Patrick Geragersian

Robust Autonomous Navigation for UAVs in Urban Environments using Machine Learning

School of Aerospace, Transport and Manufacturing

PhD
Academic Year: 2021 - 2024

Supervisor: Dr. Ivan Petrunin
Associate Supervisor: Prof. Weisi Guo
February 2024

School of Aerospace, Transport and Manufacturing

PhD

Academic Year 2021 - 2024

Patrick Geragersian

Robust Autonomous Navigation for UAVs in Urban Environments using Machine Learning

Supervisor: Dr. Ivan Petrunin
Associate Supervisor: Prof. Weisi Guo
February 2024

This thesis is submitted for fulfilment of the requirements for the degree of PhD.

© Cranfield University 2024. All rights reserved. No part of this publication may be reproduced without the written permission of the copyright owner.

Academic integrity declaration

I declare that:

- the thesis submitted has been written by me alone.
- the thesis submitted has not been previously submitted to this university or any other.
- that all content, including primary and/or secondary data, is true to the best of my knowledge.
- that all quotations and references have been duly acknowledged according to the requirements of academic research.

I understand that to knowingly submit work in violation of the above statement will be considered by examiners as academic misconduct.

Abstract

Urban canyons, characterized by their high-rise buildings and dense infrastructure, pose significant challenges to Unmanned Ariel Vehicle (UAV) navigation primarily due to the obstruction and interference of Global Navigation Satellite Systems (GNSS) signals. These challenges include multipath effects, where signals bounce off multiple surfaces before reaching the receiver, leading to inaccurate positioning data, and direct signal blockage, which causes intermittent loss of satellite visibility. The reliance on GNSS for UAV navigation in such environments is further compromised by potential denial of service, whether intentional or unintentional, through interference. These limitations highlight the need to explore alternative or supplementary navigation technologies to ensure the safe and efficient operation of UAVs in urban settings.

To address these challenges, this research begins by focusing on reducing sensor errors in both GNSS and Inertial Navigation Systems (INS). Specifically, data-driven approaches have been proposed to mitigate errors from individual sensors. For INS, a Gated Recurrent Units (GRU)-based error prediction algorithm, trained on labelled sensor data, is utilised to reduce the effects of random walk caused by the sensor's stochastic noise, which accumulates over time. For GNSS, a GRU classification algorithm is employed to detect and exclude Non-Line-Of-Sight (NLOS) signals from the list of available GNSS satellites. Once NLOS signals are removed, the remaining signals are ranked based on their contribution to Geometric Dilution of Precision (GDOP). The top 10 ranked signals are then used to compute the receiver's position, resulting in improved positioning accuracy and greater reliability over time.

Building on these initial advancements, this thesis proposes a robust federated multi-sensor fusion architecture to further enhance positioning accuracy. The proposed architecture integrates GNSS and INS positioning, with additional enhancements provided by GRU-based corrections. Furthermore, Monocular Visual Odometry (VO) and a barometer are incorporated to refine positioning in GNSS-degraded environments. A novel Bayesian Long Short-Term Memory (LSTM) model with Monte Carlo dropouts is introduced to generate stochastic outputs, enabling the system to estimate navigation integrity by predicting position distributions and calculating Protection Levels (PL). By

quantifying the reliability of navigation data, this approach ensures the safety and accuracy required for autonomous UAV operations in complex urban environments.

The proposed architecture was validated using a comprehensive simulation setup. This includes GNSS Hardware-in-the-Loop (HIL) simulations with the Spirent GSS7000 and OKTAL-SE Sim3D for realistic modelling of multipath effects, as well as high-fidelity VO sensor data generated through AirSim and Unreal Engine. These simulations replicate urban canyon scenarios, accounting for variations in GNSS signal quality and adverse weather conditions, providing a robust environment to test navigation performance.

The results indicate significant advancements in UAV navigation, including a 22.1% reduction in the 95th percentile horizontal error compared to state-of-the-art federated fusion approaches and a 98% reduction in misleading integrity information relative to traditional Extended Kalman Filters (EKF). These improvements demonstrate the architecture's ability to deliver accurate, reliable navigation with a quantifiable measure of integrity, even in the most challenging urban environments.

This research has profound implications for the future of Urban Air Mobility (UAM). UAM applications, such as passenger air taxis, last-mile deliveries, and emergency response, rely on precise and reliable navigation systems to ensure safe operation in urban settings. The ability to provide integrity information—quantifying the reliability of positioning data—is crucial for regulatory compliance, operational safety, and public trust in autonomous aerial systems. By addressing the challenges of GNSS-degraded environments and offering a robust framework for navigation with measurable integrity, this study lays a strong foundation for UAM. It enables scalable, reliable, and safe integration of UAVs into urban airspaces, bringing the vision of autonomous urban transportation closer to reality.

Acknowledgements

First and foremost, my heartfelt thanks go to Dr. Raphael Grech, Prof. Weisi Guo, Cranfield University, and Spirent Communications for their unwavering support throughout my PhD journey. Their profound academic insight and industry expertise were instrumental in guiding me through this challenging and rewarding process.

I extend my gratitude to the members of the DARTeC community for their assistance, both professionally and personally. The friendships I've forged during this time are invaluable and will be treasured forever. Special thanks to Oguz Kagan Isik for his invaluable help with facility access and guidance during challenging moments, and to Dr. Ruifan Liu for being a source of moral support and direction in both my research and personal life. Huw Whitworth's constant encouragement has been a cornerstone of my PhD experience, for which I am deeply thankful. I also extend my heartfelt appreciation to Dr. Shuoyuan Xu, who has been a pillar of support during the most challenging times of my PhD journey. His guidance and collaborative spirit have not only been invaluable but have also made the research process enjoyable. Not only have you been a great colleague, but you have been an even greater friend.

A profound appreciation is extended to Thomas Mason, Sangkeun Kang, Henry Li and Jason Wong, whose unwavering support and brotherhood have been a source of strength and joy. Our shared memories are a treasure, and I look forward to creating many more. Thomas, your support over nearly 15 years has been a constant in my life's ups and downs. You have always been there for me, and I cannot imagine it otherwise. Sangkeun Kang, in the six years of our friendship, your presence has become indispensable, enriching my life through both victories and challenges in League Of Legends and beyond. I look forward to making many more memories with you in the future. Henry, your friendship, though maddening at times, has always brought joy and a sense of belonging. I will never forget everything you have done for me as one of my best friends. Jason, although our acquaintance hasn't been as lengthy as with others, you've challenged me to expand my limits and enriched me with sincere knowledge. I look forward to our friendship enduring for many years ahead. My gratitude to you all is boundless.

To Sorin Andrei Negru, your role in my PhD journey has been paramount. Your excellence as a master's student and our shared academic endeavours have been incredibly rewarding. Not only do you show dedication to your subject and Cranfield University, but your dedication to the people around you are admirable. I look forward to continuing our collaborative efforts and friendship.

Dr. Ivan Petrunin, your guidance and support have been unparalleled. As a supervisor, your dedication to your students and research sets a benchmark for all. You are, not only an inspirational individual, but also someone deeply admired by many colleagues at Cranfield. My success at Cranfield is in no small part due to your influence, for which I am eternally grateful.

To my parents, Kenarik and Albert, your unconditional love and support have been my guiding light. Your resilience and achievements continue to inspire me, and I am filled with admiration and love for both of you. Thank you for empowering me to chase my dreams.

Lastly, to my love, Yi-Feng Wang, your presence through every step of my PhD has been a source of strength and motivation. Your support has been the bedrock of my pursuit, and this success is as much yours as it is mine. Without you, I would have never been able to carry out my PhD. I am profoundly grateful and dedicate this thesis and future success to you.

Thank you all for being part of my journey.

Table of Contents

Academic integrity declaration i
Abstract ii
Acknowledgements iv
List of Figures ix
List of Tables xii
Chapter 1 Introduction 14
1.1 Background and Motivation 14
1.2 Research Aim and Objectives 22
1.3 Publications 22
1.4 Contributions 24
1.5 Thesis Outline 25
Chapter 2 Introduction to GNSS positioning 27
2.1 Introduction to Global Navigation Satellite Systems 28
2.2 Least Squares Approach in GNSS Positioning 29
2.3 GNSS errors 30
2.4 Error Mitigation Techniques in GNSS 32
. 33
2.5 Multipath Issues in GNSS 33
2.5.1 Line-of-Sight Signals 34
2.5.2 None Line-of-Sight Signals 34
2.5.3 Multipath Signals 34
2.6 Inertial Navigation System/Inertial Measurement Unit 35
2.7 Conclusion 36
Chapter 3 Evolution of Sensor Integration Techniques: Deep Learning Techniques for Sensor Fusion 38
3.1 Introduction to Artificial Intelligence, Machine Learning, and Deep Learning 39
3.1.1 Artificial Intelligence 39
3.1.2 Machine Learning 39
3.1.3 Deep Learning (DL) 40
3.2 Existing Methods for GNSS-INS Integration 41
3.3 Limitations of Kalman Filter-Based Methods 44
3.4 Emergence of Deep Learning (DL) Approaches 45
3.5 Deep Learning Techniques for GNSS-INS Integration 45
3.6 Conclusion 47
Chapter 4 Data Generation Framework 49
4.1 Real-World Data vs. HIL Simulation 50
4.1.1 Real-World Data Collection 50
4.2 HIL Simulation 50
4.2.1 Justification for HIL Simulation in This Research 51
4.3 Data Collection Overview 53

4.4 Simulation Framework Components 54
4.4.1 SimGEN GSS7000 HIL 54
4.4.2 U-blox F9P multi-constellation GNSS receiver 56
4.4.3 Sim3D 56
4.4.4 SimSENSOR 58
4.4.5 Unreal Engine with AirSim 60
4.5 Integration of Simulation Components 61
4.5.1 Data Flow and Synchronization 61
4.6 Simulation Parameters and Conditions 62
4.6.1 Sampling Rates 62
4.6.2 Vehicle Dynamics 63
4.6.3 Extreme Conditions 63
Chapter 5 Mitigating INS Errors Using Deep Learning in GNSS-INS Systems 65
5.1 Introduction 66
5.2 Related Work 67
5.3 Proposed System 68
5.3.1 IMU Measurement Error 68
5.3.2 System Overview 69
5.4 Experimental Setup 72
5.4.1 Data capturing 72
5.4.2 Architecture Optimization 74
5.5 Results & Analysis 76
5.5.1 GRU Performance Analysis 76
5.5.2 Training and Inference Speed 83
5.5.3 Perturbation Analysis 84
5.6 Conclusion 89
Chapter 6 Machine Learning-Enhanced GNSS in Urban Canyons: A Novel Multipath Detection and Signal Ranking Approach 92
6.1 Introduction 93
6.2 Literature Review 94
6.2.1 Signal Classification Techniques 94
6.2.2 Signal Ranking Techniques 96
6.3 Proposed Architecture 97
6.3.1 High Level Overview 97
6.3.2 GRU-based Satellite Signal Classification 99
6.3.3 Discussion on using GRU for classification and regression simultaneously 101
6.3.4 Signal Ranking Using $\boldsymbol{\gamma i}$ 101
6.3.5 Position Estimation Using Least-Squares 103
6.4 Simulation Setup & Methodology 105
6.5 Results & Discussion 107
6.5.1 Classification Performance 107

6.5.2 Position Performance Comparison 111
6.6 Conclusion & Future Work 115
Chapter 7 Robust Federated Multi-Sensor Scheme: Deep learning Approach for Integrity-based Positioning in Urban Canyons 117
7.1 Introduction 119
7.2 Related Work 120
7.2.1 Multi-Sensor Fusion 120
7.2.2 Integrity Monitoring 122
7.2.3 Uncertainty-based deep learning approaches 123
7.3 Proposed Architecture 124
7.3.1 Overview 124
7.3.2 Feature-based Monocular Visual Odometry with ORB 126
7.3.3 Loosely-coupled local filter EKF for GNSS/INS integration 127
7.3.4 Bayesian-LSTM implementation for multi-sensor fusion 129
7.3.5 Integrity Analysis 131
7.4 Simulation Overview and data collection 132
7.4.1 Scenario Overview 132
7.4.2 GNSS Condition (GNSS Receiver) 133
7.4.3 Visual Odometry Weather Condition and Camera Calibration 134
7.4.4 Data Collection 136
7.4.5 Machine Learning Training Phase 137
7.5 Results & Discussion 139
7.5.1 Training and inference speed 139
7.5.2 Sensor Fusion Performance 139
7.5.3 Protection Level Estimation Performance 149
7.5.4 Comparative Performance Analysis 153
7.6 Conclusion and Future Work 156
Chapter 8 General Discussion 158
8.1 Implementation and Identified Issues 160
8.2 Worldwide Context of the Research 161
8.3 Future Innovation Requirements 162
Chapter 9 Conclusion 165
Reference 167

List of Figures

Figure 1-1 Image of obstructed sky view in downtown Toronto. Source: Author..... 15

Figure 1-2 Illustration of a) horizontal protection level and b) Stanford Diagram 21

Figure 2-1 Visual illustration of signal propagation in urban canyons under different conditions (LOS, multipath propagation, NLOS) 33

Figure 3-1 Inside a Gated Recurrent Unit Neural Network Node 47

Figure 4-1 multi-system, multi-sensor data flow 53

Figure 4-2: Image of Spirent GSS7000 Simulator [64] 55

Figure 4-3 Image of ZED-F9P [65] 56

Figure 4-4: Example Unreal Engine Environment with AirSim 60

Figure 5-1 High Level Diagram of Proposed GRU System Architecture 71

Figure 5-2 Ground Truth UAV trajectory in simulated scenarios 74

Figure 5-3 Absolute Horizontal Position Error at Each Time Interval with No GNSS Update comparison with Sensor A INS and Sensor A GRU 78

Figure 5-4 Comparison of Horizontal Position Error Prediction Performance in terms of Straight (a) and Heading Change (b) Trajectories for Sensor A INS and Sensor A GRU at 95th percentile. 79

Figure 5-5 Horizontal Position Error Box Plot Comparison between Sensor A INS, Sensor A Decision Tree, Sensor A RNN and Sensor A GRU 81

Figure 5-6 Position Error Distribution (a) (b) for Training and Testing Data with Sensor A INS and Sensor A GRU 82

Figure 5-7 Trajectory Comparison between Ground Truth, Sensor A INS and Sensor A GRU for Training (a) and Testing Data (b) 83

Figure 5-8 Comparative Analysis of Sensor Perturbation Error Performance for Sensor B GRU (a) and Sensor C GRU (b) with the Respective Difference in INS error 87

Figure 5-9 Absolute Horizontal Position Error at Each Time Interval with No GNSS Update comparison with Sensor B INS/ Sensor B GRU (a) and Sensor C INS/ Sensor C GRU 89

Figure 6-1 High level proposed architecture overview 98

Figure 6-2 Machine learning architecture with optimized parameters 100

Figure 6-3 Sim3D environment of Toulouse, France 105

Figure 6-4 Proposed trajectory for data capture and architecture testing 106

Figure 6-5 Example observable output in RINEX format..... 106

Figure 6-6 Confusion matrix for test results of decision tree classification performance 107

Figure 6-7 Confusion matrix for test results of SVM performance 108

Figure 6-8 Confusion matrix for test results of random forest classification performance 109

Figure 6-9 Confusion matrix for test results of GRU (proposed) classification performance 110

Figure 6-10 Comparison of position plot performance between ground truth and receiver in the horizontal direction 114

Figure 6-11 Comparison of position plot performance between ground truth, proposed solution (Rank + GRU) and SVM in the horizontal direction 114

Figure 6-12 Comparison of position error distribution between receiver, proposed solution (Rank + GRU) and SVM in the horizontal direction 115

Figure 7-1 Illustration of centralized (left) and federated(right) fusion architectures.. 121

Figure 7-2 Proposed high-level overview of the robust multi-sensor federated scheme. 125

Figure 7-3 Illustration of Monte Carlo dropouts to produce observation distribution. 130

Figure 7-4 Illustration of integrity regarding a) comparison of protection level and alert limit and b) Stanford diagram. 132

Figure 7-5 Simulation ground truth trajectory..... 133

Figure 7-6 Ray-tracing GNSS signals in urban canyon using OKTAL-SE Sim3D..... 134

Figure 7-7 Scenario weather condition setup in Toulouse, France with a) afternoon b) evening c) dust d) fog [117]..... 135

Figure 7-8 Visual sensor calibration step 136

Figure 7-9 Visual position accuracy comparison of horizontal positioning performance between Bayesian-LSTM and ground truth 145

Figure 7-10 Visual position accuracy comparison of vertical positioning performance between Bayesian-LSTM and ground truth 145

Figure 7-11 Comparison of absolute horizontal position error distribution between EKF, VO and Bayesian-LSTM in presence of dust 146

Figure 7-12 Comparison of absolute horizontal position error vs time between EKF, VO and Bayesian-LSTM in presence of dust 148

Figure 7-13 Comparison of absolute vertical position error vs time between local EKF, VO and Bayesian-LSTM in presence of dust 149

Figure 7-14 Categorizing horizontal protection level estimate from Bayesian-LSTM using Stanford diagram 150

Figure 7-15 Categorizing vertical protection level estimate from Bayesian-LSTM using Stanford diagram 152

Figure 7-16 Comparison of horizontal protection level estimate with actual absolute position error vs time. 152

Figure 7-17 Comparison of horizontal protection level estimate performance between Bayesian-LSTM and simple EKF integration 154

Figure 7-18 Comparison of horizontal positioning accuracy between Bayesian-LSTM, EKF and ground truth 155

Figure 7-19 Comparison of mean position errors (North, East and Down) and 95th percentile error in the horizontal and vertical direction 156

List of Tables

Table 4-I Standard IMU errors used in SimSENSOR 59

Table 5-I Sensor A Error and Bias Settings for Gyroscope and Accelerometer 72

Table 5-II Sensor B Error and Bias Settings for Gyroscope and Accelerometer 72

Table 5-III Sensor C Error and Bias Settings for Gyroscope and Accelerometer 72

Table 5-IV Comparison of Mean ($\boldsymbol{\sigma}$), Standard Deviation ($\boldsymbol{\mu}$) and 95th Percentile Position Error for INS Sensor A, GRU Training and Testing in North (N), East (E) and Down (D) 76

Table 5-V Comparison of Position Error Improvement in the North, East, Down and Horizontal (H) Direction with Sensor A, GRU Training and Testing 77

Table 5-VI Comparison of Position Error in the North, East, Vertical and Horizontal Direction with INS Sensor B, INS Sensor C, GRU Testing Data Sensor B and GRU Testing Data Sensor C 84

Table 5-VII Comparison of Mean, Standard Deviation and 95th Percentile of Position Error for INS Sensor B, INS Sensor C, GRU Testing Data Sensor B and GRU Testing Data Sensor C 85

Table 5-VIII Comparison of KL divergence between INS and GRU for Sensor A, Sensor B and Sensor C 88

Table 6-I Network performance comparison of decision tree (DT), SVM, random forest (RF) and GRU (proposed) in regard to accuracy, class precision, recall and F1 Score 111

Table 6-II Comparison of mean error for various positioning technique in the north, east, horizontal and vertical directions 112

Table 6-III comparison of error standard deviation for various positioning technique in the north, east, horizontal and vertical directions 112

Table 6-IV Comparison of 95th percentile error for various positioning technique in the north, east, horizontal and vertical directions 113

Table 6-V Comparison of percentage position error improvement for various positioning technique and VDOP/HDOP in the north, east, horizontal and vertical directions 113

Table 7-I. Comparison of absolute mean position error for various sensor/machine learning configurations and weather conditions 140

Table 7-II. Comparison of absolute standard deviation position error for various sensor/machine learning configurations and weather conditions 141

Table 7-III. Comparison of position accuracy for various sensor/machine learning configurations and weather conditions 142

Table 7-IV. Relative comparison to receiver for performance improvement of various sensor/machine learning configurations and weather conditions 143

Chapter 1 Introduction

1.1 Background and Motivation

The concept of Urban Air Mobility (UAM) has gained significant traction in the context of expanding urbanisation and the ensuing challenges in transportation efficiency and sustainability [1]. UAM encompasses the broader framework of integrating air-based transportation solutions into urban environments, which includes various types of aerial vehicles such as drones, air taxis, and autonomous flying vehicles. Unmanned Aerial Vehicles (UAVs) are a subset within this framework, focusing specifically on unmanned aircraft used for tasks ranging from delivery services to surveillance and emergency response. While UAM represents the overall system and vision for urban air transport, UAVs serve as critical components within this system, particularly in addressing specific transportation challenges [2].

