RMSE for MC runs of simulated run 4.

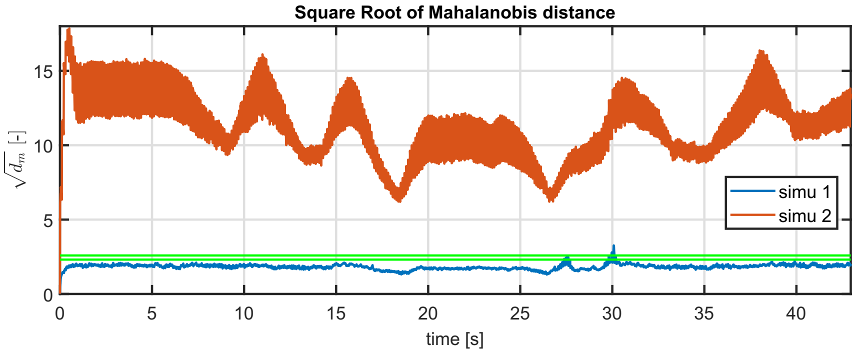

Figure 8: Mahalanobis distance of the pose in the MC runs of simulated run 4. 95 % confidence interval of χ^2-distribution for 100 MC runs in green.

deviations are constant over all observations and, thus, violate the assumption of white Gaussian noise in the KF. Given the short measurement distances of 3–5 m, the impact of the systematic deviations is about a factor of two higher on the translation compared to the orientations.

The Mahalanobis distance analysis yields similar outcomes. Results without systematic deviations confirm the filter's pessimistic uncertainty estimation, with a narrower interval for the hypothesis testing due to the repetitions providing a clearer result outside the interval.

Deviations in runs with systematic effects are significantly higher, while standard deviations increase only slightly, attributed to additional uncertainties from transformation parameters and distance offset. Three effects can be seen:

1. The initialization of the filter is visible, which can already be seen in the first seconds of the time series in Figure 7. During this period, the VCM stabilizes and the correlations among the states are determined.
2. The values of the Mahalanobis distance vary over time, while the RMSE is nearly constant. This variation stems from differing numbers of matched LiDAR observations and their distribution in the environment.
3. The time series contains an oscillation with a period of 0.1 s matching the rotation rate of the LiDAR, due to the distribution of the panel mainly in three directions in the room with different panel sizes, resulting in a recurring pattern in standard deviations. For epochs with fewer points, the Mahalanobis distance decreases as the standard deviations increase.

Table 3 shows the results for all five trajectories, confirming the trend seen for trajectory 4.

Table 3: Results of all MC runs (100 each) for all trajectories of the two versions of the simulated data set. Values are shown as mean over the whole trajectory with t for translation and o for orientation.

Traj. #	$\overline{\text{RMSE}}$				$\sqrt{\bar{d}_m}$	
	Simu 1		Simu 2		Simu 1	Simu 2
	T [mm]	*o* [mrad]	*t* [mm]	*o* [mrad]	[–]	[–]
1	0.87	0.33	3.77	0.81	1.98	13.04
2	0.74	0.32	3.74	0.70	1.87	11.13
3	0.82	0.53	3.94	0.72	1.88	11.47
4	0.79	0.36	3.92	0.76	1.84	11.37
6	0.81	0.31	3.81	0.83	2.07	12.55

3.2 Real data

Figures 9 and 10 illustrate the results for trajectory 4 using real data. The translations exhibit deviations of up to 15 mm (with an RMSE of 4.30 mm), while the orientations display deviations of up to 4 mrad (with an RMSE of 1.35 mrad). The time series reveals distinct systematic deviations, such as negative deviation along the z-axis throughout most of the trajectory. The Mahalanobis distance exceeds the hypothesis test interval for the entire trajectory, indicating an over-optimistic assessment of the uncertainty. This mirrors findings from single runs of the simulated data with systematic effects, indicating remaining systematic deviations as the likely cause.

To further investigate the observed deviations in the real data, the normalized estimation error is calculated. For each element δ_j in the deviation vector $\mathbf{\Delta}$, the corresponding normalized error $\bar{\delta}_j$ is determined by dividing it by the estimated standard deviation, as described in [53]

$$\bar{\delta}_j = \frac{\delta_j}{\sqrt{\mathbf{\Sigma}_{pose,\hat{x}\hat{x},j}}}. \tag{39}$$

Here $\mathbf{\Sigma}_{pose,\hat{x}\hat{x},j}$ represents the jth diagonal element of the estimated VCM. By normalizing the errors, the expected uncertainty in each state variable is considered, enabling a more effective comparison across different components of the pose. Figure 11 depicts the normalized errors for run 4. Ideally, these normalized errors should conform to a normal distribution, as indicated by the purple lines in the figure at a significance level of $\alpha = 5\,\%$. However, the plot reveals that all pose elements exceed the expected normal distribution interval throughout the trajectory. This observation confirms the presence of systematic deviations similar to those seen in Figures 8 and 10. The translations exhibit their highest values when the system has moved furthest into the room, while the orientation errors peak during specific movements, notably between 7 and 17 s and between 27 and 35 s.

3.3 Discussion

The findings highlight the significant impact of systematic effects on the accuracy of the positioning results. While simulations with perfectly corrected observations yield excellent results, more realistic simulations and real data trajectories exhibit increased deviations and inconsistent estimations. The investigated systematic deviations are the

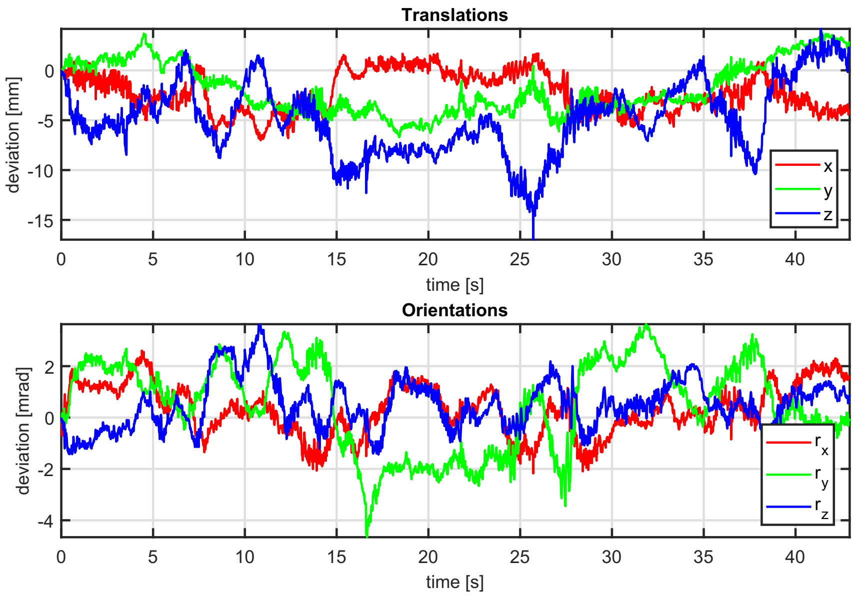

Figure 9: Deviations of the pose in run 4.

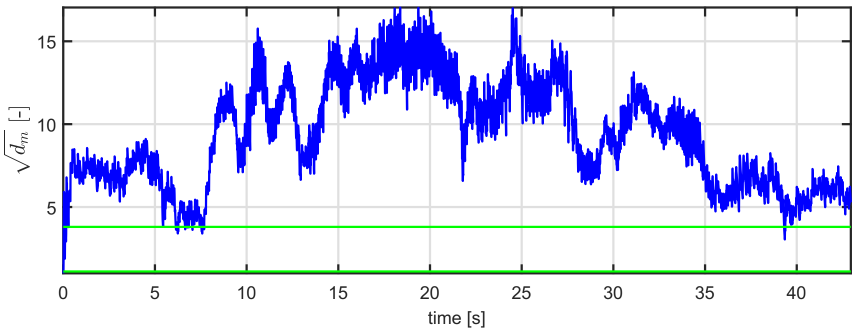

Figure 10: Mahalanobis distance of the pose in run 4. 95 % confidence interval of χ^2-distribution in green.

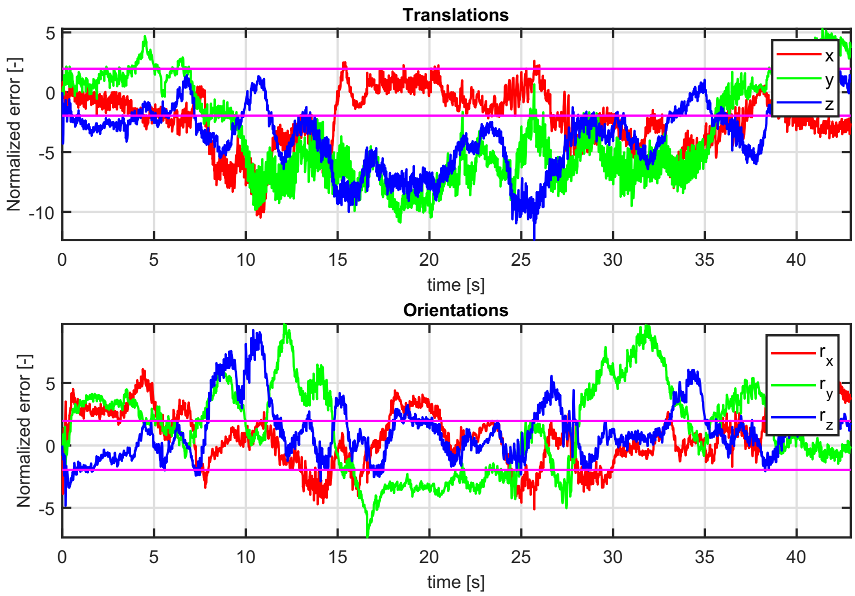

Figure 11: Normalized estimation error for the pose of system. Purple lines show an interval of a normal distribution with α =5 %.

remaining deviations of the transformation parameters from the calibration between LiDAR and IMU and the distance offset of the LiDAR observations. Both of these parameter types represent different challenges.

