

Doctoral thesis

Doctoral theses at NTNU, 2024:259

Vasileios Bosdelekidis

Integrity Monitoring for Visual and Multi-Sensor Navigation Systems

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of
Philosophiae Doctor
Faculty of Information Technology and Electrical
Engineering
Department of Engineering Cybernetics



Norwegian University of
Science and Technology

Vasileios Bosdelekidis

Integrity Monitoring for Visual and Multi-Sensor Navigation Systems

Thesis for the Degree of Philosophiae Doctor

Trondheim, June 2024

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics



Norwegian University of
Science and Technology

NTNU

Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

© Vasileios Bosdelekidis

ISBN 978-82-326-8110-5 (printed ver.)
ISBN 978-82-326-8109-9 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

ITK-nr.: 2024-04-W

Doctoral theses at NTNU, 2024:259

Printed by NTNU Grafisk senter

Summary

Abstract

Autonomous vehicle navigation in safety-critical operations is a complex task. In this domain, integrations of auxiliary sensors, such as cameras, LiDAR, or radar, with conventional sensors, such as Global Navigation Satellite System (GNSS), and Inertial Measurement Units (IMUs), has spurred intense research focus. The reason for the popularity is the enhanced accuracy and reliability of these systems, even in the face of some sensor failures. This thesis presents a methodology to address the challenges of an Integrity Monitor (IM) for these systems and of the identification of hazardous situations. Some of the research questions that guide the development of the IM framework are how to accommodate the IM for complex integrations with auxiliary sensors and, potentially in the presence of multiple simultaneous faults, how to mitigate non-Gaussian noise distributions, which are prevalent in challenging environments, and how to cope with completely unknown noise models of some sensors. Motivated by the need for scalable methods independent of specific navigation algorithms or sensors, this thesis develops an innovative IM architecture. It is evaluated specifically for a Visual Navigation (VN)/GNSS/IMU integration, where a camera serves as the auxiliary sensor. Despite the development of IM modules specific to the camera and the conventional sensors, the architecture is designed to scale, allowing for the incorporation of additional auxiliary sensors.

The developed IM architecture facilitates fault detection and integrity risk assessment by leveraging deep learning approaches and operating with raw measurements from the camera sensor. The corresponding camera-specific IM modules function independently of the presence of any other auxiliary sensors in the monitored integration. Generally, this incorporation of modules tailored to each auxiliary sensor refines Protection Level (PL) computation methodologies. Effectively, it limits the hypothesis space in fault-tolerant navigation systems and the need of uncertainty estimates from the monitored navigation algorithm or noise model knowledge of the auxiliary sensors.

Also, the aforementioned scalability of the architecture to various auxiliary sensors is achievable by developing corresponding Fault Detection and Exclusion (FDE) and PL inflation modules tailored to account for potential faults in each of those sensors.

Through experimentation on datasets collected during real-world car drives in urban environments and simulation of sensor faults or non-Gaussian noise models, the IM shows robust performance, comparable to methods where noise of all sensors is known. Moreover, the architecture facilitates the identification of challenging environmental conditions specific to individual sensors, enabling the recommendation of testing under similar conditions during the navigation algorithm development phase. This advancement, represents a significant step forward in the quest for a universal IM for navigation methods and the development of navigation systems that adhere to standardized safety requirements.

Contents

- Summary** **iii**
- Abstract iii

- Contents** **ix**

- Preface** **xi**

- 1 Introduction** **1**
- 1.1 Integrity monitoring of multi-sensor navigation systems: Frameworks and challenges 1
- 1.2 Background on integrity monitoring in the navigation of autonomous vehicles 3
- 1.2.1 On MHSS and the benefits of MHSS vs Residual Based (RB) RAIM 4
- 1.2.2 On multi-sensor IM 4
- 1.2.3 On sequential estimation navigation integrity 5
- 1.2.4 On fault detection and threshold optimization 5
- 1.2.5 On integrity monitoring of smoothing-based navigation 6
- 1.3 Implications in auxiliary sensor navigation 6
- 1.4 Research questions 8
- 1.5 Publications and Contributions 9

1.6	Outline	11
2	Navigation algorithm uncertainty and different sensor systems	15
2.1	Introduction	15
2.2	Sensor noise	15
2.3	Covariance estimation by different algorithms	17
2.3.1	Covariance estimation for map-based vision-aided navigation: A review	19
2.4	Covariance inflation for atypical sensor noise and for sensor faults - Loose and Tight Integration	20
2.5	Bank of Kalman filters, Model-based Fault Detection for Multi-sensor Data Systems	23
2.6	Chapter Summary	25
3	Navigation algorithm-agnostic IM for multiple sensor noise models	29
3.1	Introduction	29
3.2	Allocation of integrity risk	31
3.3	PL computation	32
3.4	Overbounding Time-Correlated Noise	33
3.5	LLR and LLRSD for Fault Detection	35
3.6	Experiments	36
3.6.1	Data	36
3.6.2	Experimental Setup and Parameters	37
3.6.3	Navigation Algorithms Evaluated with the IM	40
3.6.4	Results and Discussion	42
3.6.5	Quantitative Evaluation of the IM Method	48
3.7	Chapter Summary	50
4	An IM designed for scalability to auxiliary sensors	51
4.1	Introduction	51

4.2	Motivation for a sensor-tailored approach to IM of multi-sensor integrations	52
4.3	Hypothesis tree	54
4.4	Conventional and visual measurement integration	55
4.4.1	Notation and definitions	56
4.4.2	IMU measurement model in multi-sensor integrations	57
4.4.3	MHSS for camera frames	58
4.4.4	ErKF in integrations of VN with conventional sensors	59
4.5	Integrity Solution for integrations of conventional and auxiliary sensors	61
4.5.1	PLs for IMU pre-integrated measurements	61
4.5.2	PLs for VN faults	64
4.5.3	Overview of the complete IM	66
4.6	Datasets	66
4.6.1	UrbanLoco Dataset	66
4.6.2	Robotcar Dataset	67
4.6.3	KAIST Dataset	67
4.6.4	4seasons Dataset	68
4.7	Chapter Summary	68
5	Integration of conventional sensors with a camera and faults: A DNN approach for navigation anomaly detection	69
5.1	Introduction	69
5.2	DNN modeling for anomaly prediction in visual-based navigation	72
5.2.1	Isolation of anomalies in VN	72
5.2.2	Deep CNN	74
5.2.3	Output bias initialization	74
5.3	Experiments	75
5.3.1	Training data and appropriability for the problem	75
5.3.2	Anomaly labeling result	78

5.3.3	Training and evaluation of the network	79
5.4	Chapter Summary	82
6	Deep Hypothesis Testing for fault alarms in auxiliary sensors	83
6.1	Introduction	83
6.1.1	Prior work	84
6.1.2	Contribution	84
6.1.3	Chapter organization	85
6.2	Detection of anomalies in the uncertainty of visual SLAM and in the IM test statistic	85
6.2.1	Architecture for auxiliary sensor FDE	88
6.3	Anomaly detection in DHT: Grouped image features	89
6.4	Camera fault hypotheses definition	90
6.5	Ordinary distribution	91
6.6	Test for anomalies	93
6.7	Chapter Summary	94
7	Experimental evaluation	95
7.1	Introduction	95
7.2	Anomaly detection and camera exclusion results	95
7.2.1	Trajectories and system setup	95
7.2.2	Anomaly detection results	97
7.2.3	Numerical evaluation of the IM performance	99
7.3	PL inflation methodology for IMU preintegration and VN faults	105
7.3.1	Fault modes and residual computation	105
7.3.2	Datasets	105
7.3.3	The scenario of car navigation in urban environment	106
7.3.4	Simulation of faults	110
7.3.5	Discussion of identified VN faults	117

7.4 Chapter Summary	120
8 Conclusion	123
8.1 Future Directions	124
8.2 Final remarks	125
A Derivation of the positioning uncertainty in SLAM	127
A.1 1D case	127
A.2 2D case	128
B ErKF and Gauss-Markov Process Noise Overbounding	131
C LLR for faulty sensor detection	135
References	137

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of philosophiae doctor (PhD) at the Norwegian University of Science and Technology (NTNU), Trondheim. The work has been conducted at the Department of Engineering Cybernetics (ITK). Professor Tor Arne Johansen has been the main supervisor of this work with the co-supervision of Associate Professor Torleiv Håland Bryne and Associate Professor Nadezda Sokolova. Tor Arne and Torleiv are at the ITK, NTNU, and Nadezda is at SINTEF and ITK, NTNU. The work has been funded through the Research Council of Norway and the Centre for Autonomous Marine Operations and Systems (AMOS).

Acknowledgements

I would like to thank the people that have made this thesis possible. In addition, I acknowledge all the people that have provided valuable insights.

My main supervisor, Tor Arne, has given me the opportunity to pursue a Ph.D. Thank you, Tor Arne, for guiding me to specify the goals and reach useful results, and for providing thorough and so quick feedback whenever I needed it. Thank you, Nadia and Torleiv, for being my co-supervisors, offering valuable feedback, and always being

available to discuss ideas. Nadia, I appreciate you running the project and including me in the team discussions.

A big thank you Khadija Shaheen for working with me on the project during the Non-Linear Estimation course where we turned in a very concise literature review on "Model-based Fault Detection for Multi-sensor Data Fusion," a significant part of which is included as a section in this thesis.

It has been a long journey of looking at various datasets for evaluation in this thesis, including field data from other NTNU projects. I thank you Hakon Hagen Helgesen, Martin Eek Gerhardsen, and Oystein Kaarstad Helgesen for your time providing me with marine vehicle data and helping me start work with them, even though I fell short of time to fully explore navigation and IM challenges and complete experiments on this data. It would be of great value to see results on marine data at one moment in the future.

A heartfelt thank you to my good friend, Nikolaos S. Ioakeimidis, MD, for the assistance with language recommendations and proofreading parts of the dissertation.

To my office mates over the years, Behdad, Tarek, Esmee, Trym Arve and Waseem thank you for caring about my research and being very social, even though we stayed in same office for short period unfortunately. Thank you cordially to many of the colleagues in ITK for always trying to include me in your discussions and inviting me to activities after work. You make it a great working environment in ITK.

The greatest thank you to my family. My dad and mum, Neofotios and Elissavet, thank you for supporting me to reach my goals.

List of Abbreviations

AL	Alert Limit
AUPRC	Area Under Precision-Recall Curve
CI	Covariance Intersection
CNN	Convolutional Neural Network
CSS	Conventional Sensor Set
DHT	Deep Hypothesis Testing
DM-VIO	Delayed Marginalization Visual-Inertial Odometry
DNN	Deep Neural Network
EKF	Extended Kalman Filter
ErKF	Error State Kalman Filter
FDE	Fault Detection and Exclusion
FF	Fault Free
GMP	Gauss-Markov Process
GNSS	Global Navigation Satellite System
GTSAM	Georgia Tech Smoothing and Mapping
GT	Ground Truth
HMI	Hazardous Misleading Information
IMU	Inertial Measurement Unit

IM	Integrity Monitor
INS	Inertial Navigation System
KF	Kalman Filter
LLRSD	Rolling standard deviation of Log Likelihood
LLR	Log Likelihood Ratio
LSTM	Long Short Term Memory Network
MHSS	Multiple Hypothesis Solution Separation
NED	North-East-Down
PDF	Probability Density Function
PE	Position Error
PL	Protection Level
RAIM	Receiver Autonomous Integrity Monitoring
RBT	Relaxed Bound Tightness
RB	Residual Based
RNN	Recurrent Neural Network
SAM	Smoothing and Mapping
SLAM	Simultaneous Localization and Mapping
TDL	Time Distributed Layer
UAV	Unmanned Aerial Vehicle
VIO	Visual-Inertial Odometry
VN	Visual Navigation
VO	Visual Odometry

Chapter 1

Introduction

Autonomous vehicle navigation in safety-critical operations has become a focal point in research, particularly in the integration of various sensors such as cameras, GNSS, and IMUs. The motivation for integrating these sensors arises from the need to exploit their complementary capabilities, ensuring the system's availability even in conditions where one or more sensors might fail. This is particularly crucial in safety-critical operations, where the failure of navigation sensors could have severe consequences. The development of frameworks for monitoring these systems and identifying hazardous situations has, however, been limited, often tied to specific navigation algorithms or sensors. For this reason, the present thesis addresses this topic.

This chapter presents further background and motivation of the thesis, highlighting the identified research questions. It also gives a brief overview of how the thesis is organized and what methods are used.

1.1 Integrity monitoring of multi-sensor navigation systems: Frameworks and challenges

The challenges in automotive navigation are manifold. Tall structures can impact GNSS signals, leading to total denial or multipath issues. Feature-less environments or light reflections can negatively affect feature extraction and association in camera images. Meanwhile, inertial sensors will accumulate drift over time. While the fusion of complementary sensors has shown factual improvements in navigation accuracy [1], previous research has not investigated thoroughly the conditions of nominal behavior of those sensors.

Integrity Risk of a positioning system refers to the probability that an unavoidable state estimation error will cause the system's estimate to deviate from the true value by more than a predefined PL, potentially triggering an alert. The Stanford Diagram shown in Figure 1.1 helps in distinguishing the times that the system is available or unavailable as

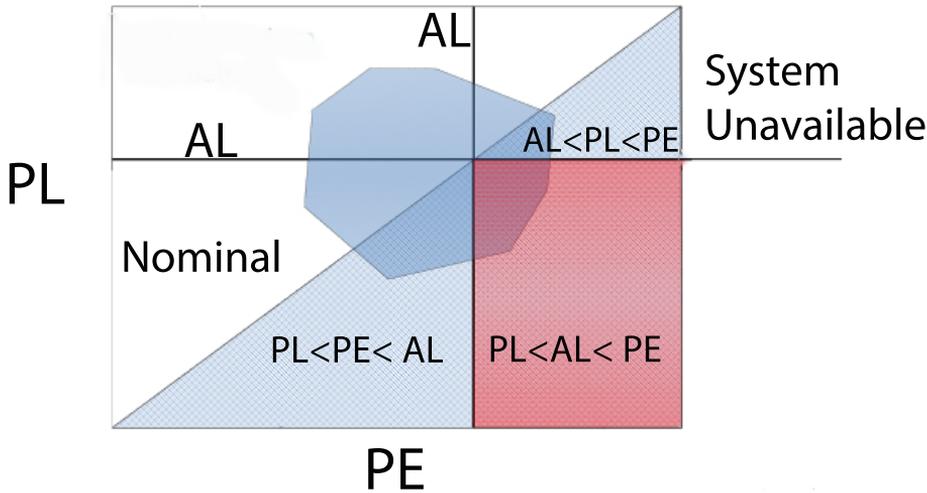


Figure 1.1: Stanford Diagram depicting a possible distribution of the points of a system (represented with the colored octagon), the system's availability regions and the regions related to integrity events. Modified version from [2].

well as two types of integrity events. The acronym PE refers to the true Position Error of the given navigation system. The upper area in the diagram, where the PL overcomes the Alert Limit (AL), is where the system becomes unavailable. Regarding the events of interest, one category pertains to instances of misleading information, occurring when $PL < PE < AL$ and $AL < PL < PE$. Still, safety is not compromised. Conversely, the Hazardous Misleading Information (HMI) event, denoted by the region highlighted in red, involves scenarios where the PE exceeds the AL without triggering a PL alert. This event holds particular significance and can compromise safety. The indicated region of nominal system operation has $PE < PL < AL$. It is worth noting that depending on the navigation problem at hand, the vehicle travel directions differ, with PL computed in each direction. For instance, in automotive navigation, the two directions of interest are usually along the direction of motion and the cross-track direction. However, the current thesis adopts the North-East-Down (NED) frame convention. One examined IM problem concerns Unmanned Aerial Vehicle (UAV) navigation, while the majority of examined problems relate to car navigation. In the latter case, the directions of interest are solely the north and east.

A report from the European GNSS Agency [3] emphasizes that a well-established and trustable framework to set integrity risks and compute PLs is mandatory for any safety critical application that uses position estimates as input. Hence, the presence of an integrity monitor stands as a fundamental requirement for the success of safety-critical operations. This necessity is emphasized in various studies such as [4], [5], and [6]. Integrity monitors play a crucial role in ensuring safe autonomous system operation.

Their implementation enables the quantification of system reliability, thus alleviating the burden of trust placed on humans and facilitating better compliance with local regulations. At the same time they can assist in understanding better the trade-off among safety and system availability [5], [6] and eventually motivate the development of more efficient navigation solutions. However, traditional methods for evaluating integrity, such as Receiver Autonomous Integrity Monitoring (RAIM) and Multiple Hypothesis Solution Separation (MHSS), have limitations, especially when dealing with complex multi-sensor navigation systems. These methods, often developed for specific sensor faults or sensor combinations, struggle to handle the complexities of sensor fusion, correlated errors, inherent non-linearities, and the diverse nature of noise and faults in different sensors and environments. The literature underscores the challenges in guaranteeing safety when employing auxiliary sensors [7, 8].

The specific integration IMU/GNSS/camera is susceptible to numerous potential sensor faults [9]. The probabilities of failure of all the individual sensors within the given integration are substantial and might exceed the acceptable integrity risk in safety-critical operations. A precise quantification, found in existing literature, is discussed later in the thesis.

Current methods aimed at evaluating the integrity of integrations involving auxiliary sensors lack comprehensive categorization and assessment of the extended range of potential sensor faults. Moreover, potential faults in IMU measurements are overlooked in the vast majority of IMs.

Addressing the aforementioned limitations is essential for the continued development and deployment of autonomous systems in safety-critical scenarios.

1.2 Background on integrity monitoring in the navigation of autonomous vehicles

Frameworks for monitoring navigation systems in autonomous vehicles and identifying hazardous situations have been constrained by their reliance on specific navigation algorithms or sensors, as highlighted by [10].

Typical methods for evaluating integrity include the RAIM technique [11], which is primarily designed for GNSS, and MHSS [12, 13], which considers multiple possible scenarios or hypotheses regarding the source of errors or faults. MHSS has been promoted several times as a promising framework for sensor-agnostic safety monitoring (e.g. [14], [15]), as far as the error covariances due to each sensor can be estimated. However, these covariances are often not available. In addition, traditional PL computation formulas developed for GNSS/IMU systems might not be reliable. These traditional formulas typically assume standard noise characteristics and known fault models, which may not hold true in more complex integrations with auxiliary sensors.

1.2.1 On MHSS and the benefits of MHSS vs Residual Based (RB) RAIM

Joerger et al. [16] designed a fault detector and a non-least-squares estimator in RAIM. In the first part of their work they focused on multi-sensor faults and that is why they derived analytical expressions of the worst-case fault magnitude and direction, along with an expression for the PL. They proved the superiority of MHSS in comparison to RB in terms of the detection statistic, something that is justified by the fact that MHSS detection is directly related to the faulty measurement hypotheses. In addition, the non-least-squares estimator, that they developed in the second part of their work, achieved the optimization of the integrity risk, at the cost of lower accuracy. A formulation of both the integrity and continuity risk was made in the subsequent work of Joerger et al. [5], and from there they defined simple formulas for both the single and multiple measurement fault case. Their theoretical proofs are very important in the IM domain and their error bounding formulas are used also in the current thesis. In the MHSS approach employed here, the corresponding formulations are derived in the positioning domain. This makes MHSS more appropriate for a universal IM as it facilitates scalability to multiple sensors and independence from the navigation algorithm.

1.2.2 On multi-sensor IM

A large number of available navigation platforms and algorithms are multi-sensor based. However, multiple modalities of the various measurements, as well as the necessity for recursive estimation place many challenges to the classical integrity monitoring methods. Multi-sensor IM has gained significant interest but, to the best of my knowledge, proposed solutions in literature maintain some dependence on a specific navigation algorithm.

Gupta and Gao [17] introduced the Particle RAIM framework. At each time the position is weighted by incorporating its likelihood of generating the subset of measurements. In a subsequent work, Mohanty et al. [18] explored a GNSS and camera integration with the Particle RAIM framework. Due to the use of the visual sensor, they accounted also for ambiguities during feature and image association, and for errors originating from sensor fusion. The Particle RAIM framework leverages collaborative estimation and integrity monitoring, and uses the full probability distribution over position, instead of position point estimates. This means that this framework captures very well the uncertainty in the state estimation algorithm.

Meng and Hsu [15] treated the system state propagation as an additional measurement within an Extended Kalman Filter (EKF) navigation algorithm. In recursive methods such as the EKF, integrity risk can originate from the propagated state, which is influenced by past epochs. Their approach allowed for the inclusion of the EKF's temporal connectivity in integrity risk calculations, something not feasible with traditional snapshot-based methods. However, this method is not applicable to navigation systems that do not use Kalman Filter (KF)-based approaches. Otherwise, for scalability

and due to the different nature of measurements, at each hypothesis a whole sensor is considered as faulty and their method works directly on output position solutions. A method to maintain navigation integrity despite the multiple-sensor measurements was developed also by Appleget et al. [14]. They emphasized that traditional MHSS is not that effective in handling sensor faults that result in insignificant differences in state estimation between the uncorrupted and the fault-free filters. For this reason, their algorithm employs the expected value of the covariance of the measurement residuals in a moving average χ^2 -test. Multiple filters are used, and residual covariance distances are calculated for each sensor/ subfilter pair. The residual measurement is based on the predicted states from the EKF.

Bhamidipati and Gao [19] utilized a linearized Graph-Simultaneous Localization and Mapping (Graph-SLAM) framework for worst-case failure mode slope analysis and, subsequently, integrity monitoring when there are multiple faults in both GNSS and vision. The visual observations are used to distinguish Line Of Sight and Non Line Of Sight satellites by recognizing the sky, formulating the GNSS measurement covariances. The state vector updates are done before performing FDE, which depends on the comparison with an empirical distribution of measurement residuals. Their method is capable to monitor multiple faults in both GNSS and vision. Nevertheless, their method was developed for specific sensors, applies measurement preprocessing, and performs estimation and IM collaboratively. Thus, it is not directly extendable to other sensor types or navigation algorithms.

1.2.3 On sequential estimation navigation integrity

Recursive integrity monitoring has been a topic of interest for many researchers, beyond those who investigated multi-sensor fusion. For example, Arana et al. [20] presented a method that can monitor multiple landmark association faults at different times. They avoid assumptions in the nature of faults, as integrity is evaluated under the worst combination of sensor faults. However, this is not a navigation algorithm-independent method as it is based on the KF innovation vectors sequence. Nonetheless, they accounted for the fact that the estimation error and the fault detection are affected by faults back in time, and this is an important consideration also in the method in this thesis.

1.2.4 On fault detection and threshold optimization

FDE is an important module in the IM process which improves the navigation solution reliability. In the presence of redundant measurements, it is typical to identify measurement faults by comparing positioning solutions when using the full measurement set and different measurement subsets. Faults that can be isolated will be excluded and will not be considered in the integrity risk evaluation.

He et al. [21] highlighted that a detection statistic should account for both optimal availability and integrity, and it should be independent of underlying statistical assumptions

for the data distribution. Taking this into consideration, they tested a nonlinear optimization algorithm for the fault detection bound, using the cumulative Log Likelihood Ratio (LLR) statistic. LLR and most other Log-Likelihood based statistics do not rely on the internal procedure of a navigation method, but only on the measurement set, the position state estimations and the uncertainties.

In addition to FDE methods operating in the position domain, there is another popular class of methods operating directly on sensor measurements [22]. More generally, data-driven FDE methodologies, have been developed to cope with the circumstance of insufficient knowledge about complex systems and unknown fault types. These methodologies solve FDE problems using multivariate statistical methods and machine learning models trained from historical input/output data. They can be more sensitive to soft or slowly growing faults than the other FDE methods [23].

1.2.5 On integrity monitoring of smoothing-based navigation

IM of visual localization algorithms is quite challenging as there is a vast amount of measurements (e.g. landmarks and features), and it is common that more than one measurement are faulty at each time [7]. Visual localization is a typical example of applications that are better solved with optimization instead of filter-based methods due their flexibility in handling complex relationships of diverse measurements and capacity for global optimization. It is therefore essential to test IM methods with an optimization-based navigation approach too. In the existing literature, an IM method targeted to a nonlinear pose optimization problem was developed by Li and Waslander [7]. They emphasized that their method cannot guarantee that a proposed bound will always be valid, as is true for any nonlinear system with outliers. They employed a variation of the Parity Space Approach as statistical tests to remove multiple outliers for a batch of measurements, before the PL calculation step. An assumption in this test is that the noise of the measurement model follows a Gaussian distribution. This reliance on Gaussian distribution highlights a notable gap in existing literature. A central contribution of the presented method in this thesis is challenging this assumption by accommodating sensor measurement noise that may not adhere to Gaussian distribution. Experiments validate the approach within an IM, specifically in scenarios where at least one sensor exhibits time-correlated noise patterns.

1.3 Implications in auxiliary sensor navigation

The integration of auxiliary sensors in navigation systems presents several implications for navigation integrity, as discussed in recent literature. These implications are pertinent especially in relatively emerging domains, e.g. autonomous car navigation, where stringent requirements are set. However, current multi-sensor fusion techniques lack measurement models with sufficient confidence, making difficult the computation of integrity bounds down to the same low integrity risks that have been successfully achieved in the more well-studied field of aviation. [24].

Figure 1.2 lists potential fault sources related to the individual sensors in the integration of IMU, GNSS, and camera, or in the fusion process. It is crucial to recognize that some faults are deterministic and some non-deterministic.

Deterministic faults are those that can be predicted or occur in a consistent and repeatable manner. These faults are generally due to systematic errors that do not vary randomly over time. Examples include calibration faults, deterministic scale factor errors, and sensor misalignment. Such faults must be eliminated as the IM does not accommodate them.

Non-deterministic faults, on the other hand, are stochastic in nature and involve random variations that are not easily predictable. These faults arise due to random noise, environmental changes, or other unpredictable factors. Examples include multipath effects in GNSS, stochastic faults in IMU, and feature extraction faults in cameras. The IM procedure is designed to monitor and mitigate these types of faults.

It is important to note that not all faults can be clearly categorized as purely deterministic or non-deterministic. Some faults may exhibit characteristics of both, depending on conditions. For example, ionospheric delays in GNSS can be deterministic when modeled using known conditions, but non-deterministic when influenced by unpredictable space weather. Similarly, feature extraction faults in cameras can be deterministic in controlled environments, but non-deterministic in varying lighting conditions. Despite the importance of understanding this distinction and its relevance to IM, the specific categorization of faults is not paramount in the context of this thesis. The current work assumes the absence of common deterministic faults in the sensor measurement sets or integration systems utilized in the experiments.

One significant implication is the increase in potential sources of faults introduced by auxiliary sensors. The IM process may be even more complicated if it is necessary to account for faults originating from multiple sensors simultaneously. Moreover, there is possibility of correlated faults from different sensors, which is a situation that challenges traditional FDE techniques [25].

Another key consideration is that the error distribution might be non-Gaussian. While the presence of auxiliary sensors can contribute to non-Gaussianity, conventional sensors like GNSS can also exhibit non-Gaussian behavior due to environmental challenges encountered during navigation in urban environments. This deviation from Gaussianity poses challenges for traditional navigation error modeling approaches, such as Kalman Filters (e.g. KF and EKF), requiring novel methods to accurately characterize and mitigate error sources. Additionally, achieving tight integrity guarantees comparable to those in aviation becomes challenging due to uncertainties in measurement models and reliability [25].

Furthermore, challenges might arise due to the reliance of an IM system on a variation of a KF algorithm, relying on the estimation of uncertainties—specifically, the variances

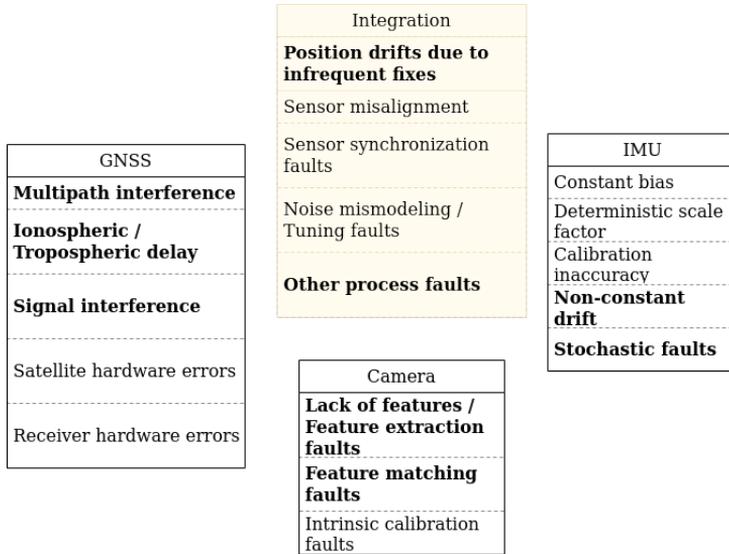


Figure 1.2: Examples of potential sensor and process fault sources in the integration of IMU, GNSS, and Camera. Sources that most commonly cause non-deterministic faults are shown in bold.

associated with the state estimates—to determine PLs. This is a very typical IM approach. However, the suboptimal uncertainty estimation raises concerns. In practice, uncertainty estimates may be inconsistent, leading to inaccuracies in assessing navigation integrity. Moreover, fault detection becomes more intricate, especially in scenarios involving multiple sensors, as traditional methods often assume one fault at a time. Another source of faults is linear approximation [25].

1.4 Research questions

Addressing the challenges mentioned in the previous section requires advancements in IM techniques capable of accommodating the complexities introduced by auxiliary sensors while ensuring accurate and reliable navigation integrity across diverse real-world scenarios. This necessitates a nuanced understanding of error sources, including those from non-GNSS measurements, to achieve accurate integrity assessments. Additionally, new methods must be developed to address situations where multiple faults occur simultaneously, and to account for non-Gaussian error distributions prevalent in urban environments and other challenging scenarios.

The motivation for this research, therefore, stems from the need to develop a method capable of accounting for faulty sensor measurements without exhaustive examination of potential fault hypotheses. This method should identify sensor faults independently, reducing reliance on specific navigation algorithm prerequisites and ensuring

scalability across various auxiliary sensors.

1.5 Publications and Contributions

The main contributions of the thesis are summed up below:

- Introduction of an IM architecture agnostic to navigation algorithm internals.
- Versatility to various sensors within the IM architecture.
- Development of a Deep Neural Network (DNN)-based method for early anomaly detection in visual-only SLAM systems, facilitating potential integration for camera FDE in existing IMs.
- Enabling the utilization of traditional IMs without modification, and the direct correlation of IM anomalies with camera image inputs, independently if other auxiliary sensors are present. Analogous methods can be developed for independent evaluation of other auxiliary sensors.
- Introduction of a PL computation approach for navigation systems integrating auxiliary with conventional sensors, isolating PL computation for auxiliary sensors and enabling tailored safety assurances.
- The developed anomaly detection methodology, which relies directly on raw camera inputs, demonstrates the added benefit of detecting hazardous environmental conditions that cause these anomalies. An indirect contribution of this thesis is, therefore, its potential to facilitate comprehensive testing during the development of navigation or IM systems under challenging conditions. However, it is worth noting that the detection process is based solely on statistical analysis. This poses certain risks, such as sensitivity to statistical model assumptions and limited context awareness of the operation environment and underlying causes of anomalies.

The thesis is based on the following list of papers published in peer-review international journals and conference proceedings:

- Paper 1: [26] V. Bosdelekidis, T.H. Bryne, N. Sokolova and T.A. Johansen, "Navigation Algorithm-Agnostic Integrity Monitoring based on Solution Separation with Constrained Computation Time and Sensor Noise Overbounding," *Journal of Intelligent & Robotic Systems* 106, 7 (2022).

Contribution: The paper introduces a method of IM for the integrated navigation of GNSS and IMU, laying the initial groundwork for the developed IM architecture in the thesis for multi-sensor integrations. The method is designed to be

agnostic to navigation algorithm internals and scalable to various sensors and noise models. The paper proposes an MHSS IM algorithm that utilizes statistics on Log Likelihoods to detect measurement faults. Also, it proposes overbounding techniques to address underestimated uncertainties due to non-standard sensor noise models, with good empirical results in the case of time-correlated noise. Additionally, the architecture permits hybrid-MHSS, where some hypotheses are evaluated with a navigation algorithm with known characteristics (referred to as the standardized algorithm), while the all-sensor-in solution is evaluated with the actual algorithm under monitoring (referred to as the monitored algorithm). In this case an Error State Kalman Filter (ErKF) is used as standardized algorithm and an implementation of Georgia Tech Smoothing and Mapping (GTSAM) as monitored.

- Paper 2: [27] V. Bosdelekidis, T. A. Johansen and N. Sokolova, "DNN-based anomaly prediction for the uncertainty in visual SLAM," 2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, Singapore, 2022.

Contribution: The paper presents a method that offers potential benefits to IM of navigation systems that include visual sensors. Specifically, it investigates early anomaly detection in position estimates of a visual-only SLAM system by leveraging a DNN. The DNN can predict statistically anomalous estimate errors where the classification is done based only on features of sequential raw image frames. This capability could facilitate testing of visual-based - in general - navigation systems under identifiable challenging conditions, potentially integrating the system for camera FDE in existing IMs. .

- Paper 3: [28] V. Bosdelekidis, T. A. Johansen, N. Sokolova and T. H. Bryne, "Solution Separation-based Integrity Monitor for Integrated GNSS/IMU/Camera Navigation: Constraining the Hypothesis Space With Deep Learning," 2023 IEEE/ION Position, Location and Navigation Symposium (PLANS), Monterey, CA, USA, 2023.

Contribution: This paper uses the previously introduced visual navigation anomaly detection module and investigates refinement based on an IM test statistic specific to the camera sensor. One contribution lies in the enhanced camera FDE running in parallel to traditional IMs without modification. It becomes sufficient to compute PLs considering only conventional sensor (e.g., IMU and GNSS) noise and faults, even in navigation systems that incorporate a camera. This streamlined process is based on the assumption that the inclusion of a camera inherently reduces uncertainty, where this uncertainty is consistent in the absence of faults. Another significant contribution of the paper is that with the camera FDE there is the capability to directly correlate IM anomalies with camera inputs. This methodology of independent evaluation can be extended to auxiliary sensors other than the camera, by developing FDE modules tailored to the other sensors.

- Paper 4: V. Bosdelekidis, T. A. Johansen and N. Sokolova, "Sensor-Tailored Integrity Monitoring for Multi-sensor Navigation: GNSS/IMU/Vision Factor Graph Example," 2024 IEEE Transactions on Robotics (T-RO)(submitted).

Contribution: The paper introduces a PL computation approach for navigation systems that integrate auxiliary with conventional sensors. The basic idea is to isolate the PL computation for the auxiliary sensors, enabling tailored safety assurances specific to each sensor. The PLs for the full system are computed using an inflation process based on the individual PL factors. The investigation in the paper is specifically for the example IMU/GNSS/Visual Navigation (VN) integration with factor graphs. Therefore, in this paper we provide evaluation results only for the camera as auxiliary sensor. Importantly, the method does not impose assumptions about the internal architecture of the monitored navigation algorithm, as it uses a dedicated risk assessment module for the camera and a standard VN algorithm with known characteristics to compute the corresponding PL. Additionally, the paper presents a method for computing the PL inflation factor for faults in preintegrated IMU measurements, commonly used in Visual-Inertial Odometry (VIO) navigation systems.

In summary, the papers introduce methodologies for FDE and PL computation within an IMU/GNSS/VN system, supported by extensive experimental validation. The IM architecture developed is readily adaptable to accommodate additional auxiliary sensors. Moreover, the papers outline the methodology for computing PL inflation terms, which adjust PLs to compensate for potential faults in noise modeling, i.e. inaccuracies in sensor noise characterization, unmodeled time-correlated noise, or environmental factors affecting sensor performance. Additionally, they propose a method for identifying hazardous conditions specific to a camera.

While the developed IM architecture showed very good performance in the experimental evaluation, it should be regarded with humility as a work in progress rather than a final solution in the domain of universal IM. Recognizing its current limitations, particularly in terms of theoretical analysis in some parts, emphasizes the need for further refinement to ensure consistent safety guarantees across all rare hazardous conditions encountered in real-world scenarios. This acknowledgment underscores the ongoing effort to enhance navigation system reliability and safety, where the proposed IM architecture could be regarded as a very good base for future advancements in universal IM.

1.6 Outline

The thesis is organized into seven main chapters, each building upon the research questions and publications discussed earlier. The formulation of a universal IM framework goes through exploration of related topics on variety of sensor noise, overconfidence in covariance estimation, sensor fault sources and variety of multi-sensor integrations.

It is, therefore, essential to explore these aspects early on in the thesis, on a theoretical level. Then, each chapter systematically addresses the development of individual modules, progressively tackling the challenges outlined earlier. The thesis introduces relatively early-on the complete IM architecture for systems employing multiple auxiliary sensors, providing a high-level overview of its constituent modules. It subsequently delves into the specifics of IMU/GNSS/Vision integration with Factor Graphs, presenting methodologies corresponding to each module in the IM architecture. The developed IM approach remains, however, scalable to other sensor integrations and independent from the monitored navigation algorithm's internals. A final chapter is dedicated in presenting overall results of applying the full IM in a real-world car driving scenario. Specifically, the chapters are the following:

- **Chapter 2: Navigation algorithm uncertainty and different sensor systems**

Chapter summary: This chapter explores navigation algorithm uncertainty and various sensor systems, providing insights into their theoretical backgrounds and specific methodologies for uncertainty calculation and sensor effect analysis. It is an important introduction to the development of an IM architecture tailored for multi-sensor navigation systems, emphasizing the implications to consider.

- **Chapter 3: Navigation algorithm-agnostic IM for multiple sensor noise models.**

Publication: Paper 1.

Chapter summary: This chapter focuses on position domain operations with a MHSS-based IM for a navigation system integrating an IMU alongside two position-fixing sensors. One of the sensors is disturbed with time correlated noise, and the chapter investigates a method to compute conservative PLs for this case. Additionally, the hybrid IM approach of Paper 1 is explained, employing a standard ErKF to assess fault-present hypotheses, independently of the type of the monitored algorithm. This approach aims to tackle the challenge posed by monitored algorithms that either do not provide outputs in the desired format (i.e., position along with uncertainty estimates) or introduce time inefficiencies during their fault-tolerant reexecutions in MHSS.

- **Chapter 4: An IM designed for scalability to auxiliary sensors**

Publication: Papers 2, 3 & 4.

Chapter summary: The chapter discusses the substantial failure probabilities of individual sensors in integrated systems comprising conventional sensors, like

IMUs and GNSS, alongside auxiliary sensors, like cameras, emphasizing the importance of fault detection and integrity risk assessment. It investigates fault modes specific to IMU preintegration and VN, deriving PL inflation factors associated with potential faults. In the proposed architecture, these factors are derived independently from the monitored algorithm utilizing simplified integrations and, then, they are corrected to correspond to conservative bounds for the monitored state estimate errors. While the chapter outlines the overall IM architecture, it defers the discussion of camera FDE to subsequent chapters.

- **Chapter 5: Integration of conventional sensors with a camera and faults: A DNN approach for navigation anomaly detection**

Publication: Paper 2.

Chapter summary: The chapter proposes a DNN approach designed to predict anomalies in position estimate errors produced by a visual-based SLAM algorithm, using data from a car driving in an urban environment. The DNN is trained offline using sets of sequential image frames as input. It can therefore assist in the detection of low-level image features or dynamic changes that might cause anomalies in VN. This early investigation lays the groundwork for developing a camera FDE technique later in the thesis. Additionally, this approach can be adapted for other auxiliary sensors by developing the corresponding anomaly prediction modules.

- **Chapter 6: Deep Hypothesis Testing for fault alarms in auxiliary sensors**

Publication: Paper 3.

Chapter summary: This chapter extends the architecture of chapter 4 for detecting faults in auxiliary sensors within multi-sensor navigation systems. The investigation is specific for the camera. It proposes this approach to address the challenge of estimating consistent position error covariance in systems that integrate VN algorithms with the conventional sensors GNSS and IMU. Building upon the DNN developed in the chapter 5 for the initial prediction of VN anomalies, this approach expands to incorporate a method to refine these predictions using an IM test statistic specific to the camera. The proposed method is contained in a specific module, named camera Deep Hypothesis Testing (DHT). Other DHT modules, designed for each auxiliary sensor, can potentially be developed similarly. Operating independently, these modules enable efficient fault detection without the need to modify existing IM frameworks.

- **Chapter 7: Experimental evaluation**

Publication: Paper 3 & 4

Chapter summary: This chapter presents the experimental evaluation of the DHT module separately for camera PL inflation and FDE, as well as PL inflation for IMU preintegration faults. PL inflation involves adjusting PLs to account for sensor faults, as detailed in Chapter 4, which also describes the PL inflation methodologies for camera and IMU. The camera FDE is described in chapter 6. The evaluation chapter includes also a brief investigation of a few cases of VN anomalies identified in the trajectories under evaluation, aiming to identify potential environmental characteristics contributing to these anomalies.

- **Chapter 8: Conclusion**

Chapter summary: This concluding chapter summarizes the key findings and contributions of the thesis, discusses potential future research directions, and reflects on the implications of the study in the field of navigation systems.