With the relentless pace of urbanisation, the concept of UAM has emerged as an innovative solution and a critical element in tackling the growing challenges of urban congestion, air quality, and transportation efficiency. The effects of traffic congestion in major metropolitan areas are estimated to cost $16.5 billion and rise to $30 billion. New road construction to alleviate issues around traffic congestion in areas such as New Jersey, USA, is set to cost $220,490 per mile [3]. Therefore, integrating UAVs into urban landscapes offers a promising avenue to alleviate ground traffic, reduce pollution, and enhance emergency response capabilities [4]. However, the realisation of this vision hinges on overcoming significant technological and regulatory hurdles, particularly in developing reliable and precise autonomous navigation systems. Drone operations' safety and ability to navigate complex urban environments with high accuracy and robustness are paramount, underscoring the urgent need for advancements in this field.

Figure 1-1 Image of obstructed sky view in downtown Toronto. Source: Author

The future impact of UAVs on the UK economy and society is also profound, with government and industry projections suggesting that UAVs could contribute up to £45 billion by 2030, representing a 2% boost to GDP [5]. Moreover, drone technologies are expected to create over 628,000 jobs, spanning logistics, agriculture, healthcare, and public services. The environmental benefits are also notable, with potential reductions of up to 2.4 million tonnes of CO2 by 2030. Integrating UAVs into critical sectors, such as healthcare (e.g., medical deliveries to remote areas) and law enforcement (e.g., enhancing police response times), highlights their growing role in essential services. However, this rapid expansion underscores the need for robust and reliable navigation systems. Inaccurate positioning could lead to mission failures, safety risks, and economic losses. Ensuring precise and resilient navigation, especially in complex environments like urban canyons, is therefore crucial to fully realising the potential of UAV technology.

Urban canyons, characterised by narrow streets flanked by tall buildings, present formidable challenges for reliable UAV navigation. This is caused by the canyons' effects on Global Navigation Satellite Systems (GNSS) signals [6]. GNSS refers to a constellation of satellites that provide real-time positioning and is a core element of Positioning, Navigation, and Timing (PNT) systems. PNT encompasses three critical components: positioning, which determines a precise location; navigation, which guides movement from one point to another; and timing, which ensures synchronisation across systems. GNSS constellations, such as Global Positioning System (GPS), Globalnaya Navigatsionnaya Sputnikovaya Sistema (GLONASS), and Galileo, transmit signals that

enable receivers to determine their position, velocity, and time, thereby supporting the PNT framework. However, in urban environments, obstructions can degrade GNSS signals, compromising PNT accuracy and affecting the ability of UAVs to navigate safely. In urban environments, surrounding structures block, reflect, or diffract signals, leading to degraded accuracy or loss of signal. Signal disturbances in urban environments are called multipath effects, a term used interchangeably to describe reflected and entirely blocked signals. In this thesis, however, we will use 'multipath' exclusively for signals that reach the receiver after being reflected. In contrast, entirely blocked signals will be termed None-line-Of-Sight (NLOS) and direct signals as Line-Of-Sight (LOS) [7].

These signal disruptions present a significant challenge for UAVs, which depend heavily on accurate positioning and navigation data for safe and efficient operation. When GNSS signals are obstructed or reflected, the receiver can no longer accurately calculate the time delay between the satellite and the UAV, which is essential for determining the pseudorange and thus position.

The global reliance on GNSS has led to increasing concerns about its vulnerabilities, with several national and international initiatives aiming to improve PNT resilience. In the UK, efforts by the UK Government and the National Timing Centre to improve economic resilience highlight the strategic importance of PNT infrastructure for both civilian and defence applications. By investing £36 million, the UK economy is aiming to prepare against the potential impact of satellite systems failures [8]. Globally, the recognition of GNSS vulnerabilities has spurred research into alternative PNT technologies. The European Commission recently published a paper assessing alternative PNT technologies, with the aim of protecting critical infrastructure from potential GNSS distribution [9].

Inertial Measurement Units (IMUs) are sensors to that can help mitigate the position accuracy issues affected by signal disturbances. IMUs typically consist of accelerometers and gyroscopes, which measure the specific force and angular rotation of the UAV. These measurements allow the system to estimate the UAV's position and movement by integrating, providing a method to predict the next position even when satellite signals are unreliable [10]. The end results are typically referred to as Inertial Navigation System (INS). While an IMU provides raw motion data, an INS integrates this data over time to

estimate the vehicle's position, velocity, and orientation. However, IMU sensors are affected by sensor bias and stochastic noises, which cause drifting of positioning predictions over time [11]. Therefore, rather than using each sensor individually or individually, methods have been proposed to integrate GNSS and INS together.

One such method is Kalman Filters (KF), which fuse data from INS and GNSS to generate more accurate position estimates [12][13]. KFs play a crucial role by combining the INS's relative positioning data with GNSS's absolute positioning, correcting for the drift and errors in the INS over time. This fusion allows for continuous and more reliable positioning, even in environments with compromised GNSS signals. There are different levels of KF integration, including loosely, tightly and ultra-tightly coupled versions. Paper [14] compares the real-world performance difference between loosely coupled and tightly coupled architectures. Whilst tightly coupled did improve RMSE positioning performance, the performance improvement was at 4.88% in the horizontal direction. Therefore, this thesis will focus on the loosely coupled variant, the most dominant version.

While KFs provide a method for fusing GNSS and INS data, they have limitations. In the case of a prolonged GNSS outage, the accuracy of the position estimate will degrade over time due to drift in the INS, as there is no external reference to correct the errors [15]. Furthermore, Kalman filters rely on the assumption that noise in the system is Gaussian, which may only sometimes hold in complex environments. These limitations highlight the need for alternative approaches, such as data-driven methods, which can leverage large amounts of data to improve positioning accuracy.

The emergence of Machine Learning (ML) offers this data-driven approach to the problem. ML's ability to process and analyse vast amounts of sensor data, learn from environmental interactions, and make real-time decisions makes it an ideal candidate for enhancing UAV navigation in complex urban landscapes [16]. Leveraging data from an array of onboard sensors such as GNSS, IMU, Barometer, Camera and ML models can aid UAVs navigate difficult environments. However, integrating ML into UAV navigation systems for urban canyon environments is challenging due to the high-density infrastructure, frequent signal occlusions, and dynamic conditions typical of such areas. Environmental factors such as fluctuating weather conditions, varying urban densities,

and the dynamic nature of urban spaces also adds complexity to UAV navigation. While urban canyons pose unique challenges, other complex environments, such as dense forests, indoor industrial facilities, and subterranean settings, also require robust navigation solutions. By focusing on urban canyons, this research addresses a representative case of environments where GNSS signals are unreliable, and machine learning-based approaches can improve robustness.. This includes utilising algorithms that can effectively process and interpret the complex and dynamic data typical of urban canyons, ensuring real-time responsiveness and decision-making of these systems.

Some may argue that the rarity of highly dense urban environments and the adaptability of future UAV designs might diminish the significance of urban canyons as a navigation problem. Moreover, advancing intelligent transportation systems (ITS) could provide additional support, such as enhanced infrastructure that aids UAV navigation or improved ground-based communication systems that mitigate the impact of GNSS limitations [17].

Despite these potential developments and future infrastructure, urban canyons will likely remain a significant challenge, particularly in expanding megacities. The increasing reliance on UAVs for applications such as emergency response, package delivery, and urban monitoring highlights the urgent need for reliable navigation solutions capable of operating in complex and dynamic environments.

By addressing these challenges through the integration of machine learning techniques, this thesis is expected to provide real-world impact by enhancing UAV performance in critical operations, where consistent navigation is essential. Scientifically, the work will contribute by utilising data-driven approaches to improve positioning accuracy, robustness, and integrity in environments prone to frequent signal disruptions. From an engineering perspective, it will establish a methodology for capturing large datasets in diverse conditions, which is vital for training and validating machine learning models. These datasets will help ensure that the developed solutions can generalize effectively across a wide range of real-world scenarios, supporting broader UAV applications.

One key aspect of our research is the role of ML models in enhancing robustness. Robustness, particularly in navigation systems, is the capacity of these systems to maintain accurate and reliable functionality, offering consistent guidance despite

environmental changes, disturbances, or measurement fluctuations. The potential of ML in this area is of interest [18].

Another crucial aspect of our research is the successful training of machine learning algorithms, which work on various scenarios. This success heavily depends on large datasets, which are important in improving generalisation performance. Techniques such as probabilistic modelling and uncertainty quantification are vital for understanding the uncertainties inherent in data, which impact the system's reliability and the accuracy of its predictions.

Moreover, replicating various real-world scenarios for testing is costly, and managing the independent variables influencing sensor disturbances presents challenges. Therefore, conducting integrity analysis is a crucial aspect of building robustness within a multi-sensor fusion framework, alongside establishing a dependable, realistic framework for multi-sensor data generation. Integrating a robust approach with effective uncertainty management improves the reliability of ML systems and supports better adaptations to new or changing environments. This emphasis on integrity aligns closely with broader aviation safety standards, such as those defined by the International Civil Aviation Organization (ICAO).

The ICAO describes critical factors related to the safety of air navigation systems. [19]. These definitions are designed to provide a platform of safety standards for aircraft operations and include key performance definitions such as Integrity, Availability, Continuity, and Accuracy. Each of these parameters plays a vital role in the safe navigation of aircraft, including UAVs, within the airspace system. Integrity in sensor fusion refers to the trust that can be placed in the correctness of the information the navigation system supplies. It is crucial as it implies the system's ability to provide timely warnings to users when it should not be used for navigation. Some of the measures of Integrity include:

1. **Protection Level (PL)**: This calculated value provides a bound on the navigation system's error with a certain confidence level. The Protection Level ensures that the Actual Error does not exceed this bound without an alert.

2. **Alert Limit (AL)**: The Alert Limit is the threshold that defines the maximum allowable error for the navigation system. If the Actual Error exceeds the Alert Limit, the system is considered to have lost integrity, and immediate action or alternative measures should be taken.

3. **Actual Error (AE)**: The Actual Error is the difference between the estimated position provided by the navigation system and the actual position of the aircraft. The system ideally aims to minimise AE to ensure accurate and reliable navigation.

To better illustrate the relationship between the Actual Error and Protection Level and where the Alert Limit falls, the Stanford Diagram is used. It is divided into four quadrants, each representing a different state of system integrity:

- **Nominal Operations (Green Zone)**: This is the ideal state where AE is less than PL and AL. The navigation system is operating correctly, and integrity is maintained.

- **Misleading Information (Orange Zone)**: This occurs when AE exceeds PL but remains below AL. In this case, the system provides information that could potentially mislead the user, but it has not yet reached a critical level that would trigger an alert.

- **Hazardous Misleading Information (Red Zone)**: This is the most critical state, where AE has exceeded both PL and AL. The system provides dangerously misleading information, and integrity is compromised without an alert.

- **System Unavailable (Yellow Zone)**: This can occur in two scenarios: either when AE is less than AL but greater than PL, meaning the system is unreliable, or when AE is more significant than AL and PL, and the system is also providing misleading information. In both cases, the system is considered unavailable for safe navigation.

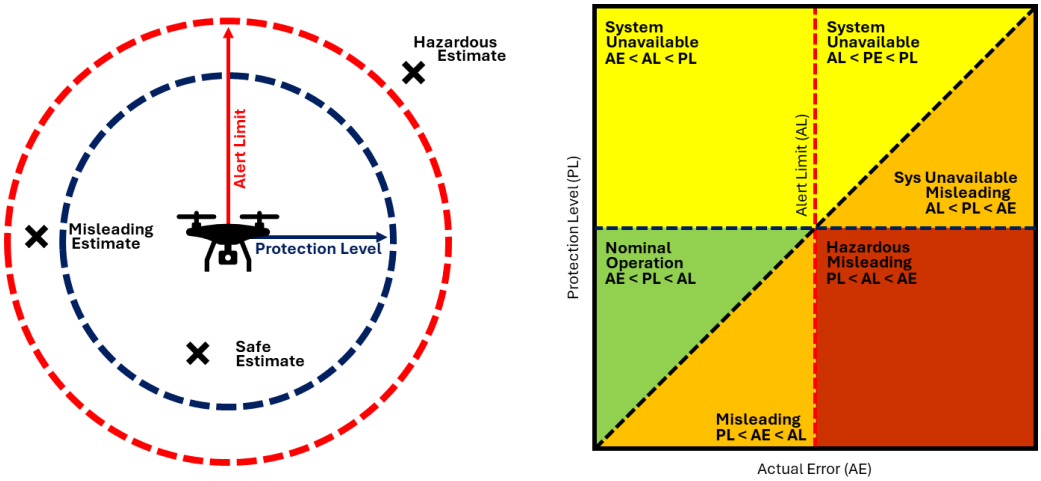


Figure 1-2 Illustration of a) horizontal protection level and b) Stanford Diagram

The Stanford Diagram is particularly useful for assessing and ensuring the safety and reliability of the GNSS-based navigation systems used by UAVs. It can also be utilised for sensor fusion systems to provide a clear visual means to monitor the integrity of the navigation system, ensuring that the Actual Error remains within acceptable bounds. This tool is invaluable for developers and operators of UAVs for real-time monitoring and decision-making, particularly in urban canyon environments where navigation signals can be unpredictable and integrity is of paramount concern.

Integrating UAVs into urban airspaces introduces a complex regulatory landscape that must be cared for. Current regulations may need to fully address the nuances of autonomous navigation and ML-based systems, creating gaps that pose challenges for widespread adoption. Collaborative efforts among stakeholders—including technologists, urban planners, and regulatory bodies—are essential to develop comprehensive regulations that ensure the safety, privacy, and efficiency of UAM. Emphasising regulatory compliance is a legal necessity and a foundational element in building public trust and ensuring the successful deployment of UAVs in urban environments.

Therefore, this research is motivated by the need for robust and safe autonomous navigation systems for UAVs in urban canyons. With the rapid growth of metropolitan areas and the increasing reliance on UAVs for various applications, from delivery services to emergency response, navigating safely and effectively in urban canyons is becoming increasingly critical.

1.2 Research Aim and Objectives

This thesis aims to explore the use of machine learning in integrity-based sensor fusion and to propose, develop and evaluate PNT navigation solutions for UAVs in urban environments, achieving cm-level accuracy, reducing GNSS dependency, and enhancing positioning performance. Based on the described aim and the outlined challenges in the text above, the specific objectives for this thesis are the following:

- **O1:** To identify and discuss candidate PNT and fusion technologies for assured PNT in safety-critical missions. The performance characteristics of these technologies and their complementary aspects will be analysed to support the development of an effective fusion solution and identify mechanisms for further improvements/adaptation of individual technologies.
- **O2:** To propose a robust sensor fusion architecture and underlying technologies for UAVs in urban canyons using deep learning approaches based on mission specifications: environment, integrity, and accuracy requirements. This method will inform the development of performance-based navigation concepts for autonomous systems.
- **O3:** To develop a sensor simulation environment for data collection and post-processing of robust sensor fusion architecture. This will provide the ability to test the proposed architecture in a repeatable manner for robust analysis of candidate fusion systems with existing methodologies.
- **O4:** To evaluate the performance of the developed architecture using hardware in the loop simulations based on existing performance metrics for comparative analysis.

Addressing these aspects can advance autonomous navigation, paving the way for a more robust alternative.

1.3 Publications

This section contains a list of conference and journal papers that have either been submitted or published, all of which directly impact this thesis's direction and completed outcome. The publications are as follows.