The simplification of the distance offset estimation, where a single offset is estimated for all LiDAR scan lines, could be enhanced by adopting an intrinsic calibration approach that provides individual estimations per scan line. The distance offset leads to larger deviations in the y-direction compared to other directions in the MC runs, influenced by the distribution of the reference panels, particularly the absence of panels behind the robot opposite to panels 21 and 22 in Figure 2.

Deviations in the transformation parameters lead to more complex deviations. These deviations depend upon the panel's constellation and the systems position along the trajectory. Since LiDAR point cloud assignment relies on the predicted pose, deviations in the pose directly affect the assignment, leading to further deviations. Additionally, the distance offset already influences the determination of the transformation parameters showing the need for a hierarchical approach, as the intrinsic calibration influences the extrinsic calibration, which in turn, both, affect the final positioning quality.

With respect to the results for the Mahalanobis distance and the normalized estimation error, the systematic deviations act as correlations between the observations. All observations are affected by the same deviations in the distance measurements and transformation parameters. Despite this, the observations are considered uncorrelated. This results in a too optimistic VCM of the observations causing the same for the VCM of the pose. The aforementioned metrics are inflated due to this underestimation of the uncertainty.

Considering these aspects, the Mahalanobis distance time series in Figure 10 should exhibit similar values during the stationary phases at the beginning and end of the trajectory, which is not the case. Possible minor effects contributing to this behavior include physical correlations among the LiDAR observations caused by similar pathways through the atmosphere and overlap on the object. However, these correlations have a limited impact, since many of the reference panels are small and, thus, only a few points are assigned to them. Additionally, the intrinsic IMU calibration is also uncertain. In the used setup, the IMU is not strictly necessary as the pose can always be determined using the LiDAR scans. This contrasts with scenarios such as long hallways or urban canyons where the IMU would play a more crucial role.

Considering the influence of the systematic deviations on the results, the usage of a KF for georeferencing is to be questioned. The assumption of white Gaussian noise is not correct with remaining systematic deviations. Either these systematics could be corrected during operation using an online estimation or a particle filter could be used to allow for arbitrary distributions modeling these effects. Further research should investigate the accuracy and consistency of these possibilities.

4 Summary and outlook

This paper presents an empirical evaluation of a trajectory computed through sensor data fusion using IMU and LiDAR observations in a laboratory environment. The fusion process includes propagating uncertainties throughout an ESKF processing chain. The evaluation compares the results against a ground truth trajectory acquired via a LT with a T-Probe. Initially, this ground truth data is employed to simulate IMU and LiDAR measurements. Subsequently, both simulated and real data sets are processed across five acquired data sets for analysis.

The results reveal a significant difference in accuracy between a simplified simulation and a more realistic simulation, as well as real-world data. The first simulation results align with the expected outcomes. However, introducing systematic deviations in the observations violates the assumption of the used ESKF, leading to higher deviations in the estimated pose and an inconsistent estimation. The similarities between the more realistic simulation and real-world data suggest that the problem is primarily caused by the remaining deviations in the transformation parameters and the distance offset. Additional minor influences include the correlations between subsequent LiDAR points and residual effects from the IMU's intrinsic calibration.

Based on these findings, further improvements to LiDAR calibrations are necessary, both before usage and during operation. This could involve an online estimation of the transformation parameters and the distance offset, dynamically updating the LiDAR calibration parameters based on the environment during operation. Additionally, future work should focus on outdoor data sets, necessitating integration of GNSS observations for georeferencing and outlier detection.

Research ethics: Not applicable.
Author contributions: Experiments, Coding, Processing, Writing: DE, HA; Proof-Reading: SV, HA, IN; Project Supervision: IN, HA. All the authors have accepted responsibility for the entire content of this submitted manuscript and approved submission.
Competing interests: The authors state no competing interests.
Research funding: This work was funded by the German Research, Foundation (DFG) as part of the Research Training Group, i.c.sens [RTG 2159].
Data availability: The raw data can be obtained on request from the corresponding author.

References

1. Thrun S, Burgard W, Fox D. Probabilistic robotics. In: Intelligent robotics and autonomous agents series, 1st ed. The MIT Press; 2005.
2. Hsu LT. Analysis and modeling GPS NLOS effect in highly urbanized area. GPS Solut 2018;22:1–12.
3. Makki A, Siddig A, Saad M, Bleakley C. Survey of WiFi positioning using time-based techniques. Comput Network 2015;88:218–33.
4. Li MG, Zhu H, You SZ, Tang CQ. UWB-based localization system aided with inertial sensor for underground coal mine applications. IEEE Sensor J 2020;20:6652–69.
5. Pöppl F, Neuner H, Mandlburger G, Pfeifer N. Integrated trajectory estimation for 3D kinematic mapping with GNSS, INS and imaging sensors: a framework and review. ISPRS J Photogrammetry Remote Sens 2023;196:287–305.
6. Schmitz B, Holst C, Medic T, Lichti DD, Kuhlmann H. How to efficiently determine the range precision of 3d terrestrial laser scanners. Sensors 2019;19:1–19.
7. Wujanz D. Terrestrial laser scanning for geodetic deformation monitoring, Reihe C, editor. München: DGK; 2016, vol 775.
8. Skaloud J, Lichti D. Rigorous approach to bore-sight self-calibration in airborne laser scanning. ISPRS J Photogrammetry Remote Sens 2006;61:47–59.
9. Hartmann J, von Gösseln I, Schild N, Dorndorf A, Paffenholz JA, Neumann I. Optimisation of the calibration process of a k-TLS based multi-sensor-system by genetic algorithms. In: The International archives of the photogrammetry, remote sensing and spatial information sciences. Copernicus GmbH; 2019:1655–62 pp.
10. Heinz E, Eling C, Wieland M, Klingbeil L, Kuhlmann H. Development, calibration and evaluation of a portable and direct georeferenced laser scanning system for kinematic 3D mapping. J Appl Geodesy 2015;9:227–43.

11. Alatise MB, Hancke GP. A review on challenges of autonomous mobile robot and sensor fusion methods. IEEE Access 2020;8:39830—46.
12. Dellaert F. Factor graphs: exploiting structure in robotics. Annu Rev Control Robot Auton Syst 2021;4:141—66.
13. Liu Z, Zhang F. BALM: bundle adjustment for lidar mapping. IEEE Rob Autom Lett 2021;6:3184—91.
14. Trusheim P, Mehltretter M, Rottensteiner F, Heipke C. Cooperative image orientation considering dynamic objects. In: ISPRS annals of the photogrammetry, remote sensing and spatial information sciences. Copernicus GmbH; 2022:169—77 pp.
15. Simon D. Optimal state estimation: Kalman, H infinity, and nonlinear approaches. Hoboken, New Jersey: John Wiley & Sons; 2006.
16. Kaczmarek A, Rohm W, Klingbeil L, Tchórzewski J. Experimental 2D extended Kalman filter sensor fusion for low-cost GNSS/IMU/Odometers precise positioning system. Measurement 2022;193:1—12.
17. Godha S, Cannon ME. GPS/MEMS INS integrated system for navigation in urban areas. GPS Solut 2007;11:193—203.
18. Qin C, Ye H, Pranata CE, Han J, Zhang S, Liu M. LINS: a lidar-inertial state estimator for robust and efficient navigation. In: 2020 IEEE international conference on robotics and automation (ICRA). IEEE; 2020:8899—906 pp.
19. Lee W, Geneva P, Yang Y, Huang G. Tightly-coupled GNSS-aided visual-inertial localization. In: 2022 international conference on robotics and automation (ICRA); 2022:9484—91 pp.
20. Schlichting A. Fahrzeuglokalisierung durch automotive Laserscanner unter Verwendung statischer Merkmale [Ph.D. thesis], Reihe C, editor. München: DGK; 2018, vol 826.
21. Esser F, Moraga JA, Klingbeil L, Kuhlmann H. Accuracy improvement of mobile laser scanning point clouds using graph-based trajectory optimization. In: Garcia-Asenjo L, Lerma JL, editors. 5th joint international symposium on deformation monitoring. València: Editorial Universitat Politècnica de València; 2023:105—12 pp.
22. Bureick J, Vogel S, Neumann I, Diener D, Alkhatib H. Geo-Referenzierung von Unmanned Aerial Systems über Laserscannermessungen und 3D-Gebäudemodelle. DVW-Schriftenr 2019;96:63—74.
23. Lucks L, Klingbeil L, Plümer L, Dehbi Y. Improving trajectory estimation using 3D city models and kinematic point clouds. Trans GIS 2021;25:238—60.
24. Mounier E, Abdelaziz SK, de Araujo PRM, Taghavikish S, Elhabiby M, Noureldin A. Utilizing LiDAR registration on 3D high accuracy digital maps for robust positioning in GNSS challenging environments. In: Proceedings of the 32nd international technical meeting of the satellite division of the institute of navigation (ION GNSS+ 2019); 2019:1366—76 pp.
25. Mounier E, Elhabiby M, Korenberg M, Noureldin A. High-precision positioning in GNSS-challenged environments: a LiDAR-based multi-sensor fusion approach with 3D digital maps registration. TechRxiv. 2023.
26. Aghili F, Su CY. Robust relative navigation by integration of ICP and adaptive Kalman filter using laser scanner and IMU. IEEE/ASME Trans Mechatron 2016;21:2015—26.
27. Schütz A, Sanchez-Morales DE, Pany T. Precise positioning through a loosely-coupled sensor fusion of GNSS-RTK, INS and LiDAR for autonomous driving. In: 2020 IEEE/ION position, location and navigation symposium (PLANS). IEEE; 2020:219—25 pp.
28. Roumeliotis SI, Sukhatme GS, Bekey GA. Circumventing dynamic modeling: evaluation of the error-state Kalman filter applied to mobile robot localization. In: 1999 IEEE international conference on robotics and automation. IEEE; 1999: 1656—63 pp.
29. Mourikis AI, Roumeliotis SI. A multi-state constraint Kalman filter for vision-aided inertial navigation. In: Proceedings 2007 IEEE international conference on robotics and automation. IEEE; 2007:3565—72 pp.
30. Dang T. An iterative parameter estimation method for observation models with nonlinear constraints. Metrol Meas Syst 2008;15:421—32.
31. Mirzaei FM, Roumeliotis SI. A Kalman filter-based algorithm for IMU-camera calibration: observability analysis and performance evaluation. IEEE Trans Robot 2008;24:1143—56.
32. Strübing T, Neumann I. Positions- und Orientierungsschätzung von LIDAR-Sensoren auf Multisensorplattformen. ZfV 2013;3:210—21.
33. El-Sheimy N, Youssef A. Inertial sensors technologies for navigation applications: state of the art and future trends. Sat Nav 2020;1:1—21.
34. Yang Y, Geneva P, Zuo X, Huang G. Online self-calibration for visual-inertial navigation: models, analysis, and degeneracy. IEEE Trans Robot 2023;39:1—20.
35. Glennie C. Rigorous 3D error analysis of kinematic scanning LIDAR systems. J Appl Geodesy 2007;1:147—57.
36. Kalenjuk S, Lienhart W. A method for efficient quality control and enhancement of mobile laser scanning data. Rem Sens 2022;14:857.
37. Hofmann S, Brenner C. Accuracy assessment of mobile mapping point clouds using the existing environment as terrestrial reference. In: International archives of the photogrammetry, remote sensing and spatial information sciences-ISPRS archives 41 (2016); 2016, vol 41:601—8 p.
38. Dreier A, Janßen J, Kuhlmann H, Klingbeil L. Quality analysis of direct georeferencing in aspects of absolute accuracy and precision for a UAV-based laser scanning system. Rem Sens 2021;13:3564.
39. Ernst D, Vogel S, Neumann I, Alkhatib H. Error state Kalman filter with implicit measurement equations for position tracking of a multi-sensor system with IMU and LiDAR. In: 2023 13th international conference on indoor positioning and indoor navigation (IPIN). IEEE; 2023:1—6 pp.
40. Parker P, editor. Inertial sensors; 2023. Available from: https://www.microstrain.com/inertial-sensors/all-sensors.
41. Velodyne. Velodyne LiDAR, Inc, editor. VLP-16 manual: user's manual and programming guide. San Jose, CA, USA: Velodyne LiDAR, Inc; 2019.
42. Metrology H, Metrology H, editors. Leica absolute tracker AT960; 2023. Available from: https://hexagon.com/products/leica-absolute-tracker-at960.
43. Garmin G, editor. GPS 18x technical specifications (data sheet); 2011. Available from: https://static.garmin.com/pumac/GPS_18x_Tech_Specs.pdf.
44. Koch KR. Introduction to Bayesian statistics, 2nd ed. Berlin, Heidelberg: Springer; 2007.