Chapter 2

Navigation algorithm uncertainty and different sensor systems

2.1 Introduction

Autonomous navigation through complex environments demands robust algorithms capable of handling diverse sensor inputs. This chapter introduces the theoretical background of navigation algorithm uncertainty, detailing the methodologies used to calculate uncertainty for different sensor systems and explaining how each sensor's characteristics influence the overall system. The exploration commences with a general analysis of sensor noise, leading into a discussion on covariance estimation in ErKF and SLAM, both of which are navigation solutions utilized or monitored in the current thesis. Another section introduces the concept of covariance inflation, a method employed to account for atypical sensor noise and potential sensor faults. Theoretical frameworks and mathematical formulations developed within that section contribute to the foundation of the proposed IM architecture in later chapters. Another section in the current chapter does a high-level exploration of loose and tight integrations of multiple sensors. The last section examines the utility of model-based fault-detection and IM with a Bank of KFs. All the concepts described in this chapter outline a high-level procedural approach to develop the IM methodology. They serve as a guide for identifying essential considerations and ensuring the robustness of the IM method in addressing potential implications.

2.2 Sensor noise

This section investigates the critical gap in universal IM frameworks for non-linear systems and sensor fusion applications, focusing specifically on the impact of sensor noise. The emphasis of the developed IM architecture is on scalability, accommodating diverse sensors and noise models.

Sensor noise encompasses the random variations or fluctuations present in sensor measurements, arising from factors such as environmental conditions, manufacturing imperfections, and electronic interference [29]. For example, in GNSS receivers, sensor noise may stem from satellite signal distortion, atmospheric conditions, and receiver hardware limitations. Sensor noise can often be modeled as additive, where the noise is directly added to the true sensor measurement or a related measurement. The latter approach aids in practical application within navigation systems. For instance, in an IMU used for navigation, although the sensor provides position, velocity, or attitude measurements, the underlying sensing principle revolves around measuring linear acceleration and angular velocity [30]. Recognizing this discrepancy is important, as noise modeling is an essential aspect for deriving PLs in an IM system. All the derivations in this thesis assume additive sensor noise.

Earlier, the thesis introduced RAIM, which serves as a fundamental IM methodology for GNSS-based navigation. It is imperative to delve into how RAIM addresses sensor noise in its computation of PLs, as this approach will be adopted in specific parts of the current thesis. RAIM incorporates a FDE module and a module to compute PLs bounding the position estimate errors. Given a navigation algorithm estimating vehicle positions and uncertainties, the part of the PL that accounts for nominal sensor noise, which is assumed to be overbounded by a Gaussian function, in the absence of sensor faults, in one direction of travel, is given below:

$$p_{L_0} = Q^{-1} \left(\frac{I_{\text{REQ}}}{2} \right) \sigma_0 \quad (2.1)$$

where Q^{-1} is the inverse tail probability of the standard normal distribution, I_{REQ} is the preset integrity risk requirement allocation under the fault-free hypothesis and σ_0 is the overbounding standard deviation of the error of a given navigation algorithm. The intuitive interpretation of the equation is that, assuming that the position error follows a standard normal distribution, then the factors $Q^{-1} \left(\frac{I_{\text{REQ}}}{2} \right)$ will bound the probability of HMI. The equation includes the division by 2 because both tails of the error distribution need to be accounted for.

Nevertheless, modern autonomous systems rely on advanced sensor fusion algorithms, encompassing non-GNSS sensors like cameras or lasers. The integration of these sensors introduces additional complexities, including unique noise characteristics, which must be adequately addressed to ensure the reliability of the IM framework. PL inflation is essential for mitigating the effects of sensor noise on the integrity of position solutions, particularly in environments where traditional PL computations may underestimate the true error bounds. A good example of such an approach is presented in section 3.4, which formulates a time-correlated noise model added in a position sensor's measurements and the relative PL inflation module.

2.3 Covariance estimation by different algorithms

Covariance stands as a key metric for quantifying uncertainty and error associated with estimated parameters. In the context of navigation, covariance matrices provide a quantitative measure of the accuracy of the estimated state variables. While specific covariance computations are not an essential part of the thesis, it is worth providing a brief description to understand the principles underlying uncertainty quantification. These principles, including the concepts of overconfidence and underconfidence, are closely related to understanding the process of computing safety bounds for navigation solutions.

The ErKF is a common approach used for state estimation in navigation systems. A standard ErKF implementation is employed throughout the thesis to derive PL inflation terms for diverse sensor integrations, as was highlighted also in the contributions section (1.5). In ErKF, the covariance matrix is dynamically updated based on the system's motion model and sensor measurements. It accounts for the propagation of uncertainty through the state variables and sensor measurements. Appendix B includes the derivation of the predicted covariance, in a similar manner as the classic KF. This covariance is given as:

$$\hat{\mathbf{P}}_{k|k-1} = \mathbf{A}_{k|k-1} \hat{\mathbf{P}}_{k-1|k-1} \mathbf{A}_{k|k-1}^T + \mathbf{Q}_{d_{k|k-1}}, \quad (2.2)$$

with $\mathbf{A}_{k|k-1}$ being the error state dynamics matrix, which is given in equation (B.5), $\mathbf{Q}_{d_{k|k-1}}$ being the discrete time model definition of the covariance matrix of the process noise, which is given in equation (B.7) and $\hat{\mathbf{P}}_{k-1|k-1}$ being the posterior covariance estimate at step $k - 1$.

The equation for the updated covariance will be:

$$\hat{\mathbf{P}}_{k|k} = (\mathbf{I} - \hat{\mathbf{W}}_k \mathbf{H}) \hat{\mathbf{P}}_{k|k-1}, \quad (2.3)$$

with \mathbf{I} being the unity matrix, $\hat{\mathbf{W}}_k$ being the kalman gain, and \mathbf{H} being the measurement matrix after linearization.

The thesis leverages popular existing implementations of visual- and visual-inertial SLAM for evaluating the developed anomaly prediction methodology and deriving PL inflation terms for integrations featuring a camera. Notably, SLAM estimated covariances are not utilized anywhere in the thesis in the IM development procedure; however, a high-level understanding of the concepts is still helpful.

The goal of SLAM using a camera is to obtain global and consistent estimate of the camera's path, localization of the camera on a map and to reconstruct a consistent map of the surrounding environment. Covariance in SLAM is typically computed

by incorporating sensor measurements, feature observations, and the robot's motion model. The SLAM algorithm iteratively refines the covariance matrix to improve the accuracy of the estimated trajectory and map. In contrast, Visual Odometry (VO) or the form that uses inertial measurements, VIO, focuses on obtaining locally consistent estimate of camera poses, achieving incremental estimates [31].

Mourikis and Roumeliotis [32] derived a closed form solution of the SLAM uncertainty, relating it with sensor noise parameters and number of features being mapped. It is also concluded that prior information about the spatial density of landmarks can be used to compute a tight upper bound on the expected covariance of the positioning errors.

Consider a robot performing SLAM in 2-D, by observing N stationary landmarks in the environment. The robot uses velocity and orientation measurements to propagate its state. Consider that \mathbf{R} is the covariance matrix of the measurements. It is a combination of co-variances due to noise in bearing and range measurements to landmarks and due to noise in orientation estimates. Define:

σ_ρ^2 : the variance of the noise in the distance measurements.

σ_ϕ^2 : the orientation uncertainty.

σ_θ^2 : the variance of the noise in the bearing measurements.

\mathbf{Q} is the power spectral density matrix of the noise of the input, and \mathbf{H} is the measurement-to-state matrix. The time evolution of the covariance of the position estimates will be:

$$\dot{\mathbf{P}}(t) = \mathbf{Q}(t) - \mathbf{P}(t)\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{P}(t) \quad (2.4)$$

The robot uses velocity and orientation measurements to propagate its state. The equation (2.4) does not have a closed form solution. However, an upper bound has been derived as:

$$\dot{\bar{\mathbf{P}}}(t) = \bar{\mathbf{Q}} - \bar{\mathbf{P}}(t)\mathbf{H}_0^T\bar{\mathbf{R}}^{-1}\mathbf{H}_0\bar{\mathbf{P}}(t), \quad (2.5)$$

which is valid in case $\bar{\mathbf{Q}} \geq \mathbf{Q}_0(t)$ and $\bar{\mathbf{R}} \geq \mathbf{R}_0(t)$, $\forall t > 0$. The $\mathbf{R}_0(t)$ is a bound for the covariance matrix aggregating the corresponding bound of the covariance of the error due to the noise in the range measurements, of the covariance due to the error in the bearing measurements, and of the covariance due to the error in the orientation estimates. An expression of $\mathbf{R}_0(t)$ is given in the equation system (A.1) of appendix A.

These bounds for the individual terms should be

$$\bar{\mathbf{R}} = (\sigma_\rho^2 + N\sigma_\phi^2\rho_0^2 + \sigma_\theta^2\rho_0^2)\mathbf{I}_{2N \times 2N}, \quad (2.6)$$

and

$$\bar{\mathbf{Q}} = \begin{bmatrix} \bar{q}\mathbf{I}_{2 \times 2} & \mathbf{0}_{2 \times 2N} \\ \mathbf{0}_{2N \times 2} & \mathbf{0}_{2N \times 2N} \end{bmatrix} \quad (2.7)$$

with $\bar{q} = \max(\delta t^2 \sigma_V^2, \delta t^2 V^2 \sigma_\phi^2)$ being an upper bound for the covariance in the inputs, assuming approximately constant velocity V .

Then, the upper bound of the steady-state normalized covariance matrix (normalization by dividing by \bar{q}), after mathematical operations, will be:

$$\mathbf{P}_{n_{SS}} = \begin{bmatrix} \sqrt{\frac{\bar{q}r}{N}}\mathbf{I}_{2 \times 2} & \mathbf{0}_{2 \times 2N} \\ \mathbf{0}_{N \times 2} & \mathbf{0}_{2N \times 2N} \end{bmatrix} + \mathbf{1}_{(N+1) \times (N+1)} \otimes \left(\mathbf{J}^T \mathbf{P}_{LL_0}^{-1} \mathbf{J} + \left(\sqrt{\frac{\bar{q}r}{N}}\mathbf{I}_{2 \times 2} + \mathbf{P}_{rr_0} \right)^{-1} \right)^{-1}, \quad (2.8)$$

with $r = \sigma_\rho^2 + N\sigma_\phi^2\rho_0^2 + \sigma_\theta^2\rho_0^2$, \mathbf{P}_{rr_0} being the initial covariance of the robot's position estimate, \mathbf{P}_{LL_0} being the initial map covariance, $\mathbf{J} = \mathbf{1}_{N \times 2} \otimes \mathbf{I}_{2 \times 2}$ and \otimes being the Kronecker matrix product.

A more detailed derivation for the 1D and 2D case can be found in Appendix A.

2.3.1 Covariance estimation for map-based vision-aided navigation: A review

The challenges with dealing with covariance estimation for map-based vision-aided navigation were highlighted in many research works.

A method to deal with the problem of covariance overconfidence in EKF for map-based vision-aided navigation was presented by Zhu and Taylor [33]. Limitations in measurement observability, linearity and uncertainty modeling result in inconsistent estimated covariance. Inconsistency can also affect negatively feature processing of the estimator itself. The Covariance Intersection (CI) technique yields conservative uncertainty estimates even if the noise sources in the states and measurements (e.g. landmark observations) are highly correlated. Unlike the EKF, CI does not assume that the uncertainty in prediction is uncorrelated with the uncertainty of measurement, nor does it assume that measurements are uncorrelated from each other. If correlations exist, the EKF may exhibit over-optimism, as its error ellipse is positioned within the overlap area between the ellipses of the prediction and the measurement. Tkocz and Janschek [34] established a monocular SLAM initialization algorithm for a consistent initial covariance, based on sensor measurements and their known uncertainties. They were interested in investigating the consistent initialization of a loosely coupled visual SLAM + IMU EKF framework aiming the scale determination. They emphasized the challenge of newly discovered features in such filter approaches and that, instead, closed-form solutions are appropriate to overcome this issue. In essence their work

utilizes a closed-form solution for the initialization of the full filter state and covariance and evaluates the initialization consistency which will affect the subsequent filter process. Their method assumes a good feature extraction and outlier detection done beforehand. The landmark's projection onto the camera's plane and the landmark's distance define the landmark position as 3 element vector.

Furthermore, Shetty's et al. research was interested in the covariance estimation for GPS-LiDAR sensor fusion. They used the Iterative Closest Point method to match the laser's point cloud with a pre-existing 3D city model in order to estimate the global pose. The overall position error covariance is constructed as a combination of covariances estimated due to different features in the point cloud (e.g. edges), however there are many assumptions used about how the features affect the error, without providing guarantees of consistency.

Finally, Carrillo et al. [35] explored various optimality criteria which can be used as metrics for uncertainty in SLAM systems. The authors proposed that the D-optimality criterion can provide the most meaningful information for robot SLAM tasks. Mathematically, the D-optimality criterion aims to minimize the determinant of the covariance matrix associated with the estimated parameters of the SLAM system (e.g. robot pose and landmark positions). The D-optimality criterion effectively minimizes the volume of the joint confidence ellipsoid of the unknown parameters.

2.4 Covariance inflation for atypical sensor noise and for sensor faults - Loose and Tight Integration

Covariance inflation refers to the process of inflating the uncertainty of sensor measurements or state estimates. This adjustment compensates for internal sensor faults, sensor noise modeling faults or other elements in the environment that cause an undetectable sensor fault, leading to inconsistent estimates of state error uncertainties. Sensor noise modeling faults occur when the actual noise characteristics deviate from the assumed model, such as when noise is non-Gaussian or time-correlated. Undetectable sensor faults are subtle errors that escape detection by standard fault detection methods, often caused by factors like multipath interference, signal blockage, or unexpected environmental changes affecting sensor performance. Let's use the term sensor fault to describe any of the aforementioned faults. Usually, there is no knowledge of the magnitude or the direction of a potential fault \mathbf{f} in a measurement set and we pursue the computation of the worst-case fault. This is done by employing the concept of fault slope:

$$\mathbf{Slope}_f = \frac{\text{effect of } \mathbf{f} \text{ on state estimation error}}{\text{effect of } \mathbf{f} \text{ on the IM test statistic}} \quad (2.9)$$

The \mathbf{Slope}_f incorporates the magnitude as well as the direction of the worst case fault. The presence of a probable fault can be captured from significant covariances of the residual of expected and actual sensor measurements. This contrasts the typical case

where covariance refers to the expected variability in non-fault situations. Additionally, as for expected measurements, I refer to those resulting from recent navigation state estimates or from measurements of other sensors. For example, in monitoring KF-based algorithms, the innovation term is often utilized as the residual [36]. In this thesis, the introduced IM method leverages the MHSS technique to generate residuals, encompassing discrepancies among redundant position sensors (as discussed in section 3.3), discrepancies between conventional sensors and VN state estimates (to be detailed in section 4.5.2), and residuals between predicted and actual preintegrated IMU measurements (to be elaborated in section 4.5.1). These residuals are crucial components within the evaluated integration example of IMU, GNSS, and VN in this thesis. Importantly, this thesis investigates fault slopes in specific directions and assumes that the maximum fault slope is associated with one direction in the residual.

Where applicable, the fault in the sensor measurements ($Slope_f$) must be corrected to reflect the caused navigation algorithm's state error. In these cases, a Jacobian for the rate of change between the state and the residual is utilized.

A high-level formulation of a PL inflation term due to one or multiple probable undetected faults can be expressed as follows:

$$PL_inflation_term = \text{Undetected state error due to sensor faults} \times \text{A penalty, given the probability of false alarm of sensor faults,}$$

Here, the first factor accounts for the $Slope_f$ associated with the underlying faults examined in each case, probably corrected to apply to the monitored algorithm. The second factor addresses the continuity requirement of the navigation solution [37]. Continuity refers to the capability of the navigation system to provide position estimates that satisfy the accuracy and integrity requirements without being interrupted during the operational period. Specific formulations for this factor are detailed during the IM development in subsequent sections of this thesis.

It is crucial to note that the method of covariance inflation and the implementation of MHSS may have varying implications depending on the type of system integration employed. Multi-sensor integration architectures typically fall into two categories: loosely-coupled and tightly-coupled [38]. In loose integrations, the identification of error sources is relatively straightforward, as the impact of individual sensors is more apparent. Conversely, defining fault hypotheses in tight integrations poses greater challenges due to the measurement coupling procedure performed internally in the navigation algorithm. Moreover, evaluating the navigation algorithm in MHSS with different subsets of sensor measurements, assumed fault-free under each hypothesis, may not be directly feasible in tight integrations.

Figure 2.1 depicts generalized diagrams of the two types of integrations. These rep-

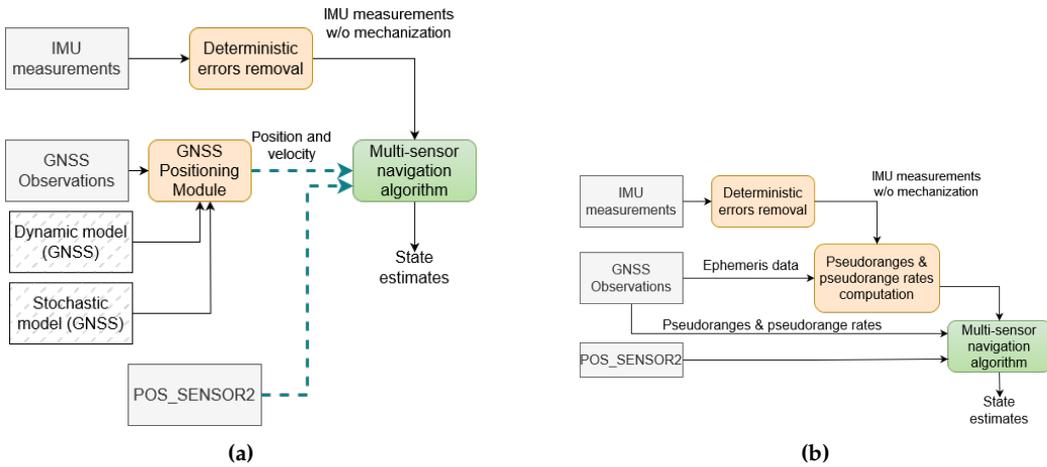


Figure 2.1: Diagrams depicting loosely- (a) and tightly-coupled (b) integration strategies of IMU, GNSS and POS_SENSOR2. The dashed lines in (a) show that execution of the navigation algorithm using either the GNSS or the POS_SENSOR2 is usually possible. Modified diagram from [38]

representations assume that in addition to the GNSS there is one more sensor providing position measurements, named as POS_SENSOR2. Consequently, redundancy exists with the GNSS sensor, allowing for IM using MHSS by assuming one of the two sensors is faulty at any given time. The IMU provides linear acceleration and angular velocity measurements, which are then processed through a mechanization procedure to derive position, velocity, and attitude. This mechanization step may occur either internally within the monitored navigation algorithm or externally. This step is not depicted in the diagram.

In the diagram depicting loosely-coupled integration, dashed cyan lines illustrate that when the navigation algorithm sufficiently functions with data solely from one position sensor, the redundant sensor may be completely omitted from informing the navigation algorithm. Conversely, in the tightly-coupled integration, integrating lower-level measurements from GNSS with IMU is imperative, a requirement that may extend to other sensors. The way that sensor measurements are utilized is very specific to the employed navigation algorithm.

In the context of this thesis, methods are investigated to define residuals associated with each sensor independently from the monitored navigation algorithm, ensuring greater adaptability and robustness across both loosely- and tightly-coupled integration architectures.

2.5 Bank of Kalman filters, Model-based Fault Detection for Multi-sensor Data Systems

Corrupted data from sensors with failures can have a severe impact on navigation system functionality, resulting in overall system performance degradation and a higher risk. In multi-sensor integrations, sensor faults pose the primary risk. IMs employ two main components to mitigate these risks: covariance inflation, which adjusts for probable navigation state faults stemming from individual sensor faults without modifying the input measurement set, and FDE techniques, which make binary decisions. These decisions entail either confirming that everything is functioning correctly and no action is needed, or identifying a fault and excluding one or more sensors or sensor measurements. In most cases, both IM components are built by using the concept of redundancy of independent information, which can be either hardware or analytical redundancy. Methods that employ hardware redundancy generally require additional hardware to generate duplicate measurements of the same quantities, such as duplicate sensors of the same type. In contrast, methods that employ analytical redundancy are generally more cost-efficient and rely on the already available hardware for the navigation system. The two most popular methods to utilize analytical redundancy are data-driven or model-based. Data-driven methods utilize data from the sensors to determine system behavior. Nevertheless, the quantity of required data is usually big and data-driven methods become inappropriate for online safety monitoring. Model-based methods require only the navigation system model and work on the principle that observed measurements are consistent with model-predicted measurements / outputs. In this domain, KF-based techniques are the standard to compute required uncertainty inflation terms in PL or FDE and there are different versions to be able to handle linear or non-linear systems.

Figure 2.2 shows the general structure of a model-based fault diagnosis system, which is made up of the residual generation and the residual evaluation components. Multi-sensor fusion processes may involve raw sensor data input or state vector inputs. The latter are estimates from systems that are informed by observations of one or more of the sensors. There are two common architectures to integrate the various sensors, and these architectures are shown in figure 2.3. In the figures we do not specify the data fusion module, which might be implementations based on, e.g. Kalman filtering or smoothing.

The figure shows two architectures; centralized fusion and distributed fusion. Centralized fusion uses the whole measurement set and achieves a high accuracy. However, it might be computationally inefficient, estimating a large state vector, and it might be unable to handle sensor inconsistencies and faults, or might result in complex failure of the central module. Distributed fusion uses multiple local fusion modules with a global fusion module. This architecture may achieve higher robustness to faults and more accurate detection of the failing sensor.

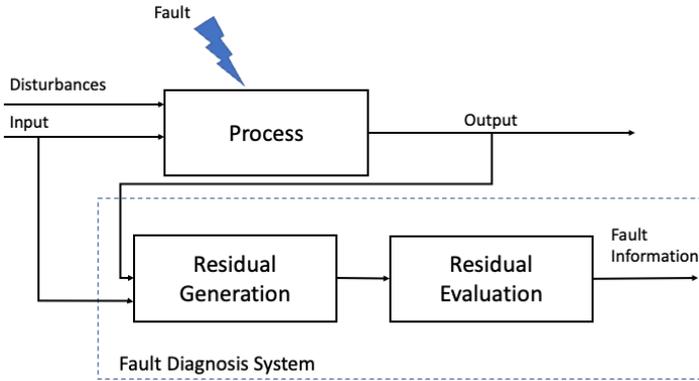


Figure 2.2: Model-based Fault Diagnosis [39].

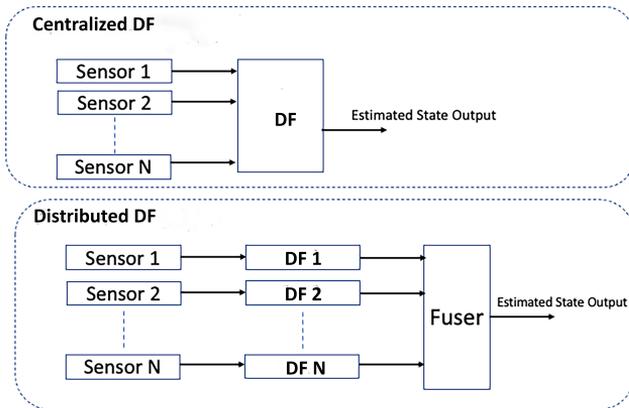


Figure 2.3: Architectures for Data Fusion. Modified diagram from [40].

There are also several algorithms for multi-sensor safety monitoring.

The first architecture, named here as information mixture, contains local fusion modules and an information mixture module analyzing local outputs, detecting faults, quantifying inconsistencies and providing a global output / estimate.

MHSS is another notable algorithm, where multiple faulty measurement hypotheses are constructed, including a fault-free hypothesis and hypotheses excluding one measurement each. It evaluates disagreement between the estimation under the fault-free hypothesis and estimations under fault hypotheses to identify faults or quantify inconsistencies for inflation term computation. The MHSS can be modified to handle multiple faults with a multi-layer approach, assuming one sensor faulty in the first layer, two sensors faulty in the second layer, and so on [41].

The Statistical Testing method relies on residuals or innovation signals of the fusion algorithm, where it focuses on measures like whiteness, mean and covariance of the residual.

In the Interacting Multiple Model algorithm, the main approach is to construct a set of possible candidate models for the true system. It consists of a bank of filters running in parallel, each designed with a unique model, and the architecture allows someone to accurately estimate the current operational mode and derive the overall correct state estimation in the presence of a fault.

The duplication/comparison is another method and uses error detection and system recovery to both detect and isolate the faulty sensors. Error detection helps to detect the erroneous state of the system. In system recovery, an error-free state can be substituted for an erroneous state. In data fusion, the duplication/comparison approach involves using at least two redundant units to perform fusion of data collected from diversified or redundant sensors to estimate system states and then comparing them to detect abnormal system behavior.

Figures 2.4-2.6 show the information-mixture , MHSS and Duplicate-Comparison algorithms.

Finally, the fault estimation method addresses the issue of distinguishing actual faults from disturbances or biases in residual, for a good quality safety monitoring. Such approaches are the one from Zarei and Shokri [44] that utilizes a Non-linear Unknown Input Observer to generate a robust residual for FDE and the one from Pazera et al. [45] which aims on simultaneous estimation of the system state and sensor faults for fault-tolerant control.

2.6 Chapter Summary

This chapter provides a theoretical exploration of navigation algorithm uncertainty and different sensor systems crucial for autonomous navigation in complex environ-

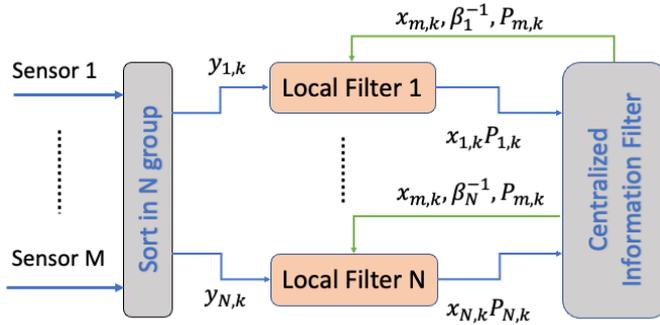


Figure 2.4: The information-mixture technique [42]: 1) the measurements from M sensors are sorted into N groups. 2) Each group of sensor $y_{i,k}$ is used to estimate the state through local filter. 3) Each local $x_{i,k}, P_{i,k}$ estimate is sent to the information mixture, and global states are calculated. 4) Finally, the global estimates $X_{m,k}, P_{m,k}$ are sent back to local filters and each local is updated. β_i is the information distribution factor of the i -th local filter, which is used to weight the local.

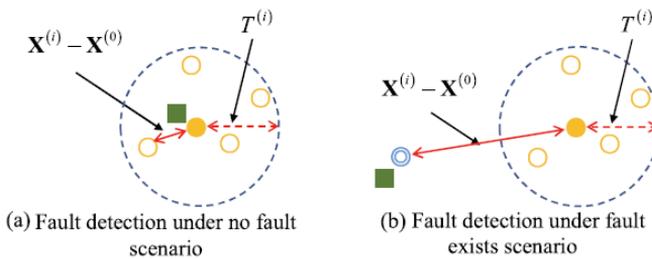


Figure 2.5: MHSS-based fault detection in positioning domain. The green rectangle, orange filled circle, orange circle and nested blue circle represent respectively ground truth position, fullset-position estimate $X^{(0)}$ under fault-free hypothesis, position estimate $X^{(i)}$ under fault-tolerant hypothesis that includes faulty sensor and position estimate under fault-tolerant hypothesis without faulty sensor. Figure from [15].

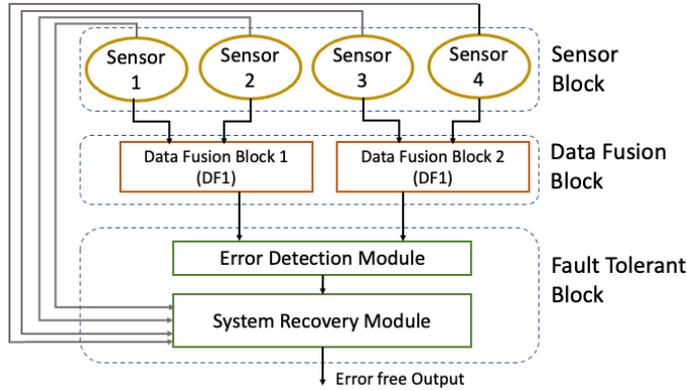


Figure 2.6: Duplication/comparison architecture [43].

ments. It introduces the theoretical background of navigation algorithm uncertainty and presents methodologies for uncertainty calculation in ErKF and SLAM, which are two popular algorithms in the domains of non-linear state estimation and VN. Also, the evaluation of the IM methodology in the present thesis uses implementations of these algorithms. The discussion delves into sensor noise and highlights its impact on navigation solutions, particularly focusing on additive noise modeling and its implications for IM. Additionally, the chapter introduces the concept of covariance inflation as a method to account for atypical sensor noise and potential sensor faults, laying the groundwork for subsequent chapters' IM architecture. The chapter continues with a comparative analysis of loose and tight integrations of multiple sensors, emphasizing their implications for IM methodologies and system robustness. The chapter concludes with a discussion of model-based FDE and IM methods for multi-sensor data systems. Overall, this chapter provides a comprehensive overview of the theoretical foundations which I considered during the development of the presented IM methodology working with multi-sensor navigation systems.

Chapter 3

Navigation algorithm-agnostic IM for multiple sensor noise models

3.1 Introduction

This chapter introduces an MHSS based framework for IM for the conventional GNSS/IMU integration with a filter-based and a smoothing-based navigation algorithm. It builds upon the preceding chapter and explores the complexities inherent in monitoring multi-sensor systems with different characteristics in terms of uncertainty outputs. In the controlled environment of the simple integration it explores also the complexities of non-standard noise, with a particular focus on time-correlated noise—a prevalent scenario in real-world applications. To investigate this case, time-correlated noise is introduced artificially in the experiments. Specifically, the chapter introduces an IM architecture which takes into account that: 1) a multi-sensor system must account for various sensor noise models which lead to inconsistent estimates of uncertainties, 2) a module must be able to detect sensor failure or sensor noise mismodeling and suggest better bound for the error, without being constantly conservative, 3) some algorithms are computationally heavy to monitor in the MHSS setting or the provided covariances cannot be interpreted in IM.

Most of the works that investigate the integrity of multi-sensor systems are constrained by linearity assumptions (e.g. the research of Meng and Hsu [15]), or they attempt to develop an algorithm that performs both the estimation and integrity monitoring collaboratively (e.g. the research of Mohanty et al. [18]). It is desirable for a universal IM framework to work with non-linear systems, that are commonly used today, and be independent from the navigation algorithm. The current study extends an MHSS method for IM to remove the linearity or Gaussian noise assumption. The evaluation of the method is with a filtering-based, as well as a smoothing based, navigation algorithm.

The development of a universal solution must account for the particular error models of various sensors. There is no widely adopted approach that exists to account for time-correlated errors with unknown parameters. The developed IM method in this chapter adopts a method by Crespillo et al. [46] to account for time-correlations by introducing overbounding hypotheses in a MHSS context. These hypotheses are used conditionally, and the decision is made automatically by a log-likelihood based algorithm. This leads to more reliable IM with tighter PLs in comparison to methods with continuous overbounding.

In addition, the developed IM enables easy integration with existing navigation algorithms. On a high-level, it utilizes a MHSS methodology, where a sensor is completely excluded in each hypothesis. Despite the limited number of fault hypotheses, some navigation algorithms are still computationally heavy to execute repeatedly in a MHSS setting, or do not provide absolute covariances, or do not provide covariances at all. Experimentation was conducted with monitoring a smoothing-based navigation algorithm. The IM method optionally uses an ErKF implementation as backend to evaluate faulty hypotheses, independently of the smoothing-based navigation algorithm used in the all-source-in hypothesis.

The experimentation in this chapter uses one simulated UAV trajectory, and one real trajectory from a car driven in an urban environment. The first experiments evaluate the proposed IM on an ErKF implementation, which is a proper choice of a modern non-linear estimator. Further experiments evaluate the IM on an algorithm based on Factor Graph Optimization, which is a very popular algorithm in the domain of smoothing-based navigation.

The main contribution of the work presented in this chapter is a novel IM architecture for the IMU/GNSS integration, which is navigation algorithm-agnostic, in terms of internal architecture, and is scalable to various sensors and sensor noise models. A limitation is that the sensors should be capable to provide independent position solutions. More specifically

- The experimentation demonstrates that an MHSS integrity monitoring algorithm can employ statistics on Log Likelihoods of various measurement subsets for a filter-based as well as a smoothing-based navigation algorithm. These metrics can identify the presence of measurement faults and improve the understanding of underestimated uncertainties provided by the navigation algorithm. The chapter proposes a simple statistic based on the rolling standard deviation of the Log Likelihood (LLRSD) of each subset, which can accurately detect inconsistencies over time. This approach is likely applicable to a wide variety of navigation algorithms, offering a simpler alternative to the more complex LLR method previously studied [21].
- Multi-sensor fusion must account for sensor noise models that can lead to under-

estimated uncertainties and invalid bounds. This occurs because certain noise models may not accurately capture the true variability and correlations present in the sensor data, resulting in overly optimistic estimates of the system's performance. This chapter proposes overbounding the common case of time-correlated noise. We observe that overbounding techniques, previously derived for KF-like methods, provide promising results with the MHSS-based architecture with ErKF backend, even when the knowledge of noise model parameters is approximate. This is an empirical result and theoretical guarantees of overbounding are not provided. Nevertheless, a significant contribution of the proposed framework is its adaptability: it allows for straightforward incorporation of different sensor noise models by constructing hypotheses to apply appropriate overbounding techniques. This is particularly useful, for example, when a sensor's noise distribution has been learned offline under nominal or non-nominal conditions.

- There is a wide variety of sensor fusion navigation algorithms, and in some cases, position covariances are unavailable or cannot be interpreted in the absolute coordinate system. For example, this can occur when using certain types of non-linear filters or when sensor data is too sparse or irregular to generate reliable covariance estimates. In these scenarios, the IM system might fail to accurately assess the reliability and integrity of the position estimates. In addition, it can be inefficient to execute a computationally heavy navigation algorithm multiple times in the MHSS setting. This chapter goes in the direction of incorporating a standardized algorithm for the evaluation of some hypotheses. It includes results for a GTSAM algorithm for IMU and GNSS fusion, whose monitoring was done with and without the ErKF for evaluation of faulty hypotheses.

The chapter begins with an analysis of the allocation of integrity risk concerning potential faults associated with the utilized sensors. The subsequent sections present the method for PL computation in the MHSS approach and the proposed IM system with correlated noise overbounding. The last two sections introduce the data used for evaluation of the proposed IM and discuss the experimental results.

The methodology was first described in Paper 1.

3.2 Allocation of integrity risk

The concept of MHSS is closely related to the allocation of an integrity risk budget, as the integrity risk tree demonstrates in figure 3.1. Instead of depicting a generic allocation among all the various errors that can contribute, this figure demonstrates the proposed, on sensor-level, allocation. It is worth mentioning that the sensor set in the diagram serves only as an example, and the investigation is only for a subset of the mentioned sensors. In the diagram, N_{ss} denotes the number of fault hypotheses. I_{REQ} is the predetermined integrity risk requirement. The integrity risk is allocated equally among all hypotheses. Here, as in many works, the integrity requirement is equally

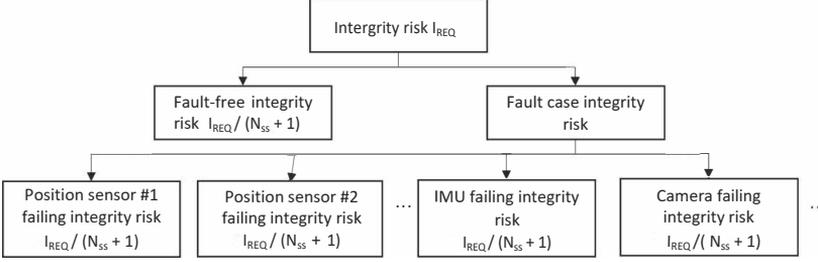


Figure 3.1: Example integrity risk allocation on sensor-level.

allocated to each hypothesis, something that may result in overly conservative bounds.

3.3 PL computation

Mathematically, the symbol p_L refers to the PL in one direction of interest, with the understanding that the study here concerns multi-dimensional navigation. It should be guaranteed that the probability of the error exceeding the PL is smaller than the integrity requirement. That is, under a fault hypothesis i , the p_L should satisfy the following inequality:

$$P(HMI | H_i) \cdot P(H_i) < I_{REQ_i} \quad (3.1)$$

HMI is the event $|\hat{x}_i - x| > \ell$ & $p_L < \ell$, where \hat{x}_i is the estimated position state of interest under hypothesis i , x is the corresponding true state and ℓ is the alert limit. I_{REQ_i} stands for the preset integrity risk requirement allocated to the i -th hypothesis. $P(H_i)$ is the probability of a fault hypothesis, which is typically determined through extensive simulation runs offline. However, in this work an arbitrary constant value of $P(H_i) = 10^{-5}$ was selected for all fault hypotheses. Then for the fault-free hypothesis, it is $P(H_0) = 1 - \sum_{i=1}^{N_{SS}} P(H_i)$.

By adding the contributions of each failure towards the integrity risk it is guaranteed that the overall PLs will bound the error and, therefore, the safety of the method. In the fault-free case, measurements follow a nominal distribution while the error is assumed to follow a Gaussian distribution.

Let's define the threshold for fault detection for a fault hypothesis i as $T_{\Delta_i} = Q^{-1} \left(\frac{P_{CONT}}{N_{SS}} \right) \sigma_{\Delta_i}$, where Q^{-1} is the inverse tail probability of the standard normal distribution, P_{CONT} is continuity risk allocated to the fault hypotheses and $\sigma_{\Delta_i}^2 = \sigma_i^2 - \sigma_0^2$, with σ_0^2 being the variance for the all-source solution, and σ_i^2 being the variance of the fault-free solution under the i th hypothesis. In the case of passing the test, a simple formula for the p_L calculation for the fault hypothesis i is given by:

$$p_{L_i} = K_{mdi} \sigma_i + T_{\Delta_i}, \quad (3.2)$$

where the factor K_{md_i} is the allowable missed detection threshold, which depends on I_{REQ_i} , and represents the number of standard deviations this threshold is away from the mean of the standard normal distribution. The standard deviation σ_i of the fault-free solution under the i th hypothesis is used to scale the threshold.

Based on the definitions, the K_{md_i} can be incorporated in formula (3.1), as follows:

$$Q(K_{md_i}) \cdot P(H_i) < I_{REQ_i} \quad (3.3)$$

By substituting K_{md_i} from (3.2) into this inequality, we get:

$$Q\left(\frac{p_{L_i} - T_{\Delta_i}}{\sigma_i}\right) \cdot P(H_i) < I_{REQ_i} \quad (3.4)$$

To find the p_{L_i} , convert the inequality to an equality, set the equally allocated requirement $I_{REQ_i} = \frac{I_{REQ}}{N_{SS}+1}$ and solve for p_{L_i} . This yields the following formula:

$$p_{L_i} = Q^{-1}\left(\frac{I_{REQ}}{P(H_i)(N_{SS} + 1)}\right) \sigma_i + T_{\Delta_i} \quad (3.5)$$

(3.5) asserts that if the position error follows a standard normal distribution then the factors $Q\left(\frac{I_{REQ}}{P(H_i)(N_{SS}+1)}\right)$ will bound the probability of HMI. The second term, T_{Δ_i} , can be considered a fault isolation threshold which has to satisfy the continuity risk requirement.

The basic equation (2.1) for the PL in the fault-free case can be modified here to $p_{L_0} = Q^{-1}\left(\frac{I_{REQ}}{2(N_{SS}+1)}\right) \sigma_0$.

3.4 Overbounding Time-Correlated Noise

It is a fact that the noise of some sensors is time-correlated and not white. The estimated covariance from some navigation algorithms will be unreliable unless the correlated noise is accounted for. Subsequently, in many cases the computed PLs will be unreliable.

A first order Gauss-Markov process (GMP) is the random process used in this work to model correlated noise. These processes are fully defined by a time constant τ , steady state variance σ^2 and initial variance σ_0^2 . The discrete first-order Gauss-Markov process is defined with the following equation for the discrete time step ζ [47]:

$$y(\zeta) = \alpha y(\zeta - 1) + u(\zeta), \quad (3.6)$$

y denotes a random variable (in this case the correlated measurement noise), $u(\zeta)$ is a

random variable that follows zero-mean Gaussian distribution with variance σ_u^2 and α is a coefficient in the range $[0, 1]$.

The correlation time T is the lag time corresponding to an autocorrelation coefficient of $1/e$, or:

$$\text{"Autocorrelation coefficient"}(y(\zeta), y(\zeta + T)) = \frac{1}{e} \quad (3.7)$$

For two samples m, n with $m > n$ the auto-correlation of $y(n)$ is

$$\begin{aligned} \text{"Autocorrelation coefficient"}(y(m), y(n)) &= \alpha^{(m-n)} \\ \Rightarrow \text{"Autocorrelation coefficient"}(y(\zeta), y(\zeta + T)) &= \alpha^T \end{aligned} \quad (3.8)$$

By combining with Eq. (3.7) it follows that: $\alpha = e^{-1/T}$.