Conference Papers:

- **C1**. P. Geragersian, I. Petrunin, W. Guo, and R. Grech, "An INS/GNSS fusion architecture in GNSS denied environment using gated recurrent unit," AIAA SCITECH 2022 Forum, Jan. 2022, doi: https://doi.org/10.2514/6.2022-1759. **(Published)**
- **C2**. P. Geragersian, I. Petrunin, W. Guo, and R. Grech, "Multipath Detection from GNSS Observables Using Gated Recurrent Unit," 2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC), Sep. 2022, doi: https://doi.org/10.1109/dasc55683.2022.9925850. **(Published)**
- **C3**. P. Geragersian, I. Petrunin, W. Guo, and R. Grech, "Uncertainty-based Sensor Fusion Architecture using Bayesian-LSTM Neural Network," AIAA SCITECH 2023 Forum, Jan. 2023, doi: https://doi.org/10.2514/6.2023-0193. **(Published)**
- **C4**. P. Geragersian, I. Petrunin, W. Guo, and R. Grech, "A Hybrid Deep Learning Approach for Robust Multi-Sensor GNSS/INS/VO Fusion in Urban Canyons," Proceedings of the Satellite Division's International Technical Meeting, Oct. 2023, doi: https://doi.org/10.33012/2023.19271. **(Published)**

Journal Paper:

- **J1.** P. Geragersian, I. Petrunin, W. Guo, and R. Grech, "Understanding and Mitigating INS Errors using Deep Learning in GNSS-INS Systems," in IEEE Transactions on Intelligent Transportation Systems **(Editorial Review Stage ID: T-ITS-23-08-2278)**
- **J2.** P. Geragersian, I. Petrunin, W. Guo, and R. Grech, "Machine Learning-Enhanced GNSS in Urban Canyons: A Novel Multipath Detection and Signal Ranking Approach." **(Editorial Review Stage ID: VT-2024-01727)**
- **J3.** P. Geragersian, I. Petrunin, W. Guo, and R. Grech, "Robust Multi-Sensor Fusion Scheme (RMSFS) for positioning in Urban Canyons using Machine Learning." **(Editorial Review Stage ID: NAVI-2024-035)**

The publications that are featured in this thesis include all the journal papers. Specifically, J1 provides an overview of the challenges facing GNSS and IMU sensors, especially in the context of Urban Canyons, where heavy reliance is placed on IMU sensors for a long duration. An approach is proposed to reduce the INS errors exhibited and investigate the environmental effects on the INS error prediction algorithm for a more robust approach.

J2 focuses on improving GNSS positioning accuracy when GNSS is available by classifying and ranking signals. J1 and J2 introduce a temporal-based machine learning approach to solve the mentioned issues. Finally, J1 and J2 are combined in a proposed robust fusion scheme detailed in J3. We introduce two additional sensors for improving positioning accuracy to the sensor fusion scheme and a novel architecture to fuse the information from the corrected sensor position estimates using a Bayesian-LSTM. The Bayesian-LSTM provides the fused position estimate, the data's uncertainty, and the covariance. This uncertainty is used to calculate the protection level to assess mission conditions. Additionally, the algorithm is tested under different weather conditions on the same trained ML network to understand the system's robustness.

1.4 Contributions

The thesis makes several substantial contributions to autonomous navigation for UAVs in urban environments, utilising machine learning techniques to enhance integrity and accuracy. The contributions address various aspects of UAV navigation, including error correction, multipath detection, and sensor fusion, summarised in logical order as follows:

- **Enhanced INS Error Correction**: Introduced the novel application of the INS error correction algorithm using a GRU deep-learning architecture. This architecture leverages temporal dependencies in INS errors caused by accelerometer and gyroscope noises and drifts. The implementation improved error reduction, with performance gains of up to 86%.
- **Impact of sensor error fluctuations on trained algorithm:** Simulations of sensor error fluctuations caused by environmental factors, such as temperature, revealed that while the GRU did not eliminate these errors, its effects were minimal on the prediction performance, preventing a decline in overall system performance.
- **GNSS signal classification using machine learning**: Developed a GRU-based classification algorithm for GNSS signals, categorising them into Line-of-Sight, Non-Line-Of-Sight, and multipath conditions.
- **Signal ranking algorithm**: NLOS signal removal method and ranking of the remaining signals after classification to optimise Geometric Dilution of Precision

(GDOP). This approach, which involves using the top ten satellites for position calculation when more than ten remain, achieved a 50% increase in accuracy over traditional classifiers like SVMs, demonstrating a reduction in error metrics.

- **Robust Multi-Sensor Fusion Scheme**: Proposed a federated multi-sensor fusion approach using a stochastic Bayesian-LSTM architecture that includes GNSS, IMU, VO, and Barometer sensors. This method enhances fault tolerance compared to traditional centralised methods and provides stochastic predictions vital for quantifying positional uncertainty. This capability is essential for integrity-based applications critical for safety in navigation systems. The architecture's robustness was validated under various conditions, including changing weather and GNSS multipath scenarios.
- **Integrity evaluation using PL calculation**: Utilised the stochastic output from the Bayesian-LSTM algorithm to calculate the protection level and map these to the Stanford diagram to compare against state-of-art architectures
- **Hardware-In-the-Loop data collection environment:** Created a HIL simulation system, incorporating UAV simulation in urban canyon to collect a range of data from different visual/non-visual sensors (either simulated or HIL-based) for large and diverse dataset to train ML networks for better generalisation.

These advancements collectively push forward the capabilities of UAV navigation systems in urban environments, providing a strong foundation for future research and application in enhancing autonomous vehicular integrity and accuracy.

1.5 Thesis Outline

This thesis comprises previous journal papers, where chapters five through seven each represent an individual paper that has been published or is under consideration for publication. Consequently, each chapter functions as an independent unit, with its introduction that sets the research stage and literature review and leads into detailed methodologies, results, discussions, and conclusions. This approach offers a clear and logical arrangement of the research content, enabling a focused exploration of the different aspects of the study within each chapter. Chapter two is an introduction to GNSS and the related problems of GNSS in urban canyons. Chapters three and four discuss the

evolving requirement for algorithms that can model complex relations and the use of hardware-in-the-loop simulation for generating datasets. Chapters five and six identify mechanisms to improve INS and GNSS positioning by adding an error correction method to counter the effects of drifting and removing NLOS signals, respectively. The literature review identified this as a point of further investigation and improvement. Chapter seven focuses on developing a robust sensor fusion architecture, which incorporates findings from chapters four and five as the first steps of the federated scheme. The second part of the scheme is local filtering to provide continuous positioning from GNSS/INS. Lastly, the local filter plus additional sensors are fused to provide positioning data and uncertainty in the data. This was identified in the literature as a step that needs to be developed, especially in ML techniques, which typically provide deterministic outputs and make it challenging to obtain uncertainties. The uncertainty estimate, hence, provides a way to measure the system's integrity needed for drones to operate autonomously in densely populated environments. Chapter eight discusses the proposed implementations and their success and further improvements. Lastly, chapter nine is the conclusion of the thesis.

Chapter 2 Introduction to GNSS positioning

This chapter introduces the fundamentals of Global Navigation Satellite Systems (GNSS), tracing their evolution from military origins to widespread civilian applications. It explains the principles of GNSS positioning through trilateration and the least squares method while addressing key error sources such as atmospheric delays, multipath interference, and satellite geometry. The chapter highlights error mitigation techniques, including dual-frequency receivers and augmentation systems. It introduces the integration of GNSS with Inertial Navigation Systems (INS) to improve accuracy in GNSS-denied environments.

2.1 Introduction to Global Navigation Satellite Systems

GNSS is a technology that is utilised extensively in everyday life [20]. From its initial concept as a positioning system for military operations in the 1980s [21] to today's vast use cases, such as timing information utilised in the New York Stock Exchange [22] and National Grid for coordinating the phase of alternating current across the network [23]. Therefore, its consistent and accurate operation is essential in daily life.

The origins of satellite navigation trace back to the late 1950s, when the Soviet Union launched Sputnik I, the first artificial satellite. Scientists tracking its signals noted that the Doppler shift in its transmitted frequency could be used to determine its position [24]. This realisation set the foundation for satellite-based navigation, which later evolved into comprehensive systems for global positioning. The first operational system, the Global Positioning System (GPS), was developed by the United States Department of Defense in the 1970s [25]. Initially intended for military applications, GPS became publicly accessible in the 1980s, enabling its adoption for civilian use worldwide. Subsequent development of other systems, including GLONASS (Russia), Galileo (European Union), and BeiDou (China), has expanded global coverage and improved system redundancy.

Global Navigation Satellite Systems operate by determining a user's position through a process known as trilateration. Satellites in orbit transmit radio frequency signals, including their position and precise transmission time. A GNSS receiver calculates the distance to each satellite by measuring the time delay between signal transmission and reception. These distances, known as pseudoranges, are combined to compute the user's position relative to the satellites. With signals from at least four satellites, the receiver can solve for its three-dimensional position (latitude, longitude, and altitude) and clock bias, an error introduced by the receiver's internal clock.

The fundamental equation governing GNSS positioning is derived from the propagation of electromagnetic waves. Assuming a signal travels at the speed of light, c, the pseudorange ρ to a satellite is expressed as [26]:

$$\rho = c \cdot (t_r - t_s) + c \cdot \Delta t \tag{2.1}$$

where t_s is the satellite transmission time, t_r is the reception time, and Δt accounts for the receiver clock bias. For simplicity, this equation does not initially include error

sources. The receiver calculates ρ for multiple satellites and solves a system of nonlinear equations to estimate its position (x, y, z) and clock bias $c \cdot \Delta t$. Expanding the system of equations:

$$\rho_i = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} + c \cdot \Delta t \tag{2.2}$$

Here, (x_i, y_i, z_i) represents the coordinates of each satellite, and ρ_i represents the pseudorange to each satellite. Solving this system requires numerical techniques such as least squares estimation.

2.2 Least Squares Approach in GNSS Positioning

For n satellites, the system of equations becomes overdetermined, as there are more equations than unknowns. This overdetermination allows for the minimization of measurement errors using the least squares approach, a standard method in GNSS positioning. The least squares approach estimates the receiver's position and clock bias by minimizing the sum of the squared residuals between the measured pseudoranges and the modelled pseudoranges. To achieve this, the pseudorange equations are linearized using a Taylor series expansion around an initial approximate position ($\hat{x}, \hat{y}, \hat{z}, \Delta\hat{t}$). The linearized form of the pseudorange equation for the i -th satellite is expressed as [27]:

$$\Delta\rho_i = \frac{\partial p_i}{\partial x}\Delta x + \frac{\partial p_i}{\partial y}\Delta y + \frac{\partial p_i}{\partial z}\Delta z + \frac{\partial p_i}{\partial \Delta t}\Delta\Delta t \tag{2.3}$$

Where $\Delta p_i = p_i - \hat{p}_i$ represents the difference between the measured pseudorange p_i and the modelled pseudorange $\hat{p}_i$ and $\Delta x, \Delta y, \Delta z, \Delta\Delta t$ are corrections to the approximate position and clock bias. The partial derivatives, which define the geometry of the satellite-receiver system, are calculated as follows:

$$\frac{\partial p_i}{\partial x} = \frac{x - x_i}{\hat{p}_i}, \frac{\partial p_i}{\partial y} = \frac{y - y_i}{\hat{p}_i}, \frac{\partial p_i}{\partial z} = \frac{z - z_i}{\hat{p}_i}, \frac{\partial p_i}{\partial \Delta t} = c \tag{2.4}$$

Here $\hat{\rho}_i = \sqrt{(\hat{x} - x_i)^2 + (\hat{y} - y_i)^2 + (\hat{z} - z_i)^2}$ is the modelled pseudorange based on the approximate position.

For satellites, this system of linearised equations can be expressed in matrix form:

$$\boldsymbol{\Delta\rho} = \boldsymbol{H\Delta p}$$

where $\boldsymbol{\Delta\rho}$ is the vector of pseudorange residuals, $\boldsymbol{\Delta p}$ is the vector of corrections to the approximate position, and $\boldsymbol{H}$ is the design matrix containing the partial derivatives for each satellite:

$$H = \begin{bmatrix} \frac{\partial \rho_1}{\partial x} & \frac{\partial \rho_1}{\partial y} & \frac{\partial \rho_1}{\partial z} & \frac{\partial \rho_1}{\partial \Delta t} \\ \frac{\partial \rho_2}{\partial x} & \frac{\partial \rho_2}{\partial y} & \frac{\partial \rho_2}{\partial z} & \frac{\partial \rho_2}{\partial \Delta t} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \rho_n}{\partial x} & \frac{\partial \rho_n}{\partial y} & \frac{\partial \rho_n}{\partial z} & \frac{\partial \rho_n}{\partial \Delta t} \end{bmatrix} \tag{2.5}$$

The least squares solution to this linearized system is obtained by solving the normal equation:

$$\boldsymbol{\Delta p = (H^T H)^{-1} H^T \Delta \rho} \tag{2.6}$$

The solution provides corrections to the initial position estimate, which are iteratively applied until the residuals converge to within a predefined threshold. The iterative process ensures that the final solution closely approximates the true position and clock bias.

The effectiveness of the least squares approach depends on several factors, including the quality of the pseudorange measurements and the spatial distribution of the satellites, quantified by the Geometric Dilution of Precision (GDOP). Favourable satellite geometry reduces the amplification of measurement errors, improving the reliability of the solution. However, limitations arise from unmodeled errors such as multipath interference and atmospheric delays. These sources of error introduce biases in the measured pseudoranges, which can propagate through the least squares solution and affect positional accuracy.

The least squares approach is fundamental to GNSS positioning by providing a framework for incorporating multiple measurements and minimizing their discrepancies. Its implementation highlights the interplay between measurement models, numerical methods, and error mitigation strategies in achieving accurate and reliable navigation solutions.

2.3 GNSS errors

GNSS signals, however, do not propagate through an ideal environment. Numerous error sources introduce inaccuracies into the measurements, necessitating correction models or

augmentation systems to achieve precise positioning. Errors in satellite clock and orbital parameters affect the transmitted signals. While satellite clock drift is corrected through synchronization with atomic clocks, ephemeris errors arise from deviations in predicted orbital positions, which are regularly updated by ground control [28].

GNSS signals, however, do not propagate through an ideal environment. Numerous error sources introduce inaccuracies into the measurements, necessitating correction models or augmentation systems to achieve precise positioning. Errors in satellite clock and orbital parameters affect the transmitted signals. While satellite clock drift is corrected through synchronization with atomic clocks, ephemeris errors arise from deviations in predicted orbital positions, which are regularly updated by ground control.

Other sources of error include ionospheric and tropospheric delays, which occur as the signal traverses the Earth's atmosphere. The ionospheric delay is caused by the interaction of GNSS signals with free electrons in the ionosphere, leading to frequency-dependent refraction. Tropospheric delay arises from the refraction of signals in the neutral atmosphere, primarily due to variations in pressure, temperature, and water vapor content. Multipath effects, caused by signals reflecting off surfaces such as buildings, water, or the ground before reaching the receiver, introduce distortions and positional errors. Receiver noise, resulting from imperfections in the electronic components of the GNSS receiver, further contributes to inaccuracies.

Additionally, errors can arise from satellite geometry, quantified by Geometric Dilution of Precision (GDOP). Poor satellite geometry—where satellites are clustered in one part of the sky rather than distributed across a wide field of view—can amplify the impact of measurement errors. User-Equivalent Range Errors (UERE), which aggregate various sources of signal distortion and noise, ultimately affect the computed position. Lastly, human and system-induced errors, including signal processing inaccuracies, software bugs, and incorrect configurations, can degrade GNSS performance.

Each of these error sources is addressed to varying degrees through advanced correction techniques and augmentation systems, ensuring GNSS can deliver reliable positioning and navigation services across diverse environments and applications.

2.4 Error Mitigation Techniques in GNSS

Numerous error mitigation techniques have been developed over time to enhance the accuracy and reliability of GNSS. These techniques address a variety of error sources, including ionospheric and tropospheric delays, satellite clock drift, ephemeris errors, and receiver noise. Despite significant advancements, certain challenges—such as multipath effects—remain difficult to eliminate fully [11].

One of the most widely used methods to mitigate ionospheric errors is the use of dual-frequency GNSS receivers [29]. The ionospheric delay can be estimated and corrected by comparing the propagation delays of signals transmitted on two different frequencies (commonly referred to as L1 and L2 in GPS). This approach exploits the dispersive nature of the ionosphere, where the delay is inversely proportional to the square of the signal frequency.

Tropospheric delays, in contrast, are largely non-dispersive and require empirical models or real-time atmospheric data for correction. Models such as the Hopfield and Saastamoinen models estimate tropospheric delay based on pressure, temperature, and humidity data [30]. Advanced receivers may also use local meteorological measurements to improve accuracy.

To address satellite-related errors, Satellite-Based Augmentation Systems (SBAS), such as the Wide Area Augmentation System (WAAS) in the United States and the European Geostationary Navigation Overlay Service (EGNOS) in Europe, provide correction signals [31]. These systems monitor GNSS satellites from a network of ground stations, compute corrections for satellite clock and orbital errors, and transmit them to users in real-time. For applications requiring even higher precision, Real-Time Kinematic (RTK) GNSS is commonly employed. RTK uses a fixed base station with a known position to provide centimeter-level corrections to nearby receivers by transmitting differential corrections [32].

Another notable development is the use of multi-constellation and multi-frequency GNSS. By leveraging signals from multiple constellations—such as GPS, GLONASS, Galileo, and BeiDou—users gain redundancy and improved accuracy, especially in challenging environments where signals from one system may be obstructed. Similarly,

multi-frequency capability allows for more robust mitigation of ionospheric effects and better handling of multipath errors [33].

Despite these advancements, environment-related errors—particularly multipath interference—remain one of the most persistent challenges in GNSS positioning. These errors occur when GNSS signals reflect off surfaces such as buildings, trees, water, or the ground before reaching the receiver, leading to distortions in the measured pseudorange. The receiver cannot distinguish between the direct and reflected signals, resulting in inaccuracies. Multipath errors are especially problematic in urban environments, where tall structures create complex signal reflections, and in areas near water bodies, where specular reflections exacerbate the issue

2.5 .Multipath Issues in GNSS

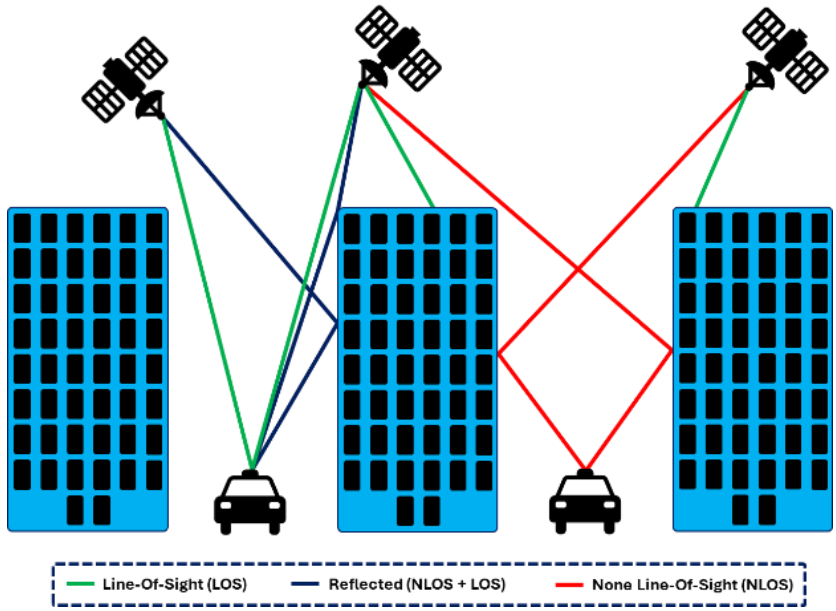


Figure 2-1 Visual illustration of signal propagation in urban canyons under different conditions (LOS, multipath propagation, NLOS)

Multipath interference is a significant challenge in GNSS systems, arising from the interaction of signals from the same satellite that have travelled along different paths before reaching the receiver. These signal paths include the direct line-of-sight (LOS) path, as well as reflected and diffracted paths, which contribute to interference. Understanding the propagation of these signals is crucial for mitigating their impact on GNSS accuracy [34]. In multipath scenarios, the key signal categories are:

2.5.1 Line-of-Sight Signals

The Line-of-Sight signal is the ideal case in GNSS, where the signal travels along the shortest, unobstructed path from the satellite to the receiver. It experiences minimal distortion aside from the delays caused by ionospheric and tropospheric effects, which can be corrected using models or augmentation systems. The direct signal is the most accurate source of information for calculating the pseudorange and, by extension, the user's position.