45. Hartmann J, Paffenholz JA, Strübing T, Neumann I. Determination of position and orientation of LiDAR sensors on multisensor platforms. J Survey Eng 2017;143:1−11.
46. Ernst D, Alkhatib H, Neumann I, Vogel S. Analysis of multiple positions for the intrinsic and extrinsic calibration of a multi-beam LiDAR. In: 2022 25th international conference on information fusion (FUSION); 2022:01−8 pp.
47. Heiker A. Mutual validation of earth orientation parameters, geophysical excitation functions and second degree gravity field coefficients, Reihe C, editor. München: DGK; 2013, vol 697.
48. Ernst D, Vogel S, Alkhatib H, Neumann I. Intrinsische und extrinsische Kalibrierung eines Velodyne VLP-16. In: Luhmann T, Schumacher C, editors. Beiträge der Oldenburger 3D-Tage 2022. Wichmann Verlag; 2022, vol 20:186−93 pp.
49. Fong WT, Ong SK, Nee AYC. Methods for in-field user calibration of an inertial measurement unit without external equipment. Meas Sci Technol 2008;19:085202. https://doi.org/10.1088/0957-0233/19/8/085202.
50. El-Sheimy N, Hou H, Niu X. Analysis and modeling of inertial sensors using allan variance. IEEE Trans Instrum Meas 2008;57:140−9.
51. Solà J. Quaternion kinematics for the error-state Kalman filter. arXiv. 2017.
52. Xu W, Zhang F. FAST-LIO: a fast, robust LiDAR-inertial odometry package by tightly-coupled iterated Kalman filter. IEEE Rob Autom Lett 2021;6:3317−24.
53. Bar-Shalom Y, Li XR, Kirubarajan T. Estimation with applications to tracking and navigation: theory algorithms and software. New York: John Wiley & Sons; 2004.

8 Appendix

Algorithm 1: Dual Filter for Fusion of IMU with Implicit Measurement Model in Update

Required:

Initial State: $\hat{\boldsymbol{x}}_+^{(0)}, \mathbf{Q}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}}^{(0)}, \hat{\boldsymbol{x}}_s^{(0)}, \mathbf{Q}_{\hat{\boldsymbol{x}}_s\hat{\boldsymbol{x}}_s}^{(0)}$

Sensor Information: $\mathbf{Q}_\mathbf{I}, \boldsymbol{u}_o, \boldsymbol{\ell}, \mathbf{V}_i$ for all epochs k

Begin

- **For** k = *1 .. K*
  - Prediction:
  - $\hat{\boldsymbol{a}}^{(k)} \leftarrow \boldsymbol{a}_o^{(k)} - \boldsymbol{a}_b^{(k-1)} - \left(\mathbf{R}_{\mathrm{b}}^{\mathrm{l},(k)}\right)^T \boldsymbol{g}_\mathrm{l}$
  - $\hat{\boldsymbol{\omega}}^{(k)} \leftarrow \boldsymbol{\omega}_o^{(k)} - \boldsymbol{\omega}_b^{(k-1)}$
  - $\hat{\boldsymbol{x}}_-^{(k)} \leftarrow \boldsymbol{F}_{\boldsymbol{x}}^{(k)} \hat{\boldsymbol{x}}_+^{(k-1)}$
  - $\boldsymbol{F}_{\boldsymbol{x}} \leftarrow \left.\frac{\partial f}{\partial \delta \boldsymbol{x}}\right|_{\boldsymbol{x},\boldsymbol{u}^{(k)}}$ and $\boldsymbol{F}_\mathbf{I}^{(k)} \leftarrow \left.\frac{\partial f}{\partial \delta \boldsymbol{u}}\right|_{\boldsymbol{x},\boldsymbol{u}^{(k)}}$
  - $\mathbf{Q}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}},-}^{(k)} \leftarrow \boldsymbol{F}_{\boldsymbol{x}}^{(k)} \mathbf{Q}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}}^{(k-1)} \left(\boldsymbol{F}_{\boldsymbol{x}}^{(k)}\right)^T + \boldsymbol{F}_\mathbf{I}^{(k)} \mathbf{Q}_\mathbf{I} \left(\boldsymbol{F}_\mathbf{I}^{(k)}\right)^T$
  - **If** *observations available* **then**
    - Update:
    - $\hat{\boldsymbol{x}}_+^{(k,m=1)} \leftarrow \hat{\boldsymbol{x}}_-^{(k)}$, $\delta\boldsymbol{x}^{(k,m=0)} \leftarrow \mathbf{0}$ and $\hat{\boldsymbol{\ell}}^{(k,m=1)} \leftarrow \boldsymbol{\ell}^{(k)}$
    - **repeat**
      - $\mathbf{Q}_{\boldsymbol{\ell\ell}}^{(k)} \leftarrow \sum_{i=1}^{z} \hat{\boldsymbol{x}}_s^{(k-1)} \mathbf{V}_i^{(k)}$
      - $\boldsymbol{h}^{(k,m)} \leftarrow \boldsymbol{h}\left(\hat{\boldsymbol{x}}_+^{(k,m)}, \hat{\boldsymbol{\ell}}^{(k,m)}\right)$
      - $\mathbf{A}^{(k,m)} \leftarrow \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_+^{(k,m)}, \hat{\boldsymbol{\ell}}^{(k,m)}} \cdot \left.\frac{\partial \boldsymbol{x}}{\partial \delta \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_+^{(k,m)}}$
      - $\mathbf{B}^{(k,m)} \leftarrow \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\ell}}\right|_{\hat{\boldsymbol{x}}_+^{(k,m)}, \hat{\boldsymbol{\ell}}^{(k,m)}}$
      - $\mathbf{O}^{(k,m)} \leftarrow \mathbf{A}^{(k,m)} \mathbf{Q}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}},-}^{(k)} \left(\mathbf{A}^{(k,m)}\right)^T$
      - $\mathbf{Q}_{\mathbf{hh}}^{(k,m)} \leftarrow \mathbf{B}^{(k,m)} \mathbf{Q}_{\boldsymbol{\ell\ell}}^{(k)} \left(\mathbf{B}^{(k,m)}\right)^T$
      - $\mathbf{Q}_{\boldsymbol{rr}}^{(k,m)} \leftarrow \mathbf{O}^{(k,m)} + \mathbf{Q}_{\mathbf{hh}}^{(k,m)}$
      - $\boldsymbol{K}^{(k,m)} \leftarrow \mathbf{Q}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}},-}^{(k)} \left(\mathbf{A}^{(k,m)}\right)^T \left(\mathbf{Q}_{\boldsymbol{rr}}^{(k,m)}\right)^{-1}$
      - $\boldsymbol{r}^{(k,m)} \leftarrow \mathbf{B}^{(k,m)} \left(\boldsymbol{\ell}^{(k,m)} - \boldsymbol{\ell}^{(k)}\right) + \mathbf{A}^{(k,m)} \delta\boldsymbol{x}^{(k,m-1)}$
      - $\boldsymbol{w}^{(k,m)} \leftarrow \boldsymbol{h}^{(k,m)} + \boldsymbol{r}^{(k,m)}$
      - $\delta\boldsymbol{x}^{(k,m)} \leftarrow -\boldsymbol{K}^{(k,m)} \boldsymbol{w}^{(k,m)}$
      - $\hat{\boldsymbol{v}}^{(k,m)} \leftarrow -\mathbf{Q}_{\boldsymbol{\ell\ell}}^{(k)} \left(\mathbf{B}^{(k,m)}\right)^T \left(\mathbf{Q}_{\boldsymbol{rr}}^{(k,m)}\right)^{-1} \boldsymbol{w}^{(k,m)}$
      - $\hat{\boldsymbol{\ell}}^{(k,m+1)} \leftarrow \boldsymbol{\ell}^{(k)} + \hat{\boldsymbol{v}}^{(k,m)}$
      - $\hat{\boldsymbol{x}}_+^{(k,m+1)} \leftarrow \hat{\boldsymbol{x}}_-^{(k)} \boxplus \delta\boldsymbol{x}^{(k,m)}$
    - **until** $\|\delta\mathbf{x}^{(k,m+1)} \boxminus \delta\mathbf{x}^{(k,m)}\| < \epsilon$