Crespillo et al. [46] designed a GNSS + INS integration scheme where noise processes are correlated and the parameter values are known only to reside in a (wide) range of values. Langel et al. [48] formulated a way to guarantee that the estimated covariance from a KF overbounds the actual error distribution of the estimate. They based their proof on the propagation of the error matrix $E = \Sigma - P$, where P is the estimated covariance matrix of the KF and Σ is the covariance matrix to define, so that $\Sigma \geq P$. To achieve that, they concluded that the process noise power spectral density matrix Q (not to be confused with the tail probability Q of the standard distribution defined earlier) should be populated with the upper bound values of its individual elements. This requires to use the maximum time constant of the range and the upper bound for the uncertain steady-state variance parameter of the correlated noise, inflated by the ratio of the maximum and minimum values of the time constant:

$$\tau = \tau_{max} \quad (3.9)$$

$$\sigma^2 = \sigma_{max}^2 (\tau_{max}/\tau_{min}) \quad (3.10)$$

Finally, the initial variance of the GMP is determined in a way to satisfy the condition $E(0) \geq 0$ which is another prerequisite to have that $\Sigma \geq P$ at all $t > 0$:

$$\sigma_0^2 \geq \frac{2\sigma_{max}^2}{1 + (\tau_{min}/\tau_{max})} \quad (3.11)$$

Let σ_0^2 be equal to this lower bound. Formal proofs are provided by Langel et al. [48].

Later sections of this chapter provide some empirical results after utilization of this method with the IM with ErKF back-end. Specifically, in the simulation experiments run, a position sensor, that behaves similarly to a GNSS, is used with noise that follows a GMP. As the paper by Langel et al. [48] demonstrated, the GMP noise in each sensor can be sufficiently accounted for by augmenting the state vector with additional bias states and a subsequent inflation of the process noise covariance matrix Q . Time-correlated noise is just an example of the challenges that can deteriorate the quality of IM. The proposed architecture is compatible with additional overbounding techniques to handle other challenging cases.

Appendix B serves as an introduction to the ErKF implementation and the strategy to use the GMP parameter values determined in Eq. (3.9), (3.10) and (3.11) to inflate Q .

3.5 LLR and LLRSD for Fault Detection

He et al. [21] introduced a method to optimize the fault detection threshold with the objective to be suitable for a non-Gaussian test statistic and to enhance system integrity and availability. Their method uses the cumulative LLR and employs a genetic algorithm to determine the fault detection threshold value under various availability constraints, effectively identifying likely faulty sensors. An intuitive understanding of how LLR can be used in fault detection is the following: In the absence of the faulty measurement, one expects a better agreement of the remaining measurements in comparison to the fault-free hypothesis, i.e. the LLR will be higher in this case. This chapter shows empirically that a simpler metric (only the faulty hypothesis is used in computations), called LLRSD, provides a better indication of the times when a measurement is faulty. For completeness, the reader may consult Appendix C which includes an introduction to the LLR metric. The formulation of the LLRSD metric follows in this section.

For a time interval a to b , let us accumulate the Log- Likelihoods of a measurement subset y^j containing all measurements except the j th. This is the nominator of the LLR metric too (Eq. (C.2)):

$$\mathcal{L}\mathcal{L}_a^{(b,j)} = \sum_{k=a}^b \ln p(y_k^j | x_k) \quad (3.12)$$

Then, similarly to Appendix C, at each time k get the maximum, $\psi_k^j = \max_{k-x+1 \leq a \leq k} \{ \mathcal{L}\mathcal{L}_a^{(k,j)} \}$, $0 \leq j \leq m$, with x the accumulation time window size and m the total number of measurements.

Using a window of configurable size N it is attempted to find how much the maximum log-likelihood of subset j at current time (ψ_k^j) deviates from log-likelihoods at previous times, inside the window.

$$\text{LLRSD}_k^j = \sqrt{\frac{1}{N} \sum_{z=k-N}^k (\psi_z^j - \mu_k^j)^2} \quad (3.13)$$

where $\mu_k^j = \sum_{z=k-N}^k \frac{\psi_z^j}{N}$ is the mean of the maximum log-likelihoods in the window.

When using LLRSD, a valid hypothesis that excludes a faulty sensor would lead to low LLRSD values, indicating that the maximum log-likelihoods at different time points are closely clustered around their mean. This lower deviation implies a higher level of agreement among the remaining measurements.

The experimental results, presented in section 3.6.4, showed a higher consistency in correctly detecting a faulty sensor when using this metric than when using the LLR metric.

3.6 Experiments

This section introduces the trajectories, sensor combinations and integration strategies as well as the navigation algorithms that were evaluated with the developed IM method. Table 3.1 will summarize the IM results for each trajectory and monitored navigation algorithm.

3.6.1 Data

The experiments utilize two trajectories. The first trajectory corresponds to a 3D UAV scenario with a GNSS and an IMU sensor with perfectly aligned reference frames. The noise processes of the sensors follow a Gaussian distribution, and the trajectory was created in MATLAB. This trajectory is given in Figure 3.2. The second trajectory includes data collected from a car driving in an urban environment with, among others, measurements from a GNSS receiver and an inertial sensor module. This trajectory is included in the KITTI raw dataset [49]. It is shown in Fig. 3.3. The experimentation is limited to a loose integration of position fix and IMU sensors. The name POS_SENSOR1 is used for the purpose of the text and refers to the GNSS sensor. The next section describes the navigation algorithms under evaluation. For the testing of the fault detection method and the overbounding of correlated noise we synthesize the measurements of an additional position sensor called POS_SENSOR2. Specifically, the time correlation between successive position measurement errors is modeled employing a Gauss-Markov process, as described earlier, according to the Eq. (3.6), and this noise is added around the measurements of POS_SENSOR1. The parameters of the noise are as follows:

$$T = 300s$$

$$\sigma_u = 0.3m$$

The frequencies of each sensor measurements are:

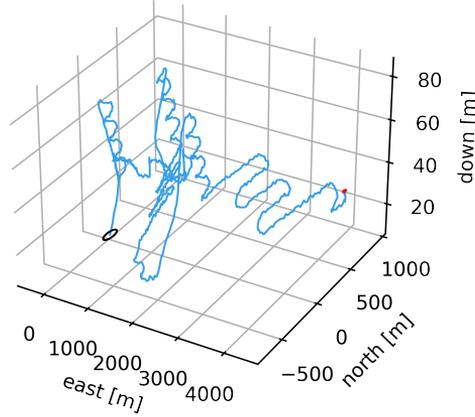


Figure 3.2: 3D simulated trajectory of a UAV in NED coordinates. The starting and ending positions are indicated with a black circle and a red dot respectively.

IMU: 10 Hz

POS_SENSOR1: 1 Hz

POS_SENSOR2: 2 Hz

The difference in frequency between the two position sensors was accounted for in the measurement error matrices. Specifically, the measurement noise covariance matrix for POS_SENSOR2, denoted as \mathbf{R}_{pos2} , was derived from the matrix for POS_SENSOR1, denoted as \mathbf{R}_{pos1} , by scaling it with a factor of 2 to match the higher frequency of POS_SENSOR2 [50]. That is, $\mathbf{R}_{\text{pos2}} = \mathbf{R}_{\text{pos1}}/0.5$. The elements in \mathbf{R}_{pos1} have been tuned for the simple POS_SENSOR1 + INS integration and based on estimated uncertainty consistency tests.

Figure 3.4 depicts the measurements of POS_SENSOR1 and POS_SENSOR2.

Both trajectories contain also Ground Truth (GT) measurements, with respect to the NED frame, utilized for the computation of the estimate error.

3.6.2 Experimental Setup and Parameters

The experiments were conducted using the following parameters and settings to evaluate the performance of the ErKF and GTSAM algorithms for different sensor integrations (POS_SENSOR1/POS_SENSOR2/IMU, POS_SENSOR1/IMU, POS_SENSOR2/IMU).

Sensor Noise Characteristics

KITTI Dataset

- POS_SENSOR1 (GNSS) Noise: Gaussian white noise with standard deviation 0.3

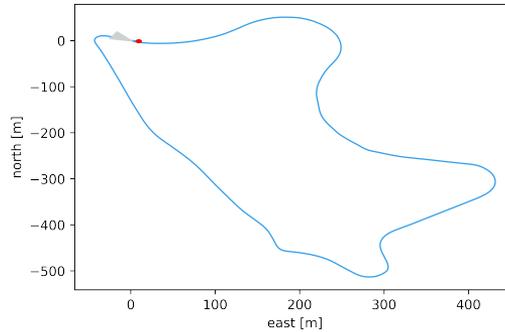


Figure 3.3: 2D collected data in NED coordinates from a real car drive. Trajectory from the KITTI raw dataset. An arrow indicates the starting position and direction, and the red dot indicates the ending position. Thus, the starting and ending position of the car are almost the same

m.

- IMU Noise (continuous): Accelerometer noise standard deviation 0.112 m/s^2 , Gyroscope noise standard deviation $1.22 * 10^{-7} \text{ rad/s}$.

UAV Dataset

- POS_SENSOR1 (GNSS) Noise: Gaussian white noise with standard deviation 0.3 m in North and East direction, and 0.5 m in Down direction.
- IMU Noise (continuous): Accelerometer noise standard deviation 0.022 m/s^2 , Gyroscope noise standard deviation $1.22 * 10^{-7} \text{ rad/s}$.

GMP noise injection for POS_SENSOR2

- Lag time T : 300 s
- Desired discrete time step Δt : 0.5 s
- $\alpha = e^{-\Delta t/T}$
- Gaussian distribution noise scale: 0.3 m

PL Computation Parameters

- Integrity risk requirement: $2.7 * 10^{-8}$ per hour
- Continuity risk requirement: $8 * 10^{-6}$ per hour
- Probability of fault hypotheses: 10^{-5}

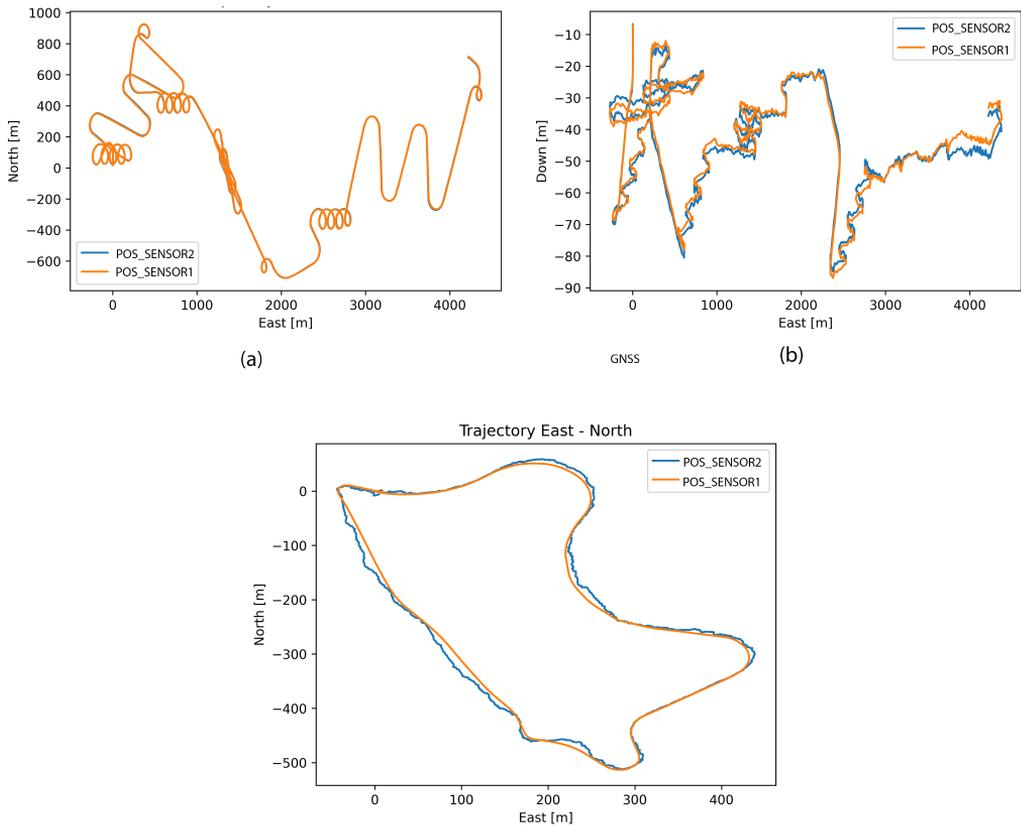


Figure 3.4: The measurements of POS_SENSOR1 and POS_SENSOR2. The coordinates were converted to the NED system. (a) Measurements in the 3D UAV trajectory North vs East direction (b) Measurements in the 3D UAV trajectory East vs Down direction (c) Measurements in the 2D KITTI trajectory North vs East direction

Fault Detection Configuration

- LLRSD Window Size: 10 samples
- LLRSD Threshold: 0.075 (empirically determined)

3.6.3 Navigation Algorithms Evaluated with the IM

The evaluation of the developed IM was on a loosely coupled sensor integration and on two types of navigation algorithms. The first was an ErKF and the second one was GTSAM, which is based on smoothing. Both will be described more analytically promptly.

A fault hypothesis is constructed after considering the full measurement set of a sensor as faulty, thus, excluding it. The MHSS-based IM executes the navigation algorithm for the original measurement set (all sensors are fault-free) and for the subsets under the fault hypotheses. The IMU is assumed fault-free in all cases to ensure redundancy of measurements at each time. Therefore, there are totally three hypotheses in the experiments:

- H_0 (All-source): all sensors are assumed non-faulty and used in the navigation algorithm.
- H_1 (Out-POS_SENSOR1): exclusion of the POS_SENSOR1.
- H_2 (Out-POS_SENSOR2): exclusion of the POS_SENSOR2.

ErKF

The ErKF is a successor of the original KF and the EKF for the cases when either the process or the measurement model or both are nonlinear [51, 52]. In this approach, the error in the states is estimated using a KF, instead of the state itself. The linearity in the error state dynamics means that the application of KF is feasible. Thus, ErKF takes advantage of KF's optimal estimations [51]. This filter is typically used for the fusion of IMU data with any other kinds of sensors. The experiments use this filter to loosely integrate one or two position sensors with the IMU sensor and validate the proposed IM. Appendix B serves as a brief introduction to the ErKF implementation.

GTSAM

GTSAM is a C++ library that implements smoothing and mapping (SAM) in robotics and vision, using Factor Graphs and Bayes Networks as the underlying computing paradigm rather than sparse matrices [53].

The experiments evaluate the IM with GTSAM with the IMU + position sensor integration. Although this integration is not state-of-the-art, it is a demonstrative example of

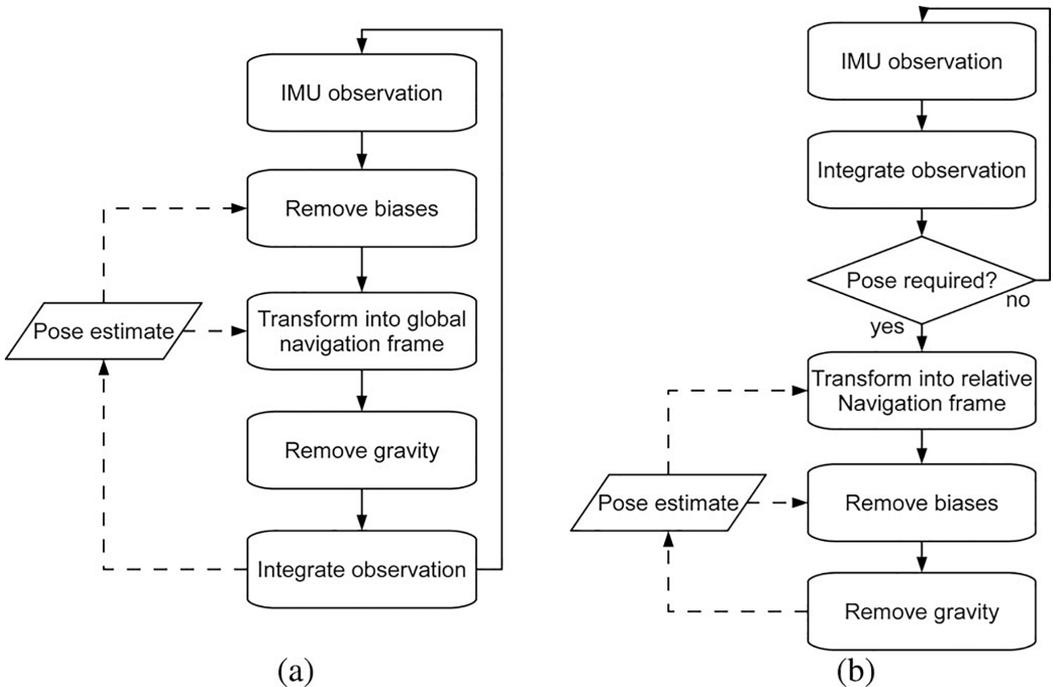


Figure 3.5: Flowcharts comparing the steps in standard inertial integration (a) with those implemented in GTSAM's IMUFactor (b). Figure from Lupton and Sukkarieh [54].

the straight-forward applicability of the method with more complex integrations used in smoothing-based navigation. The GTSAM experiments use the implementation of an IMU factor as proposed by Lupton and Sukkarieh [54]. Figure 3.5 compares the steps followed in standard inertial integration with those implemented in the IMU factor. This architecture allows the reparameterization of the navigation frame and the pre-integration of the inertial observations which facilitates the initialization of the system (all initial conditions are linearly dependent on the estimated states). In addition, the gravity vector must be considered after the observations are already integrated instead of accounting it during integration of the inertial observations in the velocity equation.

Like most nonlinear optimization libraries, GTSAM optimizes for a change with respect to a linearization point. This is very important to note, because in the case of GTSAM the covariance matrices are given in relative, and not absolute, coordinates [53].

IM of the GTSAM Navigation Algorithm using the ErKF Backend

The final examination is about how the IM with ErKF backend behaves in the case of the GTSAM navigation algorithm. Here, the ErKF performs the same integration as described earlier, aiming to estimate the covariances used to compute the bounding PLs of the estimation error in GTSAM. Specifically, the optimization procedure in GTSAM is

executed for the all-sensor input, while the ErKF estimates the covariances when leave-one-out sensor input is utilized to simulate fault hypotheses. This is an alternative implementation of the IM and, as mentioned before, can result in higher efficiency, depending on the output format and time complexity of the monitored navigation algorithm.

3.6.4 Results and Discussion

ErKF Experiments

Figures 3.6 and 3.7 present, for each of the trajectories (2 or 3 dimensions), the computed PL and the position error of the estimates relative to the GT positions and for different integrations of the POS_SENSOR1 and / or POS_SENSOR2 with the IMU. It is reminded that the motivation for including the second position sensor is to enable the use of typical multiple-hypothesis monitoring, which relies on measurement redundancy of different sensors to evaluate different fault hypotheses. In this work, when solely the POS_SENSOR1 or the POS_SENSOR2 is used with the IMU, the evaluation is limited to the fault-free hypothesis. The filter has quite good knowledge of the noise of the POS_SENSOR1 and the IMU, although the parametrization is not optimal. In the case of the simulated trajectory, it is observed that the IM fails to calculate PLs that reliably bound the error and ensure the integrity of the navigation solution when the POS_SENSOR2 is used. This is especially visible in Fig. 3.6c, where the POS_SENSOR1 has been completely excluded. In the case of the KITTI trajectory, we observed that the IM achieved sufficient bounds when MHSS was employed in the presence of both POS_SENSOR1 and POS_SENSOR2.

The remaining analysis in this section aims to evaluate the ability of the LLRSD and LLR metrics to detect the POS_SENSOR2 noise mismodeling fault.

Figure 3.8a shows for the UAV scenario the Pearson correlation coefficient of the LLRSD or the LLR metric with the absolute position error. This analysis is done for the integration of the POS_SENSOR1, the POS_SENSOR2, and the IMU, and for each hypothesis (where either POS_SENSOR1 or POS_SENSOR2 is assumed fault-free). In both cases, there is observable a weak to medium positive correlation among the metric in hypothesis $H1$ and the error in hypothesis $H0$ on the East axis only. Additionally, there is a weak negative correlation among the metric in hypothesis $H2$ and the error in hypothesis $H1$.

Figure 3.8b depicts the mean and standard deviation values of the LLRSD and LLR metric for each hypothesis, at the times when the PLs do not bound the error. The advantage of using the LLRSD metric over the LLR metric is visible in this figure where it is easy to identify that $H2$ is a valid hypothesis. In addition, the standard deviations (shown with the black vertical lines on the bars) are in general lower for the LLRSD metric and the $H1$ or $H2$ hypotheses. The lower standard deviations indicate that the LLRSD is more consistent in detecting the specific noise mismodeling fault, throughout

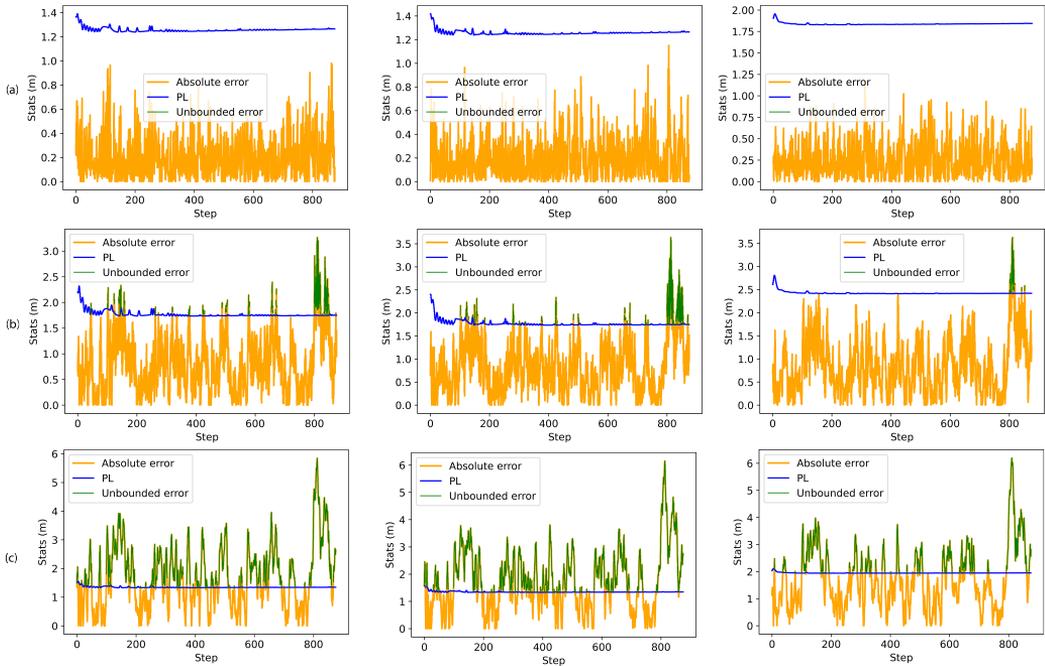


Figure 3.6: PL and absolute errors for various settings of POS_SENSOR1 + POS_SENSOR2 + IMU loose integration with ErKF for the UAV simulated trajectory. The parts of the error plot that exceed the corresponding PL are marked with green. Each column corresponds to one direction, North (left), East (middle), Down (right). (a) Utilization of measurements only from the POS_SENSOR1 and the IMU. (b) Utilization of measurements from both the POS_SENSOR1 and POS_SENSOR2, along with the IMU. (c) Utilization of measurements only from the POS_SENSOR2.

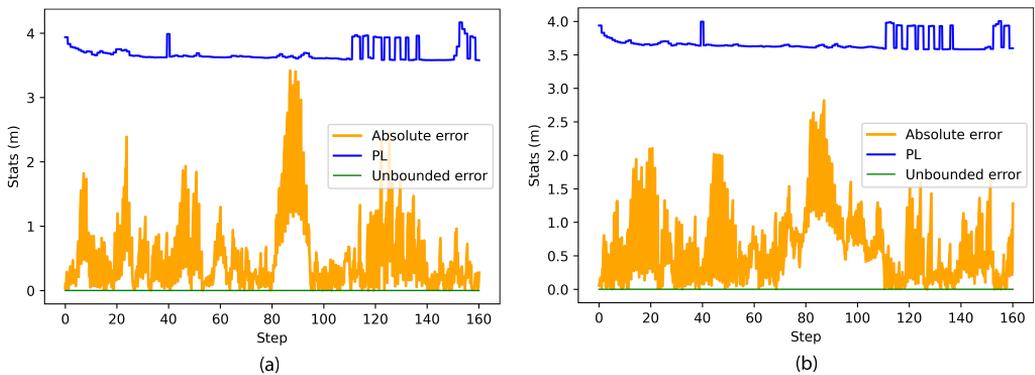


Figure 3.7: PL and errors for the POS_SENSOR1 + POS_SENSOR2 + IMU loose integration with ErKF for the KITTI trajectory. (a) North axis (b) East axis.

the experimentation time.

Figure 3.9 presents the results after applying the overbounding technique exclusively for the uncertainty of the measurements of the position sensors (see Section 3.4 and Appendix B), and for the times when the LLRSD value exceeds a preset threshold. Overbounding means that a higher bound is computed for the hypothesis that includes the faulty sensor. The LLRSD statistic can identify a faulty sensor, with the assumption that at most one sensor is faulty at each time. It is noted that the LLRSD threshold used is 0.075, selected empirically based on the LLRSD values obtained under each hypothesis during the current experiments. This threshold was chosen for a good separation of the values obtained during times of high estimated position error and low estimated position error under the all-source hypothesis. A future analysis would ideally investigate ways to set the LLRSD threshold with the objective to balance sensitivity to faults with the risk of discontinuity in system operation.

The plots in Figure 3.9 indicate a significant improvement in the IM system's ability to bound the position error. Another observation is that the overbounding technique generally avoids loose bounds during periods of low error.

Figure 3.10 presents the evaluation of LLR and LLRSD for the KITTI trajectory. The figure and Table 3.1 show that overbounding did result in unnecessarily looser bounds. This was expected as the monitored algorithm was already configured with large enough measurement uncertainties for the IMU and POS_SENSOR1 sensors. This is an important result that shows that overbounding might deteriorate the IM performance depending on how the tuning of the parameters was done in the first place and how uncertain are the correlated noise parameters.

GTSAM Experiments

Figure 3.11 depicts the PLs and errors for the UAV trajectory for the IMU + POS_SENSOR1 + POS_SENSOR2 integration. It is apparent that it is overconfident to use the covariance returned from the GTSAM software. However, the LLRSD is still a simple way to detect that the noise of POS_SENSOR2 is mismodeled. This is confirmed by Figure 3.12 which shows the mean and standard deviation of the LLRSD and LLR metrics.

Experiments on GTSAM using the ErKF backend for IM

The reduction of the execution time was apparent when this implementation was used instead of the GTSAM-based evaluation for all hypotheses, as Table 3.1 demonstrates. Figure 3.13 shows that PLs bound quite well the error in contrast to the PLs computed when purely GTSAM-based IM is used (Figure 3.11). However, the uncertainty is underestimated around the end of the trajectory. It is emphasized again that the overbounding technique was proven for KF-like algorithms.

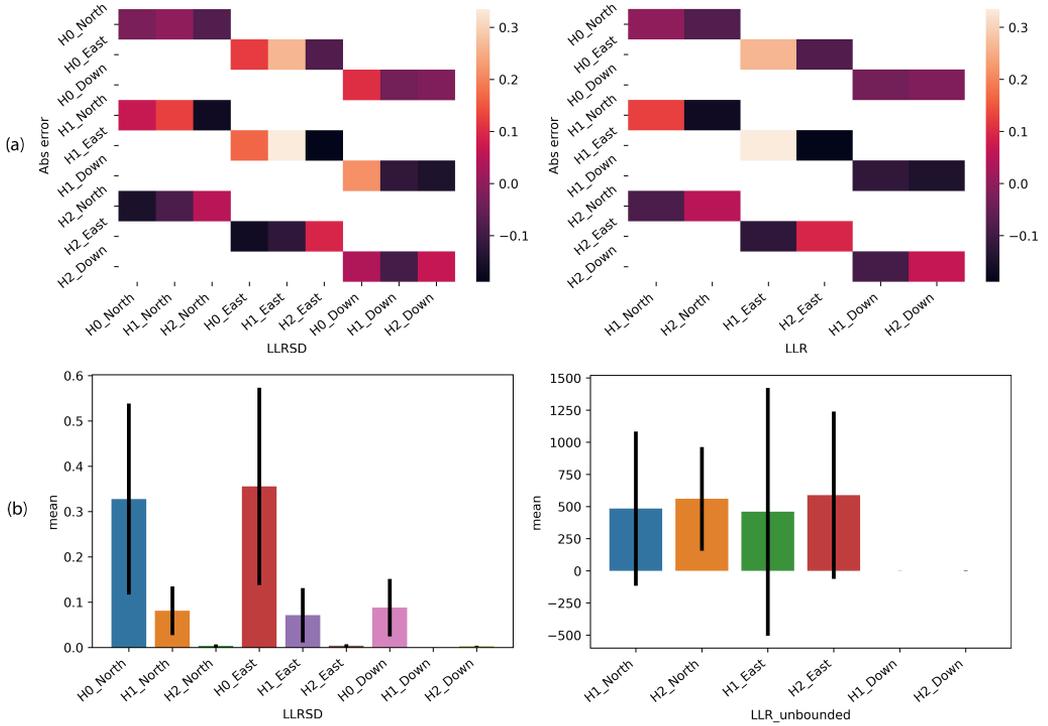


Figure 3.8: Comparison between the LLRSD and LLR metric for the POS_SENSOR1 + POS_SENSOR2 + IMU integration in ErKF for the UAV trajectory. The analysis was done for each hypothesis (denoted with H* in the labels) and each axis (North, East, Down) (a) Pearson correlation coefficient among the LLRSD or LLR metric and the absolute position error. (b) Mean and standard deviation of the LLRSD or the LLR metric in the regions where bounding failed (see Fig. 3.6). Each standard deviation is half the length of the black vertical line depicted on top of the corresponding bar.

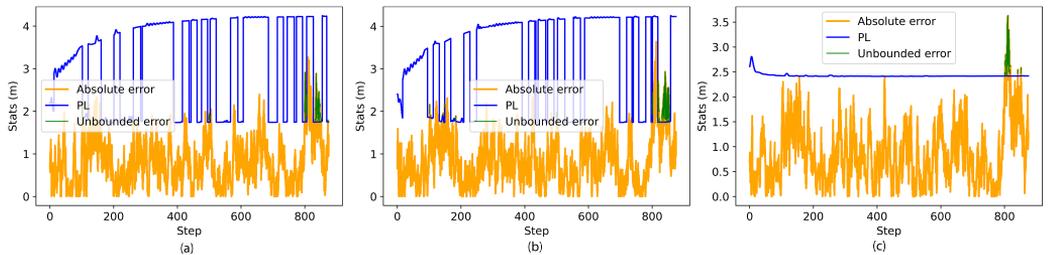


Figure 3.9: IM of the ErKF with conditional overbounding for the UAV trajectory. The PLs and the true errors are shown for the (a) North axis (b) East axis (c) Down axis.

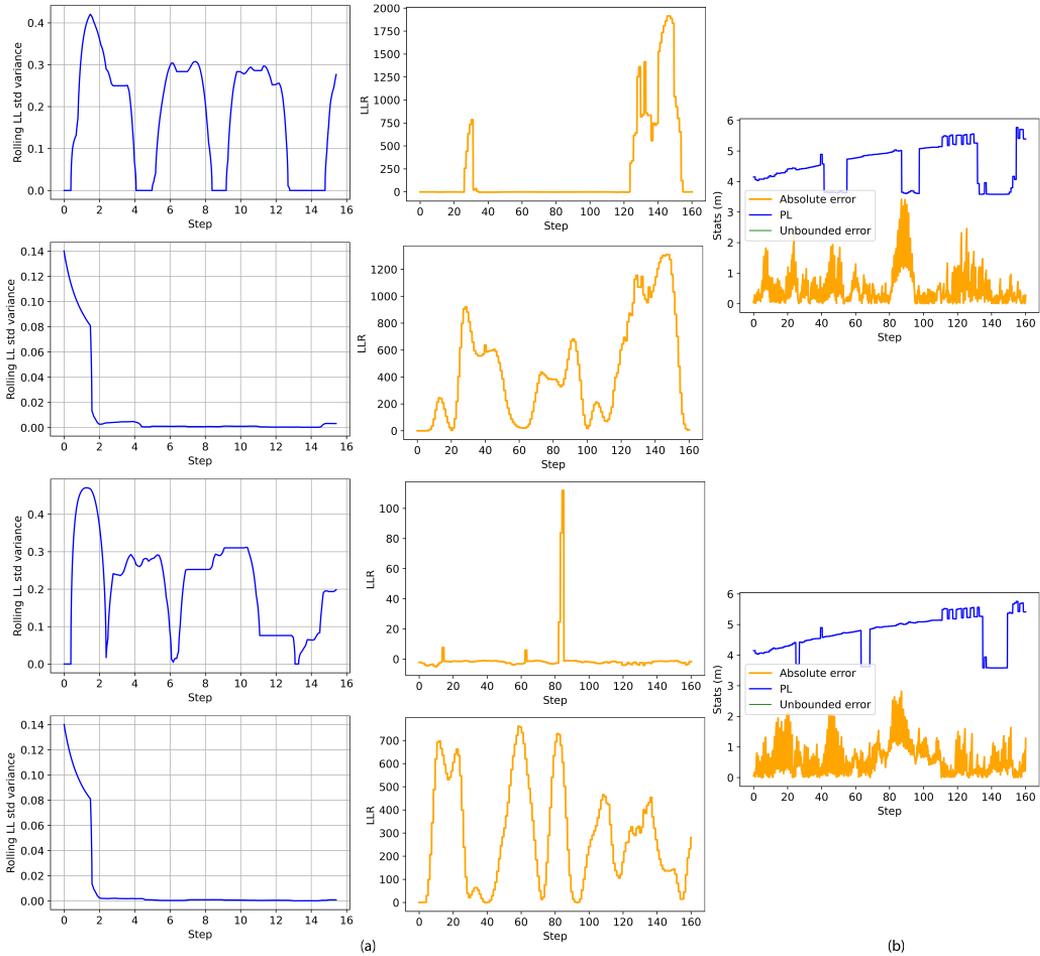


Figure 3.10: Evaluation of IM of the ErKF with overbounding for the KITTI trajectory (a) LLRSD (left column) and LLR (right column). Each row from top to bottom depicts the results for the following hypotheses and axes: Out-POS_SENSOR1 & North axis, Out-POS_SENSOR2 & North axis, Out-POS_SENSOR1 & East axis, Out-POS_SENSOR2 & East axis (b) PL and error for the POS_SENSOR1 + POS_SENSOR2 + IMU integration after conditional overbounding for the North (top row) and East (bottom row) axes. The decision for overbounding was only based on the LLRSD metric.

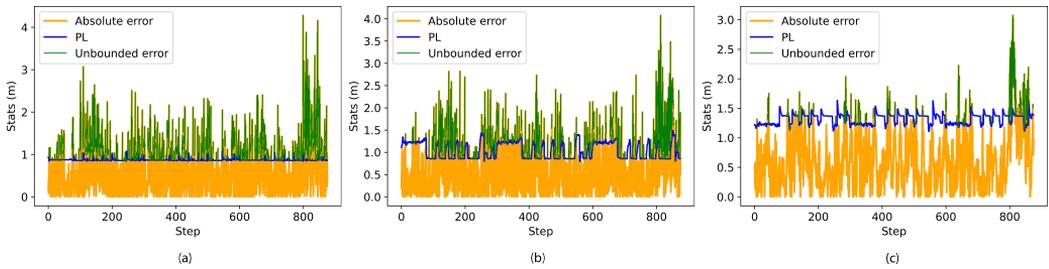


Figure 3.11: PL and errors for the POS_SENSOR1 + POS_SENSOR2 + IMU loose integration with GTSAM for the UAV trajectory (a) North axis. (b) East axis (c) Down axis.

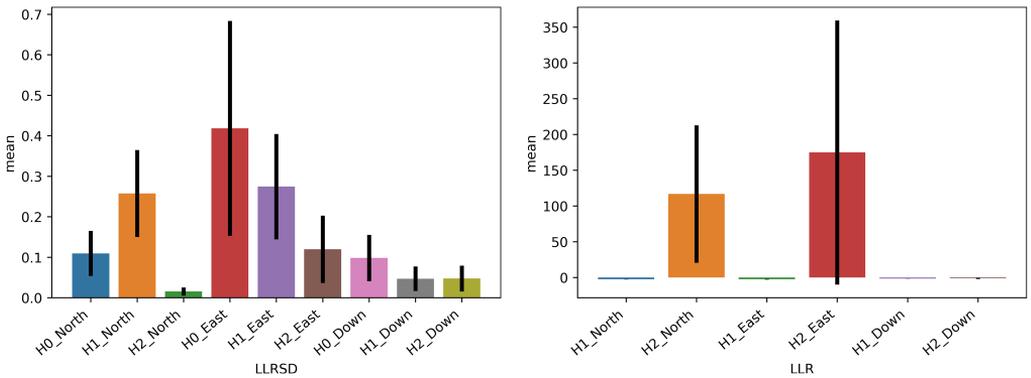


Figure 3.12: LLRS and LLR mean and standard deviation (black lines on the bars) for the POS_SENSOR1 + POS_SENSOR2 + IMU loose integration with GTSAM for the UAV trajectory

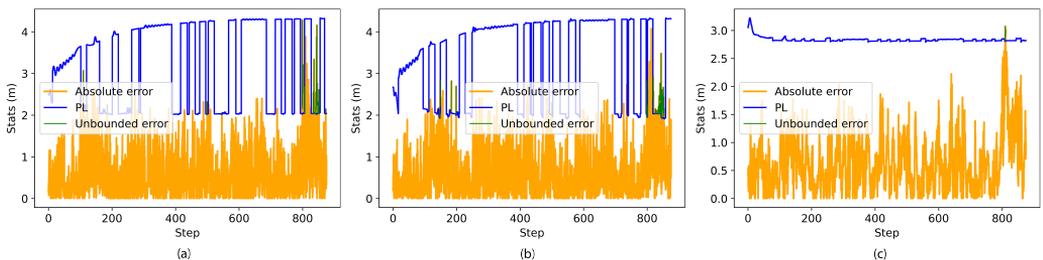


Figure 3.13: PL and errors for the integration of the POS_SENSOR1, POS_SENSOR2 and IMU, in GTSAM for the UAV trajectory. The experiment uses the ErKF for the computation of the bounds in the fault hypotheses' case. The filter underestimates the noise of the POS_SENSOR2. We also use the conditional overbounding technique.

3.6.5 Quantitative Evaluation of the IM Method

Li and Waslander [7] proposed a relaxed tightness metric to quantitatively evaluate the performance of PL in terms of the proportion of time the position error of the navigation solution is sufficiently bounded and the tightness of those bounds. They noted that in nonlinear systems with outlier measurements, it is not feasible to guarantee that the error will always be adequately bounded by a given PL. They proposed a novel Relaxed Bound Tightness (RBT) metric which is calculated as follows:

$$Z = \sqrt{\frac{\sum_{i=1}^N \rho \left(\frac{p_{L_i} - |e_i|}{\sigma_i} \right)^2}{N}}, \quad (3.14)$$

where p_{L_i} and e_i are respectively the PL and the error for a sample time i in one direction, N is number of samples, σ_i is the error covariance for the sample. ρ is a weight function that should penalize bounding failures more than loose bounds. It is given as:

$$\rho = \begin{cases} 1 & \text{if } v_i \geq |e_i| \\ \tau & \text{if } v_i < |e_i| \end{cases} \quad (3.15)$$

To compute meaningful absolute RBT values the constant τ should be selected in a way to minimize Z given an ideal bound v^* . After assuming that the error follows a Gaussian distribution without outliers, the latter is calculated as the quantile function for a Gaussian distribution, depending on the predefined minimum probability of error bounding. However, in this study, τ takes a value in the range between 1^2 and 60^2 to run the evaluation and compare the IM performance for different sensor integration schemes and navigation algorithms. The selection of this range of values is intended to compare the metric values when equal weight is given to bounding failures and loose bounds, or a large to very large penalty is applied to bounding failures. The maximum value of the range and the specific penalty values were not selected based on specific criteria. Table 3.1 presents the results along with execution times.