For accurate positioning, the receiver assumes that the measured pseudorange corresponds to the direct signal. Any interference from reflected signals alters the timing information and causes errors. The strength and integrity of the direct signal depend on the satellite's visibility and the receiver's environment. In open environments with a clear view of the sky, the direct signal is dominant and reliable. However, in urban areas or dense forests, the direct signal may be weakened or blocked entirely, increasing the likelihood of multipath interference.

2.5.2 None Line-of-Sight Signals

The None Line-of-Sight signal is a version of the satellite signal that has bounced off a surface—such as a building, water body, or the ground—before reaching the receiver. Because the None Line-of-Sight signal travels a longer path, it introduces a delay relative to the direct signal. This delay is the primary cause of multipath error, as the receiver cannot always distinguish between the direct and reflected signals.

None Line-of-Sight signals are particularly problematic in environments where surfaces are smooth and specular, such as glass buildings or calm water bodies, which create strong reflections. These reflections can appear similar to the direct signal in terms of amplitude and frequency, making them difficult to filter out using conventional methods. The phase of the reflected signal may also be shifted depending on the reflection angle and the surface material, further complicating its impact on the receiver.

2.5.3 Multipath Signals

Multipath signals refer to multiple signals from the same satellite that have taken different paths—typically including the direct and one or more reflected paths—before reaching

the receiver. Unlike the simple distinction between direct and reflected signals, multipath involves a scenario where the receiver is processing a composite of signals arriving at slightly different times and phases. These signals can combine constructively or destructively, depending on their relative amplitudes and phases, leading to complex distortions.

Multipath effects manifest in two primary ways:

- Code-Phase Multipath [35]: In code-based GNSS receivers, which measure pseudoranges based on the time delay of the signal code, multipath causes the receiver to detect a composite signal. This results in errors that typically range from a few meters to tens of meters.
- Carrier-Phase Multipath [36]: For high-precision GNSS receivers, which rely on the phase of the carrier wave, multipath introduces phase ambiguities. Although the carrier-phase errors are generally smaller (on the order of centimeters to decimeters), they are more critical in applications like geodetic surveys and autonomous systems.

The severity of multipath depends on factors such as the satellite-receiver geometry, the number of reflective surfaces in the environment, and the receiver's signal processing capabilities. Multipath is particularly challenging in environments where signals are reflected from multiple surfaces, creating numerous delayed replicas of the direct signal.

2.6 Inertial Navigation System/Inertial Measurement Unit

INS can be utilised with GNSS information to mitigate some of the accuracy issues related to GNSS-based navigation and improve receiver localisation. This is commonly referred to as INS/GNSS fusion [37]. However, INS also exhibits inherent errors that can affect the accuracy of the combined system. These errors include bias, drift, and noise in Inertial Measurement Units (IMUs) and errors in the INS integration algorithm.

The distinction between IMU and INS is essential. An IMU provides raw data on linear acceleration and angular velocity, whereas an INS utilises this data to compute position, velocity, and orientation relative to a known reference frame. The errors in inertial sensors are typically modelled as a random walk process, a statistical model in which the error grows over time as the square root of the elapsed time [38]. This type of error is referred

to as "white noise" due to its flat power spectral density, meaning that it is equally distributed across all frequencies. The random walk model is often used to represent the errors in gyroscopes and accelerometers, which can accumulate over time and result in a drift in the estimated orientation and position [11].

Under nominal circumstances, INS and GNSS errors are independent of each other. As such, the integration of these systems can improve navigational accuracy and reliability [39]. However, in Urban Canyons, where GNSS availability is limited, the integration will consist of either heavily degraded GNSS performance or no GNSS positioning. Given the errors mentioned above, this will cause a drift in position and a reduction in the availability or accuracy, impacting mission safety [7]. Therefore, providing error corrections for INS position and velocity estimates throughout the GNSS outage is beneficial to continue the mission.

Various error correction techniques have been proposed in the literature to enhance the accuracy and availability of navigation schemes dependent upon GNSS and INS. These techniques can be classified into filtering, non-linear optimisation, and machine learning [40]. Kalman filtering is a well-known approach for combining multiple sensors and modelling the uncertainty of the estimates [41]. Non-linear optimisation methods such as least squares and maximum likelihood estimate the optimal solution by minimising the residual error between the measurements and the model [42]. Machine learning techniques, on the other hand, can learn patterns and relationships from data and make predictions or decisions based on these patterns [43]. The integration of INS using some of these techniques will be discussed further in the next chapter.

2.7 Conclusion

This chapter has introduced GNSS, detailing the evolution of GNSS from its military origins to its widespread civilian use. Key principles of GNSS positioning, such as trilateration and the least squares method, were explained alongside error sources and mitigation techniques. The persistent challenge of multipath interference, particularly in urban environments, was highlighted, emphasizing the need for robust correction methods. INS was introduced as a complementary system to GNSS, capable of providing continuous navigation in GNSS-denied environments, albeit with its own inherent errors.

The integration of these two systems forms the foundation for reliable positioning solutions, setting the stage for advanced techniques discussed in subsequent chapters.

Chapter 3 Evolution of Sensor Integration Techniques: Deep Learning Techniques for Sensor Fusion

This chapter explores the evolution from traditional KF-based methods to advanced Deep Learning (DL) approaches for GNSS-INS integration. It introduces the foundational concepts of Artificial Intelligence (AI), ML, and DL, highlighting their distinctions and interrelationships. Subsequently, it reviews existing methods for GNSS-INS integration, discussing their advantages and inherent limitations. Finally, it delves into how DL, especially Recurrent Neural Networks (RNNs), offers promising solutions to overcome the challenges posed by traditional methods, particularly in scenarios where GNSS signals are unreliable or unavailable.

3.1 Introduction to Artificial Intelligence, Machine Learning, and Deep Learning

3.1.1 Artificial Intelligence

Artificial Intelligence (AI) is a multidisciplinary field that aims to create machines or systems capable of performing tasks that typically require human intelligence. These tasks encompass reasoning, learning, decision-making, perception, and natural language processing [44]. AI systems are designed to interpret complex data, adapt to new inputs, and perform human-like functions, augmenting or replicating human cognitive abilities.

AI can be broadly categorised into two types:

- **Narrow AI**: Systems engineered to perform tasks proficiently, such as facial recognition, speech recognition, or strategic game playing. These systems operate within a limited scope and cannot generalise their intelligence to unrelated domains [45].
- **General AI**: Hypothetical systems that can understand, learn, and apply intelligence across various tasks, emulating human cognitive versatility. General AI remains a theoretical construct and has not been realised in practical applications [46].

3.1.2 Machine Learning

Machine Learning (ML) is a subset of Artificial Intelligence that focuses on developing algorithms that allow programs to learn from data and make decisions or predictions. ML systems improve their performance iteratively by identifying patterns and relationships within datasets rather than depending on explicit programming for each task [47]. This process enables ML algorithms to make data-driven decisions.

ML algorithms are generally classified into three main categories: Supervised Learning, Unsupervised Learning, and Reinforcement Learning. Supervised Learning involves training models on a labelled dataset, where each input is paired with a corresponding output label. The goal is to learn a mapping from inputs to outputs that can generalise to previously unseen data. Everyday supervised learning tasks include regression, which

predicts continuous values, and classification, which assigns inputs to discrete categories [48].

Unsupervised Learning deals with unlabelled data and aims to discover inherent structures or patterns without predefined labels. Techniques such as clustering group similar data points, while dimensionality reduction methods simplify data representation by reducing the number of variables. Reinforcement Learning involves an agent interacting with an environment to learn optimal behaviours through trial and error. The agent receives feedback through rewards or penalties and learns a policy that maximises cumulative rewards over time [49].

3.1.3 Deep Learning (DL)

Deep learning is a specialized subdomain of machine learning that utilizes artificial neural networks composed of multiple layers to model complex patterns and relationships in data. These architectures, referred to as deep neural networks (DNNs), are loosely inspired by the biological neural networks of the human brain. By leveraging large volumes of data, deep learning models autonomously learn hierarchical representations, progressively capturing features at varying levels of abstraction.

A distinguishing feature of deep learning is its capacity for hierarchical feature extraction. In image processing, for instance, lower layers may detect basic elements such as edges or textures, while deeper layers discern more abstract patterns like object parts or entire objects. Nonlinear transformations, achieved through activation functions, enable deep networks to approximate intricate, highly nonlinear relationships between inputs and outputs, thereby improving their ability to model complex real-world phenomena.

Moreover, deep learning frameworks facilitate end-to-end learning, wherein models directly map raw inputs to outputs without requiring extensive manual feature engineering or domain-specific preprocessing. This characteristic, combined with the ability to generalize across diverse domains, has made deep learning an effective solution for tasks involving unstructured data, such as image recognition, natural language processing, and autonomous navigation.

3.2 Existing Methods for GNSS-INS Integration

The Kalman filter, a foundational tool in estimation theory, was introduced by Rudolf E. Kálmán in 1960 in his landmark paper, "A New Approach to Linear Filtering and Prediction Problems" [50]. It provides a computationally efficient recursive method to estimate the state of a linear dynamic system subject to random noise. The innovation of the Kalman filter lies in its probabilistic framework, which iteratively combines model-based predictions with noisy observations to refine state estimates. This recursive nature eliminates the need to retain extensive historical data, making it practical and robust for real-time applications.

The Kalman filter operates in two key steps: prediction and update. Based on a mathematical model in the prediction step, the system's state is propagated forward in time, and the associated uncertainty is estimated. Mathematically, this is expressed as:

$$State\ Prediction: \boldsymbol{x}_{k|k-1} = \boldsymbol{A}\boldsymbol{x}_{k-1|k-1} + \boldsymbol{B}\boldsymbol{u}_k \quad (\mathbf{3.1})$$

$$Covariance\ Prediction: \boldsymbol{P}_{k|k-1} = \boldsymbol{A}\boldsymbol{P}_{k-1|k-1}\boldsymbol{A}^{\boldsymbol{T}} + \boldsymbol{Q} \quad (\mathbf{3.2})$$

where $\boldsymbol{x}_{k|k-1}$ represents the predicted state estimate, $\boldsymbol{A}$ is the state-transition matrix, $\boldsymbol{B}$ accounts for control inputs $\boldsymbol{u}_k$, $\boldsymbol{P}_{k-1|k-1}$ is the covariance of the predicted state estimate, and $\boldsymbol{Q}$ is the process noise covariance.

In the update step, the system refines its state estimate and associated uncertainty by incorporating observations to correct the prediction. This process is mathematically described as follows:

$$Measurement\ Prediction: \boldsymbol{y}_{\mathrm{k}} = \boldsymbol{H}\,\boldsymbol{x}_{\mathrm{k}}|_{\mathrm{k-1}} + \boldsymbol{v}_{\boldsymbol{k}} \quad (\mathbf{3.3})$$

$$Kalman\ Gain: \boldsymbol{K}_{\mathrm{k}} = \boldsymbol{P}_{\mathrm{k}}|_{\mathrm{k}}^{-1}\,\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{P}_{\mathrm{k}}\big|_{\mathrm{k}}^{-1}\,\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R})^{-1} \quad (\mathbf{3.4})$$

$$State\ Update: \boldsymbol{x}_{\mathrm{k}}|_{\mathrm{k}} = \boldsymbol{x}_{\mathrm{k}}|_{\mathrm{k}}^{-1} + \boldsymbol{K}_{\mathrm{k}}\,(\boldsymbol{z}_{\mathrm{k}} - \boldsymbol{H}\,\boldsymbol{x}_{\mathrm{k}}\big|_{\mathrm{k}}^{-1}) \quad (\mathbf{3.5})$$

Here, $\boldsymbol{H}$ is the observation model, $\boldsymbol{v}_{\mathrm{k}}$ represents measurement noise, $\boldsymbol{K}_{\mathrm{k}}$ is the Kalman gain balancing model predictions with observations, $\boldsymbol{z}_{\mathrm{k}}$ is the actual measurement and $\boldsymbol{R}$ is the measurement of noise covariance. The Kalman gain, $\boldsymbol{K}_{\mathrm{k}}$, adapts dynamically to ensure the best weighting between the model and the observed data.

The Kalman filter minimises the estimation error covariance $\boldsymbol{P}$ in a least-squares sense, providing statistically optimal state estimates under the assumptions of Gaussian noise.

One of the most impactful applications of the Kalman filter is in sensor fusion, particularly for integrating GNSS and INS. GNSS provides accurate, low-frequency position and velocity measurements. At the same time, INS delivers high-frequency data on acceleration and orientation but suffers from drift over time due to sensor biases and noise. The Kalman filter offers a solution to navigation problems by fusing these complementary data sources.

In GNSS/INS integration, the state vector typically includes terms such as position, velocity, and bias. During the prediction step, the INS measurements propagate the state forward. In the update step, GNSS observations correct the drift and biases in the INS data. This combination ensures reliable and continuous navigation even in challenging environments such as urban canyons or during GNSS signal loss.

The mathematical formulation for GNSS/INS integration extends the Kalman filter to include sensor biases [26]:

$$\boldsymbol{x}_{\mathrm{k}} = [\boldsymbol{r}_{\mathrm{k}}\boldsymbol{v}_{\mathrm{k}}\boldsymbol{b}_{\mathrm{k}}]^{\mathrm{T}} \tag{3.6}$$

$$A = \begin{bmatrix} I & \Delta t & 0 \\ 0 & I & -\Delta t\, I \\ 0 & 0 & I \end{bmatrix} \tag{3.7}$$

Here, $\boldsymbol{r}_{\mathrm{k}}$ and $\boldsymbol{v}_{\mathrm{k}}$ denote position and velocity, respectively, $\boldsymbol{b}_{\mathrm{k}}$ represents biases and Δt is the time interval between updates. GNSS observations, incorporated through the observation model H, correct the predicted states, while the process and measurement noise covariances, Q and R, are calibrated based on sensor specifications. By integrating predictive modelling with real-time corrections, the Kalman filter has become an indispensable tool for navigation systems, offering precision and reliability across diverse operational scenarios.

Kalman filtering is a widely used method for fusing multiple sensors and modelling the uncertainty of the estimates. It is based on the Bayesian approach and the linear Gaussian model, which assume that the system and measurement models are linear, and the noise is Gaussian. Kalman filtering can provide optimal estimates in the minimum mean square

error (MMSE) sense, but it requires the system and measurement models to be known and the noise to be Gaussian. In navigation, this method is used to estimate the motion of a vehicle. However, these motions or error models may not be linear. To address these limitations, various Extended Kalman filtering (EKF) [51] and Unscented Kalman filtering (UKF) methods have been proposed, which can handle non-linear models and noise by linearising the models around the current estimate or by propagating a set of sample points (sigma points) through the non-linear models [52]. Nonetheless, EKF linearisation using first−order Taylor series around the state estimate introduces other errors, and UKFs are sensitive to selecting appropriate sigma points.

Non-linear optimisation methods such as least squares estimate the integrated position by minimising the sum of the squared errors between the observed GNSS measurements and the predicted measurements from the inertial navigation system. [53]. Maximum Likelihood Estimation (MLE) seeks to find the parameter values that maximise the likelihood of the observed data given a statistical model. In the context of INS/GNSS integration, MLE aims to find the parameters (e.g., INS biases, scale factors, GNSS receiver clock bias, etc.) that maximise the likelihood of the observed GNSS measurements given the INS predictions. [54]. These methods can handle non-linear models and noise, but they require the model to be known and the measurements to be consistent. They also suffer from slow convergence and sensitivity to initialisation, which can limit their effectiveness in GNSS/INS integration.

Paper [55] proposes a multi-sensor fusion system using an Inertial Measurement Unit (IMU), GNSS, and Visual Odometer (VO) for use in urban canyons using a non-holonomic EKF. Non-holonomic Extended Kalman Filter (EKF) is an estimation algorithm that explicitly addresses the motion constraints associated with non-holonomic systems, such as wheeled or tracked vehicles, where specific directions of motion are restricted. The author demonstrates that the systems provide a 3D Root Mean-Squared Error of 3.9m in noisy GNSS urban areas with a further reduction of 3D RMSE of 1.29m 0.00073 m/s and 0.486 degrees when GNSS outages occur. However, this system requires a third sensor (VO) whose performance under different weather or environmental conditions was not explored [56] in. Furthermore, as this filter uses an EKF, the sensor errors are assumed to be of Gaussian distribution, which is not always the case [57].

In paper [37], the author implements a tightly coupled GNSS/INS integration by adopting a closed-loop error-state EKF. Using chi-square distribution probability, the fault detection algorithm compares the calculated statistic against a threshold. The advantage of using tightly coupled fusion architectures is reduced processing errors exhibited in other coupling scenarios, allowing for better positioning prediction. Furthermore, a fault detection system allows erroneous data to be detected and processed appropriately. However, tightly coupled systems require raw sensor data that is unavailable from commercial sensors. Furthermore, the chi-squared test may not detect slow ramp errors that could occur over time [58]. This is especially true in INS systems, as slow sensor drift is typical.