$$\boldsymbol{L}^{(k)} \leftarrow \boldsymbol{I} - \mathbf{K}^{(k,m)}\mathbf{A}^{(k,m)}$$

$$\boldsymbol{Q}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}}^{(k)} \leftarrow \boldsymbol{L}^{(k)}\boldsymbol{Q}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}},-}^{(k)}\left(\boldsymbol{L}^{(k)}\right)^T + \mathbf{K}^{(k,m)}\mathbf{Q}_{\mathbf{hh}}^{(k,m)}\left(\mathbf{K}^{(k,m)}\right)^T$$

$$\hat{\boldsymbol{x}}_+^{(k)} \leftarrow \hat{\boldsymbol{x}}_+^{(k,m+1)}$$

VCE:

$$\hat{\boldsymbol{v}}_t^{(k)} = \left(\mathbf{Q}_{\boldsymbol{hh}}^{(k)}\right)^{-1}\boldsymbol{B}^{(k)}\hat{\boldsymbol{v}}^{(k)}$$

for $f = 1 \,..\, z$ **do**

$$\mathbf{Q}_{\boldsymbol{hh},f}^{(k)} \leftarrow \boldsymbol{B}^{(k)}\left(\boldsymbol{\Delta}\hat{\boldsymbol{\sigma}}_f\mathbf{V}_{\mathrm{f}}^{(k)}\right)\left(\boldsymbol{B}^{(k)}\right)^T$$

$$\hat{q}_f^{(k)} \leftarrow \left(\hat{\boldsymbol{v}}_t^{(k)}\right)^T\mathbf{Q}_{\boldsymbol{hh},f}^{(k)}\hat{\boldsymbol{v}}_t^{(k)}$$

for $r = 1 \,..\, z$ **do**

for $c = 1 \,..\, z$ **do**

$$\hat{s}_{(r,c)}^{(k)} \leftarrow \operatorname{trace}\left(\left(\mathbf{Q}_{\boldsymbol{rr}}^{(k)}\right)^{-1}\mathbf{Q}_{\boldsymbol{hh},c}^{(k)}\left(\mathbf{Q}_{\boldsymbol{rr}}^{(k)}\right)^{-1}\mathbf{Q}_{\boldsymbol{hh},r}^{(k)}\right)$$

$$\boldsymbol{Q}_{\hat{\boldsymbol{x}}_s\hat{\boldsymbol{x}}_s}^{(k)} \leftarrow \left(\hat{\mathbf{S}}^{(k)} + \left(\boldsymbol{Q}_{\hat{\boldsymbol{x}}_s\hat{\boldsymbol{x}}_s}^{(k-1)}\right)^{-1}\right)^{-1}$$

$$\hat{\boldsymbol{x}}_s^{(k)} \leftarrow \mathbf{Q}_{\hat{\boldsymbol{x}}_s\hat{\boldsymbol{x}}_s}^{(k)}\left(\hat{\mathbf{Q}}^{(k)} + \left(\mathbf{Q}_{\hat{\boldsymbol{x}}_s\hat{\boldsymbol{x}}_s}^{(k-1)}\right)^{-1}\hat{\boldsymbol{x}}_s^{(k-1)}\right)$$

else $\rightarrow$ no observations available

$$\boldsymbol{Q}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}}^{(k)} \leftarrow \boldsymbol{Q}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}},-}^{(k)}$$

$$\hat{\boldsymbol{x}}_+^{(k)} \leftarrow \hat{\boldsymbol{x}}_-^{(k)}$$

Returns: $\hat{\boldsymbol{x}}_+$, $\boldsymbol{Q}_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}}$, $\hat{\boldsymbol{x}}_s$, $\boldsymbol{Q}_{\hat{\boldsymbol{x}}_s\hat{\boldsymbol{x}}_s}$ for all (k)

Bibliography

[Alkhatib 2008] Alkhatib, Hamza: *On Monte Carlo methods with applications to the current satellite gravity missions (PhD thesis)*, Rheinische Friedrich-Wilhelms-Universität Bonn, Landwirtschaftliche Fakultät, IGG - Institut für Geodäsie und Geoinformation, Dissertation, 2008. – URL `https://bonndoc.ulb.uni-bonn.de/xmlui/handle/20.500.11811/1389`

[Amiri-Simkooei et al. 2009] Amiri-Simkooei, A. R. ; Teunissen, P. J. G. ; Tiberius, C. C. J. M.: Application of Least-Squares Variance Component Estimation to GPS Observables. In: *Journal of Surveying Engineering* 135 (2009), Nr. 4, S. 149–160

[Amiri-Simkooei 2007] Amiri-Simkooei, AliReza: *Least-squares variance component estimation: Theory and GPS applications (PhD thesis)*. Delft : Nederlandse Commissie voor Geodesie, 2007

[Andrew 2002] Andrew, Alex M.: Applied Interval Analysis: With Examples in Parameter and State Estimation, Robust Control and Robotics. In: *Kybernetes* 31 (2002), Nr. 5

[Axmann et al. 2023] Axmann, Jeldrik ; Moftizadeh, Rozhin ; Su, Jingyao ; Tennstedt, Benjamin ; Zou, Qianqian ; Yuan, Yunshuang ; Ernst, Dominik ; Alkhatib, Hamza ; Brenner, Claus ; Schön, Steffen: LUCOOP: Leibniz University Cooperative Perception and Urban Navigation Dataset. In: *2023 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2023, S. 1–8

[Baarda 1967] Baarda, W.: *Statistical concepts in geodesy*. Bd. 2. Netherlands Geodetic Commission, 1967

[Balangé et al. 2021] Balangé, Laura ; Zhang, Li ; Schwieger, Volker: First Step Towards the Technical Quality Concept for Integrative Computational Design and Construction. In: *Contributions to International Conferences on Engineering Surveying* (2021), S. 118–127

[Bar-Shalom et al. 2004] Bar-Shalom, Yaakov ; Li, X. R. ; Kirubarajan, Thiagalingam: *Estimation with applications to tracking and navigation: theory algorithms and software*. New York : John Wiley & Sons, 2004

[Barron 2019] Barron, Jonathan T.: A General and Adaptive Robust Loss Function. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2019, S. 4326–4334

[Bentley 1975] Bentley, Jon L.: Multidimensional binary search trees used for associative searching. In: *Communications of the ACM* 18 (1975), Nr. 9, S. 509–517

[Bergelt et al. 2017] Bergelt, Rene ; Khan, Owes ; Hardt, Wolfram: Improving the intrinsic calibration of a Velodyne LiDAR sensor. In: *2017 IEEE SENSORS*, IEEE, 2017, S. 1–3

[Brazeal et al. 2021] Brazeal, Ryan G. ; Wilkinson, Benjamin E. ; Hochmair, Hartwig H.: A Rigorous Observation Model for the Risley Prism-Based Livox Mid-40 Lidar Sensor. In: *Sensors (Basel, Switzerland)* 21 (2021), Nr. 14, S. 1–20

[Bureick et al. 2019] Bureick, Johannes ; Vogel, Sören ; Neumann, Ingo ; Unger, Jakob ; Alkhatib, Hamza: Georeferencing of an Unmanned Aerial System by Means of an Iterated Extended Kalman Filter Using a 3D City Model. In: *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science* 87 (2019), Nr. 5-6, S. 229–247

[Caspary & Wang 1998] CASPARY, Wilhelm ; WANG, Jianguo: Redundanzanteile und Varianzkomponenten im Kalman Filter. In: *Zeitschrift für Vermessungswesen* 123 (1998), Nr. 4, S. 121–128

[Clearpath Robotics 2021] CLEARPATH ROBOTICS ; CLEARPATH ROBOTICS (Hrsg.): *Husky Unmanned Ground Vehicle robot*. 2021. – URL `https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/`. – Zugriffsdatum: 2021-07-23

[Coenen & Rottensteiner 2021] COENEN, Max ; ROTTENSTEINER, Franz: Pose estimation and 3D reconstruction of vehicles from stereo-images using a subcategory-aware shape prior. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 181 (2021), S. 27–47