Overbounding is justifiable in safety-critical applications where bounding the error over long periods is a strict requirement, which corresponds to higher τ . In contrast, unconditional overbounding methods may be considered a hurdle when tighter PLs are needed (lower τ). In addition, one should consider that overbounding leads to longer execution times when IM is run sequentially, mainly due to the evaluation of more hypotheses. In the absence of bounding failures, the value of τ does not affect the RBT value, as is confirmed from the results. Finally, Figure 3.14 plots the RBT values for the simulated UAV trajectory for a few selected τ values in the range $[1, 60^2]$. The figure shows that additional criterium for the decision to utilize overbounding or not should be the algorithm that is monitored. Overbounding becomes justifiable

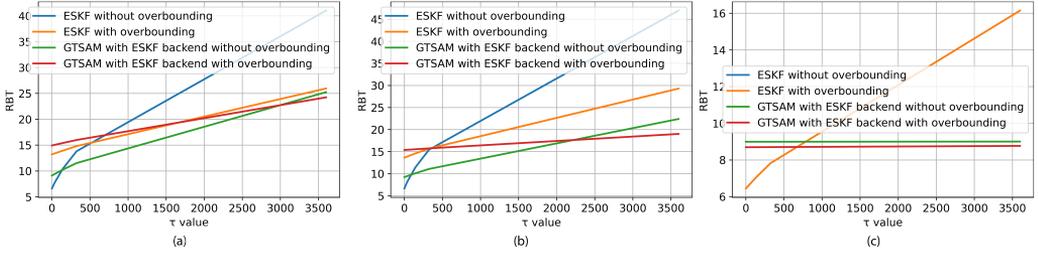


Figure 3.14: RBT values of examined IM algorithms, for various values of the unbounded error penalty constant τ , for the UAV simulation trajectory. (a) North axis (b) East axis (c) Down axis.

already for a value of $\tau = 500$ for the ErKF case, whereas, for the GTSAM with ErKF backend, it is not justifiable until a value of $\tau = 2000$. This is because the position error was generally lower in the GTSAM case. To generalize, the results highlight that overbounding methods can be avoided in IM if the tuning of the noise parameters in the navigation algorithm has been done optimally (i.e. the position error is low) or if the noise parameters have been set conservatively (i.e. PLs will be already loose).

Table 3.1: Comparison of RBT values for two trajectories with various algorithms or IM setups

Trajectory	ErKF		GTSAM		GTSAM + ErKF hypotheses	
	w/o FDO	with FDO	w/o FDO	with FDO	w/o FDO	with FDO
UAV	Maximum RBT for $\tau = 12^2$					
	11.4	+27.8%	+349%	+349%	-10%	+37.3%
	Maximum RBT for $\tau = 60^2$					
	46.96	-37.6%	+445%	+445%	-46.2%	-50%
Full estimation + IM Avg execution time per step (s)						
	0.1183	0.1483	0.2264	0.2718	0.1551	0.1482
KITTI	Maximum RBT for $\tau = 12^2$					
	15.73	+12.7%	+71%	+71%	+22%	+31.3%
	Maximum RBT for $\tau = 60^2$					
	15.73	+12.7%	+755%	+755%	+22%	+31.3%
Full estimation + IM Avg execution time per step (s)						
	0.07	0.075	0.089	0.117	0.074	0.087

3.7 Chapter Summary

This chapter focuses on designing an IM solution for multi-sensor navigation systems operating in the position domain, using a MHSS approach. The IM system assumes an entire sensor measurement set is faulty under each hypothesis and works directly with the output position states and uncertainties from the underlying navigation algorithm. Due to the computational demands of some monitored navigation algorithms and the variability in how they provide covariances, the chapter proposes including a standard and low-complexity filter in the IM as backend, specifically the ErKF, for evaluating all but the fault-free hypotheses.

The IM is not dependent on the internal implementation of the navigation algorithm. The challenge of having independent execution of the IM lies in the potentially unreliable uncertainties provided by the navigation algorithm, which may not be detected by consistency checks. Although IM of solutions that run collaboratively with the navigation algorithm are generally more efficient, the chapter proposes that measures based on measurements' log likelihood can provide a good understanding of sensor measurement uncertainties. This is especially helpful when the navigation algorithm has only a very approximate knowledge of the measurement noise models.

The proposed IM method includes also an overbounding technique which is applied conditionally, based on the rolling standard deviation of log-likelihoods. Experimentation is done by synthesizing a measurement that includes time correlated noise, something that is unknown in the navigation algorithm. Instead, the algorithm has some knowledge of broad ranges of values where the noise parameters lie. The results demonstrate the effectiveness of log-likelihood based measurements in providing sufficient indications of underestimated uncertainties. Theoretical results from previous literature on overbounding the error of KF estimates in presence of correlated sensor noise were applied here with the ErKF. The results are promising: PLs that are bounding increasing errors and that maintain tightness during periods of low error.

As the experiments confirmed, the proposed method's architecture achieves the goal of universality in terms of navigation algorithm's internal architecture as well as computational complexity, while it facilitates extendibility to a variety of sensors with challenging noise models. A limitation of the method is that the included sensors should provide independent position solutions which is not true in many occasions, and, therefore, this remains a topic for future research.

The developed framework, working with a limited number of hypotheses on sensor-level, serves as a strong foundation for further IM development in the current thesis, particularly in handling the integration of IMU/GNSS/Camera in subsequent chapters.

Chapter 4

An IM designed for scalability to auxiliary sensors

4.1 Introduction

Main motivation for an integrated system of conventional sensors, e.g. IMU/GNSS, and a camera, is to exploit the complimentary capabilities of the sensors, i.e. availability of the system in conditions where one or more of the sensors fail. Tall structures might be affecting the quality of a GNSS signal (total denial, multipath), mostly featureless environments or light reflections might affect negatively the quality of feature extraction and association in camera images, while un-aided inertial sensors accumulate drift. Despite the factual improvements in the navigation accuracy due to fusion of complimentary sensors [1], previous research has not investigated thoroughly the conditions of nominal behavior of those sensors.

The probability of sensor failure in the GNSS/IMU/Camera integration is significant, potentially surpassing acceptable risk levels in safety-critical operations. GNSS satellite failure probability has been found to reach magnitudes of 10^{-3} [55], without accounting for failures arising from signal environment or user segment equipment issues. Additionally, accounting for the state error due to undetected fault in IMU measurements in the PL computation is also crucial, with the fault probability of IMU ranging in 10^{-3} to 10^{-5} per hour [56].

Most MHSS-based IMs for GNSS/IMU coupled integrations account only for GNSS faults [56]. Looking into Visual-Inertial navigation systems, IMU preintegration [57, 58] plays a pivotal role and faults in IMU measurements should be accounted for.

The inclusion of a camera increases substantially the number of sources of faults. Among the most critical error sources in VN is the occurrence of incorrect feature associations, with Zhu et al. ([59]) concluding to a probability on the order of 10^{-1} for

this type of fault, depending on the distance to and the complexity of the landmark. More broadly, a few works, such as [60] and [61], have made efforts to categorize the various faults related to VN, assess their impact, and explore methods for mitigation. In addition, the camera introduces challenges in the definition of a nominal error model (error distribution) during feature extraction as that model depends on the environmental conditions while capturing images (e.g. different illumination conditions), the VN method (direct vs feature based) and the specifics of the utilized feature detector [62]. Therefore, a method for fault detection in the new sensor set is essential. Also, typical RAIM methods are not well suited to account for the non-standard sensor noise and extended set of undetected faults for PL computation.

This chapter includes the description of the IMU and VN models in the multi-sensor integration, the respective residuals, and the PL inflation terms for IMU and VN. Subsequent chapters will delve into the specifics of VN anomaly detection and camera FDE. The proposed architecture addresses several limitations found in previous works regarding multi-sensor navigation IM. All modules build upon the foundation laid out by the IM system to ensure reliable navigation in complex urban environments and the evaluation is based on real-world car drives. An overview of the evaluation datasets throughout the thesis is provided in this chapter. The experimental results for the developed IM architecture are presented collectively in the end of the thesis, in Chapter 7.

This chapter is primarily based on Paper 4, while also leveraging parts from Papers 2 and 3.

4.2 Motivation for a sensor-tailored approach to IM of multi-sensor integrations

This section provides a concise overview of previous research addressing camera measurement faults and their impact on output images and VN. It also reviews relative IM approaches, both in the specific context of VN and in dealing with numerous sensor fault modes, in general. The proposed IM architecture addresses several limitations found in previous works regarding multi-sensor navigation IM.

The review of integrity methods for VN from Zhu et al. [62] separated the class of methods that try to guarantee integrity by exploiting visual measurements in very specific navigation problems from those that consider the integrity of the VN itself. The former class includes methods like the one investigating integrity during lane tracking [63] or exploitation of synthetic visual-based range measurements to aid GNSS IM during aircraft landing [64]. Nonetheless, more related to the work in this thesis is the latter class of methods; Fu et al. [65] studied the integrity of stereo VO systems under various conditions (e.g. environmental), implementing limit checks to identify landmark match errors. They treated a 3D landmark matching pair as the visual measurement, with the difference between pair members as residual, overbounded in a similar manner to

ARAIM. They utilized innovation-like metrics by comparing a transformed representation of the local scene based on state estimates with the measured image. Gupta and Gao [66] proposed a method for computing PLs using vehicular state estimates from camera images and a 3D environmental map obtained with a LiDAR. The process is to construct local depth maps for multiple candidate states, comparing them with the measured image to derive error distributions. Finally, some methods in the literature employ outlier detection techniques for the feature point measurements in images and a similar to MHSS framework for clustered feature points (e.g. the method from Wang et al. [67]).

Zhu et al. [68] delved into the challenges posed by the substantial number of fault modes in multi-sensor solutions, which can also depend on the operational environment. Their method builds upon a wide array of previous greedy search based FDE methods. However, their emphasis lies in evaluating the integrity performance after potentially removing certain measurements.

Given an observation matrix, which stacks Jacobian matrices from all participating sensors, it builds a residual based test statistic and a test threshold based on all measurements. If the test fails, the method proceeds by excluding the measurements contributing the most to the test statistic. The process iterates with a new threshold for the remaining measurements. The authors of [68] acknowledged some limitations in their method; Firstly, the utilized greedy FDE still needs to search over all fault modes and, secondly, large variations in the magnitude of elements within the observation matrix, stemming from different sensors, can lead to insufficient redundancy for effective fault detection.

The literature highlights the challenges in robustly guaranteeing safety when employing auxiliary sensors, primarily due to their non-standard noise characteristics, susceptibility to various sources of faults, and inherent non-linearities. Most of the methods are developed to address specific sensor faults or demand explicit fault descriptions. Some of the methods are developed to work with very specific sensors or sensor combinations, relying on the low level measurements provided by those sensors. However, such approaches can still hinder applicability when dealing with a wide array of auxiliary sensors. It should prove beneficial to develop a method capable of accounting for faulty sensor measurements without necessitating exhaustive examination of all potential fault hypotheses. This method should also identify sensor faults independently of redundancy with other sensors and reduce reliance on navigation algorithm prerequisites, such as uncertainty estimation, or knowledge of sensor noise characteristics.

This chapter proposes a method for protection bound computation appropriate for integrations of auxiliary sensors with conventional sensors, such as GNSS and IMU. The method incorporates simple integrations of the participating sensors in a multi-level IM approach; It quantifies the effect that the various measurement subsets, corresponding to the various fault modes, have on the integrity risk of standardized and simpler, than

the monitored, navigation algorithms.

The method emphasizes the separation of the IM procedure for each auxiliary sensor from other auxiliary sensors, facilitating the determination of safety guarantees specifically-targeted to the sensor. Eventually, integrity risk bounds derived for simpler integrations are corrected appropriately to correspond to conservative bounds for the position errors of the original monitored algorithm. The correction procedure involves an offline sensitivity analysis, focusing on how changes in the outputs of the simplified integration impact the outputs of the monitored algorithm. Both outputs exist in the position space, simplifying the analysis. The sensitivity analysis must be conducted once for a predefined set of trajectories and input measurement perturbations for any navigation algorithm to be monitored. Two examples of this sensitivity analysis described in this chapter are for perturbations in the IMU and in the camera measurements which will be described in sections 4.5.1 and 4.5.2 respectively.

The experiments in chapter 7 validate the proposed PL computation procedures for a GNSS/IMU/VN integration with factor graphs as an illustrative navigation method. Notably, the IM does not impose any assumptions about the internal architecture of the navigation algorithm, making it applicable to various algorithms. Briefly, the advantages of the proposed IM architecture are:

- It eliminates the need to model specific sensor faults, focusing on their direct impact on navigation.
- It enables the discovery of hazardous operating conditions for auxiliary sensors through a dedicated risk assessment module for each sensor, along with evaluations of dedicated standardized navigation algorithms relative to the sensor.
- Based on the above, it also offers applicability for various auxiliary sensors.

4.3 Hypothesis tree

Figure 4.1 shows the multi-level approach as a hypothesis tree, in the general case of integrating an IMU with multiple position and auxiliary sensors. The monitored navigation algorithm runs as normal for the full measurement set. The Conventional Sensor Set (CSS) includes sensors that provide a measurement of the position (e.g. GNSS) and inertial sensors (e.g. IMU). The integrity of this integration can be verified via a typical MHSS method, validating the hypotheses $H_{OUT_POS0}, H_{OUT_POS1}, \dots$ of one position sensor failing at a time. A bank of ErKFs is considered to be appropriate for the integrations of the measurement subsets, to find inconsistencies among the various position estimates. This chapter does not include a methodology for the CSS IM, as it is a well examined topic [69], and one methodology has been proposed in chapter 3. On the right side of the tree, the method validates the hypothesis H_{CC} , which considers as fault-free the camera and the CSS. The integrity of this integration

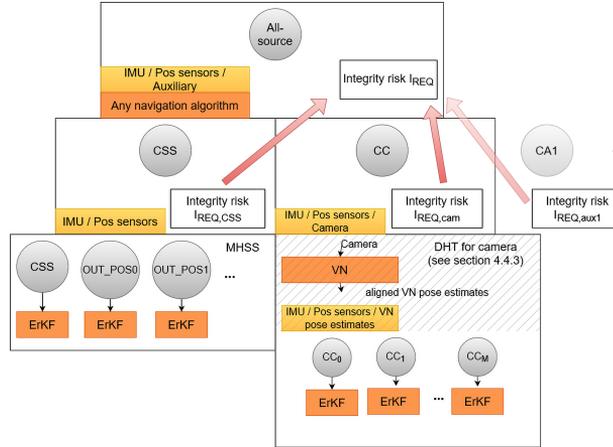


Figure 4.1: Hypothesis tree.

can be validated in the DHT module, which is described analytically in section 4.4.3 of this chapter and in section 6.3 in the chapter that focuses on camera FDE. In DHT for camera, a typical VN algorithm, e.g. VIO, can estimate poses of the camera sensor. An MHSS approach will validate the consistency of VN poses with CSS measurements, in hypotheses $H_{CC_0}, H_{CC_1}, \dots, H_{CC_M}$ that assume subsets of the VN poses, in a limited window of M samples, as faulty. A bank of ErKFs is appropriate for this integration too and to quantify inconsistencies in the outputs. The red arrows signify that the integrity risk bounds computed on each level to account for faults in the simpler integrations are corrected to correspond to conservative bounds in the monitored navigation algorithm. Section 4.5.2 describes analytically the procedure for correcting the integrity risk bound under H_{CC} . The figure shows also, with lower opacity, that the tree can be extended to additional auxiliary sensors, where the validation happens for the measurement subsets CSS & Auxiliary sensor 1, CSS & Auxiliary sensor 2 etc. under corresponding hypotheses H_{CA1}, H_{CA2}, \dots . Finally, the method accounts for probable faults in IMU preintegration with a residual-based method, that is not part of the hypothesis tree. Section 4.5.1 describes that method.

4.4 Conventional and visual measurement integration

For the GNSS/IMU/VN integration, the proposed method investigates the four "on sensor level" fault modes; FF , $GNSS - F$, $IMU - F$ and $VN - F$. Here, FF stands for the Fault-Free mode, while the names of the other three modes include a sensor identifier and the letter F, indicating that the underlying sensor is faulty. The current work, for simplicity, does not consider cases where more than one sensors fail at the same time and assumes that the presented fault modes are independent. The section introduces first the sensor measurements and the coordinate frames. Then, it discusses the measurement models within multi-sensor integrations, demonstrated in yellow

rectangles in figure 4.1, specifically for the three sensors of interest; GNSS, IMU and camera .

4.4.1 Notation and definitions

The origin of the body-fixed coordinate frame (b) is on the IMU sensor, and the x-axis points towards the forward direction, the y-axis to the right of the vehicle, and the z-axis towards the ground. Depending on the vehicle setup in different datasets, a transformation of the data to this specific axis convention (Forward Right Down) has to be carried out.

The world coordinate frame (w) for the vehicle (local world frame) is fixed at the starting location, and has the NED axis convention. The GNSS position measurements, as well as all position estimates from the navigation algorithms evaluated here, are expressed in this frame.

The camera coordinate frame (c) has its origin on the camera sensor. Commonly, camera pose estimates from a VN algorithm are relative to the first camera frame (c_0), which is aligned with the camera position and orientation at the beginning of the image sequence. A transformation of these estimates to the world frame (w) is necessary, to incorporate them into the bank of ErKFs within the "DHT for camera" module.

In the notation used for variables in all equations in this chapter, left-side subscripts within parentheses represent the reference frame, while a combination of subscript and superscript within parentheses signifies a transformation between two coordinate frames. A superscript within parentheses on the right side indicates that the variable is defined under a specific hypothesis. Subscripts on the right side of variables specify the sensor type (if applicable) associated with that variable and the step (e.g., of measurement or estimation) at which the variable is defined.

Given the rotation ${}^{(w)}\mathbf{R}_k^{(b)}$ of the frame (b) to the frame (w) at each step k , the models for the linear acceleration and the angular velocity of the IMU follow below:

$${}^{(b)}\tilde{\mathbf{a}}_k = {}^{(w)}\mathbf{R}_k^T {}^{(b)}\mathbf{a}_k - {}^{(w)}\mathbf{g} + \mathbf{b}_{a_k} + \mathbf{n}_a \quad (4.1)$$

$${}^{(b)}\tilde{\boldsymbol{\omega}}_k = {}^{(b)}\boldsymbol{\omega}_k + \mathbf{b}_{w_k} + \mathbf{n}_w \quad (4.2)$$

The terms \mathbf{b}_{a_k} and \mathbf{b}_{w_k} are, respectively, the random biases of the linear accelerations and angular rates, while the terms \mathbf{n}_a and \mathbf{n}_w are white Gaussian noise vectors, and ${}^{(w)}\mathbf{g}$ is the gravity vector.

Let ${}^{(w)}\mathbf{x}_C = \begin{bmatrix} {}^{(w)}\mathbf{p}_C \\ {}^{(w)}\boldsymbol{\phi}_C \end{bmatrix}$ be the camera pose relative to (w), where ${}^{(w)}\mathbf{p}_C$ is the camera position, and ${}^{(w)}\boldsymbol{\phi}_C$ is the camera attitude vector. The measurement of an extracted

feature point on the image plane is given as:

$$\mathbf{z}_l = \pi_l({}_{(w)}\mathbf{x}_C) + \mathbf{v}_l + \mathbf{b}_l, \quad (4.3)$$

where $\pi_l(\mathbf{x}_C)$ is the 2D projection of the feature l 's world coordinates on the image plane, \mathbf{v}_l is zero-mean Gaussian noise, and \mathbf{b}_l is a bias in the feature location. It is worth noting that the variable \mathbf{x}_C is in the equation because the projection depends on ${}_{(w)}\mathbf{p}_C$ and the rotation matrix between camera body frame and global frame.

4.4.2 IMU measurement model in multi-sensor integrations

It is possible to utilize IMU measurements, between any two camera image acquisition times, to propagate position, velocity and orientation in the world frame at time j . This propagation is achieved by combining these measurements into a single relative motion constraint, with this integration procedure occurring in a local frame. The key advantage is an enhancement in the efficiency of optimization-based systems, as it avoids the need for re-propagating the states, even when there are changes in the linearization point. The procedure is known as IMU preintegration.

Assuming that the IMU measurement biases at time i are known, the compound IMU measurements, expressed as relative motion constraints between the two times, are written in the equation system (4.4) [58]. It is worth noting that from now on the text simplifies the notation by omitting the coordinate frame scripts. The symbol \mathbf{R} refers to the rotation matrix from (b) to (w) , all IMU measurements are expressed in (b) , while gravity, position and velocity vectors are expressed in (w) .

$$\begin{aligned} \Delta\tilde{\mathbf{R}}_{ij} &= \Delta\mathbf{R}_{ij} \text{Exp}(\delta\phi_{ij}) \\ &= \left[\prod_{k=i}^{j-1} \text{Exp}((\tilde{\omega}'_k - \mathbf{b}_{w,k})\Delta t) \right] \text{Exp}(\delta\phi_{ij}) \\ \Delta\tilde{\mathbf{v}}_{ij} &= \Delta\mathbf{v}_{ij} + \delta\mathbf{v}_{ij} \\ &= \left[\sum_{k=i}^{j-1} \Delta\mathbf{R}_{ik}(\tilde{\mathbf{a}}'_k - \mathbf{b}_{a,k})\Delta t \right] + \delta\mathbf{v}_{ij} \\ \Delta\tilde{\mathbf{p}}_{ij} &= \Delta\mathbf{p}_{ij} + \delta\mathbf{p}_{ij} \\ &= \left[\sum_{k=i}^{j-1} \frac{1}{2} \Delta\mathbf{R}_{ik}(\tilde{\mathbf{a}}'_k - \mathbf{b}_{a,k})\Delta t^2 \right] + \delta\mathbf{p}_{ij}, \end{aligned} \quad (4.4)$$

for the relative rotation of the body frame from time i to time j , $\Delta\mathbf{R}_{ij} = \mathbf{R}_i^T \mathbf{R}_j$, and the position and velocity evolution, $\Delta\mathbf{p}_{ij}$ and $\Delta\mathbf{v}_{ij}$. Additionally, $\Delta\mathbf{R}_{ik} = \mathbf{R}_i^T \mathbf{R}_k$. The integer $k = i, \dots, j-1$ is the time step of IMU measurements. The formula assumes white noise in individual IMU measurements, which is incorporated in the terms $\delta\phi_{ij}$,

$\delta \mathbf{v}_{ij}$ and $\delta \mathbf{p}_{ij}$. This separation of the terms on the right hand side of (4.4) is possible after isolating mathematically the noise terms \mathbf{n}_a and \mathbf{n}_w from $\tilde{\mathbf{a}}_k$ and $\tilde{\boldsymbol{\omega}}_k$ in equations (4.1). The two biased but non-noisy IMU measurements are indicated as $\tilde{\mathbf{a}}'_k$ and $\tilde{\boldsymbol{\omega}}'_k$. Finally, the term Δt denotes the time interval between two inertial measurements.

Importantly, the above equations rely solely on IMU measurements between times i and j and not on the position and velocity at time i , which would indicate reliance on the linearization point at time i .

4.4.3 MHSS for camera frames

This section discusses the VN model, and the relation of navigation faults to the full GNSS/IMU/VN integration. The faults are associated with sequences of camera frames, instead of individual frames.

For each image, the camera pose estimate is determined with respect to the reference frame (c_0), that is, w.r.t. the first camera frame. A typical VN system will utilize a technique for feature extraction and matching to features in earlier images. Visual odometry tracks matched features through sequences of frames, estimating the displacement of the camera. The factors that can deteriorate the feature extraction and matching procedures are, among others, ambiguities in the scene, lack of texture, bad illumination and occlusions. The big number of sources of possible faults would require to consider hypotheses from a complex space. The first step in the method is, therefore, to simplify the hypothesis space of faults by relating them with the drifts in camera pose estimates that they might cause in VN algorithms.

The configurable parameter M , shown in figure 4.1, dictates both the number of VN poses assessed for faults (as accumulated drifts within the window of VN poses) and the frequency of the test, i.e., the method conducts the test every M 'th image. In each time step $i \in [1, M]$ inside the considered horizon, let L_i be the total number of feature correspondences obtained until that step in the horizon. The feature correspondences are represented by their projections on each image. The complete vector of stacked feature projections at step i will be $\mathbf{Z}_i = \begin{bmatrix} \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_{L_i} \end{bmatrix}$, or by grouping them in their i corresponding

images:

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_1 \\ \vdots \\ \mathbf{Z}_i \end{bmatrix}, \quad (4.5)$$

where $\mathbf{Z}_{i'}$, $i' = 1, \dots, i$ is a vector with feature point correspondences found in image obtained at time i' .

In the general case, the VN algorithm seeks a solution to the optimization problem $\hat{\mathbf{x}}_C = \arg \min_{\mathbf{x}} \|\mathbf{z} - \pi(\mathbf{x})\|_{\Sigma_{\mathbf{u}}}^2$, where $\hat{\mathbf{x}}_C = \begin{bmatrix} \hat{\mathbf{p}}_C \\ \hat{\phi}_C \end{bmatrix}$ is the camera pose estimate at the

current step, which stacks the position $\hat{\mathbf{p}}_C$ and attitude $\hat{\phi}_C$ vectors, $\pi(\mathbf{x}) = \begin{bmatrix} \pi_1(\mathbf{x}) \\ \vdots \\ \pi_L(\mathbf{x}) \end{bmatrix}$, and

$\Sigma_{\mathbf{u}}$ is the covariance matrix of the stacked noise vector $\mathbf{u} = \begin{bmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_L \end{bmatrix}$. The number L is the number of feature correspondences used in the optimization problem and is something determined internally in the navigation algorithm. Importantly, that number does not affect the formulations.

The fault vector affecting the pose solution at step i in the current horizon will be determined by instantaneous or slowly growing faults (drifts), as well as pose corrections, due to past feature correspondences. Without loss of generality, we limit the analysis at the part of the fault that is due to the past L_i feature correspondences and construct the corresponding fault hypotheses.

Let $\hat{\mathbf{X}}_{C,M}$ represent the VN pose estimates assessed for faults in the current horizon of size M . The method employs $M + 1$ hypotheses, with each assuming that the subset $\mathbf{X}^{(CC,i)} \subset \mathbf{X}_{C,M}$, $i = 0, 1, \dots, M$ of VN poses remains fault-free. In this context, "fault-free" signifies that camera measurements within the current horizon do not introduce new erroneous drifts in the poses, specifically rapid drifts due to faulty image feature extraction or association. However, these poses may still be affected by prior drifts or faults that occurred prior to the current horizon. The rule for selecting VN poses to include in $\mathbf{X}^{(CC,i)}$ is the following:

$$\mathbf{X}^{(CC,i)} = \{\hat{\mathbf{x}}_{C,i+1-M}, \dots, \hat{\mathbf{x}}_{C,i}\} \quad (4.6)$$

For example, under $H_{CC,0}$ all VN poses are excluded (as $0 + i \leq M$ for any $i = 1, \dots, M$). Under $H_{CC,1}$ only the M -th VN pose is assumed fault-free, while under $H_{CC,M}$ all VN poses from index 1 to M are included. The remaining text identifies the hypothesis $H_{CC,M}$ as the all-source hypothesis in DHT, and uses the simpler notation H_{CC} .

4.4.4 ErKF in integrations of VN with conventional sensors

The previous section defined a hypothesis set of faulty VN pose estimates, where these faults appear predominantly as drifts spanning M previous estimates. This section describes the method to test the hypotheses. The test is feasible by performing standard loosely coupled integration of VN poses with conventional sensors in an ErKF. This approach maintains computational simplicity. Notably, testing these hypotheses using the actual monitored algorithm, without knowing the specifics of the GNSS/IMU/VN

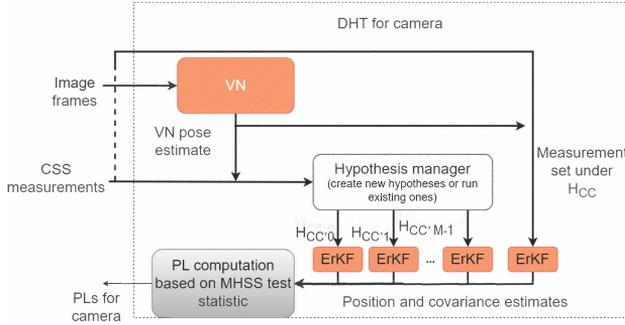


Figure 4.2: The DHT for camera.

integration, might be impossible.

The DHT module employs a bank of ErKFs to test the hypotheses $H_{CC,\iota}$, $\iota = 1, 2, \dots, M + 1$, integrating the M VN pose estimates with measurements from CSS, obtained in the current horizon. The outputs from the bank of ErKFs are the position solutions and estimated uncertainties under each hypothesis. The IM will use only the outputs at the end of the horizon, as section 4.5.2 will describe. Figure 4.2 depicts the bank of ErKF filters used in the DHT module. The measurement model assumes Gaussian white noise $\mathbf{n}_C \sim \mathcal{N}(0, \mathbf{R}_C)$ in the VN pose fixes, with \mathbf{R}_C being a time-invariant covariance matrix, and includes a time-varying fault vector \mathbf{f}_C :

$$\zeta_C = h_C(\boldsymbol{\chi}) + \mathbf{f}_C + \mathbf{n}_C \quad (4.7)$$

The above formulations omitted explicit notation for each step. Similarly, a GNSS fix is modeled as:

$$\zeta_P = h_P(\boldsymbol{\chi}) + \mathbf{f}_P + \mathbf{n}_P, \quad (4.8)$$

where \mathbf{n}_P is a noise factor in the GNSS position measurements, which is assumed to be zero-mean Gaussian white noise, and \mathbf{f}_P is a time-varying fault vector in the GNSS position measurements. The $\boldsymbol{\chi}$ denotes the state at each time, a vector which stacks the position \mathbf{p} , velocity \mathbf{v} , attitude quaternion \mathbf{q} , acceleration bias \mathbf{b}_a and angular velocity bias \mathbf{b}_w vectors:

$$\boldsymbol{\chi} = (\mathbf{p}, \mathbf{v}, \mathbf{q}, \mathbf{b}_a, \mathbf{b}_w) \quad (4.9)$$

The non-linear functions h_C and h_P are mapping the state to a VN pose or GNSS measurement, respectively.

A fault-tolerant ErKF under hypothesis $H_{CC,\iota}$ ignores in the integration the VN pose estimates that are excluded under that hypothesis. At the start step $i = 0$ of each measurement horizon, comprising M VN poses, all state and covariance estimates across filters are reset to align with the filter operating under H_{CC} :

$$\begin{aligned}\hat{\boldsymbol{\chi}}_{C,\kappa-M}^{(CC,t)} &= \hat{\boldsymbol{\chi}}_{C,\kappa-M} \\ \hat{\mathbf{P}}_{C,\kappa-M}^{(CC,t)} &= \hat{\mathbf{P}}_{C,\kappa-M},\end{aligned}\quad (4.10)$$

$\forall t \in [1, M], \forall \kappa \in \mathbf{K}$, where \mathbf{K} denotes the set of camera image indices when the method performs the test. The reset occurs regardless of any fault detections in the previous horizon.

Fundamentally, the ErKF propagates the error quantities $\delta \mathbf{p}, \delta \mathbf{v}, \delta \boldsymbol{\theta}, \delta \mathbf{b}_a$ and $\delta \mathbf{b}_w$ instead of estimating their actual values. $\delta \boldsymbol{\theta}$ denotes the error in the attitude in an angle-axis representation [70]. As the equations (4.7) and (4.8) relate the measurements with the true states, an additional step is to relate the true state with the error state:

$$\boldsymbol{\chi} = \begin{bmatrix} \hat{\mathbf{p}} + \delta \hat{\mathbf{p}} \\ \hat{\mathbf{v}} + \delta \hat{\mathbf{v}} \\ \hat{\mathbf{q}} \otimes \begin{bmatrix} 1 \\ \frac{1}{2} \delta \hat{\boldsymbol{\theta}} \end{bmatrix} \\ \hat{\mathbf{b}}_a + \delta \hat{\mathbf{b}}_a \\ \hat{\mathbf{b}}_w + \delta \hat{\mathbf{b}}_w \end{bmatrix}, \quad (4.11)$$

where \otimes is the quaternion product. Now, the linearization for the VN and GNSS measurement models is done about the error state $\delta \hat{\boldsymbol{\chi}}^{(CC,t)} = 0$ under each hypothesis. This concludes the formulation of the measurement models and standard integration of conventional and visual measurements in the ErKF.

4.5 Integrity Solution for integrations of conventional and auxiliary sensors

Section 4.5.1 formulates the protection bound computation to account for IMU pre-integration faults. Section 4.5.2 formulates the protection bound computation for camera-only navigation faults, based on the VN model. Lastly, section 4.5.3 concludes the methodology with the overall diagram, which indicates also the scalability of the method to auxiliary sensors other than the camera.

The equations in the subsequent sections will utilize the check accent (\checkmark) to indicate variables associated with the actual monitored navigation algorithm, such as the monitored algorithm's state estimates.

4.5.1 PLs for IMU pre-integrated measurements

Formulating the residual of IMU preintegrated measurements, in relation to its expectation, and finding its contribution to the undetected position state estimate error under the all-source hypothesis, is essential in computing PLs that bound that part of the state estimate error. Let the index λ identify the steps of camera image acquisition. The predicted position estimate, in the body frame, under the all-source hypothesis at

time $j = t_{\lambda+1}$ of acquisition of an image frame, based on the estimate at the time of acquisition of the previous image frame, $i = t_{\lambda}$, is:

$$\check{\mathbf{R}}_i^T \check{\mathbf{p}}_j = \check{\mathbf{R}}_i^T (\check{\mathbf{p}}_i + \check{\mathbf{v}}_i t_{ij} - \frac{1}{2} \mathbf{g} t_{ij}^2) \quad (4.12)$$

where $t_{ij} = t_{\lambda+1} - t_{\lambda}$. By rearranging the terms, the predicted position change due to acceleration, between the two times will be:

$$\Delta \check{\mathbf{p}}_{ij} = \check{\mathbf{R}}_i^T (\check{\mathbf{p}}_j - \check{\mathbf{p}}_i - \check{\mathbf{v}}_i t_{ij} + \frac{1}{2} \mathbf{g} t_{ij}^2) \quad (4.13)$$

Therefore, the residual \mathbf{r}_{B_j} of the observed preintegrated IMU measurement relevant to the position state will be:

$$\begin{aligned} \mathbf{r}_{B_j} &= \Delta \check{\mathbf{p}}_{ij} - \Delta \tilde{\mathbf{p}}_{ij} \\ &= \check{\mathbf{R}}_i^{-1} (\check{\mathbf{p}}_j - \check{\mathbf{p}}_i - \check{\mathbf{v}}_i t_{ij} + \frac{1}{2} \mathbf{g} t_{ij}^2) - \Delta \tilde{\mathbf{p}}_{ij} \end{aligned} \quad (4.14)$$

The residual, \mathbf{r}_{B_j} , affects the position state error of the monitored navigation algorithm through the Jacobian \mathbf{J}_{B_j} that relates \mathbf{r}_{B_j} with the position state $\check{\mathbf{p}}$, at any estimation step. The method to define the Jacobian \mathbf{J}_{B_j} resembles an offline sensitivity analysis and may entail using outputs from a Monte Carlo simulation. This simulation introduces symmetrical perturbations to IMU measurements, and Algorithm 1 describes the perturbation method for one input in the simulation.

Algorithm 1 Computing perturbation for preintegrated IMU residual

Given: IMU measurements $\mathbf{y}_{B_k} = [a_1, a_2, a_3, \omega_1, \omega_2, \omega_3]_k, \forall k$

foreach *interval*[i, j] **between two frames** i and j

select randomly a perturbation δy_{B_j} for linear acceleration measurements, where the maximum value of the perturbation is limited to one standard deviation of the nominal IMU noise.

foreach IMU measurement \mathbf{y}_{B_k} obtained in interval [i, j] do

 perturbed $\mathbf{y}_{B_k}^+ \leftarrow \mathbf{y}_{B_k} + [\delta y_{B_j}, \delta y_{B_j}, \delta y_{B_j} \mathbf{0}_{1 \times 3}]$

 perturbed $\mathbf{y}_{B_k}^- \leftarrow \mathbf{y}_{B_k} - [\delta y_{B_j}, \delta y_{B_j}, \delta y_{B_j} \mathbf{0}_{1 \times 3}]$

Then rerun the monitored navigation algorithms using as input the perturbed IMU measurements $\mathbf{y}_{B_k}^+$ and $\mathbf{y}_{B_k}^-$, $\forall k$. Recompute the states $\check{\mathbf{p}}_k$ and $\check{\mathbf{v}}_k$ and predicted motions $\Delta \check{\mathbf{p}}'_{ij}$

 Recompute the IMU preintegration terms $\Delta \check{\mathbf{p}}'_{ij}$, as in (4.4)

 Recompute the residuals as in equation (4.14)

The reader should pay attention to the alternation of subscripts k and j in all equations. An IMU state and a PL are available at every estimation step k . However, preintegration and Jacobian computation occur at the time j when an image frame is acquired, considering the IMU measurements gathered between the previous image at step i and the current image at step j . Likewise, a preintegration fault is only available at step j , and not at each step k .

The following equation gives the position state residual:

$$\mathbf{J}_{B_i} \delta \check{\mathbf{p}}_{B_k} = \mathbf{r}_{B_j} \quad (4.15)$$

Now, position state errors $\check{\mathbf{f}}_{p_{B_k}}$, might be caused by IMU preintegration nominal noise \mathbf{n}_{B_j} or faults \mathbf{f}_{B_j} :

$$\mathbf{J}_{B_j} (\delta \check{\mathbf{p}}_{B_k} + \check{\mathbf{f}}_{p_{B_k}}) = \mathbf{r}_{B_j} + \mathbf{n}_{B_j} + \mathbf{f}_{B_j} \quad (4.16)$$

Consider one of the axes in the local world frame (w) (i.e., North or East axis for motion on a 2D plane) and denote it as d . The test statistic τ_{d,B_j} for the position state along the direction of the axis d , affected by the q -th element in the fault vector \mathbf{f}_{B_j} , represents the detected portion of that fault. This test statistic is given by:

$$\tau_{d,B_j} = \sigma'_{q,r_{B_j}}, \quad (4.17)$$

with $\sigma'_{q,r_{B_j}}$ denoting a value that is inversely proportional to the standard deviation of the q -th residual element. This is obtained as the square root of the (q, q) element in the inverse of the covariance matrix of the residual \mathbf{r}_{B_j} .

The position state estimation error due to the faults follows from equation (4.16):

$$\check{\mathbf{f}}_{p_{B_k}} = \mathbf{J}_{B_j}^+ \mathbf{f}_{B_j} \quad (4.18)$$

This equation utilizes the pseudoinverse $\mathbf{J}_{B_j}^+ = (\mathbf{J}_{B_j}^T \mathbf{J}_{B_j})^{-1} \mathbf{J}_{B_j}^T$ of \mathbf{J}_{B_j} which exists always and is unique.

The part of the position state residual, due to IMU preintegration faults, that remains undetected is given as a ratio of the position state estimate error (due to the faults) and the test statistic. This ratio defines the characteristic fault slope, as introduced in [37]. An assumption is that the maximum faults in the position states precisely correspond to the maximum faults in \mathbf{r}_{B_j} . This means that the maximum fault directions coincide with the axes in (b_k) , and explicit checks for the direction of maximum faults are obviated. In one direction, and after replacing (4.17) and (4.18), the factors f_{q,B_j} cancel each other:

$$Slope_{d,B_k} = \frac{\check{f}_{d,p_{B_k}}}{f_{q,B_j} \cdot \tau_{d,B_j}} = \frac{[J_{B_j}^+]_{dq}}{\sigma'_{q,r_{B_j}}}, \quad (4.19)$$

where the element (d, q) of $J_{B_j}^+$ is symbolized as $[J_{B_j}^+]_{dq}$.

The PL computation incorporates the resulting fault slope in (4.19), factoring in the continuity risk requirement associated with preintegrated IMU measurements.

$$PL_{d,B_k} = Slope_{d,B_k} \cdot Q^{-1}\{C_{REQ,B}\}, \quad (4.20)$$

with Q^{-1} being the inverse tail probability of the standard normal distribution and $C_{REQ,B}$ being the allocated continuity risk requirement. Thus, the term $Q^{-1}\{C_{REQ,B}\}$ in the above equation serves to constrain the PL, considering the probability of false alarms of faults in preintegrated measurements under fault-free conditions.

4.5.2 PLs for VN faults

Camera faults might result in instantaneous or accumulating faults in the VN pose trajectory. The position state estimate $\mathbf{p}_{CC,\kappa}$ under hypothesis H_{CC} , obtained after a sequence of M VN poses, will include an unknown fault with vector \mathbf{f}_{C_κ} . The subscript κ denotes that the fault is visible at step κ , which is the time of acquisition of the last camera measurement in the sequence to test. The PL computation cannot exclusively account for individual camera image or VN pose faults in the sequence, as these types of faults are inherently correlated to each other due to their continuous and interdependent nature. Rather, according to the hypothesis space defined in section 4.4.3, the PL computation will account for the uncertainty in the resulting state (at step κ) introduced by the overall inconsistency in the input VN pose sequence. This inconsistency is quantifiable by comparing the states under each DHT hypothesis.

Running the ErKF estimation under each DHT hypothesis H_{CC_i} , as is visualized in figure 4.2, will result in a position state error $\mathbf{f}_{C_\kappa}^{(CC,i)}$. The camera faults are resulting in position state estimate errors $\check{\mathbf{f}}_{p_{C_\kappa}}^{(CC,i)}$ also for the actual monitored navigation algorithm, and this error can be approximated via $\mathbf{f}_{C_\kappa}^{(CC,i)}$ as:

$$\check{\mathbf{f}}_{p_{C_\kappa}}^{(CC,i)} = J_{C_\kappa}^+ \mathbf{f}_{C_\kappa}^{(CC,i)}, \quad (4.21)$$

with $J_{C_\kappa}^+$ being the pseudoinverse of the Jacobian that relates the ErKF's and monitored navigation algorithm's position state. The sensitivity analysis for Jacobian determination may involve various perturbations to camera measurements, as well as potentially to other participating sensors. Algorithm 2 outlines the computation of perturbations

in monitored and ErKF states for the example case of perturbing individual camera measurements with motion blur. More sophisticated procedures for rendering motion blur using the actual trajectory [71] are beyond the scope of this thesis.