3.3 Limitations of Kalman Filter-Based Methods

Despite the widespread adoption of KF-based methods in GNSS-INS integration, these approaches exhibit several inherent limitations that can compromise performance in real-world scenarios. The KF operates optimally under the assumptions of linearity in system dynamics and measurement models, as well as Gaussian-distributed and uncorrelated process and measurement noises. However, practical navigation systems often violate these assumptions due to several reasons.

The nonlinearity of system dynamics is one such challenge. The motion of a vehicle or platform, particularly in complex manoeuvres or high-dynamic environments, is inherently nonlinear. The standard KF cannot handle highly nonlinear relationships between the state variables and the measurements, leading to suboptimal estimates or filter divergence. Non-Gaussian and correlated noise further complicate matters. Sensor measurements and environmental disturbances may exhibit non-Gaussian noise characteristics and temporal correlations. Factors such as multipath effects in GNSS signals, sensor bias drifts, and correlated noise in inertial sensors challenge KF's assumption of white Gaussian noise, affecting estimation accuracy [59][60].

These limitations highlight the challenges traditional KF-based GNSS-INS integration methods face in handling the complexities of real-world navigation scenarios. The need for approaches that accommodate nonlinearity, non-Gaussian noise and modelling uncertainties has become increasingly apparent.

3.4 Emergence of Deep Learning (DL) Approaches

In response to traditional methods' limitations, there has been a growing interest in leveraging ML techniques for GNSS-INS integration. ML approaches can potentially model complex, nonlinear relationships directly from data without relying on explicit analytical models. This data-driven paradigm shift enables the system to learn patterns and adapt to varying conditions that are challenging to capture with conventional modelling.

Key motivations for adopting ML in GNSS-INS integration include handling nonlinearities, modelling unknown dynamics, adaptability and learning capability, and recent computational improvements. ML algorithms are inherently capable of approximating nonlinear functions. This property allows them to model the intricate relationships between sensor measurements and nonlinear navigation states. ML approaches can also learn system dynamics and sensor characteristics from historical data, capturing behaviours that are challenging to model analytically due to their complexity or stochastic nature. By learning directly from data, ML models can adapt to the statistical properties of the noise in the measurements, including non-Gaussian and correlated noise patterns. Moreover, ML models can be updated and retrained as new data become available, allowing the system to adapt to environmental changes, sensor degradation, or system modifications over time. Advances in hardware acceleration and efficient algorithms have made it feasible to deploy ML models in real-time applications, addressing concerns about computational complexity. Lastly, ML models, such as recurrent-based networks, can use time-dependent data to capture temporal patterns, learn sequential dependencies, and make predictions based on the evolution of input over time. Such an approach can aid in understanding relationships, such as the accumulation of errors in INS over time, enabling more accurate predictions and effective correction strategies.

3.5 Deep Learning Techniques for GNSS-INS Integration

Recurrent Neural Networks (RNNs), a class of deep learning models designed for sequential data, are well-suited for processing time-series sensor data inherent in navigation systems. Derived from feedforward neural networks by David Rumelhart [61], they use part of the previous state outputs as part of the current input and, therefore,

establish a relationship between the output and the input. These specific neural networks have been used primarily in handwriting and speech recognition. Recurrent Neural Networks have been used for sensor fusion in the past. The RNN algorithm is shown in Equations 1 and 2 [62]:

$$h_t = \sigma_h(W_{hh}h_{t-1} + W_{xh}X_t + b_h) \tag{3.8}$$

$$y_t = \sigma_y\text{-}W_y h_t + b_y \tag{3.9}$$

where h_t is the hidden layer vector, y_t is the output vector, X_t is the input vector, W_{hh}, W_{xh}, W_y are the parameter matrices, b_h, b_y are the bias terms and σ_h, σ_y are the activation functions. RNNs take the information from the previous state h_{t-1} and multiply it with a weight matrix. The same is done for the new input X_t and is then combined with the previous state to create the new hidden state h_t. Because RNN is structured, information from the past can be linked with information from the current time step.

The activation functions (Tanh) assess the sum weights of the input and decide which information is needed. However, RNNs suffer from vanishing and exploding gradient problems due to the weight matrix used to multiply with the previous state. This may only provide a short-term memory for the past information being inputted. It also means that some information can be propagated even though it is useless. Two types derived from RNNs (GRU and LSTMs) solve these issues. An example of GRU architecture is shown below.

LSTMs and GRUs use gates to determine whether the input information should be kept or removed. In a GRU, two gates deal with this. An update gate is used to help the neural network determine how much information from the previous step needs to be passed to the next step. The formula for this is presented below [63]:

$$z_t = \sigma(W^z X_t + U^z h_{t-1}) \tag{3.10}$$

The reset gate is used to determine how much of the past information should be forgotten to improve the performance of the GRU. The formula is presented below:

$$r_t = \sigma(W^r X_t + U^r h_{t-1}) \tag{3.11}$$

These gates help remove the issues related to exploding and vanishing gradients. However, the additional gates add computational complexity to the neural network, which may slow down the training process. LSTMs, on the other hand, use three gates to solve these issues: the input gate, output gate, and forget gate. Because LSTMs use three gates, they are more computationally expensive than GRUs but can provide more accurate information for training the model.

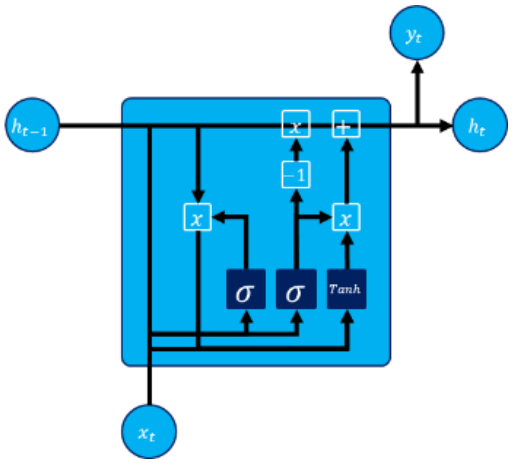


Figure 3-1 Inside a Gated Recurrent Unit Neural Network Node

Deep learning models can be used to predict and correct sensor errors in GNSS-INS integration, particularly during GNSS signal outages or in environments with degraded GNSS performance. By training on historical sensor data, these models learn the patterns of inertial sensor drift and biases or classify GNSS signals, enabling the prediction of errors based on past observations. These capabilities allow for the compensation of INS drift (without reliance on external GNSS updates) or improving GNSS positioning accuracy, enhancing the robustness of the navigation system.

Moreover, deep learning facilitates the fusion of heterogeneous sensor data beyond traditional GNSS and INS measurements. They can be used as part of a larger federated system to improve navigation accuracy.

3.6 Conclusion

This chapter has traced the evolution of sensor integration techniques for GNSS-INS, from traditional Kalman filter-based methods to advanced deep learning approaches. While Kalman filtering remains a cornerstone in sensor fusion, its limitations in handling

nonlinearity, noise characteristics, and modelling uncertainties necessitate more robust solutions. Deep learning models, particularly recurrent architectures such as RNNs, GRUs, and LSTMs, offer a data-driven alternative by directly learning complex, nonlinear relationships from time-series data. These models can effectively predict and correct INS drift during GNSS signal outages, improving overall system robustness.

Beyond addressing traditional challenges, deep learning expands the scope of sensor fusion by enabling the integration of diverse sensor modalities, such as vision-based sensors. By learning spatial and temporal patterns, hybrid architectures that combine models show promise in enhancing navigation accuracy in GNSS-denied environments. This progression toward data-driven methodologies marks a significant step in improving UAV navigation in complex urban settings and other challenging environments

Despite deep learning's promising capabilities in GNSS-INS integration, several challenges must be addressed to realise its full potential. Training deep learning models requires large amounts of high-quality data that capture the diversity of real-world operating conditions. Collecting and annotating such datasets can be resource-intensive. The next chapter will introduce the Hardware-in-the-loop data collection system created to collect extensive and realistic datasets to improve machine learning performance.

Chapter 4 Data Generation Framework

This chapter outlines the data collection methodology employed in this research, focusing on differentiating between real-world data, data generated through Hardware-in-the-Loop (HIL) simulation and synthetic data. HIL simulation is a realistic alternative to real-world data collection because it integrates fundamental hardware components with a controlled, simulated environment. This approach enables the capture of authentic hardware responses while benefiting from the flexibility and repeatability of simulated environments. The justification for HIL simulation arises from the need to test systems under various impractical conditions to replicate consistently in real-world scenarios.

The research employs a comprehensive simulation framework, incorporating SimGEN GSS7000 HIL, Sim3D for environmental modelling, SimSENSOR for inertial sensor outputs, and Unreal Engine with AirSim for dynamic scene generation. Each component plays a critical role in creating a high-fidelity simulation environment. A significant aspect of the framework is the inclusion of non-Gaussian noise models, which accurately mimic the complex error patterns encountered in real-world sensors, thereby enhancing the validity of the results.

4.1 Real-World Data vs. HIL Simulation

4.1.1 Real-World Data Collection

Real-world data collection is considered a fundamental method for capturing the actual performance of systems in their natural environments. This data type is generated by deploying sensors and systems in real-world conditions, allowing them to interact directly with environmental factors such as weather, physical obstacles, and signal interference. The main advantage of real-world data is its authenticity. The data inherently includes all the unpredictable variables and noise present in real-life scenarios, which provides a comprehensive understanding of how the system performs under uncontrolled conditions [64].

However, real-world data collection has its challenges. One primary difficulty is the cost of conducting large-scale field tests, especially when testing under extreme or unusual conditions. The time required to collect extensive datasets in the field can be prohibitive, and logistical issues further complicate the process. Safety concerns also arise when testing systems in high-risk environments, such as during severe weather or in scenarios involving high-speed dynamics. Moreover, the reproducibility of real-world data is limited, as it is nearly impossible to recreate identical environmental conditions for repeated experiments. This lack of control over variables makes it difficult to isolate specific factors for detailed analysis.

4.2 HIL Simulation

HIL simulation offers a practical and realistic alternative to real-world data collection. This approach integrates real hardware components, such as sensors or control systems, into a simulated environment. The simulation provides virtual representations of environmental factors, allowing the hardware to operate as it would in actual conditions. The advantage of HIL simulation lies in its ability to control and manipulate ecological variables systematically, enabling a thorough examination of system behaviour under a wide range of conditions [65].

HIL simulation is particularly valuable when real-world testing is too expensive, time-consuming, or hazardous. For instance, simulation without risking personnel or equipment can test a system’s performance under extreme weather conditions, such as

high winds or low visibility. Additionally, HIL simulation offers the benefit of reproducibility, as identical tests can be run repeatedly with complete control over the conditions. This ensures consistent and reliable results, allowing for a more precise analysis of how specific variables affect system performance.

While HIL simulation can closely mimic real-world conditions, one challenge is ensuring that the simulated environment accurately reflects the complexities of the real world. No matter how advanced, simulations may miss certain subtleties or interactions that could influence real-world outcomes. Nonetheless, modern HIL systems, particularly those incorporating non-Gaussian noise models, can produce highly realistic data comparable to real-world data in many respects.

4.2.1 Justification for HIL Simulation in This Research

The research objectives necessitate testing the system under diverse and challenging conditions, some of which are impractical or impossible to replicate in real-world scenarios. Hardware-in-the-loop (HIL) simulation provides a compelling solution by offering a controlled yet highly realistic environment to evaluate system performance rigorously. This approach bridges the gap between purely simulated and real-world testing by combining the flexibility and repeatability of simulation with the physical integration of hardware components, enabling the system to interact with simulated inputs in real-time.

One key advantage of HIL simulation is its ability to precisely manipulate environmental variables, sensor errors, and noise models. This capability is particularly valuable for investigating extreme or rare conditions, such as severe multipath interference in urban canyons, ionospheric scintillation in equatorial regions, or signal blockages in mountainous terrain. Testing these scenarios in the real world would be logistically complex, resource-intensive, and, in some cases, infeasible. By contrast, HIL simulation allows researchers to configure these conditions with precision and observe how the system responds, enabling detailed performance analysis and the identification of potential vulnerabilities.

Another critical advantage is the repeatability offered by HIL simulation. Specific test conditions can be replicated precisely and consistently, allowing researchers to isolate

and study particular system behaviours. This repeatability is essential for ensuring the robustness and reliability of the system under study, as it provides a systematic way to identify and address performance issues. Additionally, HIL simulation supports iterative testing, enabling researchers to refine algorithms and hardware designs in response to observed results.

The simulation framework employed in this research integrates advanced tools, including SimGEN GSS7000 HIL, Sim3D, SimSENSOR, and Unreal Engine with AirSim. Together, these tools provide a comprehensive testing environment that combines high-fidelity GNSS signal simulation, three-dimensional ray tracing for multipath analysis, sensor error modelling, and realistic virtual environments for system interaction. Including non-Gaussian noise models further enhances the realism of the simulations, allowing for a more accurate representation of real-world conditions. For example, these tools can simulate time-varying multipath effects, atmospheric noise, and hardware-specific inaccuracies critical for assessing system performance in complex scenarios.

HIL simulation also offers a more efficient and cost-effective alternative to real-world data collection. Field tests often require significant logistical coordination, high costs, and extended timelines, mainly for testing in remote or extreme environments. HIL simulation mitigates these challenges by enabling comprehensive laboratory testing, where conditions can be easily adjusted and experiments repeated as needed. Furthermore, it reduces the risk associated with testing systems in real-world scenarios, mainly when those systems are in the early stages of development.

By providing a controlled yet realistic testing environment, HIL simulation supports the research objectives more effectively than either real-world testing or purely software-based simulations alone. It offers the precision and repeatability required for systematic analysis while maintaining a level of realism sufficient to generate actionable insights. This approach ensures that the system is rigorously evaluated under a wide range of conditions, ultimately contributing to the development of robust and reliable solutions that can perform effectively in real-world applications.

4.3 Data Collection Overview

Objective Three (O3) proposes creating a simulation framework for real-time recording of outputs from multiple sensors, using various techniques to enhance realism. This is shown in Figure 4-1. It includes using Hardware-in-the-Loop simulation, where GNSS receivers process simulated GNSS Radio Frequencies (RF) based on predefined parameters and trajectories. As a result, the output RF mimics the signals an antenna receives and is processed to produce position data. This data is then compared to the simulation's ground truth to analyse GNSS receiver errors. Ray-tracing software calculates signal propagation conditions to simulate multipath conditions in urban canyons more realistically, considering reflection and refraction. This approach improves the simulation's realism and tests the system's robustness.

In addition, stochastic and deterministic error models for IMUs are employed to enhance the realism of simulations for accelerometer, gyroscope, and barometer measurements from a moving vehicle.

A virtual environment generates visual data for VO measurements using a camera mounted on a virtual drone. This data, stored with timestamps, allows for correlation with other sensor readings. The dynamic environment features changing weather conditions and map environments to test system robustness further.

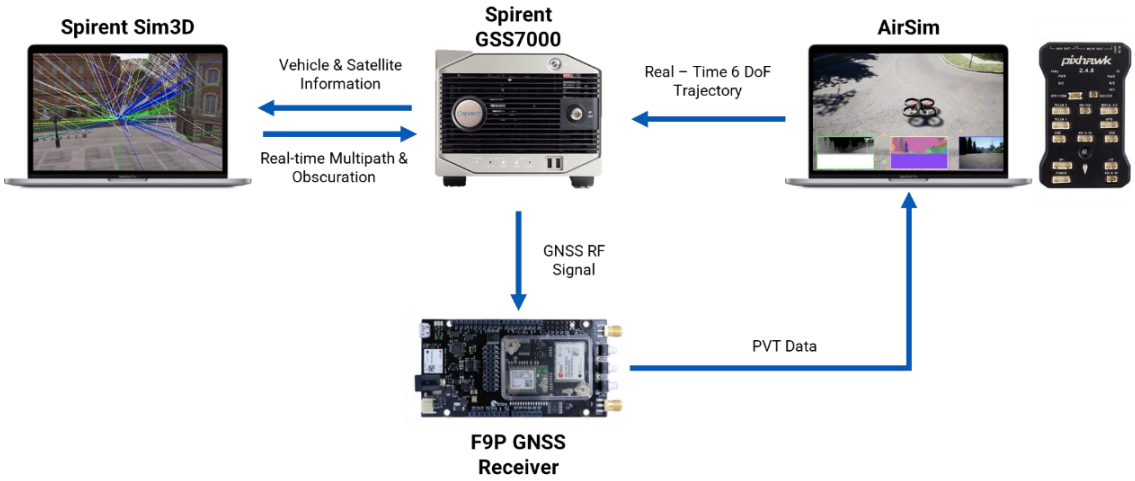


Figure 4-1 multi-system, multi-sensor data flow

4.4 Simulation Framework Components

4.4.1 SimGEN GSS7000 HIL

The SimGEN GSS7000 system is a central component of the GNSS simulation framework, offering the capability to generate complex and realistic GNSS signals. Its functionality involves producing multi-constellation GNSS signals based on predefined trajectories and environmental parameters. These signals are then transmitted as real-time radio frequency (RF) outputs to GNSS receivers, enabling these receivers to behave as they would under real-world conditions. The hardware integration ensures high-fidelity interaction between the simulated environment and the GNSS receiver, capturing the nuances of signal propagation and processing in a controlled setting.

One of the features of SimGEN GSS7000 is its ability to simulate satellite orbits, signal impairments, and atmospheric effects with a high degree of realism. The system accurately models the motion of GNSS satellites and the resulting Doppler effects experienced by signals as they travel to receivers. Additionally, it incorporates detailed atmospheric effects, including ionospheric delays, scintillation, and tropospheric attenuation, which are critical for understanding signal behaviour in different environmental conditions. For example, ionospheric scintillation can significantly degrade GNSS signals in equatorial or high-latitude regions, and its modelling often involves Rician or Nakagami distributions to represent amplitude and phase variations. Tropospheric noise, on the other hand, is modelled based on elevation angle and weather conditions, adding another layer of complexity to the simulation.

Including non-Gaussian noise models in the SimGEN GSS7000 further enhances its capability to simulate real-world conditions. Unlike traditional simulators that assume Gaussian distributions for noise, SimGEN supports noise characteristics that deviate from this assumption, providing a more authentic representation of the errors encountered by GNSS receivers in practice. Examples include oscillator noise, which exhibits a power-law spectral density, leading to phase noise with time-domain characteristics that are not Gaussian, and receiver noise, which can introduce skewness or kurtosis in the signal. Integrating diverse noise models is crucial for testing GNSS receivers in urban or

obstructed environments, where multipath effects, signal blockages, and reflections result in highly variable and non-Gaussian error distributions.