[Dammert et al. 2025] DAMMERT, Lucas ; THALMANN, Tomas ; MONETTI, David ; NEUNER, Hans-Berndt ; MANDLBURGER, Gottfried: A Review on UAS Trajectory Estimation Using Decentralized Multi-Sensor Systems Based on Robotic Total Stations. In: *Sensors (Basel, Switzerland)* 25 (2025), Nr. 13, S. 1–22. – URL `https://www.mdpi.com/1424-8220/25/13/3838`

[Dang 2008] DANG, Thao: An iterative parameter estimation method for observation models with nonlinear constraints. In: *Metrology and Measurement Systems* 15 (2008), Nr. 4, S. 421–432

[Debeunne & Vivet 2020] DEBEUNNE, César ; VIVET, Damien: A Review of Visual-LiDAR Fusion based Simultaneous Localization and Mapping. In: *Sensors (Basel, Switzerland)* 20 (2020), Nr. 7, S. 1–20. – URL `https://www.mdpi.com/684622`

[Dosovitskiy et al. 2017] DOSOVITSKIY, Alexey ; ROS, German ; CODEVILLA, Felipe ; LOPEZ, Antonio ; KOLTUN, Vladlen: CARLA: An Open Urban Driving Simulator. In: *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, S. 1–16

[Dreier et al. 2021] DREIER, Ansgar ; JANSSEN, Jannik ; KUHLMANN, Heiner ; KLINGBEIL, Lasse: Quality Analysis of Direct Georeferencing in Aspects of Absolute Accuracy and Precision for a UAV-Based Laser Scanning System. In: *Remote Sensing* 13 (2021), Nr. 18. – URL `https://www.mdpi.com/2072-4292/13/18/3564`

[Dreier et al. 2024] DREIER, Ansgar ; JOST, Berit ; KUHLMANN, Heiner ; KLINGBEIL, Lasse: Investigations of the scan characteristics with special focus on multi-target capability for the 2D laser scanner RIEGL miniVUX-2UAV. In: *Journal of Applied Geodesy* 18 (2024), Nr. 1, S. 97–113. – URL `https://www.degruyter.com/document/doi/10.1515/jag-2022-0029/html`

[Drixler 1993] DRIXLER, Erwin: *Analyse der Form und Lage von Objekten im Raum: PhD Thesis*. Bd. 409. München : DGK, Reihe C, 1993

[Durrant-Whyte & Henderson 2016] DURRANT-WHYTE, Hugh ; HENDERSON, Thomas C.: Multisensor Data Fusion. In: *Springer Handbook of Robotics* (2016), S. 867–896

[Esser et al. 2023] ESSER, Felix ; KLINGBEIL, Lasse ; ZABAWA, Lina ; KUHLMANN, Heiner: Quality Analysis of a High-Precision Kinematic Laser Scanning System for the Use of Spatio-Temporal Plant and Organ-Level Phenotyping in the Field. In: *Remote Sensing* 15 (2023), Nr. 4, S. 1–21

[Esser et al. 2024] ESSER, Felix ; TOMBRINK, Gereon ; CORNELISSEN, Andre ; KLINGBEIL, Lasse ; KUHLMANN, Heiner: System Calibration of a Field Phenotyping Robot with Multiple High-Precision Profile Laser Scanners. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2024, S. 8471–8477

[Ettlinger et al. 2018] Ettlinger, Andreas ; Neuner, Hans ; Burgess, Thomas: Development of a Kalman Filter in the Gauss-Helmert Model for Reliability Analysis in Orientation Determination with Smartphone Sensors. In: *Sensors (Basel, Switzerland)* 18 (2018), Nr. 2, S. 414. – URL https://www.mdpi.com/1424-8220/18/2/414

[Fang et al. 2022] Fang, Xing ; Hu, Yu ; Zeng, Wenxian ; Akyilmaz, O.: Weighted least-squares fitting of circles with variance component estimation. In: *Measurement* 205 (2022), S. 1–11

[Förstner & Wrobel 2016] Förstner, Wolfgang ; Wrobel, Bernhard P.: *Photogrammetric Computer Vision*. Bd. 11. Cham : Springer International Publishing, 2016

[Geneva et al. 2018] Geneva, Patrick ; Eckenhoff, Kevin ; Yang, Yulin ; Huang, Guoquan: LIPS: LiDAR-Inertial 3D Plane SLAM. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, S. 123–130

[Ghallabi et al. 2018] Ghallabi, Farouk ; Nashashibi, Fawzi ; El-Haj-Shhade, Ghayath ; Mittet, Marie-Anne: LIDAR-Based Lane Marking Detection For Vehicle Positioning in an HD Map. In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018, S. 2209–2214

[Glennie 2007] Glennie, Craig: Rigorous 3D error analysis of kinematic scanning LIDAR systems. In: *Journal of Applied Geodesy* 1 (2007), Nr. 3, S. 147–157. – URL https://www.degruyter.com/document/doi/10.1515/jag.2007.017/html

[Glennie & Lichti 2010] Glennie, Craig ; Lichti, Derek D.: Static Calibration and Analysis of the Velodyne HDL-64E S2 for High Accuracy Mobile Scanning. In: *Remote Sensing* 2 (2010), Nr. 6, S. 1610–1624. – URL https://www.mdpi.com/17498

[Glennie et al. 2016] Glennie, Craig L. ; Kusari, A. ; Facchin, A.: Calibration and Stability Analysis of the VLP-16 Laser Scanner. In: *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences* XL-3/W4 (2016), S. 55–60

[Hartmann et al. 2024] Hartmann, Jan ; Ernst, Dominik ; Neumann, Ingo ; Alkhatib, Hamza: PointNet-based modeling of systematic distance deviations for improved TLS accuracy. In: *Journal of Applied Geodesy* 18 (2024), Nr. 4, S. 613–628. – URL https://www.degruyter.com/document/doi/10.1515/jag-2023-0097/xml

[Hartmann et al. 2023] Hartmann, Jan ; Heiken, Max ; Alkhatib, Hamza ; Neumann, Ingo: Automatic quality assessment of terrestrial laser scans. In: *Journal of Applied Geodesy* 17 (2023), Nr. 4, S. 333–353

[Hartmann 2023] Hartmann, Jens: *Hochgenaue 3D-Objekterfassung durch kinematisches terrestrisches Laserscanning. PhD thesis*. Bd. 905. DGK, Reihe C, 2023

[Hartmann et al. 2019] Hartmann, Jens ; Gösseln, Ilka von ; Schild, Niklas ; Dorndorf, Alexander ; Paffenholz, Jens-André ; Neumann, Ingo: Optimisation of the Calibration Process of a k-TLS based Multi-Sensor-System by Genetic Algorithms. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2019, S. 1655–1662

[Hartmann et al. 2017] Hartmann, Jens ; Paffenholz, Jens-André ; Strübing, Thorsten ; Neumann, Ingo: Determination of Position and Orientation of LiDAR Sensors on Multisensor Platforms. In: *Journal of Surveying Engineering* 143 (2017), Nr. 4, S. 1–11

[Hartmann et al. 2018] Hartmann, Jens ; Trusheim, Philipp ; Alkhatib, Hamza ; Paffenholz, Jens-André ; Diener, Dmitri ; Neumann, Ingo: High Accurate Pointwise (Geo-) Referencing of a k-TLS based Multi-Sensor-System. In: *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences* 4 (2018), Nr. 4, S. 81–88

[Heiker 2013] Heiker, Andrea: *Mutual validation of Earth orientation parameters, geophysical excitation functions and second degree gravity field coefficients*. Bd. 697. München : DGK, Reihe C, 2013

[Heinz 2021] Heinz, Erik: *Beiträge zur Kalibrierung und Evaluierung von Multisensorsystemen für kinematisches Laserscanning (Dissertation)*. Universitäts-und Landesbibliothek Bonn, 2021

[Heinz et al. 2015] Heinz, Erik ; Eling, Christian ; Wieland, Markus ; Klingbeil, Lasse ; Kuhlmann, Heiner: Development, calibration and evaluation of a portable and direct georeferenced laser scanning system for kinematic 3D mapping. In: *Journal of Applied Geodesy* 9 (2015), Nr. 4, S. 227–243

[Hexagon Metrology 2023] Hexagon Metrology ; Hexagon Metrology (Hrsg.): *Leica Absolute Tracker AT960*. 2023. – URL `https://hexagon.com/products/leica-absolute-tracker-at960`. – Zugriffsdatum: 2023-01-27

[Hillemann et al. 2019] Hillemann, Markus ; Weinmann, Martin ; Mueller, Markus S. ; Jutzi, Boris: Automatic Extrinsic Self-Calibration of Mobile Mapping Systems Based on Geometric 3D Features. In: *Remote Sensing* 11 (2019), Nr. 16, S. 1–29

[Hofmann 2017] Hofmann, Sabine: *Potential von LiDAR mobile mapping für hochgenaue Karten (PhD thesis)*. Bd. 801. München : DGK, Reihe C, 2017

[Hofmann-Wellenhof et al. 2012] Hofmann-Wellenhof, Bernhard ; Lichtenegger, Herbert ; Collins, James: *Global positioning system: theory and practice*. Springer Science & Business Media, 2012

[Holland & Welsch 1977] Holland, Paul W. ; Welsch, Roy E.: Robust regression using iteratively reweighted least-squares. In: *Communications in Statistics - Theory and Methods* 6 (1977), Nr. 9, S. 813–827

[Horn 1987] Horn, Berthold K.: Closed-form solution of absolute orientation using unit quaternions. In: *Journal of the optical society of America A* 4 (1987), Nr. 4, S. 629–642

[Hsu 2018] Hsu, Li-Ta: Analysis and modeling GPS NLOS effect in highly urbanized area. In: *GPS solutions* 22 (2018), Nr. 7, S. 1–12