Algorithm 2 Perturbing states in the monitored navigation algorithm and ErKF due to artificial motion blur.

Given: Vehicle drive with CSS measurements and camera images in vector form, $\xi_\lambda \in \mathbb{R}^m$, where m is the image dimension and λ is the image index.

Select and normalize the convolution kernel for simulating motion blur in the environment, E.g., a 3×3 horizontal kernel will be $\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$.

foreach λ **do**

 Compute the perturbed image ξ_λ^+ as the convolution of ξ_λ with the kernel h :

$$\xi_\lambda^+(x, y) = \sum_u \sum_v h(u, v) \xi_\lambda(x - u, y - v),$$

 where x and y denote the pixel coordinates in the image, and u, v are the coordinates in the kernel.

Then rerun the monitored navigation algorithm with perturbed camera measurements ξ_λ^+ , $\forall \lambda$.

 Recompute the VN pose states $\mathbf{y}_{C_\lambda}^+$ for the perturbed camera measurements .

 Rerun the ErKF under H_{CC} with perturbed VN poses $\mathbf{y}_{C_\lambda}^+$ while keeping the CSS unchanged.

The test statistic for detectable faults takes into account the ErKF position solutions $\mathbf{p}_{C_k}^{(CC_i)}$ under the DHT hypotheses. Significant disparities in the estimated position posteriors among different solutions indicate potential uncertainty in the position estimates under hypothesis H_{CC} . Similarly to the theory in typical MHSS, define the residual in DHT as:

$$\mathbf{r}_{C_k}^{(CC_i)} = |\mathbf{p}_{C_k}^{(CC_i)} - \mathbf{p}_{C_k}| \quad (4.22)$$

where $\mathbf{p}_{C_k}^{(CC_i)}$ and \mathbf{p}_{C_k} are the estimated ErKF position states under H_{CC_i} and H_{CC} , respectively.

Then, the test statistic corresponding to the ErKF position state along an axis d will be:

$$\tau_{d, C_k}^{(CC_i)} = \frac{1}{\sigma_{d, r_{C_k}}^{(\Delta CC_i)}} \quad (4.23)$$

where $\sigma_{d,r_{C_k}}^{(\Delta CC_i)} = \sqrt{\sigma_{d,r_{C_k}}^{(CC_i)^2} - \sigma_{d,r_{C_k}}^2}$.

For the derivation of the maximum fault slope, it is sufficient once again to assume that the maximum position fault occurs in the same direction as the axes of the NED frame, which is the reference for the ErKF position estimates. Therefore:

$$Slope_{d,C_k}^{(CC,i)} = \frac{[J_{C_k}^+]_{dd} \cdot f_{d,C_k}^{(CC,i)}}{f_{d,C_k}^{(CC,i)} \cdot \tau_{d,C_k}^{(CC,i)}} = [J_{C_k}^+]_{dd} \cdot \sigma_{d,r_{C_k}}^{(\Delta CC_i)} \quad (4.24)$$

Finally, the PL factor that accounts for camera faults will be:

$$PL_{d,C_k}^{(CC,i)} = Slope_{d,C_k}^{(CC,i)} \cdot Q^{-1}\{C_{REQ,C}^{(CC,i)}\}, \quad (4.25)$$

with $C_{REQ,C}$ being the continuity risk requirement associated with faults in the VN pose sequences.

4.5.3 Overview of the complete IM

Figure 4.3 shows a diagram of the proposed IM, making clear the inputs to each module, as well as emphasizing that the only expected output from the monitored navigation algorithm is the vehicle position estimate. Furthermore, the diagram suggests the potential incorporation of a prediction module for auxiliary faults and the generation of fault alerts. These parts are indicated with lower opacity. As mentioned before, dedicated modules for the camera, concerning VN anomaly detection and camera FDE, will be elaborated upon in subsequent chapters. The diagram also implies a conditional inflation process if these alerts are utilized.

4.6 Datasets

The evaluation of various methods in this thesis relies on datasets collected from car drives in urban environments, each posing several challenges. These datasets comprise camera images, GNSS readings, and IMU measurements, facilitating the development and validation of different algorithms and techniques.

4.6.1 UrbanLoco Dataset

The UrbanLoco dataset [72] offers rich sensor data captured during car drives in dense urban environments. The data utilized in this thesis originates from a car navigating through a city in California. The dataset is distributed with GT positions from a SPAN-CPT module that integrates a GNSS and an INS. It encompasses scenarios with dynamic objects, illumination changes, and repetitive patterns. It comprises sensor modalities such as stereo cameras, GNSS, and IMUs.

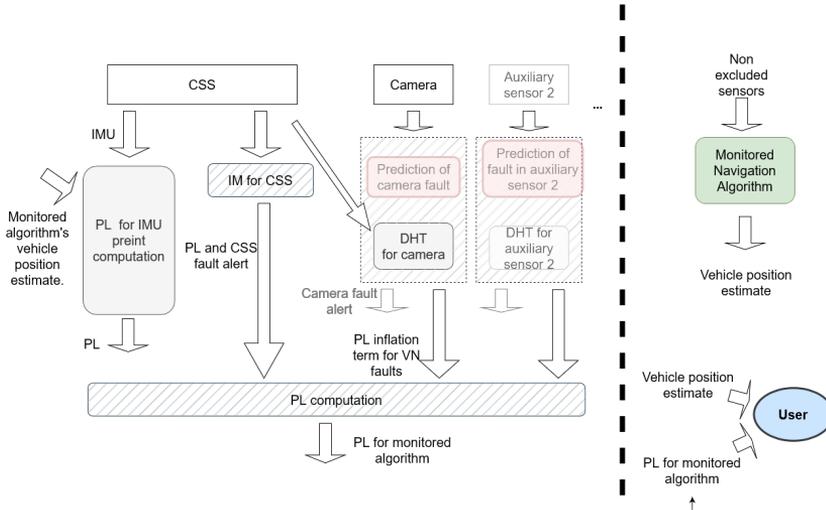


Figure 4.3: Diagram for the complete IM.

4.6.2 Robotcar Dataset

The Robotcar dataset [73] provides sensor data recorded by a car driving in Oxford, UK, over the course of a year. This dataset captures various weather conditions and road scenarios, allowing for robust testing of algorithms under different environmental factors. It includes sensor data from cameras, GNSS, and IMUs. Lastly, it provides a ground truth generated using post-processed raw GPS, IMU, and static GNSS base station recordings [74].

4.6.3 KAIST Dataset

The KAIST dataset [75] features sensor data collected from car drives in urban areas, including trajectories with complex traffic patterns and illumination variations. It includes sensor data from stereo cameras, GNSS and IMU. The available data is from multiple cities and countries, although the evaluations in the thesis use only data captured in South Korea. KAIST provides reference data from a pose-graph SLAM solution that fuses various sensors. Although we use the provided reference data for all trajectories, the accuracy might vary and be significantly deteriorated under conditions with GNSS unavailability [75]. In addition, an analysis of the GT revealed that the reference does not cover some utilized trajectories completely, and, in these cases, it was necessary to fill the gaps by the estimates from a simple integration of GNSS/IMU in ErKF. Therefore, the reference for trajectories in this dataset is inaccurate in some parts.

4.6.4 4seasons Dataset

The 4seasons dataset [76] comprises sensor data recorded from car drives inside Munich, Germany, capturing instances of driving within the old town area. This dataset offers multiple real-world runs along the same route, allowing exploration of diverse weather and illumination conditions. It includes sensor inputs of cameras, GNSS, and IMUs.

The utilization of these datasets in the evaluation of different methods throughout this thesis provides an assessment of algorithm performance across diverse urban driving scenarios. Each method leverages the unique characteristics of these datasets to address specific challenges and achieve robustness in real-world applications.

4.7 Chapter Summary

In addressing the challenges of safety assurance in integrated navigation systems, the study in this chapter introduces a method that redefines hazard assessment in the presence of auxiliary sensors. Traditional approaches often struggle with dealing with multiple sensor faults and scalability limitations. However, the proposed method simplifies this by calculating protection boundaries for simpler and standardized sensor integrations, and methodically using these boundaries to inflate the bounds for the full integration. The introduced multi-level IM architecture eliminates the necessity of modeling specific sensor faults. Instead, it focuses the attention on their direct influence on standardized navigation algorithms linked to each auxiliary sensor. Moreover, it streamlines the identification of hazardous operating conditions related to each sensor, which is a significant step towards more efficient testing of autonomous vehicles under specific environmental conditions.

The experimentation, presented in chapter 7, will demonstrate the method's applicability to GNSS/IMU/Vision integrations with factor graphs. However, the proposed IM architecture is agnostic towards the internal architecture of the monitored navigation algorithm. Furthermore, the experimentation will demonstrate the method's adaptability to diverse faults by quantifying their impact on integrity risk.

Chapter 5

Integration of conventional sensors with a camera and faults: A DNN approach for navigation anomaly detection

5.1 Introduction

Estimating global positions from visual measurements entails many more sources of error in comparison to estimating from GNSS/IMU measurements, for example raw image noise or feature extraction and association errors [77]. Therefore, bounding the true position error due to each sensor in the GNSS/IMU/Camera integration is challenging. The extension of the MHSS framework to navigation systems that are based on auxiliary sensors (e.g. camera, LiDAR) is also challenging due to the additional computational complexity, since in a MHSS framework a computationally heavy navigation algorithm has to be executed multiple times at each step.

Another requirement to guarantee reliable safety monitoring is the identification of the rare cases that cause a navigation system to fail, in order to conduct extensive testing of the framework under those conditions. The requirements of real universality of IM and identification of environment conditions that cause sensor faults or degradation, without the need of efforteous manual selection of relevant features, inspired the work presented in this chapter. The focus is on camera-based navigation, although the method can be extended to any sensor.

SLAM is a very popular framework when it comes to VN. Consequently, there is a variety of SLAM methods, and the development of a safety monitor that can be integrated with a majority of these methods may be impossible. Methods that allow

the uncertainty estimation in SLAM have been examined in section 2.3. However, many existing SLAM implementations do not readily provide these uncertainty estimates. In some cases, uncertainty estimation would require modifications in the internals of those implementations, or, at least, to obtain some internal matrices to do the computations.

Previous methods in the literature that utilize covariance prediction attempted to relate features of the sensor input to covariance matrices and managed to overcome, in a large degree, the above limitations. An example is the method from Hu and Kantor [78] who predicted the variation of covariance of a Gaussian error distribution. Later contributions that use offline training, and the features are directly derived from the input measurements, are promising to predict the exact error covariance, based solely on the input measurements, without the requirement for manual identification of relevant features (e.g. [79] and [80]).

A DNN-based anomaly detection approach gives a simpler solution to the problem, as the model can be trained to learn when the distribution of estimated position errors should have fatter tails than the normal distribution to reliably compute error bounds. Anomaly detection in a DNN setting attempts to learn a feature representation of the raw inputs in the dataset, in a way that anomalous instances are distinguishable from normal instances. Other methods attempt to learn directly an anomaly score mapping function $\tau(\cdot) : X \rightarrow \mathbb{R}$ [81]. The most popular approach for learning-based anomaly detection is to use autoencoders, where, in an unsupervised manner, the network is trained to reconstruct normal data from their low dimensional representations. The reconstruction error for anomalous data will be very high. The biggest disadvantage of using this approach in the context of this work is that the knowledge of which data is normal, in terms of not causing unbounded errors from a navigation algorithm, is very limited.

Wen and Keyes [82] proposed an anomaly detection method based on Convolutional Neural Networks (CNNs), utilizing transfer learning from a larger dataset, as the occurrence of anomalies is very rare. However, pre-trained models are usually available for specific type of data, whereas the detection of anomalies that can affect negatively a safety monitor is a problem that lacks similar data. Conventional neural network methods tend also to neglect past information, which makes them inappropriate for learning long-term dependencies among sensor measurements that cause anomalies. DNNs are more appropriate than previous methods to capture temporal dependencies as well as correlations between observations in the data. Long Short Term Memory Networks (LSTMs) are designed to model short-term as well as long-term data dependencies by controlling the addition and forgetting mechanisms of new and old information [83]. LSTMs have found large utilization in recent studies on anomaly detection. The largest focus was to detect abnormalities in sensor measurement time-series extending the framework of autoencoders, as in [84] and [85], or in the time-series of a specific IM test statistic (e.g. [86]). Literature on directly associating navigation faults with anomalies in single-sensor readings is still lacking, whereas a few works have used neural network

architectures for prediction of failures, based on sets of measurements and probable actions undertaken by the system (e.g. [87]). Wyk et al. [88] tackled the problem of identifying anomalous sensor readings during automated vehicle navigation, via combined CNN and KF-based anomaly detection. However, a central assumption is redundancy in sensor measurements and that a KF is applicable with the sensor input at hand. An interesting conclusion of their experiments is the superiority of CNNs in comparison to a combination of Recurrent Neural Networks (RNNs) with LSTMs, when there are normal values between consecutive anomalous values. However, they did not evaluate the combination of CNNs with LSTM, as is used in the current work.

In comparison to previous research, the presented approach in this chapter achieves clearer quantification of anomalies associated with input images during SLAM navigation, based on the error of position estimates to a reference trajectory.

An open research topic is dealing with the problem of labeling measurement anomalies or outliers in a time-series without misclassifying inliers. The most relevant approaches to this problem utilize hypothesis testing. An example is the work of Tong and Barfoot [89] where their statistical testing approach deals also with the problem of misclassified inliers in a sequence of error samples that fail the test. Nonetheless, in the work in the present thesis, the start and end of anomalies is identified with a simpler statistical method. The automated detection of additional unseen anomalies in currently unlabeled data ([90]) remains a subject for future investigation.

The main contributions and potential benefits of the work presented herein can be summarized as follows:

1. Early anomaly prediction in the position errors of camera-based navigation, taking into account low-level image features and presence of dynamic objects.
2. The algorithm can classify one or more subsequent outliers and image features that are likely the real origin of SLAM failures.
3. The method can benefit existing sensor- and navigation algorithm agnostic IM systems by alarming for unbounded covariance. Although the DNN was developed for visual input, the same logic can be applied for any type of sensor.
4. The developed network expects raw sensor inputs and classifies the output of a SLAM algorithm. Therefore, the robustness of various SLAM algorithms can be evaluated objectively with the same network and under the challenging conditions present in a standard dataset.

Section 5.2 describes the offline methodology to label the training dataset (5.2.1), the utilized DNN model (5.2.2) and the bias initialization procedure to cope with the class imbalance problem (5.2.3). Section 5.3 outlines the datasets used to evaluate the model's performance (5.3.1), the result of the statistical anomaly labeling for the creation

of training data (5.3.2) and the performance of the model on the training, validation and test sets (5.3.3). Section 5.4 concludes the chapter.

The chapter is based on the published paper 2.

5.2 DNN modeling for anomaly prediction in visual-based navigation

In the following, a raw measurement sample refers to M_{vis} sequential images. Let K be a set of pairs of measurement samples and associated true errors of positions estimated by a SLAM algorithm, at each time step i . The errors are assumed to follow a normal distribution $\mathcal{N}(0, \mathbf{R}_i)$ in the nominal case, with \mathbf{R}_i the covariance matrix at step i . Then, the objective is to optimize for the parameters of a DNN to predict that specific features in input images will cause an error to fall outside the distribution \mathcal{N} . For example, a sample $\xi_i \dots \xi_{i+M_{vis}}$ starting at image i contains the images $\xi_k \in \mathbb{R}^m, k = i, i+1, \dots, i+M_{vis}$ stacked in a vector, with m being the number of pixels in the image (or a down-scaled version of it). Then a set of low dimensional features are obtained, which are vectorized such that for each image $f(\xi_k) \in \mathbb{R}^r, r \ll m$. Let \mathbf{W} be a weight matrix and \mathbf{b} a bias which can shift the neural network activation function to the left or right and is also learned by the neural network. Then the hypothesis for a sample i will be:

$$h_i(\xi_i, \dots, \xi_{i+M_{vis}}) = g(\mathbf{W} \cdot \Lambda(f(\xi_i), \dots, f(\xi_{i+M_{vis}})) + \mathbf{b}) \quad (5.1)$$

The formula abstracts the specifics of the hidden layers that are applied between the input and output layer. Λ is a function that combines features of all images in the set and finds the relationship between each other. g was selected to be the Sigmoid function, as the desired range of values for the binary classification problem is $[0, 1]$. Section 5.2.2 will give an idea of the layers that compute function Λ . It is attempted to estimate the parameters \mathbf{W} and \mathbf{b} by optimizing the cost function:

$$J(\mathbf{W}, \mathbf{b}) = \frac{1}{K} \sum_{i=1}^K Cost(h_i, y_i), \quad (5.2)$$

with y_i denoting the true label corresponding to sample i . The labeling of normal or anomalous cases had to be done as a preprocessing step, and the procedure is described in section 5.2.1. Section 5.2.2 describes the DNN architecture to learn relevant features, while section 5.2.3 explains the method to initialize realistically the bias of the output due to the imbalance of the two classes.

5.2.1 Isolation of anomalies in VN

The targeted problem is to identify feature levels in images that can cause a sudden increase in the estimated absolute position error of a SLAM algorithm of interest. A GT is therefore required for training the model. In that GT the input images are associated with the label "anomaly - 1" or "normal - 0", depending if they cause an anomaly in the

error distribution of the SLAM estimates. The SLAM algorithm should be executed with the optimal calibration parameters for this purpose. However, sensor faults and degradation may happen very rarely, and, many times, there is only a small effect by challenging conditions of very short duration. Therefore, it is important to predict how the SLAM algorithm would behave if some specific conditions were persistent. The requirement to predict image features that cause suddenly or progressively an anomaly creates a trade-off for the anomaly isolation algorithm.

The algorithm that is adopted in this work to detect anomalies in the error samples is a modification of the peak detection algorithm of Brakel [91]. The detection depends on the z-score of a position error sample, that is, the number of standard deviations that the error sample is above the error mean [92]. Here, a moving average, that is tolerant to outliers, is used as reference. The outliers affect the moving average in a small degree, although a slow adaptation is allowed, assuming that only measurements at the beginning of the peak cause the fault. In case of multidimensional positioning the maximum among all axes errors is selected, and the anomaly labeling is done based on that error.

Let N be the number of error samples which coincides with the number of steps in the estimated camera trajectory. Let e_1, e_2, \dots, e_N denote the sequence of error samples and $e_1^*, e_2^*, \dots, e_N^*$ a sequence with weighted error samples, so that, for each $j \in [1, N]$

$$e_j^* = \begin{cases} e_j, & \text{if } j \text{ an outlier} \\ \alpha e_j + (1 - \alpha)e_{j-1}^*, & \text{otherwise} \end{cases}, \quad (5.3)$$

with α being a parameter that sets the influence of an outlier to the mean. A reasonable value for α for non-stationary signals is in the range of $[0.01, 0.1]$ and is expected to be set as zero for a stationary signal.

Then at each step i and for a configurable horizon of size L samples the mean will be:

$$m_i = avg(e_{i-L+1}^*, \dots, e_i^*) \quad (5.4)$$

and the standard deviation:

$$\sigma_i = std(e_{i-L+1}^*, \dots, e_i^*) \quad (5.5)$$

A sample is flagged as an outlier when its z-score is above a threshold. This threshold can be selected based on the assumption that the error follows a normal distribution and the expected number of anomalous samples. Z-score tables, like the one in [93], can be used to select a threshold based on the anticipated probability of anomaly.

It is seen that the outliers are included slowly in the computation of the moving mean. In that way, the algorithm can classify subsequent outliers until the mean has been adapted to include them. The algorithm will stop detecting after a sequence of outliers

has adapted the mean value. Although this algorithm can prevent some false positives, it is important to force an earlier finish of labeling anomalies.

The method uses a simple method for stopping the labeling of outliers based on the dynamics of the error. Specifically when L errors have been added in a sequence, with the first element corresponding to the latest error, an anomaly is still valid if the error appears to be still increasing. We evaluate if the maximum error is the current one or at least one of the latest ones, to tolerate for noise:

$$\arg \max_{j \in i, \dots, i-L+1} e_j^* < T_p, \quad (5.6)$$

where $T_p \in \mathbb{Z}^+$ a positive integer used as threshold to tolerate that any of the latest T_p errors is the maximum. In this way, the method labels as anomalies only the first samples that lead to a peak in an error plot, while the method without the stopping rule labels all images associated with an error peak.

5.2.2 Deep CNN

To learn the relationship of low-level image features and transient changes in the environment, the DNN architecture of Figure 5.1 was implemented. All input images are downsampled using the OpenCV library [94] to low-dimensional images of width W and height H , where the downsampling method is bilinear interpolation.

A Time Distributed Layer (TDL) [95], shown as the large rectangle in Figure 5.1, was selected to compare a set of M sequential images and learn, in this way, the effect of transient changes (e.g. dynamic objects) to the predicted output. In the TDL the same layers are applied to each image to extract relevant features, but one set of optimal parameters is produced for all images in the sample. The extraction of low level features from the images is achieved by combining two convolutional layers with non-linear activation function and two subsequent max-pooling layers. The TDL keeps a 1-1 relation of input image and corresponding output. The LSTM layer is introduced to learn the temporal dependence among observations, e.g. image frames in chronological order [95]. A flattening or pooling operation after the TDL is introduced since the requirement is to have only one dimension per output from the Time Distributed wrapping to insert them to the LSTM layer.

Finally, a fully connected layer of 256 units and a drop-out layer are added in the output.

5.2.3 Output bias initialization

The anomalies are significantly fewer than the normal cases, and this can be an apparent issue if a strict anomaly labeling method is selected. Using a zero-bias can make it difficult to obtain good convergence initially. In contrast, acknowledging the class imbalance, an initial bias b_0 can be obtained based on the probability of a positive class [96]:

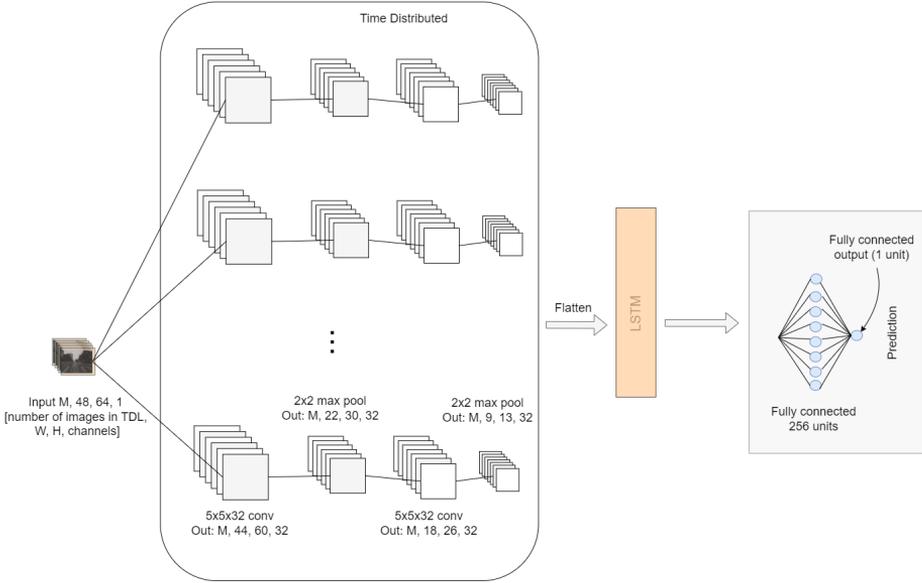


Figure 5.1: DNN diagram. The indicated output dimensions might be inaccurate and they depend on the padding and stride parameters used. Modified architecture from Liu et al. [79] to use TDL. In that paper they used leaky rectify activation as non linear activation function.

$$\begin{aligned}
 p_0 &= pos/(pos + neg) = 1/(1 + e^{-b_0}) \\
 \Rightarrow b_0 &= -\ln\left(\frac{1}{p_0} - 1\right) \\
 \Rightarrow b_0 &= \ln\frac{pos}{neg},
 \end{aligned} \tag{5.7}$$

where pos and neg are respectively the number of positive (anomalous) and negative (normal) examples in the training set.

5.3 Experiments

5.3.1 Training data and appropriability for the problem

The input to the CNN is sequential images and position error pairs $\{\{\xi_i \dots \xi_{i+M_{vis}}\}, e_i | i \in [1, K]\}$. The number of frames in each sample was pre-selected to be $M_{vis} = 7$. The Visual SLAM algorithm ORBSLAM2 [97] is executed for the computation of the estimated camera positions. Then the errors to a reference trajectory can be computed. The training can be done offline, for any SLAM algorithm, by using the same set of data every time. The current dataset for training consists of three trajectories from two different sources. In all cases a car is driving in an urban environment, with illumina-



Figure 5.2: Example images from the experimentation datasets. Two trajectories are evaluated from UrbanLoco and one from KAIST.

tion challenges, repetitive patterns and dynamic motion of pedestrians and other cars. All the data are open-source and the sources are UrbanLoco [72] and Complex Urban Dataset / KAIST [75]. The datasets were introduced in section 4.6.

The images' aspect ratio varies between the datasets. Figure 5.2 shows some example images from both datasets. A frame rate of 10 Hz is used for both datasets, where downsampling is employed if necessary. This chapter evaluates the monocular SLAM case. If a dataset contains data exclusively from a front facing stereo camera, then only images from the left camera are used. Both datasets provide their own extrinsic and intrinsic camera parameters.

Figures 5.3, 5.4 and 5.5 depict the GNSS, GT and estimated camera positions from ORBSLAM2 (labeled as "cam" in the figures) relative to the initial position, in NED coordinates, for the three trajectories. Computation of the optimal transform to align the camera poses with the reference was achieved with the Umeyama method [98]. The gray arrow in the figures shows the starting position, and the red circle the ending position.

Figure 5.6 compares the true absolute errors for the first UrbanLoco trajectory of the position estimates from three navigation solutions; using the camera alone in ORBSLAM2, integrating an IMU and GNSS, or integrating an IMU, a GNSS and the camera position estimates from ORBSLAM2. One can see that the inclusion of camera may deteriorate the navigation performance some times, leading to large position errors. However, many times a camera can complement the IMU and GNSS, showing comparable or better performance than the IMU and GNSS integration. A camera can be very as-

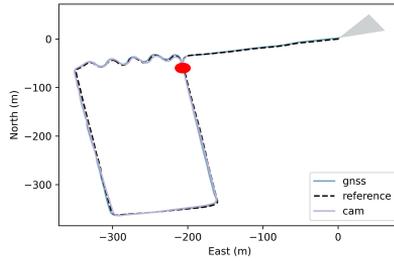


Figure 5.3: The first trajectory from UrbanLoco. GNSS, GT and estimated camera positions from ORBSLAM2.

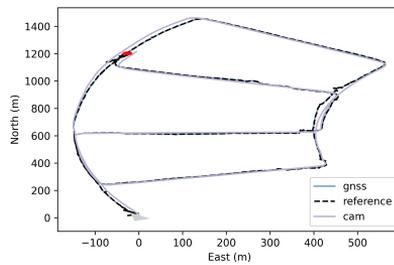


Figure 5.4: The trajectory from KAIST. GNSS, GT and estimated camera positions from ORBSLAM2.

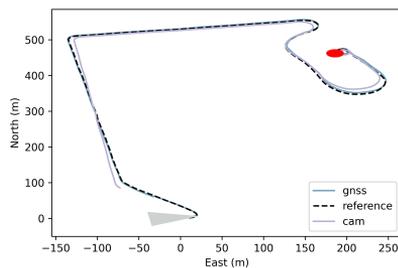


Figure 5.5: The second trajectory from UrbanLoco. GNSS, GT and estimated camera positions from ORBSLAM2.

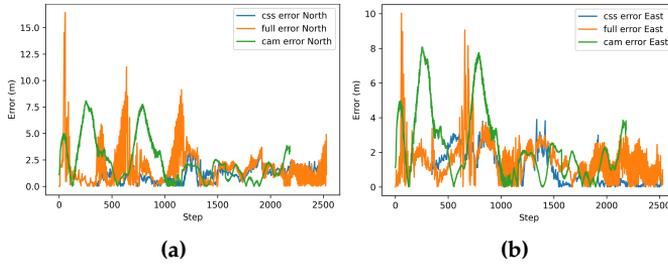


Figure 5.6: Position errors over time obtained for three navigation solutions. "css error" is the error from the integration of GNSS and IMU in ErKF. In the "full error" the ORBSLAM2 camera position estimates are integrated with the IMU and GNSS. "cam error" refers to the error of position estimates from ORBSLAM2. The errors in (a) north and (b) east axis are shown.

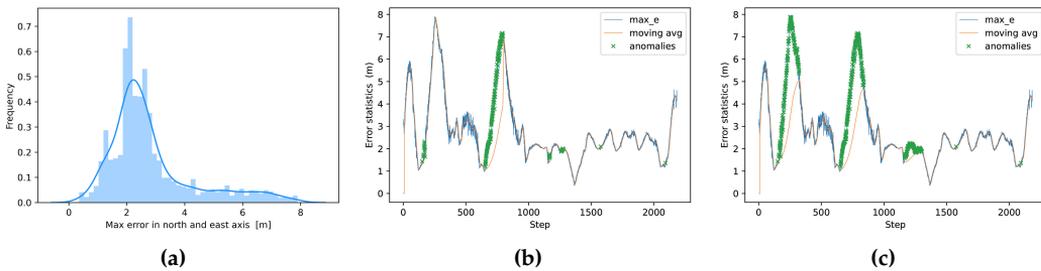


Figure 5.7: For the first UrbanLoco trajectory (a) error distribution of the estimated camera position, (b) marked anomalies with the anomaly ending rule, and (c) marked anomalies without the anomaly ending rule.

sistive in cases of GNSS or IMU unavailability or faults. Therefore, the integration of auxiliary sensors with conventional sensors (e.g. IMU and GNSS) may lead to superior performance, although the detection of anomalous measurements independently from each sensor is an essential prerequisite.

During the experimentation it is expected that the majority of the true position error samples is concentrated in a small region. As the estimated camera position might be prone to errors due to intrinsic calibration or alignment parameter inaccuracies, the distribution is not necessarily concentrated close to zero. Figures 5.7(a) and 5.8(a) show that distribution for the first UrbanLoco and KAIST trajectories respectively. Since the problem is 2-dimensional, the illustrated error is the maximum observed among the North and East axes at each time step.

5.3.2 Anomaly labeling result

The CNN is fed with GT labels that correspond to each image and show if it causes an anomaly or not. The decision to label an image as anomalous or not depends on both individual intuition and the labeling algorithm outlined in section 5.2.1. Figures 5.7(b),

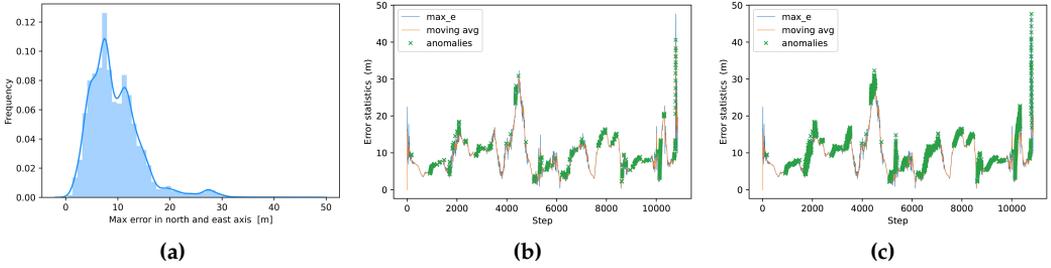


Figure 5.8: For the KAIST trajectory (a) error distribution of the estimated camera position, (b) marked anomalies with the anomaly ending rule, and (c) marked anomalies without the anomaly ending rule.

5.7(c), 5.8(b) and 5.8(c) illustrate for the two trajectories the samples that are labeled as anomalous or normal, with or without the ending rule, together with the error plot and the moving mean. The z-score threshold was selected as 4.5 and influence parameter as $\alpha = 0.01$. The anomaly ending rule is used in all further experiments and the threshold used was $T_p = 4$. Importantly, different parameters can lead to more or fewer labeled anomalies, where, based on the given problem, one might select to be more or less conservative.

5.3.3 Training and evaluation of the network

Each sample can contain subsequent images from the same trajectory. A set of 7400 samples of image sequences from the UrbanLoco and KAIST trajectories (1900 and 1000 samples from the two UrbanLoco trajectories and 4500 samples from the KAIST trajectory) was split randomly in the training (66.6%) and test sets (33.3%), with a fixed seed so that the DNN never sees the test set during training. The samples that appear as anomalies when labeling without the ending rule and as normal when applying the ending rule (see figures 5.7 and 5.8) are not considered for training or validation in the experiments. 30% of the training samples were selected for validation, using the cross-validation method. The size of a batch was selected as 50 samples, and main criterion is that enough positive examples are included, although large batch sizes might cause memory exhaustion. The network is trained for 6000 epochs, binary cross-entropy is used as the loss function and the optimization is accomplished with the Adam optimizer with learning rate 10^{-5} .

Figure 5.9 evaluates the evolution of the loss, recall and precision of the DNN during training and validation. The two latter metrics indicate the presence of false alarms and of missed anomalies, respectively. The DNN generalizes quite well to the validation set, although the significant fluctuations indicate some sensitivity to noise. Despite the continuous improvement of the precision and recall, the validation loss curve seems to flatten after a while. This indicates that although the classifier makes correct predictions, the margin between the calculated class probabilities does not become

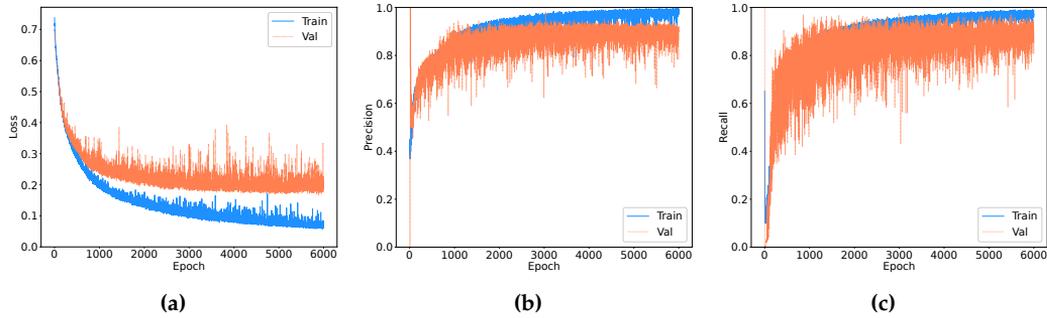


Figure 5.9: Evolution of the (a) loss, (b) precision and (c) recall of the DNN with TDL during training, where, with cross-validation, 1/3 of the training samples are selected for validation.

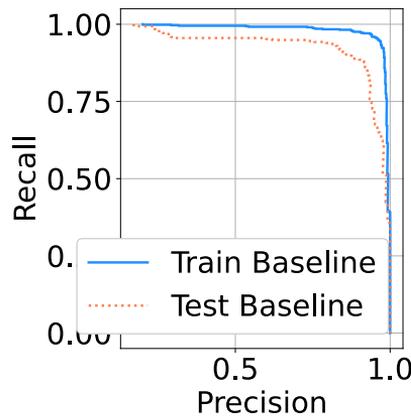


Figure 5.10: AUPRC performance of the network for the train and test sets, where the precision-recall points are obtained using various classification thresholds.

larger. Steps that can improve the method further are denoising, regularization and increased dataset size. Finally, considering the context of the problem, someone might want to improve further from the recall of 0.9 which was achieved until now for the test set.

Figure 5.10 plots the Area Under Precision-Recall Curve (AUPRC). It shows the performance that can be achieved for different values of the classification threshold. In this problem, the presence of False Negatives is usually far more costly than the presence of False Positives, although one would also like to avoid many false alarms that cause an IM to stop the autonomous operation.

For a classification threshold of 0.5, Table 5.1 shows the confusion matrices of the predictions of the model on the test set. In addition, the performance of the CNN with TDL is compared with the performance of a CNN that does not take into account

Table 5.1: Confusion matrix for the predictions of the test samples with and without TDL in the network, where the counts for the latter architecture are shown in parentheses.

		Prediction outcome		total
		p	n	
actual value	p'	283 (203)	31 (111)	314
	n'	42 (20)	1559 (1581)	1601
total		325 (223)	1590 (1692)	

Table 5.2: Performance on the test set for different values of thresholds T_z and T_p . The percentage of labeled anomalies to normal samples is also shown.

T_z	T_p	% anomalies	Precision	Recall
3	5	57	0.92	0.9
4.5	5	23	0.89	0.87
4.5	4	20	0.87	0.9
4.5	3	15	0.79	0.87
5	4	11	0.88	0.91

temporal dependence among frames, i.e. with the TDL removed. It is visible that the CNN performs better in learning anomalies when the TDL is present.

Finally, Table 5.2 compares the performance of the model on the test set for different values of the z-score threshold T_z and anomaly ending tolerance threshold T_p . The performance is similar and remains good in all cases, at least for the recall metric. It is observable that tuning the threshold parameters can lead to more or less distinguishable anomalies. Comparing the third and fifth rows of the table it is observable that labeling using stricter thresholds ($T_z = 5$ and $T_p = 4$) resulted in slightly better model performance than using the originally selected parameter set ($T_z = 4.5$ and $T_p = 4$). However, one might select the thresholds that are not too conservative to avoid mislabeling anomalies as normal, even if the performance of the model is slightly worse

5.4 Chapter Summary

This chapter investigated a deep CNN for associating anomalous increases of the estimation error from a visual SLAM algorithm to low level image features. The CNN is trained in a supervised manner with several image trajectories captured by ground vehicles in dense urban environments, whereas the class labels are specified based on the statistical properties of the position error. A TDL is included to learn temporal dependences among sequential image frames. This allows the investigation of the effect that dynamic objects have on the accuracy of visual-based position estimates, as well as the early prediction of anomalies. The results demonstrated a good performance of the CNN on test data and a tangible improvement in learning anomalies in comparison to a CNN without the TDL. The approach introduced here is a noteworthy addition to VN anomaly prediction. The forthcoming chapter will build upon this methodology to develop the camera FDE approach within the IM framework.

Chapter 6

Deep Hypothesis Testing for fault alarms in auxiliary sensors

6.1 Introduction

The IM approach presented in section 3 relies on estimation of consistent position error covariance due to each sensor in the IMU/GNSS integration in the fault-free case. The covariances of multi-sensor navigation systems, such as GNSS/IMU/Camera, are frequently not readily accessible, often necessitating internal modifications within the navigation algorithm to obtain them. Furthermore, the extension of traditional MHSS-based IMs to the examined multi-sensor systems might be unrealizable due to a large and intractable number of fault sources, resulting in many fault-tolerant hypotheses to evaluate, or in the development of IMs that are undertested under challenging environmental conditions. In fact, the reliability of VN algorithms in non-ideal situations is a topic which is researched very sparsely. One work from Bednář, Petrlík, Vivaldini and Saska [99] concluded that different algorithms perform very differently in real environments depending on the orientation of the camera, the presence of sunlight and the type of vehicle motion. Yet, an extensive set of conditions that characterize camera performance is still not available.

A possible solution to overcome the above mentioned limitations is to limit the hypothesis space of assumed fault-free sensor measurement subsets in MHSS, by evaluating the navigation faults that each auxiliary sensor causes, independently from other sensors. This is achieved with the DHT module, introduced in chapter 4. We refer to a navigation fault or anomaly as the event where the true error of a navigation solution is not drawn from the typical distribution of errors. We denote as IM anomaly the event that the true error of a navigation solution exceeds a calculated PL. IM anomalies are usually happening due to violation of assumptions during PL computation, e.g. the presence of non-linearities, or due to insufficient compensation for every sensor's noise

and faults. The main objective of the presented method is to prevent IM anomalies originating from insufficient compensation due to unknown noise of some sensors or due to the limited fault hypothesis space. The method proves that navigation anomaly prediction, based on individual auxiliary sensor's measurements, can be applied for this purpose.

6.1.1 Prior work

The utilization of deep learning for anomaly detection in multivariate time-series data has shown impressive results in past works [100]. Here, the emphasis is shifted towards its application within IM contexts. Sun [101] utilized multi-layer LSTM to classify an input sequence of GNSS measurements as faulty or fault-free, based on irregularities in the distribution of observations of an IM test statistic. The model is trained with labeled data which consist of raw GNSS measurements and the fault-alarm or fault-free label. Kim and Cho [86] utilized a Time Delayed Neural Network to detect dissimilarities of the current trend of an IM statistic compared to a past trend, where the system was operating under normal conditions. Gupta and Gao [102] utilized deep learning to produce position error and error covariance estimates, that can be used in PL calculation, for a camera + LiDAR navigation system. This is done by comparing static images with a local depth map produced around the current state estimate and given the LiDAR map. A shortcoming of the mentioned research is that they are very specific to the sensors or the IM used. In addition they do not focus on understanding the direct effect that environmental conditions have on navigation and IM anomalies due to sensory faults.