The default settings of the SimGEN GSS7000 are designed to provide a robust and reliable baseline for simulation scenarios while allowing extensive customisation for specific use cases. These defaults typically include parameters for multi-constellation signal generation, including GPS, GLONASS, Galileo, and BeiDou systems, with satellite orbits and clock models pre-configured based on standard almanac data. Atmospheric effects, such as ionospheric and tropospheric delays, are also enabled by default, with settings based on established atmospheric models. Additionally, essential Gaussian noise is included in the default setup to represent typical receiver environments, while users can enable more complex noise characteristics, such as scintillation or multipath effects, for advanced testing scenarios.

The SimGEN GSS7000 provides researchers and engineers with a detailed and flexible simulation environment that allows them to test GNSS-based systems under various conditions. This includes scenarios ranging from ideal open-sky environments to highly obstructed urban canyons or GNSS-denied environments. The ability to precisely control and replicate signal conditions is invaluable for validating receiver performance, optimising algorithms, and developing mitigation strategies for real-world challenges in GNSS applications.

Figure 4-2: Image of Spirent GSS7000 Simulator [66]

4.4.2 U-blox F9P multi-constellation GNSS receiver

The decision to utilise the U-blox F9P multi-constellation GNSS receiver over other options, such as the U-blox M8 or products from different vendors, was based on several key factors pertinent to our research objectives. Firstly, the U-blox F9P offers multi-constellation capability, supporting GPS, GLONASS, Galileo, and BeiDou simultaneously. This feature significantly enhances the availability and accuracy of positioning data, particularly in challenging environments such as urban canyons, where signal blockage and multipath effects are prevalent. This, therefore, improves the simulation capabilities and widens testing circumstances, aligning it more realistically to real-world testing.

Another crucial consideration is the F9P's support for dual-frequency reception (L1/L2), which mitigates ionospheric errors and enhances positioning reliability. This feature is essential for simulating high-precision GNSS applications, where accuracy and reliability are important safety factors.

Furthermore, the integration capabilities of the U-blox F9P with our existing simulation framework, including hardware-in-the-loop (HIL) systems, were evaluated. Hobbyist communities and open-source software supported the extraction of raw and calculated values.

Figure 4-3 Image of ZED-F9P [67]

4.4.3 Sim3D

Sim3D is a specialised ray-tracing software that plays a critical role in the GNSS signal simulation framework, particularly when paired with the SimGEN GSS7000 simulator. Its primary function is to model the propagation of GNSS signals in three-dimensional environments, capturing the complex interactions that occur between these signals and

their surroundings. By leveraging advanced ray-tracing algorithms, Sim3D calculates how GNSS signals are reflected, diffracted, or attenuated as they encounter physical obstacles, such as buildings, vehicles, terrain features, or vegetation. These interactions are collectively known as multipath effects and are among the most significant sources of error in GNSS systems, particularly in urban or obstructed environments.

In urban environments, for instance, GNSS signals often traverse areas densely populated with reflective and obstructive surfaces. In such conditions, signals may reach the receiver via multiple paths due to reflections off nearby buildings or other structures, or they may be diffracted around edges. Sim3D is designed to simulate these interactions with high fidelity, accounting for both line-of-sight (LOS) and non-line-of-sight (NLOS) conditions. This allows for a detailed examination of the multipath-induced errors that degrade the accuracy of GNSS positioning, timing, and velocity measurements. Unlike traditional simulation methods, which often rely on simplified Gaussian error models, Sim3D produces error distributions that more closely reflect the statistical and spatial complexities observed in real-world environments. This includes the generation of non-Gaussian, asymmetric error profiles that arise from dynamic urban environments, where signal interactions are highly variable and context-dependent.

The capacity of Sim3D to simulate these interactions with such precision is essential for understanding the performance of GNSS systems under challenging conditions. It provides a robust framework for testing and evaluating GNSS-based navigation systems, particularly those designed for use in environments prone to high levels of multipath interference and NLOS signal propagation. By incorporating detailed environmental models and simulating the physics of signal propagation, Sim3D contributes to the development of more resilient and accurate GNSS receivers. This is especially significant for applications where reliability under adverse conditions is paramount, such as autonomous navigation in urban canyons, aviation operations near airports, and surveying in environments with significant physical obstructions.

Furthermore, the inclusion of Sim3D in the simulation process enables researchers to explore the interactions between GNSS signals and specific environmental features in ways that are not feasible in real-world testing. For example, by manipulating the virtual environment, it is possible to isolate the effects of individual factors, such as building

density or signal frequency, on GNSS performance. This provides valuable insights into how environmental conditions influence system performance, allowing for the design of mitigation strategies and the optimisation of receiver algorithms. By expanding the scope of GNSS simulations to include these real-world complexities, Sim3D represents a significant advancement in the field of GNSS testing and evaluation [68].

4.4.4 SimSENSOR

SimSENSOR is another component of the simulation framework, providing high-fidelity simulations of inertial sensor outputs for navigation systems. This tool is specifically designed to replicate the behaviour of standard sensors, including accelerometers, gyroscopes, and magnetometers, which are critical for estimating a system's position, velocity, and orientation. Inertial sensors are especially important in scenarios where GNSS signals are unavailable, degraded, or unreliable, such as in urban canyons, tunnels, or during periods of intentional signal interference. By generating realistic inertial sensor data, SimSENSOR enables comprehensive testing and validation of navigation systems under such challenging conditions.

At the core of SimSENSOR is its ability to simulate both deterministic and stochastic error models, ensuring that the generated outputs accurately represent the real-world behaviour of inertial sensors. Deterministic errors include biases, scale factor variations, and cross-axis sensitivity, which are predictable and specific to the sensor's design and manufacturing. Stochastic errors, on the other hand, are inherently random and include phenomena such as random walks, bias instability, and quantization noise. These errors significantly impact the accuracy of inertial sensors, particularly over time, as they tend to accumulate in the absence of external corrections.

One of the features of SimSENSOR is its ability to incorporate Allan variance-based modelling. Allan variance is a critical metric for characterizing the noise and stability of inertial sensors over different time scales. By modelling sensor behaviour using parameters derived from Allan variances, such as random walk coefficients and bias instability, SimSENSOR generates data that closely mimics the real-world performance of MEMS sensors. This capability is essential for testing systems that rely heavily on inertial sensors for accurate navigation, particularly in GNSS-denied environments. For example, long-term drift in gyroscopes and accelerometers can lead to significant

deviations in position and orientation estimates, and SimSENSOR provides a reliable means to evaluate and mitigate such effects.

SimSENSOR also supports dynamic simulations, enabling it to generate sensor outputs that correspond to specific motion profiles of the simulated vehicle or object being tested. This includes modelling linear and rotational dynamics, such as acceleration, deceleration, and angular velocity changes. The ability to integrate dynamic profiles ensures that the simulated data reflects the complex interactions between sensor outputs and the system's motion. This is particularly important for testing navigation algorithms that fuse inertial data with other sources, such as GNSS or visual odometry.

Furthermore, SimSENSOR's versatility extends to simulating scenarios with varying environmental conditions and sensor configurations. For instance, it can replicate the effects of temperature variations on sensor performance or simulate sensor degradation over time. These capabilities allow researchers to evaluate the robustness of navigation systems under a wide range of operating conditions. By generating high-quality sensor data with precisely controlled error characteristics, SimSENSOR enables a deeper understanding of system behaviour, facilitating the development of more accurate and resilient navigation solutions.

Incorporating SimSENSOR into the simulation framework provides a comprehensive approach to evaluating navigation systems, particularly in scenarios where reliance on inertial sensors is critical. Its ability to replicate realistic sensor behavior, model complex error characteristics, and simulate dynamic motion profiles ensures that navigation algorithms and hardware are rigorously tested and validated. This, in turn, enhances the overall reliability and performance of systems deployed in real-world environments, where navigation challenges are often unpredictable and multifaceted.

The default values for SimSENSOR are as follows:

Table 4-I Standard IMU errors used in SimSENSOR

IMU Sensor A			
Accelerometer		**Gyroscope**	
Scaling factor (ppm)	600	**Scaling factor (ppm)**	500
Bias (ug)	20	**Bias (deg/h)**	3
ARW (ug/sqrt(Hz))	100	**GRW (deg/s/sqrt(h))**	0.004
Update rate (Hz)	100		

However, if a test has variations on the above values, these will be mentioned in the chapter explicitly.

4.4.5 Unreal Engine with AirSim

The Unreal Engine and the AirSim plugin provide a high-fidelity 3D simulation environment for testing various systems in dynamic, visually realistic scenarios. Unreal Engine is widely recognised for its ability to generate photorealistic graphics and complex physics interactions, making it an ideal platform for simulating environments that closely resemble the real world. AirSim extends the capabilities of Unreal Engine by adding support for simulating vehicles, such as drones and cars, within these environments.

Unreal Engine with AirSim is critical in generating visual data, such as camera images and depth information, used in algorithms like Visual Odometry (VO) and other vision-based navigation systems. The system can simulate dynamic environments, including moving objects, weather conditions, and lighting variations, all of which can affect the performance of navigation algorithms. Non-Gaussian noise models are also integrated into this component, particularly in the form of visual artefacts such as motion blur, lighting distortions, and other imperfections that influence the quality of visual data. This makes the simulated environment more realistic and ensures that the visual sensors are tested under conditions representative of the real world.

Figure 4-4: Example Unreal Engine Environment with AirSim

4.5 Integration of Simulation Components

4.5.1 Data Flow and Synchronization

The simulation framework operates across three primary PCs, each handling a specific component: Sim3D, SimGEN GSS7000, and Unreal Engine with AirSim. These components communicate using established protocols to ensure synchronised operation and data consistency throughout the system.

Sim3D runs on a dedicated PC and communicates directly with the SimGEN GSS7000 using a high-speed Ethernet connection. The data exchange leverages TCP/IP for reliable transfer of environmental factors, including ray-traced signal reflections, blockages, and multipath effects. These factors are dynamically computed in Sim3D and integrated into the GNSS signal generation process within SimGEN. The modified GNSS signals are then output as RF signals for hardware receivers or as digital streams over the same Ethernet connection for software analysis.

A second PC hosts the Unreal Engine and AirSim, which generate visual data such as camera images and depth information and simulate dynamic environmental conditions like weather and lighting. This PC is connected to SimGEN through TCP/IP for trajectory updates and environmental parameter synchronisation. The visual data produced by Unreal Engine, including images and motion profiles, is streamed to the main data aggregation system via UDP to minimise latency and support real-time processing requirements.

SimSENSOR is integrated into the system to simulate inertial sensor outputs. It communicates with the other components through a serial connection using UART and transmits data reflecting motion profiles, including accelerometer and gyroscope readings. The inertial data is synchronised with the GNSS and visual data to ensure coherent system behaviour, particularly in scenarios where GNSS signals are degraded or unavailable.

Time synchronisation across all components is achieved using Precision Time Protocol (PTP), ensuring that the data streams from SimGEN, Sim3D, and Unreal Engine with AirSim are temporally aligned. This synchronisation is crucial for accurately reproducing

real-world scenarios, where sensor data must be consistent for effective fusion and analysis.

The simulation setup's modular design allows for seamless integration of components while ensuring robust communication. The protocols, TCP/IP for reliable data transfer and UDP for low-latency streaming ensure optimal performance in high-fidelity and real-time testing scenarios.

4.6 Simulation Parameters and Conditions

4.6.1 Sampling Rates

The sampling rates chosen for the various sensors in the simulation framework are designed to match real-world conditions as closely as possible. For GNSS receivers, the simulation framework employs a sampling rate that captures the dynamic behaviour of satellite signals at high precision. The GNSS signals are typically sampled at around 1 Hz to 10 Hz, corresponding to the standard sampling rate used in most real-world GNSS receivers. This allows the simulation to replicate real-time GNSS data acquisition, ensuring that the signal dynamics, such as satellite movement and changes in signal strength due to atmospheric conditions, are accurately captured.

Inertial sensors, such as accelerometers and gyroscopes, require higher sampling rates to represent motion accurately. In this simulation, inertial sensors are typically sampled at frequencies between 100 Hz and 1000 Hz. This high sampling rate is necessary because the sensors measure rapid acceleration and rotational motion changes, which occur much more frequently than GNSS updates. Using a high sampling rate, the simulation framework ensures that the data generated by SimSENSOR closely reflects real-world IMU behaviour.

The frame rate is a crucial parameter for cameras and other visual sensors. It is set to match real-world conditions, typically between 30 frames per second (fps) and 60 fps. This ensures that the visual data, including camera images and depth information generated by Unreal Engine with AirSim, corresponds to the performance of actual cameras used in vision-based navigation systems. Combining these sampling rates creates a cohesive and realistic dataset across all simulated sensors, allowing for comprehensive testing and validation of multi-sensor systems.

4.6.2 Vehicle Dynamics

The simulation framework includes detailed modelling of vehicle dynamics, which is essential for generating realistic sensor data. The vehicle's speed, acceleration, and trajectory are carefully controlled to ensure that the motion profiles simulated within the framework reflect real-world driving or flight conditions. The simulation covers various vehicle dynamics, from low-speed operations in constrained environments to high-speed motion in open spaces. For example, speeds in the simulation can range from a few meters per second for ground vehicles operating in urban environments to over 50 meters per second for aerial vehicles such as drones.

The acceleration profiles used in the simulation include both steady acceleration and deceleration scenarios, as well as sudden changes in speed, such as those encountered during evasive manoeuvres or when reacting to obstacles. These profiles are critical for testing the system's ability to handle rapid changes in motion, which can introduce significant challenges for GNSS and inertial navigation systems. By simulating a wide range of vehicle dynamics, the framework ensures that the system is tested under realistic and diverse conditions, providing valuable insights into its performance across various operational scenarios.

4.6.3 Extreme Conditions

The simulation framework is designed to model a range of extreme conditions that may not be feasible to test in real-world environments due to safety or logistical constraints. One such condition is adverse weather. The simulation incorporates weather scenarios, including rain, fog, and snow, which can significantly impact sensor performance. For instance, rain can cause signal attenuation for GNSS receivers, while fog may obscure visual sensors, leading to degraded vision-based navigation performance.

Lighting conditions are another crucial factor considered in the simulation. The framework models different times of day, including daylight, twilight, and nighttime. Artificial lighting, such as streetlights or vehicle headlights, is also incorporated into nighttime simulations. These variations in lighting are essential for testing vision-based systems, which are highly sensitive to changes in illumination. The presence of dynamic

lighting artifacts, such as glare or shadows, introduces additional challenges for navigation systems, making the simulation more representative of real-world conditions.

The simulation also includes scenarios involving signal degradation. For GNSS systems, signal blockage and multipath effects are simulated, especially in urban environments where buildings and other structures interfere with satellite signals.

Chapter 5 Mitigating INS Errors Using Deep Learning in GNSS-INS Systems

Abstract

Integrating Inertial Navigation Systems (INS) and Global Navigation Satellite Systems (GNSS) has gained significant attention for improving navigation accuracy in transportation, aviation, and military applications. However, GNSS signals are subject to multipath errors, particularly in urban environments, where the "urban canyon" effect degrades the availability and accuracy of GNSS signals. For temperature variation, we know this affects the error model of the INS as it impacts accelerometers and gyroscopes. The literature does not show precisely how this can impact prediction performance in the deep learning-enabled INS. Existing methods primarily focus on the impact of INS errors on Adaptive Kalman Filters (AKF). Still, AKF may not sufficiently model a convolved non-linear relationship between historical errors and updates.

Here, we emulate the error fluctuations by generating three equivalent IMU error models. To address the research gap, this chapter presents a GRU-based error correction method for improving the accuracy and reliability of our GNSS-INS system in urban canyon environments. The data was generated through simulations on the Spirent GSS 7000 simulator and SimSENSOR software. The experimental results show an increase in positioning performance compared to INS by up to 93%. The GRU-based error correction method exhibited a 95^{th} percentile error in the horizontal direction of 18.5m compared to INS 81.9m after a GNSS outage of 100 seconds. Regarding GRU predictive performance when IMU errors fluctuate, no increase in position error is demonstrated.

5.1 Introduction

The integration of INS and GNSS has gained widespread adoption in various fields, including transportation, aviation, and military applications [69]. GNSS, comprising satellite-based systems such as GPS, GLONASS, and Galileo, offers global coverage and high precision for positioning, navigation, and timing (PNT) purposes. However, GNSS signals are subject to various errors, including atmospheric effects, ionospheric delay, and multipath interference, which can impair the accuracy of the system [34].

As mentioned in previous chapters, GNSS multipath can significantly impact the availability and accuracy of GNSS signals, particularly in densely built urban areas where the geometry of the built environment creates deep "canyons" that obstruct the direct line of sight to satellites [70] [71], Therefore, alternative sensors are needed to provide navigation aid in these challenging settings to improve navigation performance for positioning estimates during prolonged GNSS outages.

This chapter explores using a GRU-based architecture to predict and mitigate position errors in INS systems and address the abovementioned challenges. GRU allows for the temporal dependencies to be captured of the GNSS and INS errors and predicts the error correction term that can be used to correct the INS measurements. This chapter also explores the significance of perturbation from INS error sources on the accuracy of the GRU model. The chapter makes several novel contributions. First, we examine the use of GRU architecture to aid in GNSS-denied environments and demonstrates the feasibility and effectiveness of utilising GRUs for an extended period. This will be achieved by comparing the performance of the GRU with other error correction methods, such as decision trees and RNNs, in urban canyons. Secondly, a generalisation analysis will be conducted to inspect the effects of sensor error perturbations by testing the architecture trained on one sensor with another unseen sensor. Here, we imitate the error fluctuations by calculating INS positions based on the same sensor but with slight variations in stochastic and deterministic error profiles, such that we recreate the equivalent effect.

5.2 Related Work

The literature studies error correction techniques for Global Navigation Satellite Systems (GNSS) and Inertial Navigation Systems (INS) to improve their accuracy and reliability. This section reviews the existing error correction methods for GNSS and INS, focusing on the techniques applied to GNSS/INS integration.