[Hu et al. 2025] Hu, Hong ; Dong, Tanwen ; Liu, Kerui ; Fang, Yubao ; Kang, Ruihong: Least squares variance component estimation for robust GNSS/INS/LiDAR integrated with large-scale navigation. In: *Measurement Science and Technology* 36 (2025), Nr. 7, S. 1–15

[Hu et al. 2012] Hu, Jun ; Li, Zhi-Wei ; Sun, Qian ; Zhu, Jian-Jun ; Ding, Xiao-Li: Three-Dimensional Surface Displacements From InSAR and GPS Measurements With Variance Component Estimation. In: *IEEE Geoscience and Remote Sensing Letters* 9 (2012), Nr. 4, S. 754–758

[ISO 5725-1:2023 2023] ISO 5725-1:2023: *Accuracy (trueness and precision) of measurement methods and results, Part 1: General principles and definitions*. ISO, Geneva, Switzerland, 2023

[Janßen 2024] Janssen, Jannik: *Reduktion der Unsicherheiten bei der Registrierung von TLS-Punktwolken (Dissertation)*. Bonn : Universitäts-und Landesbibliothek Bonn, 2024

[JAVAD 2025] JAVAD ; JAVAD (Hrsg.): *TRIUMPH-1M Datasheet.* 2025. – URL `https://www.javad.com/product/triumph-1m/`. – Zugriffsdatum: 2025-11-18

[JCGM 2008a] JCGM: *JCGM 100: Evaluation of Measurement Data - Guide to the Expression of Uncertainty in Measurement (GUM).* Sèvres Cedex : Joint Commitee for Guides in Metrology (JCGM/WG1), 2008

[JCGM 2008b] JCGM: *JCGM 101: Evaluation of Measurement Data - Supplement 1 to the Guide to the Expression of Uncertainty in Measurement - Propagation of distributions using a Monte Carlo method.* Sèvres Cedex : Joint Commitee for Guides in Metrology (JCGM/WG1), 2008

[Jost 2023] Jost, Berit: *Strategies for the Empirical Determination of the Stochastic Properties of Terrestrial Laser Scans (Dissertation).* Universitäts-und Landesbibliothek Bonn, 2023

[Jüngling 2023] Jüngling, Moritz: *Langzeituntersuchung der intrinsischen Kalibrierparameter eines Velodyne VLP-16.* Hannover, Leibniz Universität Hannover, Bachelor thesis (unpublished), 2023

[Kalenjuk & Lienhart 2022] Kalenjuk, Slaven ; Lienhart, Werner: A Method for Efficient Quality Control and Enhancement of Mobile Laser Scanning Data. In: *Remote Sensing* 14 (2022), Nr. 4, S. 857. – URL `https://www.mdpi.com/2072-4292/14/4/857`

[Kellner 2024] Kellner, Katharina: *Analyse des Livox HAP LiDAR.* Hannover, Leibniz Universität Hannover, Bachelor thesis (unpublished), 2024

[Kerekes 2023] Kerekes, Gabriel: *An elementary error model for terrestrial laser scanning*, University of Stuttgart Germany, Dissertation, 2023

[Khami et al. 2025] Khami, Arman ; Neumann, Ingo ; Vogel, Sören: Intrinsic and extrinsic calibration of a UAV-based multi-sensor system. In: *Journal of Applied Geodesy* (2025), S. 1–19

[Kim et al. 2021] Kim, Han S. ; Kim, Yongil ; Kim, Changjae ; Choi, Kang H.: Kinematic In Situ Self-Calibration of a Backpack-Based Multi-Beam LiDAR System. In: *Applied Sciences* 11 (2021), Nr. 3, S. 945. – URL `https://www.mdpi.com/2076-3417/11/3/945`

[Kim & Chung 2018] Kim, Jiwoong ; Chung, Woojin: Robust Localization of Mobile Robots Considering Reliability of LiDAR Measurements. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, S. 6491–6496

[Klein & Filin 2011] Klein, I. ; Filin, S.: LiDAR and INS Fusion in Periosn of GPS Outages for Mobiles Laser Scanning Mapping Systems. In: *ISPRS Archives* XXXVIII-5/W12 (2011), S. 231–236

[Koch & Kusche 2002] Koch, K.-R. ; Kusche, J.: Regularization of geopotential determination from satellite data by variance components. In: *Journal of Geodesy* 76 (2002), Nr. 5, S. 259–268

[Koch 2002] Koch, Karl-Rudolf: Räumliche Helmert-Transformation variabler Koordinaten im Gauß-Helmert- und im Gauß-Markoff-Modell. In: *ZfV* 127 (2002), Nr. 3, S. 147–152

[Koch 2004] Koch, Karl-Rudolf: *Parameterschätzung und Hypothesentests in linearen Modellen. 4., bearbeitete Auflage.* Dümmler, 2004

[Koch 2007] Koch, Karl-Rudolf: *Introduction to Bayesian statistics.* 2., updated and enl. ed. Berlin and Heidelberg : Springer, 2007

[Koenig & Howard 2004] Koenig, N. ; Howard, A.: Design and use paradigms for Gazebo, an open-source multi-robot simulator. In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)* Bd. 3, URL `https://ieeexplore.ieee.org/abstract/document/1389727`, Sep. 2004, S. 2149–2154 vol.3

[Kröger et al. 2021] Kröger, Johannes ; Kersten, Tobias ; Breva, Yannick ; Schön, Steffen: Multi-frequency multi-GNSS receiver antenna calibration at IfE: Concept - calibration results - validation. In: *Advances in Space Research* 68 (2021), Nr. 12, S. 4932–4947

[Kümmerle & Kuhner 2020] Kümmerle, Julius ; Kuhner, Tilman: Unified Intrinsic and Extrinsic Camera and LiDAR Calibration under Uncertainties. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, S. 6028–6034

[Lee et al. 2022] Lee, Woosik ; Geneva, Patrick ; Yang, Yulin ; Huang, Guoquan: Tightly-coupled GNSS-aided Visual-Inertial Localization. In: *2022 International Conference on Robotics and Automation (ICRA)*, 2022, S. 9484–9491

[Lenzmann & Lenzmann 2004] Lenzmann, Lothar ; Lenzmann, Enno: Strenge Auswertung des nichtlinearen Gauß-Helmert-Modells. In: *Allgemeine Vermessungs-Nachrichten (AVN)* 5 (2004), Nr. 2, S. 68–73

[Lerke & Schwieger 2021] Lerke, Otto ; Schwieger, Volker: Analysis of a kinematic real-time robotic total station network for robot control. In: *Journal of Applied Geodesy* 15 (2021), Nr. 3, S. 169–188

[Levinson & Thrun 2010] Levinson, Jesse ; Thrun, Sebastian: Robust vehicle localization in urban environments using probabilistic maps. In: *2010 IEEE International Conference on Robotics and Automation*, IEEE, 2010, S. 4372–4378

[Levinson & Thrun 2014] Levinson, Jesse ; Thrun, Sebastian: Unsupervised Calibration for Multi-beam Lasers. In: *Springer Tracts in Advanced Robotics*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2014, S. 179–193

[Li et al. 2024] Li, Longze ; Chang, Jiang ; Vakanski, Aleksandar ; Wang, Yachun ; Yao, Tiankai ; Xian, Min: Uncertainty quantification in multivariable regression for material property prediction with Bayesian neural networks. In: *Scientific Reports* 14 (2024), Nr. 1, S. 1–15

[Li et al. 2023] Li, Shengyu ; Li, Xingxing ; Zhou, Yuxuan ; Xia, Chunxi: Targetless Extrinsic Calibration of LiDAR–IMU System Using Raw GNSS Observations for Vehicle Applications. In: *IEEE Transactions on Instrumentation and Measurement* 72 (2023), S. 1–11

[Liu & Zhang 2021] Liu, Zheng ; Zhang, Fu: BALM: Bundle Adjustment for Lidar Mapping. In: *IEEE Robotics and Automation Letters* 6 (2021), Nr. 2, S. 3184–3191

[Lou et al. 2005] Lou, Jianguang ; Tan, Tieniu ; Hu, Weiming ; Yang, Hao ; Maybank, Steven J.: 3-D model-based vehicle tracking. In: *IEEE Transactions on Image Processing* 14 (2005), Nr. 10, S. 1561–1569

[Lucks et al. 2021] Lucks, Lukas ; Klingbeil, Lasse ; Plümer, Lutz ; Dehbi, Youness: Improving trajectory estimation using 3D city models and kinematic point clouds. In: *Transactions in GIS* 25 (2021), Nr. 1, S. 238–260

[Luhmann 2000] Luhmann, Thomas: *Nahbereichsphotogrammetrie*. Wichmann Verlag, 2000

[Lv et al. 2022] Lv, Jiajun ; Zuo, Xingxing ; Hu, Kewei ; Xu, Jinhong ; Huang, Guoquan ; Liu, Yong: Observability-Aware Intrinsic and Extrinsic Calibration of LiDAR-IMU Systems. In: *IEEE Transactions on Robotics* 38 (2022), Nr. 6, S. 3734–3753

[Mallozzi et al. 2019] MALLOZZI, Piergiuseppe ; PELLICCIONE, Patrizio ; KNAUSS, Alessia ; BERGER, Christian ; MOHAMMADIHA, Nassar: Autonomous Vehicles: State of the Art, Future Trends, and Challenges. In: *Automotive Systems and Software Engineering.* Springer, Cham, 2019, S. 347–367

[Medic 2021] MEDIC, Tomislav: *Efficient calibration strategies for panoramic terrestrial laser scanners (PhD Thesis).* Universitäts-und Landesbibliothek Bonn, 2021