It is worth noting that the majority of existing anomaly detection methods are trained to find the best representation of normal data, requiring the availability of large datasets that are free from anomalies. Methods that try to learn representations of normal data might ignore the important features that differentiate normal from anomalous samples, especially when learned features are redundant, noisy or refer to a specific subset of the training set [103]. Research on supervised learning of anomalies are very few, due to the difficulty of obtaining enough anomaly examples for training, and even in those cases learning of anomalies would be limited to the already labeled anomalies. Here, the presented IM method builds upon chapter 5, where training data was labeled in few trajectories as normal or anomalous using statistical methods.

6.1.2 Contribution

At the start, the method utilizes a supervised Deep CNN model with a TDL, which is trained offline, for the prediction of anomalies in the state estimate error, based on features in the raw sensor input. Then, the method refines the predicted anomalies based on an IM test statistic, which is directly related to the sensor of concern, without the need to consider other auxiliary sensor measurements.

The first main contribution of the presented work is that, despite the extension to

auxiliary sensors, the MHSS module computes the PLs informed solely by conventional sensor measurements as before. In this way the IM scales to auxiliary sensor measurements, without modification of existing IM methods and without important computation burdens. An assumption is that the integration of more sensors in a navigation system results in smaller estimation uncertainty, with the latter being consistent in case of absence of sensor faults. Then, the PL will reliably bound the estimate error of the monitored algorithm, which is informed by the full IMU/GNSS/Auxiliary sensor measurement set. The second contribution is that the method can directly capture the relation of anomalies in IM to raw inputs of individual sensors. This allows the evaluation of environment features and conditions, including dynamic changes, that cause specific sensor faults. At the same time, the initial anomaly detection quality is independent from the quality of the IM, allowing a more objective evaluation of different IMs. The eventual fault exclusion is informed by the IM and, therefore, the DNN training can tolerate the presence of certain amount of image noise and incorrect anomaly labels.

The thesis uses the proposed method to monitor the integrity of a navigation system that integrates the GNSS and IMU sensors with a camera. The system uses car sensor data collected in urban environments. Figure 6.1 shows example images from the evaluated environments. The method employs a camera-only navigation algorithm to estimate camera poses that are used in a loose integration with the other sensors. The results show that, in the absence of faults in GNSS/IMU, for a system where the PL computation is informed from GNSS and IMU noise only, the detected anomalies in camera-only navigation correlate with IM anomalies. In addition, the results confirm an improvement of the performance of the IM system in terms of the RBT metric, when comparing it to the method that does not use camera FDE. Also, it remains computationally efficient.

6.1.3 Chapter organization

The next sections present the methodology of predicting the IM anomaly due to camera faults. They describe briefly the DNN model and the IM for collaborative learning of anomalies (6.2), introduce the camera measurement samples tested for faults (6.3), revisit the hypotheses initially introduced in chapter 4, which are tailored to test for camera-faults (6.4) and, finally, define the ordinary distribution of IM test statistics and the ordinality tests (6.5 and 6.6). Section 6.7 summarizes the chapter and discusses future works. The evaluated datasets for the presented method, anomaly prediction results and performance evaluation of the anomaly prediction and exclusion method are presented in Chapter 7.

6.2 Detection of anomalies in the uncertainty of visual SLAM and in the IM test statistic

The concept of the fault slope was introduced in equation (2.9).



Figure 6.1: Example images from the urban navigation datasets examined here. The locations are San Francisco (USA), Oxford (UK) and Dongtan (South Korea). The images are from the UrbanLoco [72], Robotcar [73] and KAIST [75] datasets.

Using that definition, in the specific investigated case, the fault *slope* for camera measurements can be computed as the ratio of the state estimate error caused by camera measurement faults to the IM test statistic measure affected by the camera measurement faults. In MHSS-based IM, the test statistic to detect faults in sensor measurements is the difference in the state estimate under all-source hypothesis H_0 and under hypothesis H_j , which is fault-tolerant to the sensor's measurements:

$$|\mathbf{x}_p^{(j)} - \mathbf{x}_p^{(0)}| \leq \mathbf{T}_{p,\Delta_j} \quad (6.1)$$

with $\mathbf{x}_p^{(j)}$ and $\mathbf{x}_p^{(0)}$ denoting the position state under fault hypothesis H_j and fault-free hypothesis H_0 , respectively. The formulas include the subscript p to highlight that the vectors refer to the elements of the position state and drop the subscript when the quantities refer to one direction of interest in the position state. \mathbf{T}_{p,Δ_j} is the detection threshold, which takes into account the continuity risk $C_{REQ,j}$ assigned to H_j . In one direction of interest, the equation for the fault *slope* in MHSS-RAIM can be simplified to [5]:

$$Slope_{f_j} = \sigma_{\Delta_j}^2 \quad (6.2)$$

where $\sigma_{\Delta_j}^2 = \sigma_j^2 - \sigma_0^2$, σ_j^2 and σ_0^2 are the estimated error covariances under H_j and H_0 , respectively.

So:

$$T_{\Delta_j} = Q^{-1}(C_{REQ,j})\sigma_{\Delta_j}^2 \quad (6.3)$$

For clarity, let AU be the set of indices for hypotheses under which the measurement set excludes the measurements from one sensor, auxiliary or conventional, but contains measurements from other auxiliary sensors. This work considers that estimation under hypotheses H_j , for all $j \in AU$ is computationally inefficient, at least when the sensor fusion algorithm of such integrations has a high computational load. In addition, there is no assumption about the internal structure of the monitored navigation algorithm which is informed by the measurement set under the all-source hypothesis H_0 . In this case, the covariance of position estimates is assumed unavailable, something that prevents the calculation of the detection threshold in (6.3). Therefore, the method performs fault detection in each auxiliary sensor's measurements individually. This is done via anomaly prediction in the estimate error with a sensor-specific DNN model and refinement based on an IM test statistic which is computed for a specialized measurement set for the sensor of concern, that excludes all other auxiliary sensor measurements. The latter procedure is implemented in the DHT module. For completeness, the subsequent portion of this section repeats the principles of the DNN model, reiterates the

faulty and excluded. This configuration enables the execution of standard ErKF, as discussed in Chapter 3. The "All-source" hypothesis H_0 assumes the fault-free operation of the entire CSS sensor set and auxiliary sensors. In the proposed architecture, the monitored navigation algorithm needs to run only once for that specific sensor set.

The DHT module is introduced in the general IM architecture, runs in parallel with the IM of CSS sensors, and executes one instance of a specific navigation algorithm for each auxiliary sensor. For example, this is a visual SLAM algorithm in the case of the camera. Although the diagram in figure 6.2 illustrates the general architecture, where many auxiliary sensors may be integrated with GNSS/IMU, the methodology presented in this chapter is again specific to camera measurements. The detailed explanation of the DHT module related to camera FDE follows.

6.3 Anomaly detection in DHT: Grouped image features

A predicted navigation anomaly from the previously described DNN model will initiate DHT to detect the specific anomalous frames that cause IM hazards due to the camera. Chapter 4 described the new hypothesis set of faults in sequential VN estimates. This and the next section will restate these hypotheses and will describe how they are used for camera FDE.

Consider that a navigation anomaly is predicted by the DNN at a step k^* . The method searches for IM anomalies around k^* in a horizon with pre-selected size M . Let us denote the full set of image frames around the anomaly as $Y_{cam, k^*-b}^{k^*+a}$, with $a - b = M$. Then, we will denote each image frame obtained in the horizon at any step k as $c_k \in Y_{cam, k^*-b}^{k^*+a}$. In a typical VN algorithm, the estimation of the camera pose w.r.t. to the first frame is achieved by tracking features observed in subsequent frames.

Chapter 4 denoted the camera pose as:

$${}_{(w)}\mathbf{x}_C = \begin{bmatrix} {}_{(w)}\mathbf{p}_C \\ {}_{(w)}\boldsymbol{\phi}_C \end{bmatrix} \in \mathbb{R}^{3+r_m} \quad (6.4)$$

with ${}_{(w)}\mathbf{p}_C \in \mathbb{R}^3$ being the camera position in the global reference frame (W) and ${}_{(w)}\boldsymbol{\phi}_C$ being the camera attitude vector parameterized with $r_m = 3$ or 4 elements, depending on the representation used. A 2D projection of a feature l to the image plane was symbolized as $\pi_l(\mathbf{x}_C)$ and the pinhole camera model can be used for the formula of $\pi_l(\mathbf{x}_C)$ [104]. The noisy measurement of a feature point extracted from the measurement image was defined in equation 4.3 and is repeated here:

$$\mathbf{z}_l = \pi_l({}_{(w)}\mathbf{x}_C) + \mathbf{V}_l + \mathbf{b}_l, \quad (6.5)$$

where \mathbf{V}_l is zero-mean Gaussian noise, and \mathbf{b}_l is a bias in the feature location. Then, L_i feature correspondences, in total, tracked in M subsequent frames, are grouped into

their respective image frames. The complete vector of stacked feature projections at

step i is $\mathbf{Z}_i = \begin{bmatrix} \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_{L_i} \end{bmatrix}$, or grouped in their i corresponding images:

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_1 \\ \vdots \\ \mathbf{Z}_i \end{bmatrix}, \quad (6.6)$$

where $\mathbf{z}_{i'}, i' = 1, \dots, i$ is a vector with feature point correspondences found in image obtained at time i' .

This notation will facilitate also here to define hypotheses by associating them clearly with image frames, which represent groups of feature correspondences potentially containing faults.

6.4 Camera fault hypotheses definition

The input measurement set to the "DHT for camera" module in figure 6.2 consists of camera and CSS measurements only. As the DHT module is part of the overall IM, define as H_{CC} the hypothesis that is associated with the all-source measurement set in the DHT module, i.e. H_{CC} is the hypothesis that all auxiliary sensors other than the camera are faulty. In DHT, a VN system utilizes the image frames to estimate camera poses, and those are compared with CSS measurements to detect inconsistencies. Such inconsistencies guide the IM anomaly detection process, in a similar way as they guided the PL inflation procedure in chapter 4.

For IM anomaly detection due to camera measurements, the method utilizes $M + 1$ hypotheses, where under hypothesis $H_{CC,\iota} (\iota = 0, 1, \dots, M)$ from 0 to M image frames are assumed to be associated with an IM anomaly. Hypothesis testing is initiated if an image frame \mathbf{c}_k is associated with an anomaly in the true error (after employing the DNN predictor) with high probability. Then, under hypothesis $H_{CC,\iota}$, the subset of frames that inform the navigation solution is $\mathbf{X}^{(CC,\iota)} \subset \mathbf{X}_{C,M}$, $\iota = 0, 1, \dots, M$ defined as:

$$\mathbf{X}^{(CC,\iota)} = \{\hat{\mathbf{x}}_{C,\iota+1-M}, \dots, \hat{\mathbf{x}}_{C,\iota}\} \quad (6.7)$$

Figure 6.3 shows the diagram of the DHT module for camera measurements. It is a slight modification of the figure 4.2, to include also the anomaly testing routine. Importantly, when developing the DHT module for other auxiliary sensors, only the VN algorithm needs to change to another sensor-specific algorithm.

Each hypothesis $H_{CC,\iota}$ is associated with a sequence with M observations $\mathbf{y}^{(CC,\iota)} = (y_1^{(CC,\iota)}, \dots, y_M^{(CC,\iota)})$, $\iota = 0, \dots, M$ of the values of an IM test statistic, which we refer to

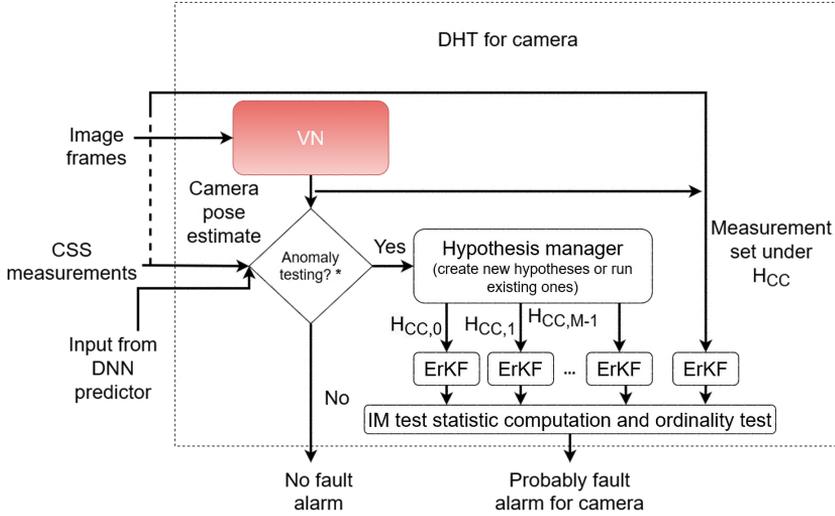


Figure 6.3: Diagram of the DHT module for fault detection in camera measurements. Different instances of ErKF are running for each hypothesis, although the implementation of the filter is always the same.

* Anomaly testing runs if either the DNN predicted a navigation anomaly or if the DHT is still evaluating a previous anomaly.

as a process under hypothesis $H_{CC,t}$. A new child process is possibly created after M steps, if the system has not returned to ordinality (section 6.5).

A subset of observations does not include a fault when the observations follow the ordinary distribution λ of IM test statistics, while, if there is one or more faults, the observation at any step $k \in [1, M]$ cannot be drawn from the ordinary distribution with large likelihood. The next section introduces the ordinary distribution and the ordinality test for the process under each hypothesis.

6.5 Ordinary distribution

According to equation (6.1), for the new hypothesis set tailored to camera measurements, at each step k the test statistic is defined as the difference between the position solution under H_{CC} and hypotheses $H_{CC,t}$:

$$\tau_{pCC,t} = |\mathbf{x}_p^{(CC,t)} - \mathbf{x}_p^{(CC)}| \quad (6.8)$$

Zhang, Wang and Gao [105] analyzed the distributions of the MHSS test statistic for different fault hypotheses and suggested that they vary due to the differences in the underlying measurement subsets. The MHSS test statistics under each fault hypothesis are normally distributed [106]. For one position direction of interest the distribution of

the IM test statistic is:

$$\tau_{CC,t} = \mathcal{N}(f_{\Delta_{CC,t}}, \sigma_{\Delta_{CC,t}}^2) \quad (6.9)$$

Here

$$\sigma_{\Delta_{CC,t}}^2 = \sigma_{CC,t}^2 - \sigma_{CC}^2$$

where

$\sigma_{CC,t}^2, \sigma_{CC}^2$ are the estimated variances in the direction of interest for the solutions under $H_{CC,t}$ and H_{CC} respectively, and

$$f_{\Delta_{CC,t}} = f_{CC} - f_{CC,t}$$

where

f_{CC} is the contribution of measurement faults to the position state of interest in the full set solution under H_{CC} , and

$f_{CC,t}$ is the contribution of faults of the fault-tolerant set of measurements under hypothesis $H_{CC,t}$ to the position state of interest in the fault-tolerant solution.

The current work deals with a set of processes, where the measurement set informing each process is different among the simultaneous processes running at each step as well as between a parent and a child process possibly created after the test of ordinality in equation (6.11). This means that a test statistic distribution determined to fit the samples of a process does not necessarily represent a distribution of ordinality that can be used for reliably testing the samples of any other new process. However, under fault-free conditions, the distributions for all-hypotheses are expected to have zero mean (according to equation (6.9), although with different variances. In this work, only one distribution of ordinality is utilized per direction, computed with samples from all processes. This distribution is found to follow a zero-mean normal distribution:

$$\lambda \sim \mathcal{N}(0, \sigma_{\lambda_a}^2) \quad (6.10)$$

where the subscript a refers to the direction of interest, i.e. North or East.

To obtain the ordinary samples and the ordinary distribution, the complete IM system with the monitored navigation algorithm is run for driving road vehicle trajectories with labeled sections where the outputs appear ordinary, after utilizing statistical methods. The labeling of ordinary sections is accomplished offline based on the sliding Z-score metric. The methodology was also used to label anomalies in section 5.2.1, therefore a

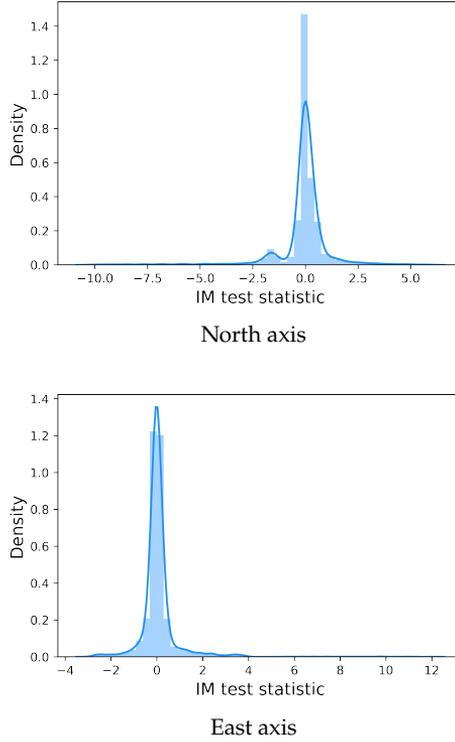


Figure 6.4: Ordinary distribution of IM test statistics, on each axis. It is computed from samples of the metric that were selected as ordinary based on statistical tests. The samples were obtained during IM of real trajectories in the case study.

definition in this chapter is omitted. Understandably, the formulation of the ordinary distribution is an open topic for more research in the future. Figure 6.4 depicts the computed ordinary distribution that was utilized in the experiments to decide anomalous samples.

6.6 Test for anomalies

The method identifies the end of an anomaly by executing a test at the finishing step $k_{1,\kappa} = \kappa M$ of the current running processes, with $\kappa \in \mathcal{Z}^+$ a positive integer. Starting from the hypothesis $H_{CC,M-1}$, that assumes that only the 0th image frame is faulty, until hypothesis $H_{CC,0}$, that assumes all M image frames are faulty, the test is:

$$y_i^{(CC,j)} \in [F_\lambda^{-1}(0.05), F_\lambda^{-1}(0.95)] \quad (6.11)$$

$$\forall i \in [1, M], \forall j \in [0, M - 1]$$

where:

$F_{\lambda}^{-1}, F_{\lambda}^{-1}$ the inverse of the Cumulative Distribution Function of the ordinarily distributed variable λ .

$f(y_i^{(CC,t)}; \lambda)$ the probability density function of sample $y_i^{(CC,t)}$ drawn from the ordinary distribution λ .

A new set of processes is created at time $k_{1,\kappa}$, until the ordinality test in equation (6.11) succeeds.

6.7 Chapter Summary

The presented method constrains the space of sensor fault hypotheses in MHSS-based IMs for multi-sensor navigation systems and is suitable for integrations of the conventional GNSS and IMU sensors with auxiliary sensors. There are two important modifications in the MHSS-based IM architecture of previous works. PL computation is only informed from the conventional sensors IMU/GNSS as before, while the IM evaluates each auxiliary sensor for FDE individually, assisted by a DNN. In this work the evaluated system is, specifically, the integration IMU/GNSS/Camera. The proposed method is scalable to integrations with multiple auxiliary sensors and does not rely on the availability of a noise model for each auxiliary sensor. An additional contribution is that the initial anomaly prediction in the distribution of navigation faults caused by the camera is done based on raw measurements. Hence, the method can facilitate future researches to quantify the direct effect of various environmental conditions on the performance of IM, via the effect on individual sensors. This may allow the development of robust IMs that are tested in a predetermined set of challenging simulated or real environmental conditions. The next chapter will delve into the full IM system evaluation, exploring the camera fault alarm capabilities of the DHT module individually. Additionally, it will compare the performance of the IM system, with camera FDE but without the PL inflation of chapter 4, in terms of PL bounding performance, to the typical MHSS-based IM, that is informed by the noise and faults of all sensors.

Chapter 7

Experimental evaluation

7.1 Introduction

This chapter presents the experimental evaluation of the DHT module separately for camera PL inflation and FDE, as well as PL inflation for IMU preintegration faults. It is reminded that the PL inflation methodologies are described in chapter 4, while camera FDE in chapter 6. The evaluation chapter includes also a brief investigation of a few cases of VN anomalies identified in the trajectories under evaluation, aiming to identify potential environmental characteristics contributing to these anomalies.

In the experiments, $I_{REQ} = 2.7 * 10^{-8}$ and $C_{REQ} = 8 * 10^{-6}$.

7.2 Anomaly detection and camera exclusion results

7.2.1 Trajectories and system setup

The first experiment concerns the VN anomaly detection and exclusion module presented in chapter 6. The performance of the proposed method is evaluated by utilizing datasets from the UrbanLoco, Robotcar and KAIST, described in section 4.6. In each dataset, the car completes one trajectory. We selected two datasets from UrbanLoco, referred to as UrbanLoco1 and UrbanLoco2 in the text, one dataset from Robotcar and one dataset from KAIST. The trajectories in the datasets are split into segments of around 1500 to 2250 steps (150s - 225s) and the evaluation is done for one segment from each trajectory, excluding some steps at the beginning to allow initialization. The segments are:

- UrbanLoco1: Steps 0-1500 in the dataset CAColiTower20190828184706, collected near Coit Tower, San Francisco: Very busy area at the beginning, many pedestrians and pedestrian crossings, illumination challenges.

- UrbanLoco2: Steps 0-1500 in the dataset CALombardStreet2019082819041, collected near Lombard street, San Francisco: Not that busy area, a lot of shadows / dark areas, hilly terrain.
- Robotcar: Steps 1000-3000 in the dataset 2015-08-14-14-54-57, collected in Oxford, UK: not busy area, roadworks, severe illumination challenges.
- KAIST: Steps 4500-6750 Suburban area, a lot of car traffic, High rising buildings with repetitive patterns, severe illumination challenges.

The experimentation utilizes the ORBSLAM2 algorithm [97] for camera pose estimation for the UrbanLoco and KAIST trajectories. The Robotcar trajectory comes with official estimation data that were obtained using a VO solution. In all cases, the Umeyama method [98] is utilized to find the optimal transformation between the reference and the corresponding camera pose estimates. In the following, the text uses the same term VN for both camera pose estimation algorithms.

At this stage, splitting of the trajectories was important to limit learning navigation anomalies occurring due to previous drifts induced by visual-only navigation. There is future work to do to label the exact frames that associate with anomalies. This will require to go from statistical methods in the overall trajectory error to relative error methods and handling of the drift.

The number of frames to test each time was selected $M = 10$. This number should be selected to be at least equal to the number of image frames M_{vis} per input sample that is fed to DNN. That number is 7, as noted in chapter 5. A key finding of that work was that larger samples lead to better anomaly learning than smaller ones, although there is an upper limit for the sample size due to limited hardware resources. A selection $M > M_{vis}$, as in this evaluation, is done to account for probably anomalous segment that begins a bit before a predicted anomaly from the DNN.

Figure 7.1 shows the reference trajectories, the evaluated segments and the aligned estimated segments from the VN system.

The IM for CSS, informed by the IMU and GNSS noise, computes the PLs based on the integrity requirement $I_{REQ} = 2.7 * 10^{-8}$. The experiments employ a formula for the computation of PLs that was previously utilized for RAIM [5]. The PL in each direction, for the fault-free case, is computed as:

$$p_L = Q^{-1}\{I_{REQ}/2\}\sigma_{CSS} \quad (7.1)$$

where σ_{CSS} is the estimated variance of the state error under H_{CSS} in the respective direction. As there is no redundancy of the measurements in the utilized CSS, the PLs do not compensate for possible faults in the measurements of those sensors. Nevertheless, we tuned their noise parameters and confirmed that there are no significant faults by utilizing consistency checks and IM.

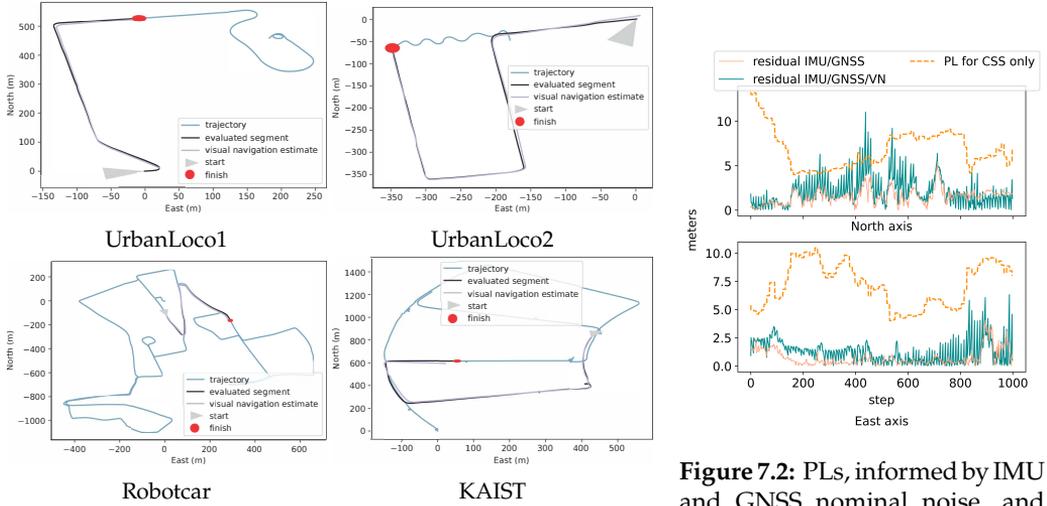


Figure 7.1: Evaluated trajectories. The figures show the full reference trajectories, the evaluated segments of each trajectory and the aligned camera poses for those segments, estimated from the VN system.

7.2.2 Anomaly detection results

This section analyzes the performance of IM anomaly detection after executing the DHT module, with or without utilization of the DNN. If the DNN is not utilized, the DHT runs uninterrupted for the detection of anomalies. Uninterrupted execution means that:

1. Anomaly testing does not stop even if the ordinality test in equation (6.11) succeeds.
2. Camera measurements are not excluded even after the detection of anomaly.

For UrbanLoco1, figure 7.2 shows the result of the IM for the integrations IMU/GNSS and IMU/GNSS/VN. In both cases, the PLs are only informed from GNSS and IMU nominal noise, and are computed based on equation (7.1). Someone observes an increase of the error at the North direction after the integration with VN, especially after step 180. This results in many IM anomalies that last until approximately step 500. For the same trajectory, figure 7.3 shows the minimum log Probability Density Function (log PDF), as defined in equation (7.2) below. It also shows the DHT processes, that the ordinality test in equation (6.11) identifies as anomalous, under hypotheses $H_{CC,0}$, $H_{CC,A}$ and $H_{CC,9}$. In this evaluation, the DHT module runs uninterrupted.

The experiments use the minimum log PDF metric, that is defined here, to visualize

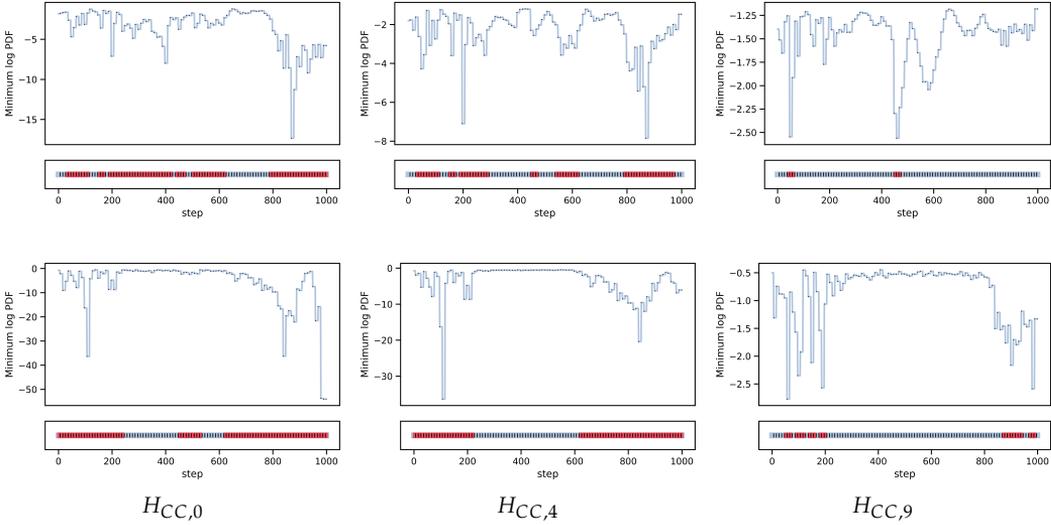


Figure 7.3: For UrbanLoco1, the plots show the values $l_{CC,\iota}(\mathbf{y}^{(CC,\iota)}; \lambda)$ for $\iota = 0, 4, 9$ over time and the processes that are detected anomalous (red) or pass the ordinality test (light blue). Processes are shown as subsequent segments separated by a vertical line at the bottom of each plot.

the likelihood that the various DHT processes under different hypotheses are ordinary. Previously, the symbol $\mathbf{y}^{(CC,\iota)}$ was used to denote the sequence of IM test statistics that are obtained during a process executed under hypothesis $H_{CC,\iota}$. The minimum log PDF of the samples in $\mathbf{y}^{(CC,\iota)}$, given the ordinary distribution λ , is:

$$l_{CC,\iota}(\mathbf{y}^{(CC,\iota)}; \lambda) = \min \{ \ln(f(y_1^{(CC,\iota)}; \lambda)), \dots, \ln(f(y_M^{(CC,\iota)}; \lambda)) \} \quad (7.2)$$

with $f(y_i^{(CC,\iota)}; \lambda)$, $i = 1, \dots, M$ being the PDF around the sample $y_i^{(CC,\iota)}$ when it is drawn from the ordinary distribution λ .

By comparing with figure 7.2, it is observable that the sharp drops of the minimum log PDF in the plots are quite precise in indicating the location of IM anomalies. In addition, processes that do not include any or include few image frames (i.e. processes under $H_{CC,0}$ and $H_{CC,4}$) appear anomalous more often than $H_{CC,9}$, that includes most of the image frames in the sequence. This is expected as for processes, like $H_{CC,9}$, where the fault-free measurement set is very similar to that under H_{CC} , the IM test statistic (equation (6.8)) is generally closer to zero. Additionally, the plots indicate that the effect of individual image frames on the build-up of an anomaly may vary. An example is

around step 450, where the plot for $H_{CC,9}$ in North direction shows that one of the tested processes is anomalous. The sharp decrease in the min log PDF is likely caused by the only image frame assumed non-faulty in the anomalous process. This is not the case for the anomaly at around step 850. The anomaly is clearly visible in the plots for $H_{CC,0}$ and $H_{CC,4}$ but not in the plot for $H_{CC,9}$. The anomaly seems to build up over more than one image frames in the processes.

The next experiment introduces the DNN in the anomaly detection procedure. This means that DHT initiates testing only after receiving a navigation anomaly alarm from the DNN. Anomaly testing stops when the ordinality test succeeds. No fault exclusion is implemented at this experiment. Figure 7.4 shows, for all evaluated trajectory segments, the PLs per axis as computed in equation (7.1). Additionally, it plots the residual of position state estimates under H_0 to reference positions. These may be compared with the PLs for bounding issues. The figure also annotates the VN anomalies detected by the DNN. The detected IM anomalous segments are marked on the residual. The DNN does not differentiate between anomalies in the North and East direction, as training was done based on the maximum of the VN system's estimate errors in the two directions. Therefore, the VN anomalies are the same in both directions. The y-axis location of VN anomalies indicates the magnitude of the VN estimate error (maximum between the two axes). Lastly, a detection for IM anomaly may happen for any of the two directions, but the method will alarm for an anomalous segment regardless.

Figure 7.5 shows, for UrbanLoco1, the result of fault exclusion using the proposed method. The PLs and the residual can be compared with figure 7.2. The fault exclusion leads to a reduction of the residual error and the prevention of some IM anomalies. However, the exclusion is insufficient to prevent all anomalies.

A general conclusion from the figures is that the proposed method performs well in identifying navigation and IM anomalies, in the vast majority of cases where there is anomaly present. The DHT module was able to mitigate for false alarms from the DNN, although there were also few cases where it falsely ignored them. Subsequently the camera measurement FDE was unsuccessful to prevent all IM anomalies, as the experimentation showed. Therefore, despite the very promising results, it is apparent that further tuning is necessary to adequately prevent all anomalies, as well as limit false positives, i.e. detection alarms for camera measurements that do not actually cause anomalies. The quantitative evaluation of the method follows in the next section.

7.2.3 Numerical evaluation of the IM performance

This section evaluates the performance of the proposed camera FDE when monitoring the GNSS/IMU/VN system and PLs are only informed by GNSS and IMU noise. Four metrics are evaluated and these are:

1. The RBT metric, proposed by Li and Waslander [7] to quantify how much of the time the error is sufficiently bounded, as well as the tightness of the bound.

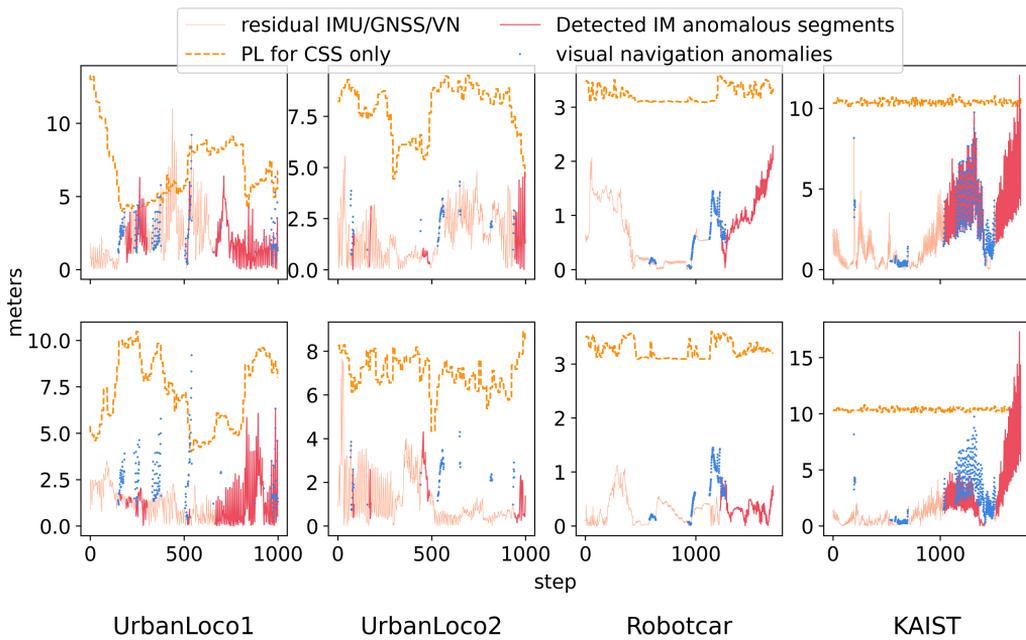


Figure 7.4: Detected anomalies, true position error of GNSS/IMU/VN integration and PLs informed by the GNSS/IMU only. Each column shows the results for one trajectory. The top row shows results for the North direction and the bottom row shows results for the East direction.

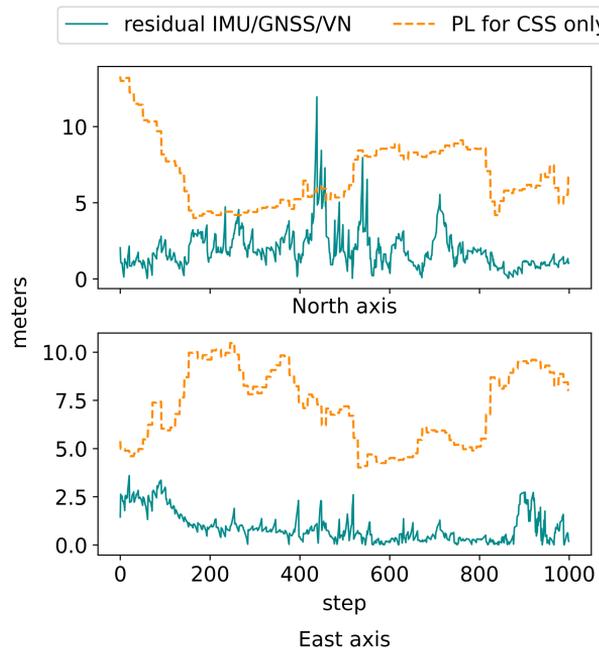


Figure 7.5: Results of IM for the IMU/GNSS/VN integration after utilizing the DHT module for VN fault exclusion. The results are for the UrbanLoco1 trajectory.

The thesis introduced the formulation of the RBT metric in section 3.6.5. In the experiments $\rho = 64$.

2. The percentage of time that the errors are bounded by the PLs.
3. The minimum position alert limit for 100% availability of the system (it is the maximum PL).
4. The average execution time.

As described in section 6.6, DHT runs until the process under $H_{CC,0}$ is found ordinary. In the experiment, new anomalies are ignored if DHT is already running by the time of their detection.

Table 7.1 summarizes the results for the various trajectory segments. As the table compares the performance of the proposed method against different IM setups, it is worth defining the names of these setups. The table and the text symbolizes as “IM for CSS” the IM that is informed only by GNSS and IMU noise. Also, it symbolizes as “GNSS/IMU/VN MHSS” the MHSS-based IM that is informed by GNSS + IMU + VN noise and faults.

In the “GNSS/IMU/VN MHSS” setup, the IM does a simplistic assumption of white noise in the VN estimates. Then, The IM compensates for the contributions of failures in the GNSS measurements or VN estimates by defining the $N_{SS} = 2$ fault hypotheses H_j ($j = 1, 2$), respectively fault tolerant to GNSS measurements or VN estimates. It assumes equal distribution of the integrity risk I_{REQ} and continuity risk to the fault-free hypothesis H_0 and the two fault hypotheses. The formulas for PL computation under fault-free and fault hypotheses were introduced in Chapter 3 within the context of MHSS-based IM for conventional integrations of position sensors with an IMU. Similarly, here, the “GNSS/IMU/VN MHSS” setup considers the GNSS and VN as redundant position sensors. Reiterating the equations from chapter 3, the PL under each fault hypothesis, in each direction, is formulated as:

$$p_{L_j} = Q^{-1}\left\{\frac{I_{REQ}}{P(H_j)(N_{SS} + 1)}\right\}\sigma_j + T_{\Delta_j} \quad (7.3)$$

where σ_j is the estimated variance of the state error under hypothesis j and T_{Δ_j} is given in equation (6.3). Under the fault-free hypothesis H_0 the PL compensates for the nominal noise of all three inputs:

$$p_{L_0} = Q^{-1}\left\{\frac{I_{REQ}}{2(N_{SS} + 1)}\right\}\sigma_0 \quad (7.4)$$

The IM selects the maximum of the three PL bounds as the final PL.

The first row in the table for each trajectory shows the performance of the proposed method, which utilizes IM for CSS and the DHT module for camera FDE. It compares it against the method where the DHT module runs continuously to find IM anomalies and performs fault exclusion, independently if the DNN predicts anomalies ("DHT, no DNN" at second row), as well as against the method which does not use the DHT module ("IM for CSS" method at third row). The last row shows the performance of the MHSS where PLs are also informed from camera faults ("GNSS/IMU/Camera MHSS"). The table shows the average metrics across both North and East axes.

The method based on "IM for CSS" without camera FDE (third row) shows the worst performance in terms of bounding the error, while, the proposed method, that uses the IM for CSS + DHT, shows comparable results to the "GNSS/IMU/VN MHSS" (fourth row) in terms of the execution time and the percentage of time with bounded error. The integration of more sensors results in lower uncertainty which is consistent when the sensors do not contain faults. The results are even better, in terms of prevention of IM anomalies, when the method runs DHT and fault exclusion, without using alarms from the DNN (second row), although with significantly higher computation time than when using the DNN (proposed method). The "GNSS/IMU/VN MHSS" shows the worst performance on the RBT metric due to very loose PLs. However, one should pay attention when interpreting the differences in the RBT metrics obtained for the IM setups compared here; The RBT metric rewards tighter PLs. It might be that the exclusion of the camera, due to a false alarm, results in larger estimate error, still bounded by the PL, and the estimated uncertainty increases too. RBT will indicate that the result is better, although the navigation algorithm is less accurate.

We believe that the results are motivating, since, compared to the "GNSS/IMU/VN MHSS" method, the proposed method scales better to integrations with more auxiliary sensors. Also, it does not assume knowledge of the noise of auxiliary sensor measurements. Finally, it achieves 100% availability for a lower AL, as it does not compensate for noise and faults of the auxiliary sensors. Therefore, the proposed method should be considered during the development of computationally efficient and sensor-agnostic IMs, although further tuning is required to prevent anomaly misdetections and insufficient camera measurement exclusion.

Table 7.1: Results of the proposed IM for the GNSS/IMU/Camera navigation system and comparison with other IM setups.