RNN-based error prediction paper from the author [72] uses accelerometer and gyroscope information to determine the non-linear relationship between the input and the output. As RNNs can utilise past information, they can provide better positioning accuracy when GNSS is unavailable. However, RNNs suffer from expanding/vanishing gradients, which limit the temporal domain to the short term [73]. Adaptions from this paper exist, including an AKF-GRU-based approach that addresses the issues of RNN by implementing a GRU-based deep learning network [74]. However, in contrast to the paper above, a different type of prediction model is used to calculate the GNSS position variation ΔPGNSS. This is then integrated over time to provide a GNSS estimation used in an AKF. The disadvantage of such a system is that for prolonged GNSS outages, the consistent integration will cause exponential drifts in the predicted GNSS position as pseudorange errors to exist and therefore reduce positioning accuracy. Furthermore, this algorithm has a real-time implementation problem and requires considerable processing power.

The literature on error correction in GNSS/INS systems indicates a spectrum of methods, each with distinct benefits and limitations. Although they struggle with non-linear models, Kalman filtering and its variants (EKF, UKF) have gained traction for their efficiency in fusing multiple sensors. Non-linear optimisation techniques like least squares and MLE offer a solution to non-linearity yet suffer from slow convergence and sensitivity to initialisation. Multi-sensor fusion systems, as explored in recent studies, display promise, but the need for additional sensors and assumptions regarding error distribution pose challenges. Tightly coupled GNSS/INS integrations reveal the potential for enhanced positioning prediction; however, their reliance on raw sensor data and inability to detect slow ramp errors present considerable drawbacks.

Machine learning approaches, such as ML-ANFIS and RBFNN networks, highlight advancements in error estimation, though they confront issues like dimensionality, design

time, and non-linear challenges. RNN based error prediction, as discussed in chapter 3, appears promising, but gradient problems and processing power requirements limit it. Across all methodologies, a recurrent theme is the challenge of non-linearity and the balance between accuracy and computational efficiency. Therefore, this chapter proposes a GRU-based INS error prediction from accelerometer and gyroscope readings. GRUs solve issues regarding exploding/vanishing gradients exhibited by RNNs and require less computational power than LSTMs or GRUs with an adapted kalam filtering-based fusion approach. Furthermore, GRUs can utilise past information and can handle non-linear relationships. Furthermore, to expand on the knowledge of the effects of IMU error variations between different environments or different units of the same type of sensor, it is essential to investigate GRU performance when sensor error perturbations are introduced.

5.3 Proposed System

5.3.1 IMU Measurement Error

Before introducing the proposed system, it is important to understand how IMU sensor errors can be considered a time-domain problem and how it fits architectures such as GRUs. IMU sensors measure the UAV's specific forces and angular velocities along the body frame. These values are integrated into the system to provide the position and velocity data to the navigation system. However, incorporating these values will lead to a divergence between ground truth and INS position and velocity output due to the biases included in the measurements. It is important to derive the position and velocity equations to understand how the errors may propagate through the calculation. The velocity of the vehicle is given by [75]:

$$\boldsymbol{v}_{eb}^{n}(k) \approx \boldsymbol{v}_{eb}^{n}(k-1) + \left(\boldsymbol{C}_{b}^{n}\boldsymbol{f}_{ib}^{b} + \boldsymbol{g}^{n} - 2\boldsymbol{\Omega}_{in}^{n}\boldsymbol{v}_{eb}^{n}(k-1)\right)\tau \tag{5.1}$$

Where $\boldsymbol{v}_{eb}^{n}$ is the UAV velocity with respect to the earth in the local navigation frame, $\boldsymbol{C}_{b}^{n}$ is the direction cosine matrix from the body frame to the local navigation frame, $\boldsymbol{f}_{ib}^{b}$ is the specific force in the body frame, $\boldsymbol{g}^{n}$ is the gravitational vector in the local navigation frame and $\boldsymbol{\Omega}_{in}^{n}$ is the skew-symmetric matrix of the earth's angular velocity.

To obtain the position of the UAV, the equation above needs to be integrated once more. Assuming that the velocity varies linearly during the integration period, the position of the vehicle is given by:

$$\boldsymbol{p}_{eb}^{n}(k) = \boldsymbol{p}_{eb}^{n}(k-1) + \ (\boldsymbol{v}_{eb}^{n}(k-1) + \boldsymbol{v}_{eb}^{n}(k)\frac{\tau}{2} \tag{5.2}$$

Where $\boldsymbol{p}_{eb}^{n}$ is the UAV's position on the earth in the local navigation frame. To understand the influence of the measurement noises experienced, the sum of the true measurement and the biases gives the actual velocity and position. This is shown in the equation below, where $\delta\boldsymbol{p}$ denotes the position measurement error:

$$\delta\boldsymbol{p}(k) = \ \delta\boldsymbol{p}_{eb}^{n}(k-1) + \left(\delta\boldsymbol{v}_{\boldsymbol{eb}}^{\boldsymbol{n}}(k-1) + \delta\boldsymbol{v}_{\boldsymbol{eb}}^{\boldsymbol{n}}(k)\right) \tag{5.3}$$

To solve this equation regarding the calculated velocity error $\delta\boldsymbol{v}_{\boldsymbol{eb}}^{\boldsymbol{n}}$, equation 5.2 is used to subtract from the measured velocity error and then substituted into equation 5.3 to obtain the below equation:

$$\delta\boldsymbol{p}(k) = \ \delta\boldsymbol{p}_{eb}^{n}(k-1) + (\boldsymbol{I_3} - 2\tau\boldsymbol{\Omega}_{in}^{n})\delta\boldsymbol{v}_{\boldsymbol{eb}}^{\boldsymbol{n}}(k-1)\tau + \boldsymbol{\delta\phi} \tag{5.4}$$

Where $\boldsymbol{\delta\phi}$ represents the system vector attitude error. Equation 8 shows the relationship between position error with the vehicle dynamics, previous INS error, and sensor measurement errors. This shows a non-linear relationship between the measurement error and the sensor noise and biases of the system, thus current techniques such as EKFs may not be enough [76] to provide the accuracy UAVs require when operating in an urban environment and experiencing GNSS outages. Additionally, prolonged GNSS outages will affect typical EKF integrations as the system will heavily rely on IMU measurement updates which will inevitably cause drifting in position estimate. Therefore, a system is required that can learn the non-linear relationship identified above.

5.3.2 System Overview

The proposed system architecture is shown in Figure 5-1. During the GRU's training phase, the accelerometer and gyroscope measure specific force and angular velocity. These two sensors make up the system's Inertial Measurement Unit (IMU).

To calculate the system INS position and velocity, the current system orientation is then used to calculate the Direction Cosine Matrix (DCM) required to translate the

accelerometer and gyroscope readings from the body frame to the navigational frame. These are then integrated to provide the position, velocity, and updated orientation information. The output from this is used as the reference INS for error estimation. A more detailed version of the steps used to calculate the current timestep velocity, position, and orientation are shown in Algorithm 1. Navigation frame acceleration, angular rate, current orientation, and time since the last GNSS update are used as the feature input for the GRU at each timestep. The target output used for training is position error derived from the difference between the INS position and GNSS ground truth data.

In neural networks, a loss function is a critical component that quantifies the discrepancy between the predicted outcomes and the actual values. It serves as a fundamental metric that the optimisation algorithm aims to minimise, thereby guiding the adjustment of the model's parameters to enhance prediction accuracy. The loss function provides a mathematical framework for assessing the neural network's performance, enabling the model to learn from its errors and iteratively improve its predictions.

For training, ADAMAX is utilised as the optimiser. ADAMAX is a variant of the ADAM (Adaptive Moment Estimation) optimisation algorithm. The choice of ADAMAX over the standard ADAM is driven by its improved performance with sparse gradients. Unlike ADAM, which can sometimes produce extensive updates when dealing with sparse gradients, ADAMAX incorporates the gradient's infinity norm (max norm), making it more robust and stable in such scenarios. The use of ADAMAX ensures a more reliable and consistent optimisation process, ultimately leading to more accurate predictions of position errors.

Algorithm 1 INS Position, Velocity & Orientation Function

function: $\mathbf{Accelerometer_t}, \mathbf{Gyroscope_t}, \mathbf{Orientation_{t-1}}, \mathbf{Velocity_{t-1}}, \mathbf{Position_{t-1}}$
  $\delta t \leftarrow$ *compute timestep*
  $\boldsymbol{g} \leftarrow$ *compute gravity* $(\mathbf{Position_{t-1}})$
  $\boldsymbol{C_b^n} \leftarrow$ *compute DCM matrix* $(\mathbf{Orientation_{t-1}})$
  $a, b \leftarrow Earth's\ radii$
  $\boldsymbol{\Omega_{en}^n} \leftarrow Earth's\ rotation\ rate$
  $R, M, N \leftarrow WGS\ 84\ Model$
  $\boldsymbol{Rotation\ Matrix}\ (\boldsymbol{RM}) \leftarrow$
  compute rotation matrix $(R, M, N, \boldsymbol{\Omega_{en}^n}, a, b, \boldsymbol{C_b^n}, \boldsymbol{g}, \mathbf{Gyroscope_t})$
  $\delta \boldsymbol{V}, \delta \boldsymbol{P}, \delta \boldsymbol{\phi} \leftarrow$ *compute change* $(\mathbf{Velocity_{t-1}}, R, M, N, \mathbf{Accelerometer_t}, \boldsymbol{RM})$
  $\mathbf{Velocity_t}, \mathbf{Position_t} \leftarrow$
  compute Pos, Vel at t $(\delta\ \mathbf{Velocity_{t-1}}, \mathbf{Position_{t-1}} \boldsymbol{V}, \delta \boldsymbol{P}, \delta t)$
  $\mathbf{Orientation_t} \leftarrow$ *compute Orientation at t* $(\boldsymbol{RM}, \mathbf{Orientation_{t-1}})$
  return: $\mathbf{Velocity_t}, \mathbf{Position_t}, \mathbf{Orientation_t}$
end function

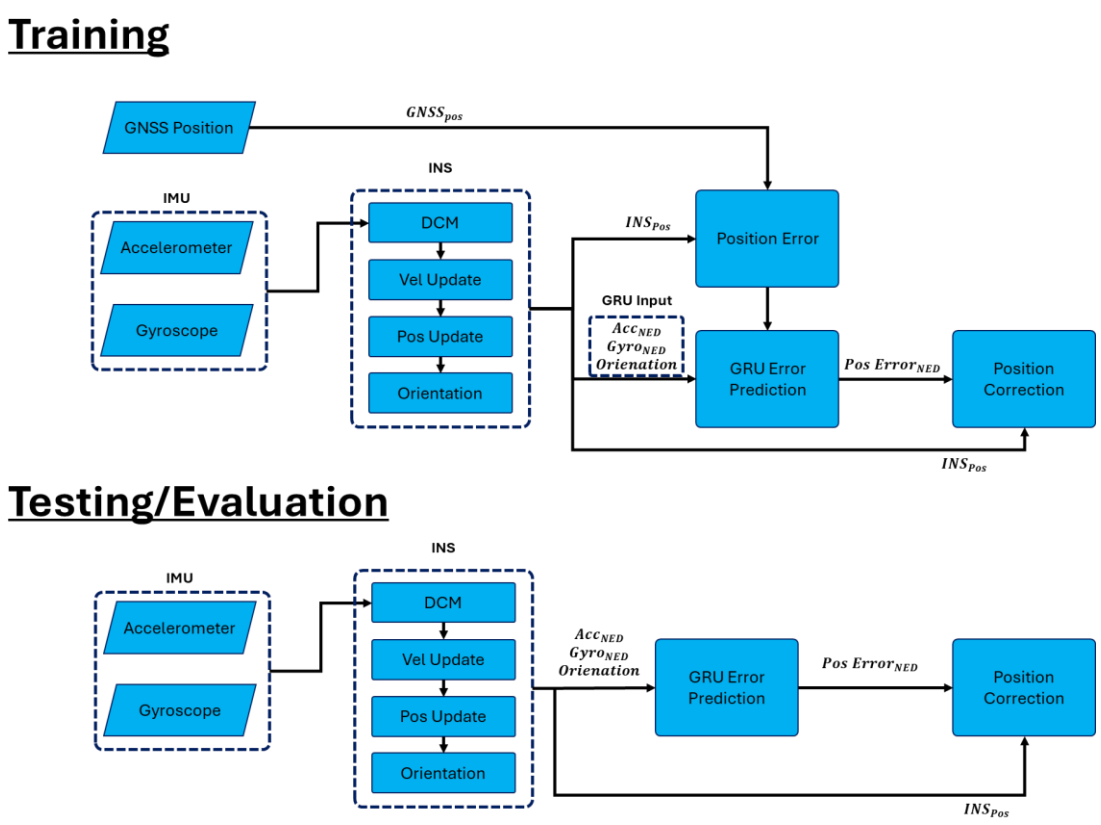


Figure 5-1 High Level Diagram of Proposed GRU System Architecture

The second mode is the testing/prediction phase of the neural network. The prediction mode is used when GNSS information is not available. The trained GRU predicts the positioning and velocity error for the given Navigation frame acceleration, angular rate, current orientation, and time since the last GNSS update. The predicted errors are then subtracted from the INS velocity and position readings to provide the system prediction. This system is designed to provide positioning errors until GNSS positioning is available again. However, the mechanism to determine GNSS positioning suitability is beyond the scope of this chapter. For evaluation purposes, the updated position prediction will be subtracted from the ground truth information available in the simulation to analyse system performance.

The choice of using Gated Recurrent Unit (GRU) architecture in this chapter was predominantly guided by its established ability to capture long-term dependencies in sequence data, a critical requirement when dealing with temporal data such as those in INS. Furthermore, GRU's comparatively simpler architecture and fewer training parameters than other recurrent networks, such as LSTM, also aided our choice.

5.4 Experimental Setup

5.4.1 Data capturing

To produce IMU data required for training, SimSENSOR is used to generate accelerometer and gyroscope measurements that the vehicle is exhibiting. The simulation uses advanced sensor modelling to replicate sensor noises and biases. Furthermore, a vehicle profile is set up to imitate drones' acceleration/deceleration curves and impulses. IMU stochastic and deterministic errors were tuned to mimic the characteristics of an Advanced Navigation Orientus IMU sensor, the specifications of which are listed in Table 5-I, Table 5-II, and Table 5-III.

Table 5-I Sensor A Error and Bias Settings for Gyroscope and Accelerometer

IMU Sensor A			
Accelerometer		**Gyroscope**	
Scaling factor (ppm)	600	**Scaling factor (ppm)**	500
Bias (ug)	20	**Bias (deg/h)**	3
ARW (ug/sqrt(Hz))	100	**GRW (deg/s/sqrt(h))**	0.004
Update rate (Hz)	100		

Table 5-II Sensor B Error and Bias Settings for Gyroscope and Accelerometer

IMU Sensor B			
Accelerometer		**Gyroscope**	
Scaling factor (ppm)	600	**Scaling factor (ppm)**	500
Bias (ug)	18	**Bias (deg/h)**	2.7
ARW (ug/sqrt(Hz))	90	**GRW (deg/s/sqrt(h))**	0.0036
Update rate (Hz)	100		

Table 5-III Sensor C Error and Bias Settings for Gyroscope and Accelerometer

IMU Sensor C			
Accelerometer		**Gyroscope**	
Scaling factor (ppm)	600	**Scaling factor (ppm)**	500
Bias (ug)	22	**Bias (deg/h)**	3.3
ARW (ug/sqrt(Hz))	110	**GRW (deg/s/sqrt(h))**	0.0044
Update rate (Hz)	100		

Variations in IMU errors for Sensor B and Sensor C are based on study [77], where five accelerometers were tested to understand their effects of temperature on the bias from the reference temperature of 20°C to 80 °C. It found that bias can increase and decrease depending on the model and manufacturing novelties in each system. In some worst-case scenarios, this can change by up to 8% from its 20°C reference. Therefore, a 10% change is used to encapsulate.

To further assess the impact of variability in the measurement errors, two other error models are used with similar error characteristics to generate IMU data to evaluate the system performance in handling errors the system is not trained for. These sensors are implemented in the simulation with a UAV characteristic profile that runs on the same trajectory with the same GNSS outage time to remove as many independent testing variables as possible and focus on the comparative performance evaluation. Running these different sensors on the same UAV with the same trajectory allows for better comparison and analysis of the performance regarding the ML prediction capabilities when IMU errors fluctuate.

Regarding the scenario setup, to generate realistic input and output data for the neural network, the simulated route (depicted in Figure 5-2) replicates various manoeuvres commonly executed by drones. The simulated route is designed to be balanced so that any potential bias introduced by the route itself is cancelled out, thus allowing us to properly study the bias introduced by the GNSS and inertial systems being monitored in this study. These include heading changes at different g load factors and straight segments with acceleration and deceleration. These manoeuvres ensure diverse changes in accelerometer and gyroscope readings, allowing both sensor errors to be modelled. Furthermore, consecutive segments allow for the isolation of random walk errors whilst heading changes represent highly non-linear situations that error correction techniques would have to face. The scenario date and time were set at 00:00:00 on September 05, 2022, with only GPS constellation available. To simulate typical GNSS interruptions, the GNSS signal is interrupted for up to 100 seconds to simulate periodic disruption. During GNSS dropout times, INS information is used to provide position information to the navigation system. The duration of 100 seconds is chosen to allow for comparison with state-of-the-art solutions and is typical in urban canyons. In contrast, literature such as [78] only compares with 30-second outages.

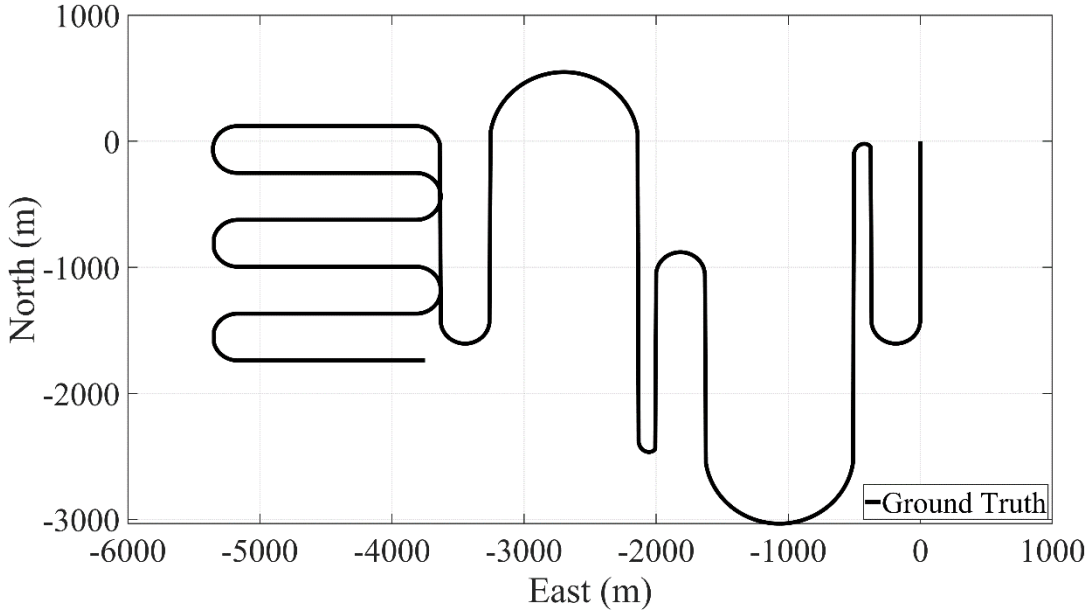


Figure 5-2 Ground Truth UAV trajectory in simulated scenarios

Data collected from the sensors and simulation systems was stored in CSV file format, with the time and date of each timestep recorded. Subsequently, the data was processed in MATLAB to generate the ground truth data that serves as the training target. The input data consisted of Navigation frame acceleration, angular rate, current orientation, and time since the last GNSS update.