[Mercedes-Benz Group 2022] MERCEDES-BENZ GROUP: *Mercedes-Benz backs redundancy for safe conditionally automated driving | Mercedes-Benz Group.* 2022. – URL `https://group.mercedes-benz.com/innovation/product-innovation/autonomous-driving/redundancy-drive-pilot.html`. – Zugriffsdatum: 2025-11-27

[Milioto et al. 2019] MILIOTO, Andres ; VIZZO, Ignacio ; BEHLEY, Jens ; STACHNISS, Cyrill: RangeNet ++: Fast and Accurate LiDAR Semantic Segmentation. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2019, S. 4213–4220

[Mishra et al. 2021] MISHRA, Subodh ; PANDEY, Gaurav ; SARIPALLI, Srikanth: Target-free Extrinsic Calibration of a 3D-Lidar and an IMU. In: *2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, IEEE, 2021, S. 1–7

[Moftizadeh 2024] MOFTIZADEH, Rozhin: *Advanced particle filtering for vehicle navigation based on collaborative information (PhD thesis).* Bd. 941. München : DGK, Reihe C, 2024

[Möser et al. 2013] MÖSER, Michael ; MÜLLER, Gerhard ; SCHLEMMER, Harald ; HEUNECKE, O ; KUHLMANN, H ; WELSCH, W ; EICHHORN, A ; NEUNER, H: Auswertung geodätischer Überwachungsmessungen. In: *Berlin: Wichmann* (2013)

[Mounier et al. 2023] MOUNIER, Eslam ; ELHABIBY, Mohamed ; KORENBERG, Michael ; NOURELDIN, Aboelmagd: *High-Precision Positioning in GNSS-Challenged Environments: A LiDAR-Based Multi-Sensor Fusion Approach with 3D Digital Maps Registration.* TechRxiv, 2023

[Mounier et al. 2025] MOUNIER, Eslam ; ELHABIBY, Mohamed ; KORENBERG, Michael ; NOURELDIN, Aboelmagd: LiDAR-based multi-sensor fusion with 3D digital maps for high-precision positioning. In: *IEEE Internet of Things Journal* 12 (2025), Nr. 6, S. 7209–7224

[Niehsen 2022] NIEHSEN, Lars: *Intrinsische Kalibrierung eines Velodyne VLP-16.* Hannover, Leibniz Universität Hannover, Bachelor thesis (unpublished), 2022

[Niemeier 2008] NIEMEIER, Wolfgang: *Ausgleichungsrechnung: Statistische Auswertemethoden.* Berlin and New York : Walter de Gruyter, 2008

[Nüchter et al. 2018] NÜCHTER, Andreas ; BLEIER, Michael ; SCHAUER, Johannes ; JANOTTA, Peter: Continuous-Time SLAM—Improving Google's Cartographer 3D Mapping. In: *Latest Developments in Reality-Based 3D Surveying and Modelling.* MDPI, 2018, S. 53–73

[Oth et al. 2013] OTH, Luc ; FURGALE, Paul ; KNEIP, Laurent ; SIEGWART, Roland: Rolling Shutter Camera Calibration. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2013, S. 1360–1367

[Othman 2021] OTHMAN, Kareem: Public acceptance and perception of autonomous vehicles: a comprehensive review. In: *AI and Ethics* 1 (2021), Nr. 3, S. 355–387

[Park & Chung 2024] PARK, Geonhyeok ; CHUNG, Woojin: Uncertainty–aware LiDAR–based localization for outdoor mobile robots. In: *Journal of Field Robotics* 41 (2024), Nr. 8, S. 2790–2804

[Pöppl et al. 2023] Pöppl, Florian ; Neuner, Hans ; Mandlburger, Gottfried ; Pfeifer, Norbert: Integrated trajectory estimation for 3D kinematic mapping with GNSS, INS and imaging sensors: A framework and review. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 196 (2023), S. 287–305

[Qin et al. 2020] Qin, Chao ; Ye, Haoyang ; Pranata, Christian E. ; Han, Jun ; Zhang, Shuyang ; Liu, Ming: LINS: A Lidar-Inertial State Estimator for Robust and Efficient Navigation. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, S. 8899–8906

[Qin et al. 2022] Qin, Peilin ; Zhang, Chuanwei ; Ma, Xiaowen ; Shi, Zhenghe: High–Precision Motion Compensation for LiDAR Based on LiDAR Odometry. In: *Wireless Communications and Mobile Computing* 2022 (2022), Nr. 1, S. 1–12

[Reid et al. 2019] Reid, Tyler G. ; Houts, Sarah E. ; Cammarata, Robert ; Mills, Graham ; Agarwal, Siddharth ; Vora, Ankit ; Pandey, Gaurav: Localization Requirements for Autonomous Vehicles. In: *SAE International Journal of Connected and Automated Vehicles* 2 (2019), Nr. 3

[Riegl 2011] Riegl: *Data Sheet VMX-250*. 2011. – URL `https://www.riegl.com/de-deutschland/news-insights/article/riegl-vmx-250-with-modular-camera-system-combined-scan-and-image-data-acquisition-in-mobile-laser-scanning`. – Zugriffsdatum: 2025-12-19

[Sánchez & Pany 2019] Sánchez, Daniela E. ; Pany, Thomas: Analysis and Calibration of the VLP-16 LiDAR for Automotive Applications. In: *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*, 2019, S. 2611–2621

[Schill et al. 2025] Schill, Florian ; Holst, Christoph ; Wujanz, Daniel ; Hartmann, Jens ; Paffenholz, Jens-André: Intensity-based stochastic model of terrestrial laser scanners: Methodological workflow, empirical derivation and practical benefit. In: *ISPRS Open Journal of Photogrammetry and Remote Sensing* 15 (2025), S. 1–17

[Schlichting et al. 2019] Schlichting, Alexander ; Zachert, Fabio ; Forouher, Dariush: Map Matching for Vehicle Localization Based on Serial Lidar Sensors. In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2019, S. 1257–1262

[Shi et al. 2020] Shi, Bo ; Bai, Yuntian ; Zhang, Shun ; Zhong, Ruofei ; Yang, Fanlin ; Song, Shizhu ; Li, Guoyu: Reference-plane-based approach for accuracy assessment of mobile mapping point clouds. In: *Measurement* 171 (2020), S. 1–11

[Simon 2006] Simon, Dan: *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. Hoboken, NJ : John Wiley & Sons, 2006

[Solà 2017] Solà, Joan: *Quaternion kinematics for the error-state Kalman filter*. arXiv, 2017

[Soudarissanane et al. 2009] Soudarissanane, Sylvie ; Lindenbergh, Roderik ; Menenti, Massimo ; Teunissen, P. J.: Incidence angle influence on the quality of terrestrial laser scanning points. In: *Proceedings ISPRS Workshop Laserscanning 2009, 1-2 Sept 2009, Paris, France*, 2009, S. 183–188

[Stempfhuber 2004] Stempfhuber, Werner V.: *Ein integritätswahrendes Messsystem für kinematische Anwendungen: PhD thesis*. Bd. 576. München : DGK, Reihe C, 2004

[Stenz et al. 2017] Stenz, Ulrich ; Hartmann, Jens ; Paffenholz, Jens-André ; Neumann, Ingo: A Framework Based on Reference Data with Superordinate Accuracy for the Quality Analysis of Terrestrial Laser Scanning-Based Multi-Sensor-Systems. In: *Sensors (Basel, Switzerland)* 17 (2017), Nr. 8, S. 1886

[Stenz et al. 2020] Stenz, Ulrich ; Hartmann, Jens ; Paffenholz, Jens-André ; Neumann, Ingo: High-Precision 3D Object Capturing with Static and Kinematic Terrestrial Laser Scanning in Industrial Applications—Approaches of Quality Assessment. In: *Remote Sensing* 12 (2020), Nr. 2

[Strübing & Neumann 2013] Strübing, Thorsten ; Neumann, Ingo: Positions- und Orientierungsschätzung von LIDAR-Sensoren auf Multisensorplattformen. In: *ZfV* 3 (2013), S. 210–221

[Su 2025] Su, Jiangyao: *Towards interval-based autonomous integrity monitoring: Error bounding and uncertainty propagation (PhD thesis).* München : DGK, Reihe C, 2025

[Thalmann & Neuner 2021] Thalmann, Thomas ; Neuner, Hans: Temporal calibration and synchronization of robotic total stations for kinematic multi-sensor-systems. In: *Journal of Applied Geodesy* 15 (2021), Nr. 1, S. 13–30. – URL `https://www.degruyterbrill.com/document/doi/10.1515/jag-2019-0070/html`

[Titterton & Weston 2004] Titterton, David H. ; Weston, John L.: *Strapdown inertial navigation technology.* Bd. 17. 2. ed. Stevenage : The Inst. of Engineering and Technology, 2004

[Tombrink et al. 2023] Tombrink, Gereon ; Dreier, Ansgar ; Klingbeil, Lasse ; Kuhlmann, Heiner: Trajectory evaluation using repeated rail-bound measurements. In: *Journal of Applied Geodesy* 17 (2023), Nr. 3, S. 205–216. – URL `https://www.degruyter.com/document/doi/10.1515/jag-2022-0027/html`

[Tombrink et al. 2025] Tombrink, Gereon ; Dreier, Ansgar ; Klingbeil, Lasse ; Kuhlmann, Heiner: Spatio-temporal trajectory alignment for trajectory evaluation. In: *Journal of Applied Geodesy* 19 (2025), Nr. 2, S. 321–331

[Toth et al. 2009] Toth, Charles ; Grejner-Brezezinska, D. A. ; Oh, J. H. ; Markiel, J. N.: Terrain-based navigation: A tool to improve navigation and feature extraction performance of mobile mapping systems. In: *Boletim de Ciências Geodésicas* 15 (2009), Nr. 5, S. 807–823