	RBT	Percentage of Time available with bounded error (%)	Available for horizontal alert limit (m)	Average execution time (s)
UrbanLoco1	IM for CSS + DHT (proposed method)	98.1	13.28	0.192
	DHT, no DNN	98.5	13.28	0.250
	IM for CSS	97.3	13.28	0.077
	GNSS/IMU/VN	100	36.96	0.129
	MHSS			
UrbanLoco2	IM for CSS + DHT (proposed method)	100	9.50	0.142
	DHT, no DNN	100	9.50	0.215
	IM for CSS	100	9.50	0.077
	GNSS/IMU/VN	100	27.76	0.106
	MHSS			
Robotcar	IM for CSS + DHT (proposed method)	100	3.6	0.097
	DHT, no DNN	100	3.6	0.155
	IM for CSS	100	3.6	0.066
	GNSS/IMU/VN	100	21.92	0.147
	MHSS			
KAIST	IM for CSS + DHT (proposed method)	100	10.85	0.149
	DHT, no DNN	100	10.85	0.217
	IM for CSS	98.5	10.85	0.071
	GNSS/IMU/VN	99.9	17.18	0.101
	MHSS			

7.3 PL inflation methodology for IMU preintegration and VN faults

The remaining of the chapter evaluates the performance of the PL inflation methodology, presented in chapter 4, on the 4seasons dataset [107] which comprises sensor data recorded from a car driving inside Munich, Germany. While the dataset includes data for many trajectories, the experimentation focuses on instances of driving within the old town area. Figure 7.6 displays a satellite image of the drive area, the trajectory, shown in light blue and the start location. This dataset encompasses several real-world drives along the same route, enabling the investigation of varied weather and illumination conditions. The validation tries to indicate safety hazards under some variations in weather conditions, the challenge with sun reflection, and the performance of the fusion of camera with GNSS and IMU in the presence of IMU biases. The validation is for an example Factor Graph-based integration, optimizing for an ImuFactor with preintegrated IMU measurements, a GNSS Factor and a VN factor that includes aligned camera pose estimates from a SLAM algorithm. This implementation can be replaced by another algorithm.

The next sections describe the experimental setup and validate the PLs under two fault modes, namely the $IMU - F$ and the $VN - F$, with respect to the true estimate error. The PLs for these modes were computed in equations (4.20) and (4.25), respectively. Furthermore, the IM accounts for the fault-free mode (FF) in relation to the CSS, exclusively, given that the nominal noise of the camera is assumed unknown; In effect, it accounts for the nominal noise of GNSS and IMU by integrating those sensors in ErKF, as demonstrated on the left side of the tree in figure 4.1. It omits PL computation under the $GNSS - F$ fault mode due to lack of redundancy of GNSS with other position sensors. Therefore, to avoid the presence of any GNSS faults in the data, the GNSS receiver positions are substituted with fault-free reference positions, distorted by nominal GNSS noise.

7.3.1 Fault modes and residual computation

In DHT residual computation, VN poses are derived using the DM-VIO algorithm [108], in visual-inertial mode. The DHT employs the ErKF to validate hypotheses. The monitored algorithm, implemented with the GTSAM Library [109], integrates the VN poses with IMU and GNSS measurements. For preintegrated residual computation, predictions of the preintegrated IMU measurements, based on the estimated position state of the monitored algorithm, are compared with the observed ones. Those are computed using the corresponding preintegration method in the GTSAM library [110].

7.3.2 Datasets

To enhance the accuracy of VN fault localization, the experimentation is conducted for drive segments instead of full drives of the trajectory in figure 7.6. Table 7.2 summarizes the selected drives and corresponding segments. The table includes also details for each drive, such as capture date, time and prevailing conditions. The segments are initially



Figure 7.6: Satellite image of the drive location in the old town of Munich.

established based on the first evaluated drive, and each segment spans 200 - 250 seconds, including both driving and stationary time. The method generates corresponding segments for other drives, ensuring consistent starting and ending locations. It is important to note that the duration of those corresponding segments varies and might be longer than 250 or shorter than 200 seconds. Corresponding segments in different drives are mentioned with the same Id. Additional segment information includes the steps of IMU data collected during driving on the segment and the driving duration in seconds.

Figure 7.7 depicts the selected segments on the drives. An arrow and a dot indicate the start and end locations, respectively, on "Old Town 1". These indicators offer insight into the travel direction across segments and the full trajectory loop: the car initially navigates segment 1, then proceeds to segment 2, concluding at a location slightly past the starting point. Notably, all segments align almost perfectly, with the only noticeable misalignment seen in the correspondence of segment 2, while the car travels from location (750, -875) to (900, -500).

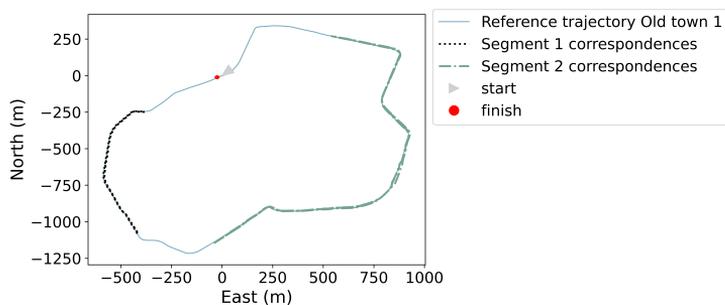
7.3.3 The scenario of car navigation in urban environment

Figure 7.8 shows the percentage rate of change of residual of preintegrated IMU measurements to the corresponding position state estimated from GTSAM, for the segments 1 and 2 in Old town 1. For example, a value of 8 on the vertical axis in one of the plots of figure 7.8 means that each change of 100% to the residual component in the

Table 7.2: Evaluated drives in 4seasons.

Drives		
Name	Datetime	Conditions
Old town 1	2021-01-07 10:49	winter,cloudy/snowy/sunny,morning
Old town 2	2021-02-25 12:34	winter,sunny,afternoon
Old town 3	2020-10-08 11:53	fall,cloudy,morning

Segments			
Id	Drive	Steps	Duration [s]
1	Old town 1	10000-30000	200
2	Old town 1	40000-65000	250
1	Old town 2	7244-26443	192
2	Old town 2	35558-75095	395.4
1	Old town 3	5907-33224	273.2
2	Old town 3	43884-82038	381.5

**Figure 7.7:** Drives and selected segments. The arrow and dot symbolize the start and end locations, respectively, on the drive "Old Town 1".

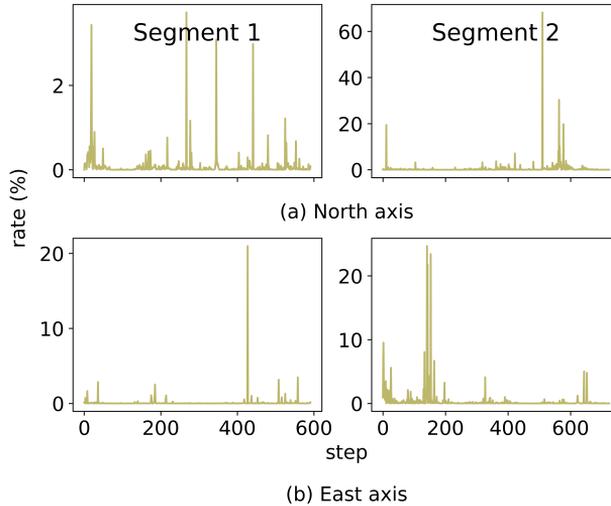


Figure 7.8: Rate of change of residual of preintegrated IMU measurements to the monitored algorithm's position state, for segment 1 (left column) and 2 (right column) in Old town 1. The first row displays results for the North axis, while the second row displays results for the East axis.

corresponding direction - North or East - leads to 8% change in the position state in the same direction. The percentage rates are used for visualization purposes but the absolute rates of change that correspond to these percentages are, in particular, the two diagonal elements in the Jacobian J_B in equation (4.19) in the North ($d = 1$) and East ($d = 2$) directions affected by the two first elements in the preintegrated IMU measurement residual ($q = 1$ and $q = 2$). Throughout all the experiments in this chapter, the values of $J_{B_{11}}$ and $J_{B_{22}}$ remain constant and have been derived using the perturbation procedure in algorithm 1 for all drives and segments in table 7.2. The procedure assists to obtain a value for the two Jacobian elements at each step in the drive segments. The final utilized values are the mean of all samples, after removing outliers. These values were found $J_{B_{11}} = 1.74$ and $J_{B_{22}} = 0.98$.

Figure 7.9 shows the result from the execution of the IM method on Segment 1 and 2 of Old town 1. The figure shows the results for the PLs under H_{CSS} , the PLs accounting for pre-integrated IMU measurement faults, the error from the ErKF under H_{CSS} , and the error from the monitored navigation algorithm. As there is no redundancy in the position sensors under H_{CSS} , where only one GNSS is available, the method does not perform MHSS, and the respective PLs account only for the nominal noise of the sensors in CSS. It is noteworthy that, in most cases, the inflation of the PLs under H_{CSS} , to account for probable pre-integrated IMU measurement faults, is unnecessary. To elaborate, consider the example from the figures. A straightforward summation of PLs for IMU pre-integration and PLs for CSS would suggest an inflation of the

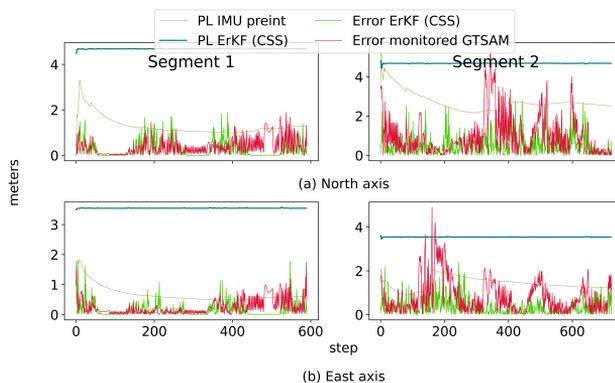


Figure 7.9: The error in position from the monitored GTSAM integration along with the error of the ErKF running for the CSS and the PLs computed to account for CSS measurement noise and for the preintegrated IMU measurement faults. The results are for Segment 1 (left column) and 2 (right column) in Old town 1.

initial bound (approximately 4 meters) by about 1 meter along the North axis for Segment 1 and by 3 meters along the North axis for Segment 2. This would significantly decrease system availability if safety requirements remained constant, without the extra conservativeness being necessary. However, the experimentation with the Segment 2 revealed an increased risk in the monitored integration of GNSS with the additional sensors, especially for the position states along the East axis. In that case the PL inflation term improves the ability to address that risk. It is interesting to examine also the computed PLs for VN faults in that case.

Figure 7.10 demonstrates the inflated PLs, accounting for faults in the preintegrated IMU measurement residual and in VN. Similarly to the computation procedure of J_B mentioned earlier, the Jacobian relating the GNSS/IMU/VN ErKF states to the monitored algorithm states has been derived using the perturbation procedure in algorithm 2 for multiple drives. This Jacobian remains constant in all experiments under faulty conditions and is the mean of all samples, after removing outliers. Its values, referring to the position directions North and East, were derived as:

$$J_{C,11} = 2.4, J_{C,22} = 2.37$$

The figure shows that the inflation leads generally to relaxed error bounds. Across most cases, the two individual PL terms, for IMU preintegration and VN, exhibit comparable numerical values. However, a notable divergence occurs in the East direction during driving on Segment 2. In this case, although both sets of PLs respond correctly to the heightened risk present around step 200 with a similar increasing trend, the PLs associated with VN display a much stronger response.

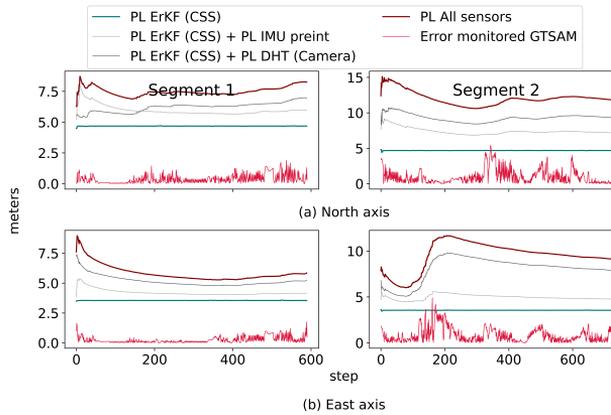


Figure 7.10: Inflated PLs to account for faults in IMU preintegrated measurements and in camera measurements. The figure shows also the true error from the monitored GTSAM integration. The results are for Segment 1 and 2 in Old town 1.

7.3.4 Simulation of faults

This section analyzes the behavior of the system while PL inflation is necessary, in the presence of substantial faults in IMU and VN estimated positions. The faults are introduced artificially in the available data and span a variety of conditions. In the following, the fault types under examination are;

- IMU constant bias; A consistent offset is added to the IMU linear acceleration measurements at each time step during a time period to simulate a steady deviation from the true acceleration values.
- IMU slowly growing and shrinking bias; Over a specified duration, a bias is gradually increased and then decreased in the IMU linear acceleration measurements to mimic a slowly growing and then diminishing deviation from the true values.
- VN position slowly growing and shrinking drift: A gradual and continuous error is introduced into the estimated positions over time, wherein the position errors slowly increase and then decrease, simulating a drift in the estimated positions.
- VN position slowly growing drift: A gradual and continuous error is introduced into the estimated positions over time. The drift does not diminish but stays constant after reaching its peak.
- VN Constant Drift Rate: Abrupt drift in VN position estimates for a defined duration.

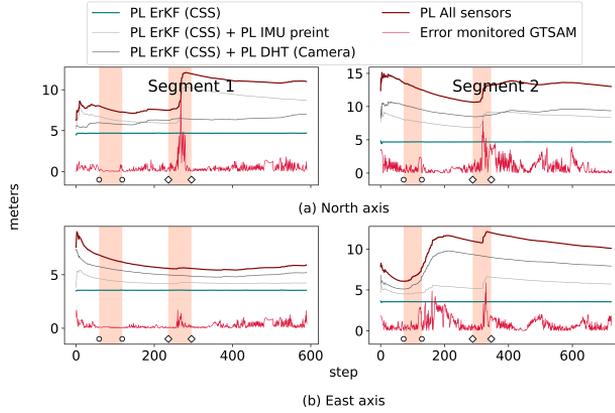


Figure 7.11: PLs during execution of experiment IMU-GROWSHRINK for drive Old town 1.

The experiments, detailed in Table 7.3, feature various combinations of these fault types. Figures 7.11 to 7.18 showcase the PL inflation process during various experiments, where the inflation encompasses individual terms addressing pre-integrated IMU measurement faults and VN position faults. These visualize the system’s response to introduced biases and drifts during the relevant time periods, indicating moments when the system may become unavailable for reasonably low alert thresholds. The figures illustrate the time horizons of application of each fault as shaded areas, and table 7.4 explains the representation.

A note is that the PL computation in all experiments utilizes the same Jacobian values J_B and J_C stated in section 7.3.3, which were calculated based on multiple drives run under normal conditions.

It is worth looking at figure 7.12 as well, which depicts the residuals of GTSAM for the integration of GNSS/IMU under the faulty conditions persisting in experiment IMU-GROWSHRINK. A comparative analysis between Figure 7.11 and Figure 7.12 affirms the increased impact of IMU faults on the simpler navigation system, which is reliant on the infrequent GNSS fixes. In that scenario, the PL inflation is also larger due to the larger influence of changes in the preintegrated IMU measurement residuals on the estimator’s state. Although the integration of the camera in this experiment showed to have a mitigating effect on the state estimate fault originating from IMU faults, further investigation is necessary to better understand the consistency issues in the complex GNSS/IMU/VN integration and the behavior of the IM system in such scenarios. This requires exploring a very extensive set of sensor fault combinations during various time-frames and evaluating various sensor frequencies. Such analysis is beyond the scope of the thesis.

A general observation in all experiments is that the application of a constant IMU bias

Table 7.3: Experiments with simulated faults

Name	IMU constant bias	IMU growing-shrinking bias	VN constant drift	VN growing-shrinking drift	VN growing drift
IMU-GROWSHRINK	2 m/s ² (20s @ 1/10)	0.4 m/s ² /s (20s @ 4/10)	-	-	-
VN-SLOWGROW	2 m/s ² (20s @ 1/10)	0.2 m/s ² /s (20s @ 4/10)	-	-	0.1 m/s (20s @ 6.5/10)
IMUVN-GROWSHRINK	2 m/s ² (20s @ 1/10)	0.2 m/s ² /s (20s @ 4/10)	-	0.2 m/s (20s @ 6.5/10)	-
IMUVN-GROWSHRINK-EXT	2 m/s ² (20s @ 1/10)	0.2 m/s ² /s (20s @ 4/10)	-	0.2 m/s (40s @ 6.5/10)	-
IMUVN_GROWSHRINK_VN_CONST	-	0.2 m/s ² /s (20s @ 4/10)	2 m (20s @ 1/10)	0.2 m/s (20s @ 6.5/10)	-

The table uses the notation (duration s @ drive proportion) which summarizes when the fault is introduced and how long it lasts in total. For example, 20s @ 4/10 means that the fault is introduced after 4/10 of the drive and lasts 20s.

The provided values for the growing and shrinking faults denote their respective rates of change. Shrinking starts after half of the total duration and maintains an equivalent rate to that of growing. It is important to note that, in all instances, the initial bias or drift is set at 0.

Table 7.4: Representation of areas with applied faults in the figures

Fault	IMU constant bias	IMU growing-shrinking bias	VN constant drift	VN growing-shrinking drift	VN growing drift
Representation	○	◇	○	◇	△

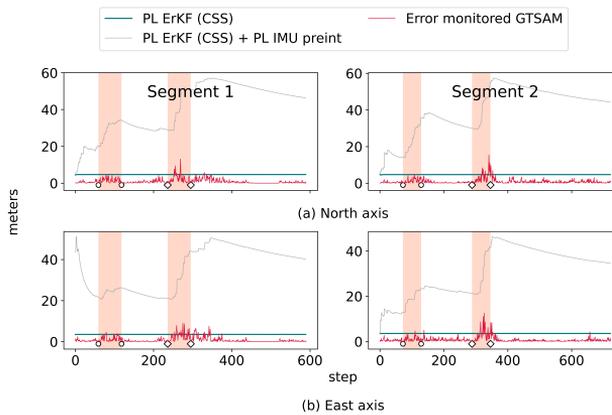


Figure 7.12: PLs during execution of experiment IMU-GROWSHRINK for drive Old town 1 where the monitored system integrates only the IMU and the GNSS in GTSAM.

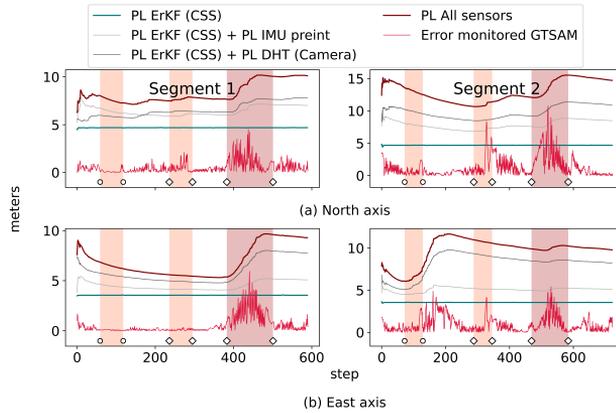


Figure 7.13: PLs during execution of experiment IMUVN-GROWSHRINK for drive Old town 1.

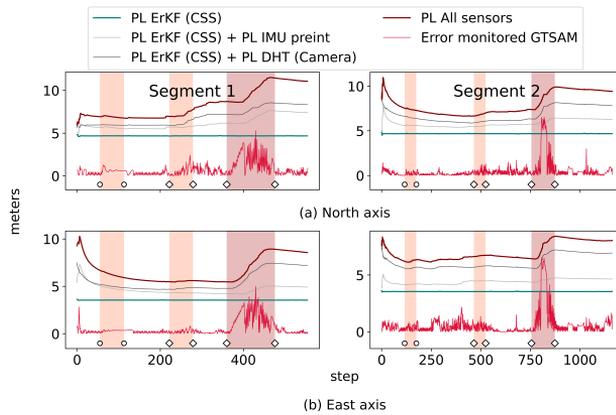


Figure 7.14: PLs during execution of experiment IMUVN-GROWSHRINK for drive Old town 2.

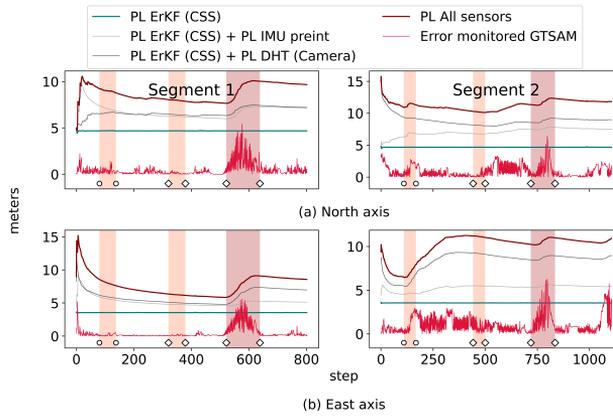


Figure 7.15: PLs during execution of experiment IMUVN-GROWSHRINK for drive Old town 3.

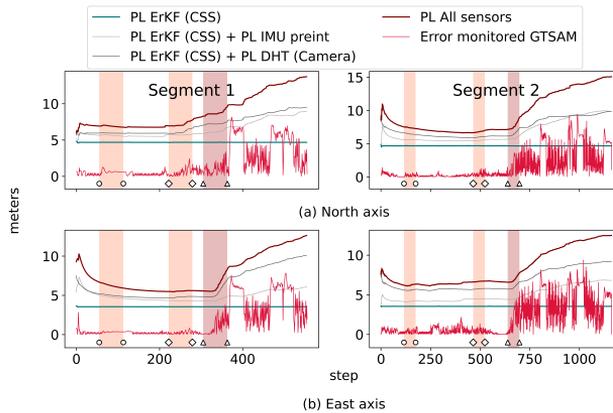


Figure 7.16: PLs during execution of experiment VN-SLOWGROW for drive Old town 2.

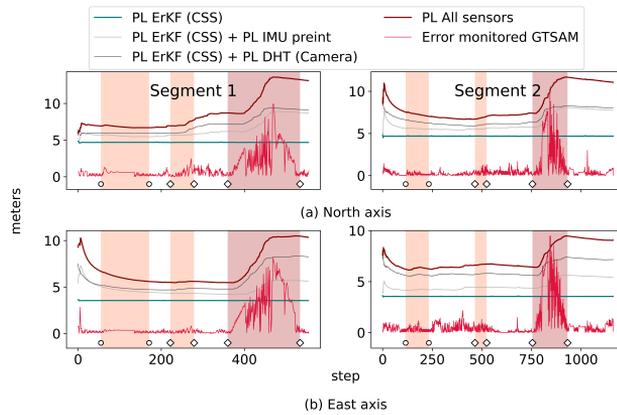


Figure 7.17: PLs during execution of experiment IMUVN-GROWSHRINK-EXT for drive Old town 2.

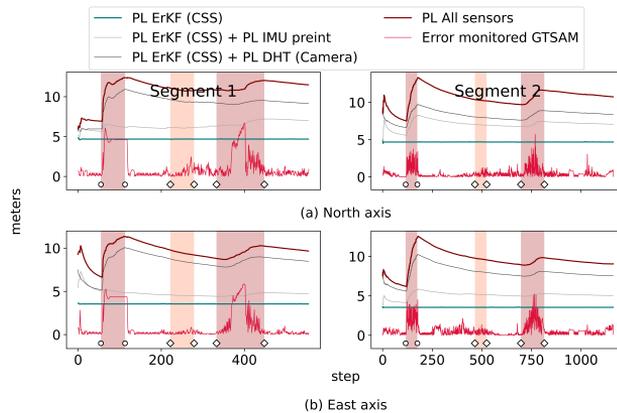


Figure 7.18: PLs during execution of experiment IMUVN-GROWSHRINK-VN-CONST for drive Old town 2.

exerts only a minimal impact on the GTSAM's error, without any tangible effect on PLs. Conversely, the application of a slowly growing and shrinking IMU bias (figures 7.11 - 7.18) affects both PL inflation terms. However, as illustrated best in figure 7.11, the effect on the PL associated with IMU preintegrated measurements is notably stronger, as expected.

A similar trend is observed when applying a VN drift. In all cases, the IM exhibits a stronger response in the PL term relevant to VN estimates.

In the VN-SLOWGROW experiment (figure 7.16), the drift does not diminish after reaching its peak. Both PL terms are substantially affected. The error in the monitored GTSAM reaches a peak shortly after the end of the drift application region. After this, it does not exhibit substantial further increase. In contrast, the PL continues to rise until the end of the experiment.

During the experiment IMUVN-GROWSHRINK-EXT on segment 2 of drive Old town 2 (figure 7.17), there was a time period of unbounded error. This is an important observation, so let us compare with figure 7.15 as well. In contrast to the experiment IMUVN-GROWSHRINK, in IMUVN-GROWSHRINK-EXT the PL stops increasing before the drift reaches its peak. This shows a tendency that the position errors of the ErKFs in the DHT module do not differ that much as time progresses and the residuals in (4.22) remain constant. Likely, the errors are not affected that much by the inclusion or exclusion of VN estimates obtained during the latest horizon, but they are affected more from the accumulated error under H_{CC} until that time. Future research could explore the potential benefits of weighting more the residuals in DHT that persist over extended periods, or comparing the residuals between position estimates under each hypothesis in DHT and the position estimates under H_{CSS} . Eventually, there is no hazardous situation in this specific case. While the error is underestimated in the East direction while driving on segment 2, the PL on the North direction still bounds the maximum error.

Finally, the results from the IMUVN-GROWSHRINK-VN-CONST experiment (figure 7.18) demonstrate that abrupt faults in the high-frequency VN fixes significantly impact fusion errors. The PL term for DHT exhibits a rapid response to the fault.

To conclude, the IM demonstrated on-time response to faults by efficiently inflating the PLs. The PLs bounded the true error in all presented experiments, except of IMUVN-GROWSHRINK-EXT. There was no hazardous outcome in any of the experiments. These results serve as a showcase of the performance of a system employing PL inflation separately for each sensor under specific fault combinations. They should be perceived as experimental evidence of the benefit of the method for autonomous systems reliant on multiple sensors. However, ensuring consistent reliability across all scenarios demands additional theoretical analysis.

7.3.5 Discussion of identified VN faults

Table 7.5 provides a comprehensive overview of the observed metrics for the specific drives and segments under evaluation, in the absence of any simulated faults. The key metrics for assessing IM performance include the RBT [111] and the Max PL. In the experiments under normal conditions there was no case of unbounded errors, therefore, higher RBT values in the table indicate looser bounds. The RBT value shown in the table is the maximum of the RBT value obtained for the PL along the North axis and the PL along the East axis. Additionally, the table records the maximum PL encountered during the run, along any axis, measured in meters. Alert limits below this value will result in system availability lower than 100%. A final important piece of information in the table is the true error of the VN system, presented as the mean and standard deviation of Euclidean distances between DM-VIO positions and the reference, measured in meters. The computation excludes position errors when the vehicle remains nearly stationary.

Table 7.5: IM and weather

Drive	Segments	RBT	Max PL (m)	VN error	
				mean (m)	std (m)
Old town 1	1	2.678	8.276	4.324	2.464
Old town 1	2	4.088	14.39	4.443	2.113
Old town 2	1	2.684	8.674	1.743	0.518
Old town 2	2	2.897	8.078	1.451	0.568
Old town 3	1	2.934	9.792	4.331	2.96
Old town 3	2	4.363	11.684	3.505	1.501

The results show significant discrepancies in the capability of VN to provide accurate position estimates among different drives. The final part of the discussion in this chapter provides insights into the prevailing scene features in some of these evaluation examples that might be affecting negatively the performance of VN. Instead of exhaustively examining every instance of substantial positional error, a developer in safety-critical systems might be looking at those scene features where recurrent and abnormal - momentary or slowly growing - errors originate. Such anomalies could stem from specific scene features or conditions and could guide the developer to more efficient testing of the VN and the IM under similar circumstances. Large VN position estimate faults are indicative of camera measurement faults and should be detected or accounted for in the PL computation. The evaluation herein is limited to very few cases and is not exhaustive of all possible VN faults.

A statistical measure to delineate anomalous VN position errors is the Z-Score $Z = (\hat{e}_S - \mu_S) / \sigma_S$ of the VN error estimates \hat{e}_S in one direction to the reference. The mean μ_S and standard deviation σ_S refer to the normal distribution of errors computed in a

horizon around the sample, while already identified outliers in the horizon contribute to the statistics in a smaller degree, as chapter 5 described. Figure 7.19 shows the true VN position estimate error over time for the three drives of segment 2. That is, each step in the plots represents instances where both VN position estimates and reference positions are available. The plots annotate the detected anomalous samples too. An examination of these plots, in conjunction with Table 7.5, reveals that the VN encounters challenges with large errors with large variances during the Old Town 1 and Old Town 3 scenarios. In contrast, under the sunny conditions of the Old Town 2 scenario, the VN demonstrates the best performance in terms of position accuracy. Interestingly, Old Town 2 is the drive that presents the highest occurrence of statistically anomalous errors.

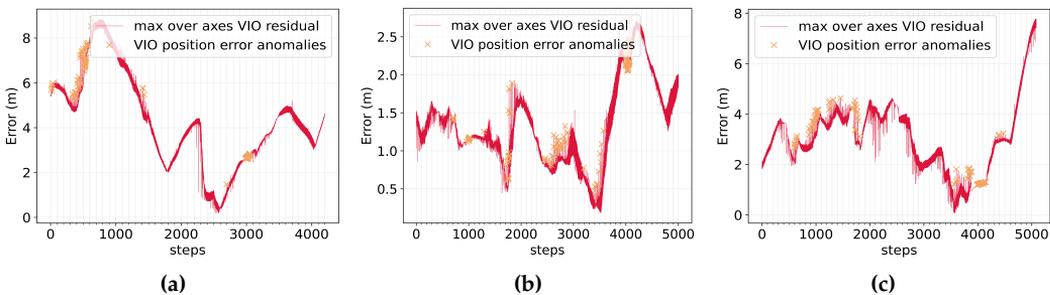


Figure 7.19: VN position estimate errors for segment 2 within all drives, annotating abnormal samples detected by their Z-Score within a constrained error sample horizon. (a) Old town 1, (b) Old town 2, and (c) Old town 3.

In figure 7.19, it is possible to identify time frames, in any of the drives, that appear particularly interesting for further analysis. After specifying the time frames of interest within one drive, the subsequent step entails mapping zero, one, or more position samples in the other two drives for the identified time frames. This mapping procedure considers that the car traverses corresponding locations in each drive in the same sequence, significantly reducing the likelihood of false mappings.

The analysis here considers three time frames of interest. The drives in bold letters are the ones considered significant for analysis around the respective time frames. The time frames in the other two drives were determined based on the mapping procedure. The time frames of interest are the following:

- Old town 1: steps 1464-1576, **Old town 2: steps 1714-1852**, Old town 3: steps 1710-1838: significant increase of the error in Old town 2 and many anomalies in Old town 2 and 3. The error is in a decreasing trend in Old town 1.
- Old town 1: steps 3355-3393, **Old town 2: steps 3943-4025**, Old town 3: steps 3832-3874: many anomalies in Old town 2 with an observable increase in the

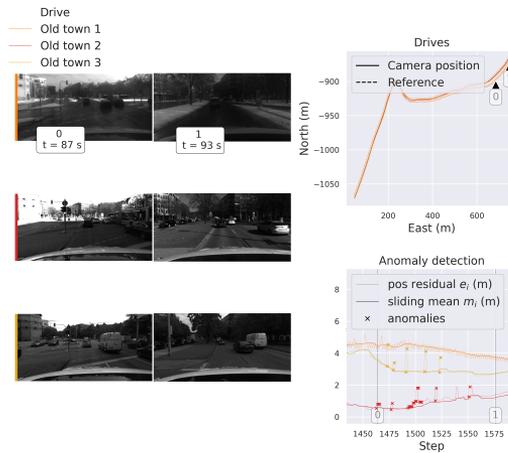


Figure 7.20: Example 1 for VN fault investigation.

error, many anomalies in Old town 3, but not significant increase in the error and no anomalies in Old town 1 with observable, though, error increase.

- Old town 1: steps 3703-4091, Old town 2: steps 4369-4867, **Old town 3: steps 4484-4968**: anomaly in Old town 3 and significant increase of the drift afterwards without recovery until the end. In contrast, Old town 1 and 2 observe a small decrease in the error.

The figures 7.20-7.22 provide a comparative analysis for the three selected time frames. The comparison includes images captured from the car's front camera and the evolution of VN residuals at specific steps within each time frame. The left side of the figures shows captured images; The top row exhibits images captured at different steps during the drive Old Town 1. The second and third rows display corresponding images from Old Town 2 and Old town 3, respectively. Each column of corresponding images is marked with a number and the time the image was captured while driving segment 2 in Old town 1. A specific color is assigned at each drive and each row with images is also marked with the color. The subfigures on the right depict information about the trajectory driven, the observed VN position residuals and the labeled anomalies during the three drives. The plots, corresponding to each drive, use the same color scheme as the images. The numbered markers on the plots show the exact location or step where the images, of the corresponding column, were obtained. The trajectory is displayed up to the location where the last image within the selected time frame was obtained. The residuals' plot is zoomed in around the locations of the images, showing information about what happens slightly before and after those locations.

In the first examined time frame (figure 7.20), the initial anomalies in Old town 2 appear to be due to the intersection and, probably, also due to unfavorable light conditions.

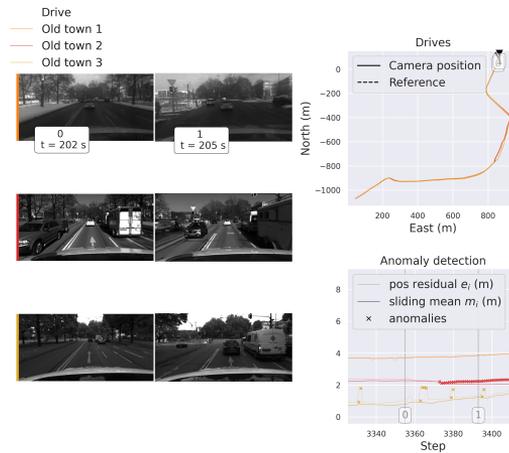


Figure 7.21: Example 2 for VN fault investigation.

The time frame spans approximately 6.4 seconds in Old town 1 and 7.4 seconds in old town 2. The latter is also the end of the anomalous region in Old town 2. Within this time frame there are many indications of anomalies in Old town 2 and 3. In both cases there is much heavier traffic than the one observed during Old town 1.

In the second examined time frame (figure 7.21) a long period of anomalous error samples is observable in Old town 2. It could be attributed to halted vehicles ahead and occlusion caused by the stationary bus/RV and van. This challenge in feature extraction and matching in the VN is further compounded by the busy nature of the intersection. The second time frame concludes as the car halts near the large vehicles. The anomalous samples persist even beyond this point.

In the final analyzed time frame (figure 7.22), anomalies become evident in the plot for Old town 3 at the commencement of the construction site (marker 0). Although the anomalous region ends quickly, a drift initiates at marker 1 after the van parks on the right, likely attributed to occlusion and mismatched features due to the construction site. The drift continues to increase for a long period (marker 2).

7.4 Chapter Summary

This chapter presents the practical application of the DHT module within the IM and evaluates the camera FDE, presented in chapter 6, and the PL inflation methodologies for potential VN and preintegrated IMU measurement faults, presented in chapter 4.

The first section incorporates the VN anomaly detection and exclusion module across datasets from UrbanLoco, Robotcar, and KAIST. The evaluation results show the system's ability to detect the vast majority of IM anomalies on-time. The IM system with anomaly exclusion shows comparable results, in terms of PL bounding performance,

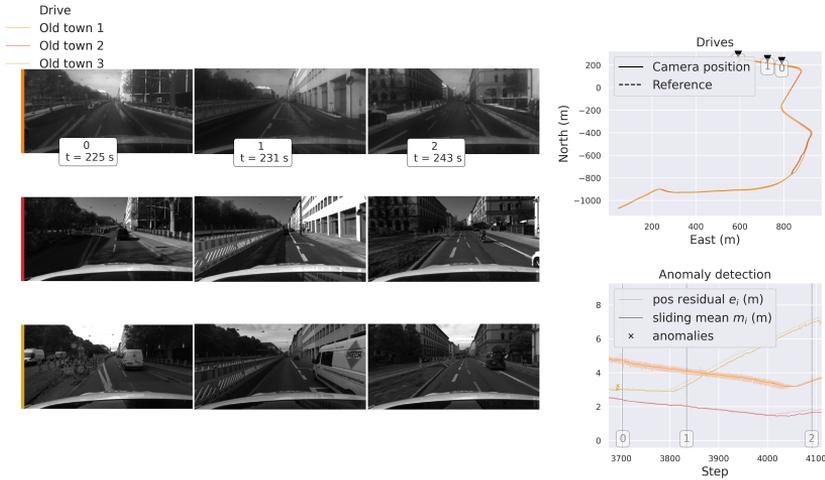


Figure 7.22: Example 3 for VN fault investigation.

to the typical MHSS-based IM which, in contrast to the presented method, is informed by the noise and faults of all sensors

The subsequent section focuses on validating the PL inflation methodology using the 4seasons dataset. The experiments highlight the methodology's ability to maintain system safety under the IMU-F and VN-F fault modes and in the presence of various types of artificially added faults, albeit revealing instances of loose PLs in fault-free scenarios. This suggests a potential for conditional inflation based on VN fault probabilities. Also, there is additional theoretical analysis required to improve the methodology and ensure sufficient bounds under different challenging conditions. Nonetheless, the results underscored the method's versatility in handling a wide array of faults. This, coupled with its ability to directly accommodate different auxiliary sensors and its independence from integration types, highlights its importance in accurately assessing the impact of faults on integrity risk. Overall, it plays a crucial role in advancing the development of a universal IM.

Moreover, the evaluation sheds light on some potentially challenging circumstances for VN navigation, linking recurrent VN anomalies to specific scene features or conditions. This is an enabler for the development of robust navigation systems capable of adapting to dynamic environmental conditions and efficient testing in predetermined challenging conditions.

In summary, the evaluation proved that the presented methodologies, in previous chapters, represent a significant stride in VN fault detection within multi-sensor integrations and the efficacy of the overall IM system.

Chapter 8

Conclusion

This thesis has presented a methodology to address critical challenges in IM of integrated navigation systems. The thesis began with a theoretical exploration of navigation algorithm uncertainty and sensor noise modeling and associated challenges with the utilization of auxiliary sensors (chapter 2). These challenges laid the groundwork for the subsequent development of the IM architecture. In my opinion, the thesis has made significant progress towards the establishment of an IM framework with scalability capabilities to multi-sensor integrations, independent from navigation algorithm internals and specific output requirements. The developed architecture and the findings in this thesis could serve as a base for future researches aiming the development of a universal IM used across various autonomous vehicle types undertaking safety critical operations.

Towards the development of the IM framework, chapter 3 extended MHSS frameworks for IM of integrations of inertial with position sensors (e.g. GNSS) to handle the case of time-correlated noise, detect sensor failures or mismodeling, and conditionally suggest error bounds inflation. Importantly, the method remains agnostic to the internal workings and computational complexity of the underlying navigation algorithm, allowing for versatility and scalability across different sensor platforms and noise models.

The chapter 4 proposed a method for PL computation suitable for systems that integrate a camera with conventional sensors, such as GNSS and IMU. This method involves a multi-level IM approach that quantifies the effect of various measurement subsets on integrity risk, allowing for safety guarantees specifically targeted to each sensor. The chapter further discussed the IMU and VN models in multi-sensor integrations, along with the respective residuals and PL inflation terms. The experimental validation of the proposed PL computation procedures for a GNSS/IMU/VN integration with factor graphs was presented in chapter 7, showcasing applicability of the developed methodology in the presence of various types of biases in IMU measurements, and VN

drifts.

Another central contribution was the development of a methodology for anomaly prediction within a third-party visual SLAM system. The methodology in chapter 5, leveraged a deep learning technique to identify anomalies in the estimates of the VN system, suggesting that similar approaches can be used for identifying anomalies in other auxiliary-sensor based navigation systems. By using convolutional and time distributed layers, the method achieved accuracy in predicting deviations from the expected VN error distribution. The method helped to learn low-level image features as well as dynamic changes in the visible area of the camera that potentially were causing the VN to fail. Such adaptable approaches, trained with auxiliary sensor measurements, have the potential to uncover correlations with anomalies in navigation, facilitating proactive anomaly detection and mitigation across diverse sensor-based navigation systems.

Furthermore, in chapter 6, the research investigated the relationship of VN fault detection and IM challenges related to the GNSS/IMU/VN integration in various urban environments. The experimentation was interested in a situation where the IM accounted for noise and faults in the CSS, but not in the camera measurements, assuming that the computed PLs would bound the estimate error of the full integration, in the absence of camera faults. This assumption is generally valid as the integration of more sensors reduces the uncertainty of the navigation algorithm, with this uncertainty being consistent in the absence of sensor faults. Therefore, in this case, an IM anomaly, i.e. a time instance with PLs that do not bound the position error, happens when a camera fault, that affects significantly the integrated system's error, is not detected and excluded. The experimentation, presented in chapter 7, demonstrated the efficacy of the developed camera FDE methodology in proactively excluding camera faults and achieving a significant increase of time without IM anomalies.