[Trusheim et al. 2024] Trusheim, Philipp ; Mehltretter, Max ; Rottensteiner, Franz ; Heipke, Christian: Cooperative Image Orientation with Dynamic Objects. In: *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science* 92 (2024), Nr. 5, S. 461–481

[ublox 2025] ublox ; ublox (Hrsg.): *ZED-F9K Datasheet.* 2025. – URL `https://www.u-blox.com/en/product/zed-f9k-module`. – Zugriffsdatum: 2025-11-18

[Velodyne 2019a] Velodyne ; Velodyne LiDAR, Inc (Hrsg.): *VLP-16 Datasheet.* 2019

[Velodyne 2019b] Velodyne ; Velodyne LiDAR, Inc (Hrsg.): *VLP-16 Manual: User's Manual and Programming Guide.* 2019

[Vennegeerts 2011] Vennegeerts, Harald: *Objektraumgestützte kinematische Georeferenzierung für Mobile-Mapping-Systeme.* Bd. 657. München : DGK, Reihe C, 2011

[Vizzo et al. 2023] Vizzo, Ignacio ; Guadagnino, Tiziano ; Mersch, Benedikt ; Wiesmann, Louis ; Behley, Jens ; Stachniss, Cyrill: KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way. In: *IEEE Robotics and Automation Letters* 8 (2023), Nr. 2, S. 1029–1036

[Vogel 2020] Vogel, Sören: *Kalman Filtering with State Constraints Applied to Multi-sensor Systems and Georeferencing: Ph.D. Thesis.* München : DGK, Reihe C, 2020

[Vogel et al. 2018] Vogel, Sören ; Alkhatib, Hamza ; Neumann, Ingo: Iterated Extended Kalman Filter with Implicit Measurement Equation and Nonlinear Constraints for Information-Based Georeferencing. In: *2018 21st International Conference on Information Fusion (FUSION)*, 2018, S. 1209–1216

[Vogel et al. 2022] Vogel, Sören ; Ernst, Dominik ; Neumann, Ingo ; Alkhatib, Hamza: Recursive Gauss-Helmert model with equality constraints applied to the efficient system calibration of a 3D laser scanner. In: *Journal of Applied Geodesy* 16 (2022), Nr. 1, S. 37–57

[Wahbah et al. 2025] Wahbah, Mohamad ; Ramme, Lukas ; Ernst, Dominik ; Vogel, Sören ; Neumann, Ingo ; Alkhatib, Hamza: Geo-referencing Autonomous Vehicles Using LoD2 and HD Maps: Performance Assessment in Simulated Urban Environments. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* X-G-2025 (2025), S. 915–922

[Wang et al. 2009] Wang, Jian-Guo ; Gopaul, Nilesh ; Scherzinger, Bruno: Simplified Algorithms of Variance Component Estimation for Static and Kinematic GPS Single Point Positioning. In: *Journal of Global Positioning Systems* 8 (2009), Nr. 1, S. 43–52

[Waymo 2025] Waymo: *Safe, Routine, Ready: Autonomous driving in five new cities.* 2025. – URL `https://waymo.com/blog/2025/11/safe-routine-ready-autonomous-driving-in-new-cities`. – Zugriffsdatum: 2025-11-27

[Wei et al. 2018] Wei, Pan ; Cagle, Lucas ; Reza, Tasmia ; Ball, John ; Gafford, James: LiDAR and camera detection fusion in a real-time industrial multi-sensor collision avoidance system. In: *Electronics* 7 (2018), Nr. 6, S. 84–116

[Winiwarter et al. 2022] Winiwarter, Lukas ; Pena, Alberto Manuel E. ; Weiser, Hannah ; Anders, Katharina ; Sánchez, Jorge M. ; Searle, Mark ; Höfle, Bernhard: Virtual laser scanning with HELIOS++: A novel take on ray tracing-based simulation of topographic full-waveform 3D laser scanning. In: *Remote Sensing of Environment* 269 (2022). – URL `https://www.sciencedirect.com/science/article/pii/S0034425721004922`. – ISSN 0034-4257

[Wujanz 2016] Wujanz, Daniel: *Terrestrial Laser Scanning for Geodetic Deformation Monitoring.* Bd. 775. München : DGK, Reihe C, 2016

[Wujanz et al. 2017] Wujanz, Daniel ; Burger, M. ; Mettenleiter, M. ; Neitzel, Frank: An intensity-based stochastic model for terrestrial laser scanners. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 125 (2017), S. 146–155

[Xu et al. 2025] Xu, Ziyang ; Wysocki, Olaf ; Holst, Christoph: *From Propagation to Prediction: Point-level Uncertainty Evaluation of MLS Point Clouds under Limited Ground Truth.* 2025

[Yang 2023] Yang, Yihui: *Towards Improved Targetless Registration and Deformation Analysis of TLS Point Clouds Using Patch-based Segmentation (PhD thesis)*. Bd. 929. München : DGK, 2023

[Yu 1992] Yu, Zongchou: A generalization theory of estimation of variance-covariance components. In: *Manuscripta geodaetica* 17 (1992), Nr. 5, S. 295–301

[Zámečníková & Neuner 2018] Zámečníková, Miriam ; Neuner, Hans: Methods for quantification of systematic distance deviations under incidence angle with scanning total stations. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 144 (2018), S. 268–284

[Zhang et al. 2021] Zhang, Chi ; Liu, Liwen ; Xue, Zhoupeng ; Guo, Kun ; Yang, Kuiyuan ; Cai, Rui ; Li, Zhiwei: Robust LiDAR Localization on an HD Vector Map without a Separate Localization Layer. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2021, S. 5536–5543

[Zhang et al. 2025] Zhang, Haoxin ; Li, Shuaixin ; Zhu, Xiaozhou ; Chen, Hongbo ; Yao, Wen: 3-D LiDAR and Monocular Camera Calibration: A Review. In: *IEEE Sensors Journal* 25 (2025), Nr. 7, S. 10530–10555

[Zhang et al. 2020] Zhang, Li ; Balangé, Laura ; Braun, Kathrin ; Di Bari, Roberta ; Horn, Rafael ; Hos, Deniz ; Kropp, Cordula ; Leistner, Philip ; Schwieger, Volker: Quality as Driver for Sustainable Construction—Holistic Quality Model and Assessment. In: *Sustainability* 12 (2020), Nr. 19, S. 7847. – URL `https://www.mdpi.com/2071-1050/12/19/7847`

[Zhao et al. 2024] Zhao, Haimei ; Zhang, Jing ; Chen, Zhuo ; Zhao, Shanshan ; Tao, Dacheng: UniMix: Towards Domain Adaptive and Generalizable LiDAR Semantic Segmentation in Adverse Weather. In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2024, S. 14781–14791

[Zhu et al. 2024] Zhu, Jun ; Li, Hongyi ; Zhang, Tao: Camera, LiDAR, and IMU Based Multi-Sensor Fusion SLAM: A Survey. In: *Tsinghua Science and Technology* 29 (2024), Nr. 2, S. 415–429

[Zhu et al. 2018] Zhu, Ni ; Marais, Juliette ; Bétaille, David ; Berbineau, Marion: GNSS position integrity in urban environments: A review of literature. In: *IEEE Transactions on Intelligent Transportation Systems* 19 (2018), Nr. 9, S. 2762–2778

[Zuo et al. 2019] Zuo, Xingxing ; Geneva, Patrick ; Lee, Woosik ; Liu, Yong ; Huang, Guoquan: LIC-Fusion: LiDAR-Inertial-Camera Odometry. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2019, S. 5848–5854

List of Figures

List of Tables

Acknowledgements

This thesis contains the results of my work at the Geodätisches Institut (GIH) at Leibniz Universität Hannover.

First of all, I'd like to thank Prof. Dr.-Ing. Ingo Neumann and Prof. Dr.-Ing. Hamza Alkhatib for their support during my studies and time of working on my dissertation. Especially, Hamza supported me early and encouraged me to pursue more than just the bachelor. Without their discussions and feedback this thesis wouldn't have been possible. I'd also like to thank Prof. Dr.-Ing. Steffen Schön and Prof. Dr.-Ing. Volker Schwieger for taking time out of their busy schedules to review my thesis. Furthermore, I'd like to thank all other PIs and colleagues from i.c.sens (RTG 2159) for the collaborative work and discussions. I am very thankful for the opportunity to focus on my thesis in the program and want to thank the DFG for the financial support.

I'd also like to thank the (former) colleagues at GIH for their support, starting with Johannes Bureick and Sören Vogel as the supervisors of my Bachelor and Master thesis, respectively. During my time at GIH, Sören, Alexander Dorndorf, Jens Hartmann, Frederic Hake and all the others were helpful and created a welcoming atmosphere.

Als Letztes möchte ich mich bei meiner Familie bedanken, die mich über all die Jahre von der Schule über die Ausbildung, das Studium und nun die Promotion begleitet und unterstützt hat.

Curriculum Vitae

Personal Information

Name:	Dominik Ernst
Date of Birth:	10.02.1995
Place of Birth:	Hannover

Course of Education

08/2005 - 05/2013	High School
08/2013 - 06/2016	Vocational training: Surveying Technician
10/2016 - 08/2021	Study of Geodesy and Geoinformatics at Leibniz Universität Hannover: Bachelor of Science (2019), Master of Science (2021)

Work Experience

06/2016 - 09/2016	Surveying Technician, RMK Celle
10/2016 - 12/2019	Work-study student, RMK Celle
11/2018 - 08/2021	Research Assistant, Geodätisches Institut, Leibniz Universität Hannover
09/2021 - 11/2025	Research Associate, Geodätisches Institut, Leibniz Universität Hannover
since 12/2025	Working Group Leader, Geodätisches Institut, Leibniz Universität Hannover