The IM architecture introduced in this thesis offers several notable advantages, including scalability to diverse sensor integrations, independence from monitored navigation algorithm internals, and robust fault detection capabilities. By employing techniques such as deep learning-based anomaly prediction and hypothesis testing individually for each auxiliary sensor, the proposed framework enables early detection and mitigation of hazardous risks in navigation associated with sensor faults. It also facilitates the identification of high-level environmental conditions that potentially cause these faults.

8.1 Future Directions

Looking ahead, several areas for future research emerge from this work. There is a need to quantify the direct effects of a wide set of environmental conditions on the performance of IM, particularly regarding individual sensors. This could include conducting thorough investigations into how factors such as weather, illuminations,

surrounding building characteristics and density, traffic and behavior of other users impact sensor and IM accuracy.

Another important area for future work is the automatic detection of unseen anomalies within the IM framework. The ability to automatically detect more unseen anomalies in unlabeled data is essential for systems that utilize auxiliary sensors affected by many environment particularities which are currently barely researched. This would lead to better understanding and definition of specific datasets to test the navigation system that utilizes a particular sensor to evaluate it in terms of integrity and continuity requirements.

Evaluation to other multiple sensor integrations, including tightly-coupled, is another aspect that requires attention in future research work. The IM framework should be tested with integrations with multiple auxiliary sensors beyond the current scope.

Moreover, there is a need for continued research to improve sensor fault modeling techniques within the IM framework. Refining these techniques is essential for enhancing the accuracy of anomaly prediction and detection. Researchers can explore advanced machine learning methods and incorporate domain-specific knowledge to better capture sensor behaviors and fault patterns. By developing more sophisticated models for sensor fault detection and diagnosis, IM frameworks can effectively mitigate integrity risks and improve overall system reliability.

Finally, enhanced performance evaluation is critical for validating the universality of the IM framework under an extensive set of various real-world conditions. Further experimentation and evaluation is needed to assess the performance of the IM in different environments, vehicle types, and sensor configurations. By conducting comprehensive testing and validation procedures, researchers can gain valuable insights into the strengths and limitations of the IM system, guiding its ongoing development.

8.2 Final remarks

In summary, this thesis represents a significant contribution to the field of IM of integrated navigation systems, offering practical solutions to critical challenges and paving the way for targeted testing, evaluation of IMs with objective criteria for universally used datasets, and effective reporting of compliance to safety requirements to relevant authorities. As we and other researchers continue to push the boundaries of innovation in this domain, the insights and methodologies presented here will serve as valuable foundations for future research and development efforts of universal IMs.

Appendix A

Derivation of the positioning uncertainty in SLAM

This appendix provides an analytical description of the closed-form solution for SLAM uncertainty of Mourikis and Roumeliotis [32], briefly introduced in Section 2.3.

A.1 1D case

The state vector is the combination of the position of the robot and the position of the N landmarks. Let the input be a measurement of the velocity of the robot, and this measurement leads the propagation of the state. The dynamics of the covariance of the position estimates $\mathbf{P}(t)$ of the robot and landmarks is:

$$\dot{\mathbf{P}}(t) = \mathbf{G}q\mathbf{G}^T - \mathbf{P}(t)\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{P}(t)$$

With $\mathbf{G} = [1 \quad \mathbf{0}_{1 \times N}]^T$, q the variance of the noise of the input (the variance is considered constant, and the noise white-gaussian with zero mean), \mathbf{R} is the covariance matrix of the noise (assumed white-gaussian with zero-mean) of the exteroceptive measurements of the relative positions -to the robot- of the landmarks at each time, and \mathbf{H} the measurement matrix (association to the state). \mathbf{H} can be viewed as the incidence matrix of a directed graph with $N + 1$ vertices, where each of N vertices (landmarks) are connected with one edge to the vertex corresponding to the robot. The normalized covariance can be defined as $\mathbf{P}_n(t) = \frac{1}{q}\mathbf{P}(t) \Rightarrow \mathbf{P}(t) = q\mathbf{P}_n(t)$. With substitution and algebraic manipulations we end up with the expression

$$\mathbf{P}_n(t) = \mathbf{U}^{-T}(\mathbf{Q}_n\mathbf{L}(t) + \mathbf{K}(t)\mathbf{P}_0)(\mathbf{K}(t) + \mathbf{U}^{-1}\mathbf{C}\mathbf{U}^{-T}\mathbf{L}(t)\mathbf{P}_0)^{-1}\mathbf{U}^{-1}$$

which is equivalent to

$$\mathbf{P}_n(t) = \mathbf{U}^{-T}\mathbf{Q}_n\mathbf{L}(t)\mathbf{K}^{-1}(t)\mathbf{U}^{-1} + \mathbf{U}^{-T}\mathbf{M}(t)\mathbf{U}^{-1},$$

with $\mathbf{C} = \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}$, $\mathbf{Q}_n = \begin{bmatrix} \mathbf{1} & \mathbf{0}_{1 \times N} \\ \mathbf{0}_{N \times 1} & \mathbf{0}_{N \times N} \end{bmatrix}$, $\mathbf{L}(t) = \begin{bmatrix} \frac{e^{\rho t} + e^{-\rho t}}{2} & \mathbf{0}_{1 \times N} \\ \mathbf{0}_{1 \times N} & \mathbf{I}_{N \times N} \end{bmatrix}$, $\mathbf{K}(t) = \begin{bmatrix} \frac{1}{\rho} \left(\frac{e^{\rho t} - e^{-\rho t}}{2} \right) & \mathbf{0}_{1 \times N} \\ \mathbf{0}_{1 \times N} & \mathbf{I}_{N \times N} \end{bmatrix}$,
 $\rho^2 = \mathbf{1}_{1 \times N} \mathbf{R}_n^{-1} \mathbf{1}_{N \times 1}$, $\mathbf{P}_0 = \mathbf{U}^T \mathbf{P}_n(0) \mathbf{U}$, \mathbf{U} is the modal matrix of $\mathbf{C} \mathbf{Q}_n$, $\mathbf{P}_n(0) = \begin{bmatrix} \mathbf{P}_{rr} & \mathbf{P}_{rL} \\ \mathbf{P}_{Lr} & \mathbf{P}_{LL} \end{bmatrix}$
 the initial value of the covariance matrix (subscripts “r” and “L” for “robot” and “Landmarks” respectively), and $\mathbf{M}(t) = \mathbf{K}^{-1}(t) \mathbf{P}_0 (\mathbf{K}(t) + \mathbf{U}^{-1} \mathbf{C} \mathbf{U}^{-T} \mathbf{L}(t) \mathbf{P}_0)^{-1}$.

The normalized covariance at steady state (at the limit $t \rightarrow \infty$) will be computed:

$$\mathbf{P}_{n_{ss}} = \lim_{t \rightarrow \infty} \mathbf{P}_n(t) = \begin{bmatrix} \frac{1}{\rho} + \mathbf{m}_{nn} & \mathbf{m}_{nn} \mathbf{1}_{1 \times N} \\ \mathbf{m}_{nn} \mathbf{1}_{N \times 1} & \mathbf{m}_{nn} \mathbf{1}_{N \times N} \end{bmatrix}$$

With $\mathbf{m}_{nn} = \frac{1}{N^2} \sum (\mathbf{F}(\infty) \mathbf{V}_{2\infty} (\mathbf{U}_{2\infty}^T \mathbf{V}_{2\infty})^{-1} \mathbf{U}_{2\infty}^T)$,

$$\mathbf{F}(\infty) = \mathbf{P}_{LL} - \rho \left(\frac{1}{1 + \rho \mathbf{P}_{rr} - \frac{2}{\rho} r^T \mathbf{P}_{Lr} + \frac{1}{\rho^3} r^T \mathbf{P}_{LL} r} \right) \left(\mathbf{P}_{Lr} - \frac{1}{\rho^2} \mathbf{P}_{LL} r \right) \left(\mathbf{P}_{rL} - \frac{1}{\rho^2} r^T \mathbf{P}_{LL} \right),$$

$\mathbf{r} = q \mathbf{R}^{-1} \mathbf{1}_{N \times 1}$, and $\mathbf{U}_2^\infty, \mathbf{V}_2^\infty$ are matrices whose column vectors are basis vectors of the left and right nullspace of $\left(q \mathbf{R}^{-1} - \frac{1}{\rho^2} r r^T \right) \mathbf{F}_1(\infty)$. Finally, multiply with q to find the non-normalized covariance at steady state.

A.2 2D case

The authors treat the velocity and orientation of the robot as measurement inputs. The covariance matrix of the process noise will be given by $\mathbf{Q}(t) = \begin{bmatrix} \mathbf{Q}_r(t) & \mathbf{0}_{2 \times 2N} \\ \mathbf{0}_{2N \times 2} & \mathbf{0}_{2N \times 2N} \end{bmatrix}$, with
 $\mathbf{Q}_r(t) = \begin{bmatrix} \cos(\hat{\phi}(t)) & -\mathbf{V}_m(t) \sin(\hat{\phi}(t)) \\ \sin(\hat{\phi}(t)) & \mathbf{V}_m(t) \cos(\hat{\phi}(t)) \end{bmatrix} \begin{bmatrix} \sigma_V^2 & 0 \\ 0 & \sigma_\phi^2 \end{bmatrix} \begin{bmatrix} \cos(\hat{\phi}(t)) & -\mathbf{V}_m(t) \sin(\hat{\phi}(t)) \\ \sin(\hat{\phi}(t)) & \mathbf{V}_m(t) \cos(\hat{\phi}(t)) \end{bmatrix}^T$, $\mathbf{V}_m(t)$
 being the velocity measurements, $\hat{\phi}$ being the robot’s orientation estimate, σ_V^2 being the variance of noise sequence affecting the velocity measurements, and σ_ϕ^2 being the variance of noise of the error in the robot’s orientation estimate. Taking into account that the measurements are correlated, the expression of the covariance matrix of all the measurements performed by the robot can be written as a combination of the covariance $\mathbf{R}_1(t)$ of the error due to the noise in the range measurements, of the covariance $\mathbf{R}_2(t)$ due to the error in the bearing measurements, and of the covariance $\mathbf{R}_3(t)$ due to the error in the orientation estimates:

$$\begin{aligned} \mathbf{R}(t) &= \mathbf{D}_{\hat{\phi}(t)}^T \mathbf{R}_0(t) \mathbf{D}_{\hat{\phi}(t)} & (\text{A.1}) \\ \mathbf{R}_0(t) &= \mathbf{R}_1(t) + \mathbf{R}_2(t) + \mathbf{R}_3(t) \\ \mathbf{R}_1(t) &= \sigma_\rho^2 \mathbf{I}_{2N \times 2N} - \mathbf{D}(t) \text{diag} \left(\frac{\sigma_{\rho_i}^2}{\rho_i^2(t)} \right) \mathbf{D}^T(t) \end{aligned}$$

$$\mathbf{R}_2(t) = \sigma_{\theta_i}^2 \mathbf{D}(t) \mathbf{D}^T(t)$$

$$\mathbf{R}_3(t) = \sigma_{\phi_i}^2 \mathbf{D}(t) \mathbf{1}_{N \times N} \mathbf{D}^T(t),$$

where $\mathbf{D}_{\hat{\phi}}(t) = \mathbf{I}_{N \times N} \otimes C(\hat{\phi}(t))$, $C(\hat{\phi}(t))$ is the rotation matrix associated with the robot's orientation estimate $\hat{\phi}$, σ_{ϕ}^2 is the orientation uncertainty (an upper bound is considered known in this case – the authors amplify with the distance between the robot and corresponding landmark), σ_{ρ}^2 is the variance of the noise in the distance measurements, $\sigma_{\theta_i}^2$ is the variance of the noise in the bearing measurements, and

$$\mathbf{D}(t) = \begin{bmatrix} J\Delta\hat{p}_1(t)^T & \dots & \mathbf{0}_{2 \times 1} \\ \vdots & \ddots & \vdots \\ \mathbf{0}_{2 \times 1} & \dots & J\Delta\hat{p}_N(t) \end{bmatrix} \text{diagonal matrix depending on the positions of the}$$

robot and landmarks. The \otimes denotes the Kronecker matrix product. Then the matrix $\mathbf{H}^T(t)\mathbf{R}^{-1}(t)\mathbf{H}(t)$ that appears in the covariance update equations of the Kalman Filter depend only on the positions of the robot and landmarks and will be $\mathbf{H}^T(t)\mathbf{R}_0^{-1}(t)\mathbf{H}_0$. Finally, the Riccati differential equation for the evolution of the covariance:

$$\dot{\mathbf{P}}(t) = \mathbf{Q}(t) - \mathbf{P}(t)\mathbf{H}_0^T(t)\mathbf{R}_0^{-1}(t)\mathbf{H}(t)\mathbf{P}(t),$$

which does not have a closed form solution. The authors derive an upper bound:

$$\dot{\bar{\mathbf{P}}}(t) = \bar{\mathbf{Q}} - \bar{\mathbf{P}}(t)\mathbf{H}_0^T\bar{\mathbf{R}}^{-1}\mathbf{H}_0\bar{\mathbf{P}}(t)$$

In case $\bar{\mathbf{R}} \geq \mathbf{R}_0(t)$ and $\bar{\mathbf{R}} \geq \mathbf{R}_0(t)$ for all $t > 0$ Considering that the upper bound of the individual terms should be $\bar{\mathbf{R}} = (\sigma_{\rho}^2 + N\sigma_{\phi}^2\rho_0^2 + \sigma_{\theta}^2\rho_0^2)\mathbf{I}_{N \times 2N}$ and $\bar{\mathbf{Q}} = \begin{bmatrix} \bar{q}\mathbf{I}_{2 \times 2} & \mathbf{0}_{2 \times 2N} \\ \mathbf{0}_{2N \times 2} & \mathbf{0}_{2N \times 2N} \end{bmatrix}$ with $\bar{q} = \max(\delta t^2\sigma_v^2, \delta t^2V^2\sigma_{\phi}^2)$ assuming approximately constant velocity V .

$$\mathbf{P}(0) = \begin{bmatrix} \mathbf{P}_{rr0} & \mathbf{0}_{2 \times 2N} \\ \mathbf{0}_{2N \times 2} & \mathbf{P}_{LL0} \end{bmatrix}$$

The upper bound of the steady-state covariance matrix will be (after similar steps as with the 1D case):

$$\mathbf{P}_{n_{ss}} = \begin{bmatrix} \sqrt{\frac{\bar{q}r}{N}}\mathbf{I}_{2 \times 2} & \mathbf{0}_{2 \times 2N} \\ \mathbf{0}_{N \times 2} & \mathbf{0}_{2N \times 2N} \end{bmatrix} + \mathbf{1}_{(N+1) \times (N+1)} \otimes \left(\mathbf{J}^T \mathbf{P}_{LL0}^{-1} \mathbf{J} + \left(\sqrt{\frac{\bar{q}r}{N}} \mathbf{I}_{(2 \times 2)} + \mathbf{P}_{rr0} \right)^{-1} \right)^{-1},$$

with $r = \sigma_{\rho}^2 + N\sigma_{\phi}^2\rho_0^2 + \sigma_{\theta}^2\rho_0^2$, \mathbf{P}_{rr0} being the initial covariance of the robot's position estimate and \mathbf{P}_{LL0} being the initial map covariance.

Appendix B

ErKF and Gauss-Markov Process Noise Overbounding

IMUs often suffer stochastic errors that cannot be estimated. It is common to model these errors as the sum of random constant turn-on-bias, a time-correlated process and a white Gaussian noise term [46]. Therefore, the angular rate and specific force measurements responsible for the linear acceleration of the sensor can be expressed as:

$$\tilde{\boldsymbol{w}}^b = \boldsymbol{w}^b + \boldsymbol{b}_{w,0} + \boldsymbol{b}_w + \boldsymbol{n}_w \quad (\text{B.1})$$

$$\tilde{\boldsymbol{f}}^b = \boldsymbol{f}^b + \boldsymbol{b}_{f,0} + \boldsymbol{b}_f + \boldsymbol{n}_f, \quad (\text{B.2})$$

where $\tilde{\boldsymbol{w}}^b$ and $\tilde{\boldsymbol{f}}^b$ are respectively the measured turn rates and specific forces in 3 axes with respect to the body frame b , \boldsymbol{w}^b and \boldsymbol{f}^b their true values, $\boldsymbol{b}_{w,0}$ and $\boldsymbol{b}_{f,0}$ are respectively the turn-on random constant bias of the angular rates and the specific forces, \boldsymbol{b}_w and \boldsymbol{b}_f are the time-correlated biases, and \boldsymbol{n}_w and \boldsymbol{n}_f are white Gaussian noise vectors. The time-correlated biases are usually modeled with a Gauss-Markov process.

The ErKF estimated error state $\delta\boldsymbol{x}$ consists of the errors in position $\delta\rho$, velocity, angles vector $\delta\boldsymbol{\theta}$, linear acceleration bias $\delta\boldsymbol{b}_a$ and angular velocity bias $\delta\boldsymbol{b}_w$. In our experiments the second position sensor is affected by time-correlated noise and, therefore, we have to account for it as well. Generally, the state vector can be augmented with additional states to account for additional time correlated errors, e.g. in position sensor's measurements. Therefore, in the general case, the state vector becomes:

$$\delta\boldsymbol{x} = (\delta\rho, \delta\boldsymbol{v}, \delta\boldsymbol{\theta}, \delta\boldsymbol{b}_a, \delta\boldsymbol{b}_w, \delta\boldsymbol{b}_{\text{pos}_1}, \dots, \delta\boldsymbol{b}_{\text{pos}_m})^T \in \mathbb{R}^{15+3m}, \quad (\text{B.3})$$

$$\mathbf{x}_t = \begin{bmatrix} \rho + \delta\rho \\ \mathbf{v} + \delta\mathbf{v} \\ \mathbf{q} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta} \end{bmatrix} \\ \mathbf{b}_a + \delta\mathbf{b}_a \\ \mathbf{b}_w + \delta\mathbf{b}_w \\ \mathbf{b}_{\text{pos}_1} + \delta\mathbf{b}_{\text{pos}_1} \\ \vdots \\ \mathbf{b}_{\text{pos}_m} + \delta\mathbf{b}_{\text{pos}_m} \end{bmatrix} \quad (\text{B.10})$$

Substituting in eq. (B.4) and by linearizing around $\delta\mathbf{x} = \mathbf{0}$ (error state is small):

$$\mathbf{z} \approx \mathbf{h}(\mathbf{x} \oplus \mathbf{0}) + \left. \frac{\partial \mathbf{h}(\mathbf{x} \oplus \delta\mathbf{x})}{\partial \delta\mathbf{x}} \right|_{\delta\mathbf{x}=\mathbf{0}} \delta\mathbf{x} + \mathbf{w} = \mathbf{h}(\mathbf{x}) + \mathbf{H}\delta\mathbf{x} + \mathbf{w}, \quad (\text{B.11})$$

with \mathbf{H} denoting the Jacobian. Having this model, we can directly retrieve the updated error state and covariance as in the EKF framework.

In the experiments presented in this paper, the fixes are directly the measurements from the position sensors and the matrix \mathbf{H} for the position update from POS_SENSOR i will be:

$$\mathbf{H}_{\text{pos}_i} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & -R(\mathbf{q})S(\mathbf{r}_{\text{pos}_i}) & \mathbf{0}_{3 \times 6} & \vdots & \mathbf{H}'_{\text{pos}_i} \end{bmatrix} \in \mathbb{R}^{3 \times (15+3m)} \quad (\text{B.12})$$

where $\mathbf{r}_{\text{pos}_i}$ the lever arm from body to POS_SENSOR i and $\mathbf{H}'_{\text{pos}_i}$ is a $3 \times 3m$ matrix which is non-zero only for the elements that correspond to $\mathbf{b}_{\text{pos}_i}$:

$$\mathbf{H}'_{\text{pos}_i} = \begin{bmatrix} \mathbf{0}_{3 \times 3(i-1)} & \mathbf{I} & \mathbf{0}_{3 \times 3(m-i)} \end{bmatrix} \quad (\text{B.13})$$

This completes an introduction to our ErKF implementation, where we focused on showing the application of the overbounding method.

Appendix C

LLR for faulty sensor detection

Let:

y_k^0 the subset that contains all measurements at time k .

$y_k^j = [\rho_k^1, \rho_k^2, \dots, \rho_k^{(j-1)}, \rho_k^{(j+1)}, \dots, \rho_k^m]$ the subset that contains all measurements except the j -th (m the total number of measurements at time k).

We define the likelihood:

$$p(y_k^j | x_k) = \prod_{w=1 \text{ and } w \neq j}^m \frac{1}{\sqrt{2\pi}c_k} e^{-\frac{\rho_k^{(w)} - \tilde{\rho}_k^{(w)}}{2c_k}}, \quad (\text{C.1})$$

where ρ_k is the observation from the sensor, $\tilde{\rho}_k$ the algorithm's estimation for the same measurement.

For a time range a to b , the cumulative LLR between two observation subsets is:

$$S_a^{(b,j)} = \sum_{k=a}^b \ln \left(\frac{p(y_k^j | x_k)}{p(y_k^0 | x_k)} \right), \quad 1 \leq j \leq m \quad (\text{C.2})$$

Define $\beta_k^j = \max_{(k-x+1 \leq a \leq k)} \{S_a^{(k,j)}\}$, where x is the accumulation time window size. And

$\beta_k = [\beta_k^1, \beta_k^2, \dots, \beta_k^m]$ is the test statistic.

So:

- If we have sensors that provide redundant measurements
- If we assume only up to one faulty sensor at each time

Then having hypotheses with one sensor excluded at each, a hypothesis that correctly assumes a sensor faulty will have much higher LLR than other hypotheses. Intuitively, it can be seen as a measurement of better agreement of all sensors.

References

- [1] S. Li, X. Li, H. Wang, Y. Zhou, and Z. Shen, "Multi-GNSS PPP/INS/Vision/LiDAR tightly integrated system for precise navigation in urban environments," *Information Fusion*.
- [2] P. Bonnifait, "Localization integrity for intelligent vehicles: How and for what?" 2022.
- [3] E. G. Agency. (2015) Report on the performance and level of integrity for safety and liability critical multi-applications. Accessed 11 Jan 2022. [Online]. Available: https://www.gsa.europa.eu/sites/default/files/calls_for_proposals/Annex%202.pdf
- [4] M. Joerger and M. Spenko. (2017) Towards navigation safety for autonomous cars. Accessed 19 Feb 2024. [Online]. Available: <https://insidegnss.com/wp-content/uploads/2018/01/novdec17-JOERGER.pdf>
- [5] M. Joerger, F.-C. Chan, and B. Pervan, "Solution separation versus residual-based RAIM." *J Inst Navig*, 2014.
- [6] S.-M. Calhoun and J. Raquet, "Integrity determination for a vision-based precision relative navigation system," in *2016 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2016, pp. 294–304.
- [7] C. Li and S.-L. Waslander, "Visual measurement integrity monitoring for UAV localization," in *2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2019.
- [8] M. V. Torres, T. E. Tabassum, I. Petrunin, and A. Tsourdos, "Hybrid multi-sensor navigation system with uncertainty correction for GNSS-denied environments," *AIAA SCITECH 2024 Forum*, 2024.

- [9] J. Lee, M. Kim, D. Min, and J. Lee, "Integrity algorithm to protect against sensor faults in tightly-coupled KF state prediction," *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*, 2019.
- [10] T. Hassan, A. El-Mowafy, and K. Wang, "A review of system integration and current integrity monitoring methods for positioning in intelligent transport systems." *IET Intell Transp Syst.*, 2020.
- [11] J. Blanch, T. Walker, P. Enge, Y. Lee, B. Pervan, M. Rippl, A. Spletter, and V. Kropp, "Baseline advanced RAIM user algorithm and possible improvements," *IEEE Transactions on Aerospace and Electronic Systems*, 2015.
- [12] S. Wang, Y. Zhai, and X. Zhan, "Implementation of solution separation-based kalman filter integrity monitoring against all-source faults for multi-sensor integrated navigation," *GPS Solutions*, 2023.
- [13] S. Baldoni, F. Battisti, M. Brizzi, and A. Neri, "GNSS-Imaging data fusion for integrity enhancement in autonomous vehicles," *IEEE Transactions on Aerospace and Electronic Systems*, 2022.
- [14] A. Appleget, R.-C. Leishman, and J. Gipson, "Evaluation of sensor-agnostic all-source residual monitoring for navigation," in *Proceedings of the 2021 International Technical Meeting of The Institute of Navigation*, 2021.
- [15] Q. Meng and L.-T. Hsu, "Integrity monitoring for all source navigation enhanced by Kalman Filter based solution separation," *IEEE Sensors Journal*, 2020.
- [16] M. Joerger, C. Chan, S. Langel, and B. Pervan, "RAIM detector and estimator design to minimize the integrity risk," in *ION GNSS*, 2012.
- [17] S. Gupta and G.-X. Gao, "Particle RAIM for integrity monitoring," in *ION GNSS+*, 2019.
- [18] A. Mohanty, S. Gupta, and G.-X. Gao, "A particle filtering framework for integrity risk of GNSS-camera sensor fusion," in *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, 2021.
- [19] S. Bhamidipati and G.-X. Gao, "SLAM-based integrity monitoring using GPS and fish-eye camera," in *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation, ION GNSS+ 2019*, 2019, pp. 4116–4129.
- [20] G.-D. Arana, O.-A. Hafez, M. Joerger, and M. Spenko, "Recursive integrity monitoring for mobile robot localization safety," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 305–311.

-
- [21] P. He, G. Liu, C. Tan, and Y.-E. Lu, "Nonlinear fault detection threshold optimization method for RAIM algorithm using a heuristic approach," *GPS Solut*, vol. 20, pp. 863–875, 2016.
- [22] R. S. Y. Young and G. A. McGraw, "Fault detection and exclusion using normalized solution separation and residual monitoring methods," *NAVIGATION: Journal of the Institute of Navigation*, 2003.
- [23] H. Yang, , C. Meng, and C. Wang, "A hybrid data-driven fault detection strategy with application to navigation sensors," *Measurement and Control*, 2020.
- [24] F. Rothmaier and J. A. Del Peral Rosado, "A parametric study on autonomous integrity monitoring using non-gnss signals," in *2023 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. IEEE, 2023.
- [25] J. A. Hage, P. Xu, P. Bonnifait, and J. Ibanez-Guzman, "Localization integrity for intelligent vehicles through fault detection and position error characterization," *IEEE Transactions on Intelligent Transportation Systems*, 2022.
- [26] V. Bosdelekidis, T. H. Bryne, N. Sokolova, and T. A. Johansen, "Navigation algorithm-agnostic integrity monitoring based on solution separation with constrained computation time and sensor noise overbounding." *J Intell Robot Syst*, 2022.
- [27] V. Bosdelekidis, T. A. Johansen, and N. Sokolova, "DNN-based anomaly prediction for the uncertainty in visual SLAM," *17th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2022.
- [28] V. Bosdelekidis, T. A. Johansen, N. Sokolova, and T. H. Bryne, "Solution separation-based integrity monitor for integrated GNSS/IMU/Camera Navigation: Constraining the hypothesis space with deep learning," *2023 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2023.
- [29] M. B. Marinov, B. Ganev, N. Djermanova, and T. D. Tashev, "Analysis of sensors noise performance using allan deviation," *2019 IEEE XXVIII International Scientific Conference Electronics (ET)*, 2019.
- [30] K. Jerath, S. Brennan, and C. Lagoa, "Bridging the gap between sensor noise modeling and sensor characterization," *Measurement*, 2018.
- [31] M. Servières, V. Renaudin, A. Dupuis, and N. Antigny, "Visual and visual-inertial slam: State of the art, classification, and experimental benchmarking," *Journal of Sensors*, 2021.
- [32] A. I. Mourikis and S. I. Roumeliotis, "Analysis of positioning uncertainty in simultaneous localization and mapping (SLAM)," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.

- [33] Z. Zhu and C. Taylor, "Conservative uncertainty estimation in map-based vision-aided navigation," *IEEE Transactions on Aerospace and Electronic Systems*, 2017.
- [34] M. Tkocz and K. Janschek, "Towards consistent state and covariance initialization for monocular SLAM filters," *Journal of Intelligent and Robotic Systems*, vol. 80, pp. 475–489, 2015.
- [35] H. Carrillo, I. Reid, and J. A. Castellanos, "On the comparison of uncertainty criteria for active slam," *International Conference on Robotics and Automation*, 2012.
- [36] C. Tanil, S. Khanafseh, M. Joerger, and B. Pervan, "Sequential integrity monitoring for Kalman Filter innovations-based detectors," in *Proceedings of the 31st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2018)*, 2018, pp. 2440–2455.
- [37] J. Blanch, A. Ene, T. Walter, and P. Enge, "An optimized multiple hypothesis RAIM algorithm for vertical guidance," *Proceedings of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation*, 2007.
- [38] T. Hassan, A. El-Mowafy, and K. Wang, "A review of system integration and current integrity monitoring methods for positioning in intelligent transport systems," *IET Intelligent Transport Systems*, 2021.
- [39] M. Zhong, T. Xue, and S. X. Ding, "A survey on model-based fault diagnosis for linear discrete time-varying systems," *Neurocomputing*, vol. 306, pp. 51–60, 2018.
- [40] K. Salahshoor, M. Mosallaei, and M. Bayat, "Centralized and decentralized process and sensor fault monitoring using data fusion based on adaptive extended kalman filter algorithm," *Measurement*, vol. 41, no. 10, pp. 1059–1076, 2008.
- [41] J. Jurado, J. Raquet, C. Schubert Kabban, and J. Gipson, "Residual-based multi-filter methodology for all-source fault detection, exclusion, and performance monitoring," *Navigation*, vol. 67, pp. 493–509, 2020.
- [42] A. Vafamand, B. Moshiri, and N. Vafamand, "Fusing unscented kalman filter to detect and isolate sensor faults in dc microgrids with cpls," *IEEE Transactions on Instrumentation and Measurement*, 2021.
- [43] M. Saied, A. R. Tabikh, C. Francis, H. Hamadi, and B. Lussier, "An informational approach for fault tolerant data fusion applied to a uav's attitude, altitude, and position estimation," *IEEE Sensors Journal*, vol. 21, no. 24, pp. 27 766–27 778, 2021.
- [44] J. Zarei and E. Shokri, "Robust sensor fault detection based on nonlinear unknown input observer," *Measurement*, vol. 48, pp. 355–367, 2014.

-
- [45] M. Pazera, M. Buciakowski, and M. Witczak, "Robust multiple sensor fault-tolerant control for dynamic non-linear systems: Application to the aerodynamical twin-rotor system," *International Journal of Applied Mathematics and Computer Science*, vol. 28, no. 2, 2018.
- [46] O. G. Crespillo, M. Joerger, and S. Langel, "Overbounding GNSS/INS integration with uncertain GNSS gauss-markov error parameters," in *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2020, pp. 481–489.
- [47] S.-C. Mohleji and G. Wang, "Modeling ADS-B position and velocity errors for airborne merging and spacing in interval management application," The MITRE Corporation, Tech. Rep., 2010.
- [48] S. Langel, O.-G. Crespillo, and M. Joerger, "Bounding sequential estimation errors due to Gauss-Markov noise with uncertain parameters," in *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*, Miami, Florida, 2019, pp. 3079–3098.
- [49] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res. (IJRR)*, 2013.
- [50] R. G. Brown and P.-Y.-C. Hwang, *Transition from the discrete to continuous filter equations*. Wiley, 2012, pp. 371–375.
- [51] V. Madyastha, V. Ravindra, S. Mallikarjunan, and A. Goyal, "Extended Kalman Filter vs. Error State Kalman Filter for aircraft attitude estimation," in *AIAA Guidance, Navigation, and Control Conference 2011*, 2011.
- [52] 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 *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, 1999, pp. 1656–1663 vol.2.
- [53] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," Georgia Tech Technical Report, GT-RIM-CP&R-2012–002, Tech. Rep., 2012.
- [54] T. Lupton and S. Sukkarieh, "Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions," *IEEE Trans. Robot.*, vol. 28, no. 1, 2012.
- [55] C. Milner, C. Macabiau, and P. Thevenon, "Bayesian inference of gnss failures," *Journal of Navigation*, 2016.
- [56] S. Wang, X. Zhan, Y. Zhai, L. Zheng, and B. Liu, "Enhancing navigation integrity for urban air mobility with redundant inertial sensors," *Aerospace Science and Technology*, vol. 126, 2022.

- [57] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, 2018.
- [58] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual-inertial odometry," *IEEE Transactions on Robotics*, 2017.
- [59] C. Zhu, M. Joerger, and M. Meurer, "Quantifying feature association error in camera-based positioning," *IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2020.
- [60] A. Ceccarelli and F. Secci, "On failures of RGB cameras and their effects in autonomous driving applications," *IEEE 31st International Symposium on Software Reliability Engineering (ISSRE)*, 2020.
- [61] W. Zhou, J. S. Berrio, S. Worrall, and E. Nebot, "Automated evaluation of semantic segmentation robustness for autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, 2020.
- [62] C. Zhu, M. Meurer, and C. Günther, "Integrity of visual navigation—developments, challenges, and prospects," *NAVIGATION: Journal of the Institute of Navigation*, 2022.
- [63] J. A. Hage, P. Xu, and P. Bonnifait, "High integrity localization with multi-lane camera measurements," *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019.
- [64] L. Fu, J. Zhang, R. Li, X. Cao, and J. Wang, "Vision-aided RAIM: A new method for GPS integrity monitoring in approach and landing phase," *Sensors*, vol. 15, pp. 22 854–22 873, 2015.
- [65] Y. Fu, S. Wang, Y. Zhai, X. Zhan, and X. Zhang, "Measurement error detection for stereo visual odometry integrity," *NAVIGATION: Journal of the Institute of Navigation*, 2022.
- [66] S. Gupta and G. Gao, "Data-driven protection levels for camera and 3d map-based safe urban localization," *NAVIGATION: Journal of the Institute of Navigation*, 2021.
- [67] S. Wang, X. Zhan, Y. Fu, and Y. Zhai, "Feature-based visual navigation integrity monitoring for urban autonomous platforms," *Aerospace Systems*, 2020.
- [68] C. Zhu, M. Meurer, and M. Joerger, "Integrity analysis for greedy search based fault exclusion with a large number of faults," *2023 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2023.
- [69] H. Jing, Y. Gao, S. Shahbeigi, and M. Dianati, "Integrity monitoring of GNSS/INS based positioning systems for autonomous vehicles: State-of-the-art and open challenges," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

-
- [70] J. L. Blanco-Claraco, "A tutorial on $\mathbb{SE}(3)$ transformation parameterizations and on-manifold optimization," *arXiv:2103.15980*, 2022.
- [71] C. Mannila, "Robustness of state-of-the-art visual odometry and SLAM systems," (Dissertation). Retrieved from <https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-328733>, 2017.
- [72] W. Wen, Y. Zhou, G. Zhang, S. Fahandezh-Saadi, X. Bai, W. Zhan, M. Tomizuka, and L.-T. Hsu, "Urbanloco: A full sensor suite dataset for mapping and localization in urban scenes," *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.
- [73] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000km: The Oxford RobotCar dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017.
- [74] W. Maddern, G. Pascoe, M. Gadd, D. Barnes, B. Yeomans, and P. Newman, "Real-time kinematic ground truth for the Oxford RobotCar dataset," *arXiv preprint arXiv: 2002.10152*, 2020.
- [75] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, and A. Kim, "Complex Urban Dataset with multi-level sensors from highly diverse urban environments," *International Journal of Robotics Research*, 2019.
- [76] P. Wenzel, R. Wang, N. Yang, Q. Cheng, Q. Khan, L. von Stumberg, N. Zeller, and D. Cremers, "4Seasons: A cross-season dataset for multi-weather SLAM in autonomous driving," *Pattern Recognition: 42nd DAGM German Conference, DAGM GCPR 2020*, 2020.
- [77] C. Zhu, M. Meurer, and C. Günther, "Integrity of visual navigation—developments, challenges, and prospects," *NAVIGATION: Journal of the Institute of Navigation*, 2022.
- [78] H. Hu and G. Kantor, "Parametric covariance prediction for heteroscedastic noise," *International Conference on Intelligent Robots and Systems (IROS)*, 2015.
- [79] K. Liu, K. Ok, W. Vega-Brown, and N. Roy, "Deep inference for covariance estimation: Learning gaussian noise models for state estimation," *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.
- [80] F. Wu, H. Luo, H. Jia, F. Zhao, Y. Xiao, and X. Gao, "Predicting the noise covariance with a multitask learning model for kalman filter-based gnss/ins integrated navigation," *IEEE Transactions on Instrumentation and Measurement*, 2021.
- [81] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Computing Surveys*, 2021.

- [82] T. Wen and R. Keyes, "Time series anomaly detection using convolutional neural networks and transfer learning," *ArXiv*, 2019.
- [83] B. Lindemann, B. Maschler, N. Sahlab, and M. Weyrich, "A survey on anomaly detection for technical systems using LSTM networks," *Computers in Industry*, 2021.
- [84] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder," *IEEE Robotics and Automation Letters*, 2017.
- [85] Y. Zhang, Y. Chen, J. Wang, and Z. Pan, "Unsupervised deep anomaly detection for multi-sensor time-series signals." *IEEE Transactions on Knowledge and Data Engineering*, 2021.
- [86] D. Kim and J. Cho, "Improvement of anomalous behavior detection of GNSS signal based on TDNN for augmentation systems," *Sensors*, 2018.
- [87] G. Kahn, P. Abbeel, and S. Levine, "LaND: Learning to navigate from disengagements," *IEEE Robotics and Automation Letters*, 2021.
- [88] F. v. Wyk, Y. Wang, A. Khojandi, and N. Masoud, "Real-time sensor anomaly detection and identification in automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2020.
- [89] C. H. Tong and T. D. Barfoot, "Batch heterogeneous outlier rejection for feature-poor SLAM," *IEEE International Conference on Robotics and Automation*, 2011.
- [90] G. Pang, A. van den Hengel, C. Shen, and L. Cao, "Toward deep supervised anomaly detection: Reinforcement learning from partially labeled anomaly data." *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery Data Mining*, 2021.
- [91] J.-P. G. v. Brakel, "Robust peak detection algorithm using z-scores," <https://stackoverflow.com/questions/22583391/peak-signal-detection-in-realtime-timeseries-data/22640362#22640362>, version: 2020-11-08.
- [92] B. Murphy and R. Barr, "The z-score," <https://mat117.wisconsin.edu/4-the-z-score/>, accessed: 2022-06-17.
- [93] A. Alvarado, "Z-table," https://castle.eiu.edu/~aalvarado2/z_table.pdf, accessed: 2022-06-17.
- [94] G. Bradski, "The OpenCV library," *Journal of Software Tools*, 2000.

-
- [95] A. Ravi and F. Karray, "Exploring convolutional recurrent architectures for anomaly detection in videos: a comparative study," *Discover Artificial Intelligence*, 2021.
- [96] A. Karpathy, "A recipe for training neural networks," <https://karpathy.github.io/2019/04/25/recipe/>, version: 2019-04-25.
- [97] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras," *IEEE Transactions on Robotics*, 2017.
- [98] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1991.
- [99] J. Bednář, M. Petrlík, K. C. T. Vivaldini, and M. Saska, "Deployment of reliable visual inertial odometry approaches for unmanned aerial vehicles in real-world environment," *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2022.
- [100] K. Choi, J. Yi, C. Park, and S. Yoon, "Deep learning for anomaly detection in time-series data: Review, analysis, and guidelines." *IEEE Access*, 2021.
- [101] Y. Sun, "RAIM-NET: A deep neural network for receiver autonomous integrity monitoring." *Remote Sens.*, 2020.
- [102] S. Gupta and G. Gao, "Data-driven protection levels for camera and 3D map-based safe urban localization." *NAVIGATION: Journal of the Institute of Navigation*, 2021.
- [103] B. Sofman, B. Neuman, A. Stentz, and J. A. Bagnell, "Anytime online novelty and change detection for mobile robots." *J. Field Robotics*, 2011.
- [104] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.
- [105] S. Zhang, Z. Wang, and Z. Gao, "Research on exact thresholds for ARAIM MHSS fault monitoring." *Mathematical Problems in Engineering*, 2019.
- [106] M. Joerger, S. Stevanovic, F.-C. Chan, S. Langel, and B. Pervan, "Integrity risk and continuity risk for fault detection and exclusion using solution separation ARAIM." *26th International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS*, 2013.
- [107] P. Wenzel, R. Wang, N. Yang, Q. Cheng, Q. Khan, L. von Stumberg, N. Zeller, and D. Cremers, "4Seasons: A cross-season dataset for multi-weather SLAM in autonomous driving," *Pattern Recognition: 42nd DAGM German Conference, DAGM GCPR 2020*, 2020.

- [108] L. v. Stumberg and D. Cremers, "Dm-vio: Delayed marginalization visual-inertial odometry," *IEEE Robotics and Automation Letters*, 2022.
- [109] F. Dellaert and G. Contributors, "borglab/gtsam," May 2022. [Online]. Available: <https://github.com/borglab/gtsam>
- [110] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation," *Robotics: Science and Systems (RSS) conference*, 2015.
- [111] C. Li and S. L. Waslander, "Visual measurement integrity monitoring for uav localization," *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2019.

ISBN 978-82-326-8110-5 (printed ver.)
ISBN 978-82-326-8109-9 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)



NTNU

Norwegian University of
Science and Technology