To pre-process data for the neural network, a batch generator generates training batches with a batch size of 128. The data is standardised using a z-score for feature inputs and target outputs. Z-score is defined as:

$$\boldsymbol{z} = \frac{x - \mu}{\sqrt{\left(\frac{\Sigma_{i=1}^{n}(x_i - \mu)^2}{n-1}\right)}} \tag{5.5}$$

where μ is the mean value of the array, and x_i is the value in the array sequence.

5.4.2 Architecture Optimization

The machine learning algorithm is optimised by implementing random searches to identify the optimal number of hidden units, the number of GRU layers, and the optimal activation function. This process, which utilises only a limited portion of the total data, prioritised minimising computational cost and maximising prediction accuracy. The final

layers, consist of an input layer with the inputs of an accelerometer, angular rate, and orientation as the features in the columns of the matrix, as well as timesteps with the recorded sensor data in the rows. Two hidden layers, comprising 256 and 128 units, respectively, and an unrolled length of 5 represent the sequence length input in each input from the batch, with a dropout rate of 15%. A tanh activation function provides a non-linear relationship between the input and output. A dense layer connects all the nodes and provides the regressed output of the neural network. For optimising the weights and biases, ADAMAX is used with an initial learning rate learning rate of 0.001. Mean squared error loss is used as the algorithm's loss function. The trained weights and biases are then saved in a .h5 file format to evaluate the test data.

The test data is then standardised with the previous parameters used in training and then fed through the machine learning algorithm, where the output is recorded and then compared against the actual position error derived from the difference between ground truth and INS position prediction. This data is then evaluated to find the prediction error's mean, standard deviation and 95th percentile error. This is then compared to the actual position error. Furthermore, error distribution histograms, box plots, plots, and mistakes over time are used to evaluate and compare the architecture to analyse the system performance better.

Regarding the investigation into the effects of perturbation errors in IMU sensors, the two sensors will be run through the simulation described above. Their accelerometer, angular rate, and orientation information will then be used as the inputs for the neural network to calculate INS position error. The data will then be evaluated similarly to how this was carried out for Sensor A regarding the prediction performance of the GRU. Furthermore, comparison the INS position error of Sensor A-B and A-C will be compared against the residual error from GRU prediction of Sensor A-B and A-C to understand how the change in sensor error characteristics affects the prediction performance.

5.5 Results & Analysis

5.5.1 GRU Performance Analysis

Table 5-IV shows a comparison of the Mean, Standard Deviation (SD) and 95th percentile error in the North, East and Vertical directions for Sensor A INS, GRU training and GRU testing data. Cells shown in green indicate the best performance in class comparison. Comparing INS and GRU, it can be observed that the GRU performs better than INS only throughout each metric. This indicates that the GRU can predict INS errors caused by the iterative acceleration and angular rate data integration. This is expected as the GRU has access to past errors indicated by the system and can utilise this to understand and mitigate these errors. Furthermore, we can also observe that the metrics recorded from the 20% Testing data are similar in magnitude to the 80% Training data. This shows that the trained architecture has not overfitted to training data and can, therefore, generalise the situation with great accuracy.

Table 5-IV Comparison of Mean (σ), Standard Deviation (μ) and 95th Percentile Position Error for INS Sensor A, GRU Training and Testing in North (N), East (E) and Down (D)

	σ error(m)			μ(m)			95th Error (m)		
	N	*E*	*D*	*N*	*E*	*D*	*N*	*E*	*D*
INS Sensor A	13.6	14.2	0.77	16.7	18.2	1.12	63.6	68.8	4.14
Train GRU A	1.46	1.70	0.14	4.09	2.77	0.23	13.7	10	0.82
Test GRU A	0.92	2.29	0.15	1.28	4.59	0.23	4.76	16.1	0.83

Table 5-V shows a comparison of the Improvements in the North, East and Vertical performance with the reference point being the INS Sensor A data. Analysing the data shows that utilising GRU error correction outputs in training and testing increases positioning performance compared to INS-only position predictions by up to 93%. Furthermore, the table also provides information on the Horizontal Mean, Standard Deviation (SD) and 95th percentile error with the respective improvement in performance. Horizontal improvements of up to 82% are observed with the Training data and 80% for the testing data, showing a consistency in performance that indicates no overfitting.

Table 5-V Comparison of Position Error Improvement in the North, East, Down and Horizontal (H) Direction with Sensor A, GRU Training and Testing

	Improvement (%)				Position Error (m)		
	N	*E*	*D*	*H*	σ	μ	95^{th}
INS Sensor A	N/A	N/A	N/A	N/A	20.8	23.7	91.9
Train GRU Sensor A	78	85	80	82	2.47	4.82	17
Test GRU Sensor A	93	77	80	80	2.57	4.71	18.5

In Figure 5-3, the Inertial Navigation System (INS) and the proposed Gated Recurrent Unit (GRU) system are compared in terms of horizontal 95th percentile position errors, considering the last time since the Global Navigation Satellite System (GNSS) update. The results demonstrate that the GRU architecture regularly reduces position errors compared to the INS across all time intervals. This implies that the GRU system remains reliable even during extended GNSS outages.

The GRU architecture can operate longer without GNSS positioning than INS for various position error threshold values (1 m, 2 m, 5 m, or 10 m). Notably, at the 10m threshold, the GRU provides an additional 42 seconds compared to the INS. Mainly, when the GNSS outage lasts for 100 seconds with an averaged variety of manoeuvres performed by the drone, the horizontal error for the INS amounts to 129.16 m, while the GRU achieves a lower error of 25.73 m. This highlights a reduction of 103.43 m in error over a 100-second GNSS outage period when utilising the GRU system.

Furthermore, a notable observation from the log graph is the linear relationship that emerges after approximately 25 seconds between horizontal position error and time. This indicates an exponential growth pattern where the position error increases exponentially as time progresses. This exponential relationship can be attributed to the random walk effect exhibited by INS systems due to the inherent sensor biases and errors discussed in the introduction.

Certain intervals exhibit fluctuating gradients when the GRU performance is examined more closely. This observation suggests potential underrepresentation of scenarios involving turning radiuses with varying g-loads, while overrepresentation of straight segments where the system performs better.

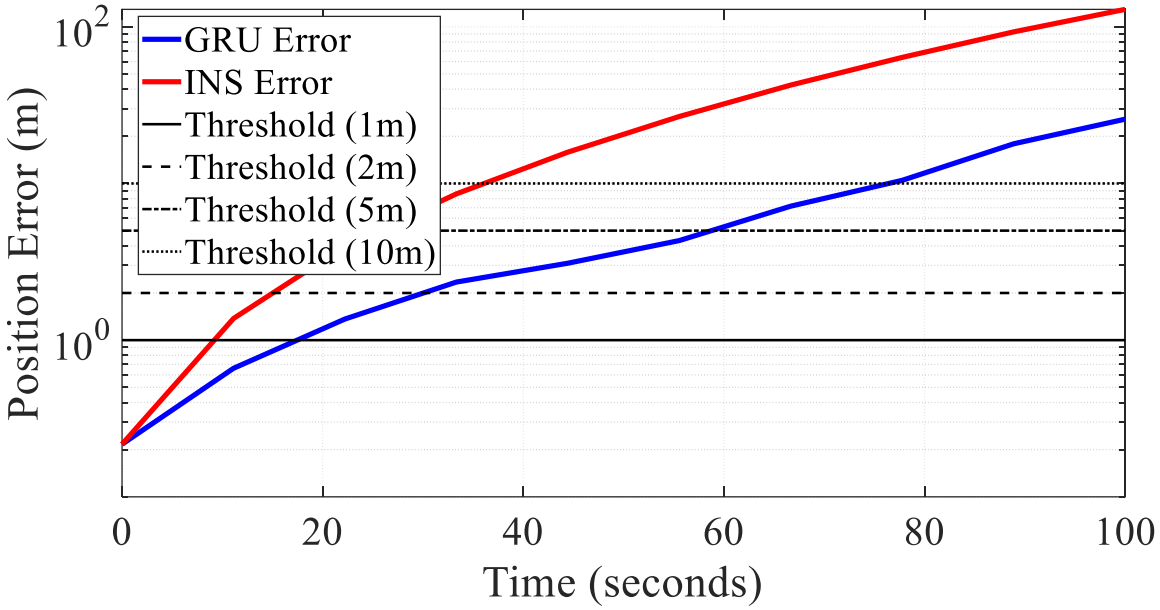


Figure 5-3 Absolute Horizontal Position Error at Each Time Interval with No GNSS Update comparison with Sensor A INS and Sensor A GRU

When comparing absolute position errors with time since the last GNSS update for straight and turning segments shown in Figure 5-4a and Figure 5-4b, respectively, straight segments exhibit fewer position errors for the same time duration as heading changes. The data collected here is the average of all radii turns and straight segments combined, as each is unique due to its direction and g-load profile. Therefore, due to the highly nonlinear data from the accelerometer and gyroscope when heading changes occur with conduction to the random walk effects, a more challenging circumstance is exhibited for the GRU to predict position and velocity errors. This is further backed up by the magnitude of INS errors when comparing straight segments (39.89m) with heading changes (98.22m). Compared to INS, there is a clear advantage to utilising GRU for longer durations of GNSS outages. Nonetheless, some outstanding explanations remain regarding spikes at 0, 10 and 55 seconds in Fig. 8a and the period after 80 seconds in Fig. 8b.

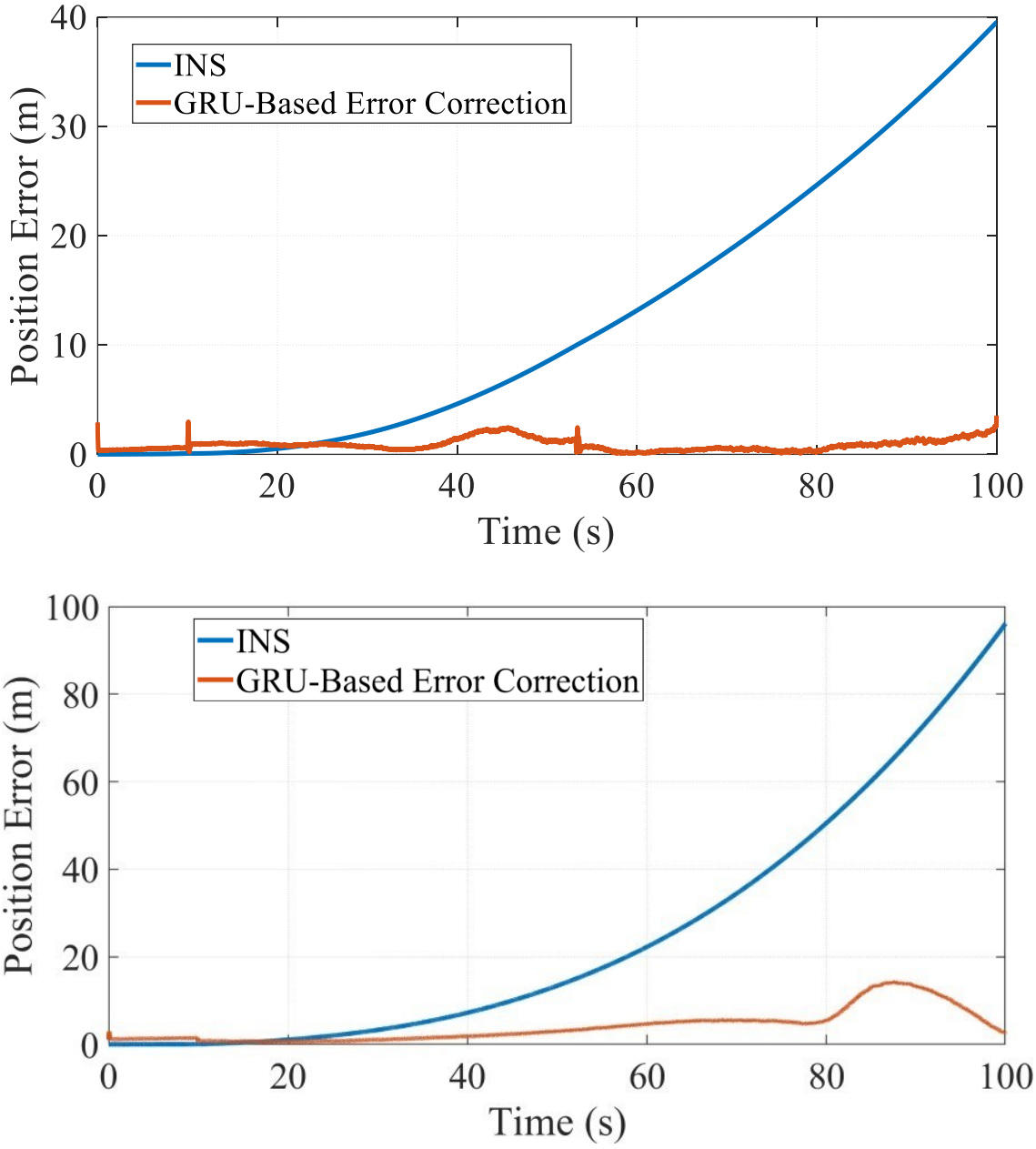


Figure 5-4 Comparison of Horizontal Position Error Prediction Performance in terms of Straight (a) and Heading Change (b) Trajectories for Sensor A INS and Sensor A GRU at 95th percentile.

Figure 5-5 presents a comparison between the Inertial Navigation System (INS), Decision Tree, Recurrent Neural Network (RNN), and the proposed Gated Recurrent Unit (GRU) network. Notably, the GRU architecture demonstrates superior performance when compared to existing solutions. One solution compared in particular is a reference to the publication [72] which discusses the use of RNNs to predict INS errors. Therefore, the architecture was replicated using one RNN layer (10 hidden nodes) and a tanh activation

function. The limitations of INS, as discussed earlier, arise from the constant integration at each time epoch, resulting in a random walk effect that exponentially increases position error. In contrast, as a learning algorithm, the Decision Tree can reduce position errors in the horizontal direction. However, both the RNN and GRU architectures outperform the Decision Tree, thanks to their capacity to leverage past error dependencies, thereby leading to a reduction in position error.

Moreover, Decision Trees are susceptible to noisy data because they tend to overfit by creating complex structures that fit the noise in the training data, are unstable with small changes leading to different trees, may choose splits that reflect noise rather than underlying patterns, and have high variance, likely changing significantly with different, especially noisy, training sets. Consequently, their performance becomes unstable when sensor errors and biases fluctuate, making them less suitable for environments affected by external factors such as weather conditions. On the other hand, when comparing RNN and GRU, a further reduction in error is observed. This improvement is attributed to the implementation of gates in GRU architectures, which enhances the longevity of memory properties associated with RNNs. Consequently, the proposed GRU architecture exhibits superior prediction performance and significantly reduces position errors. Furthermore, RNNs, with their issues of exploding/vanishing gradients, will have a more challenging time predicting position error in trajectories that predominantly use heading changes carried out over longer distances.

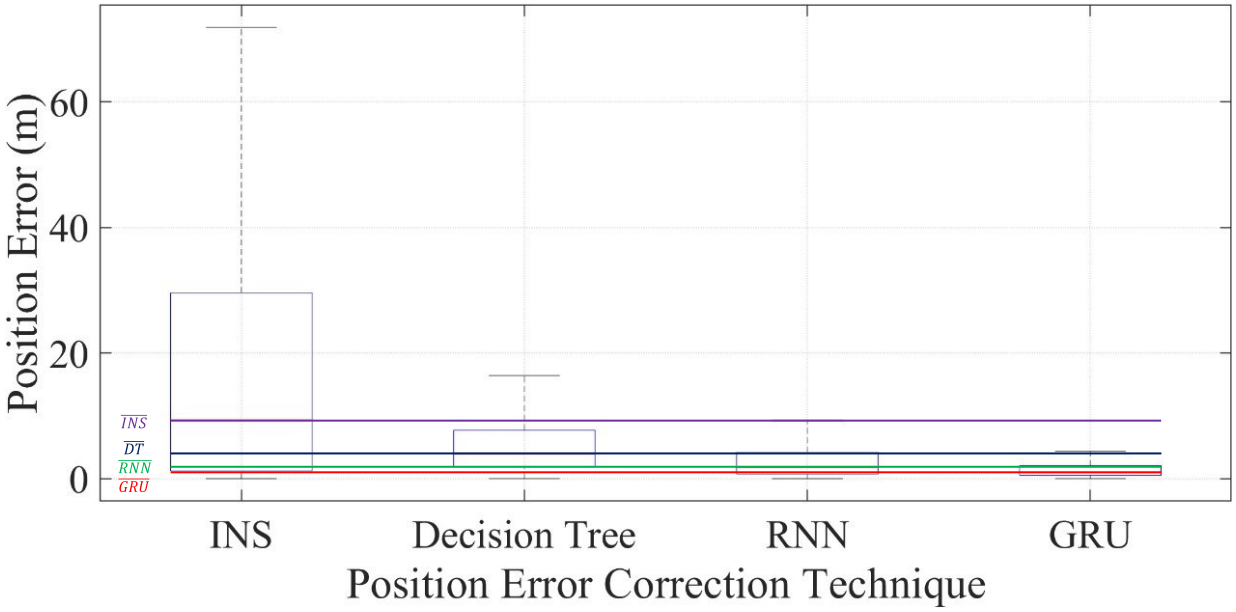
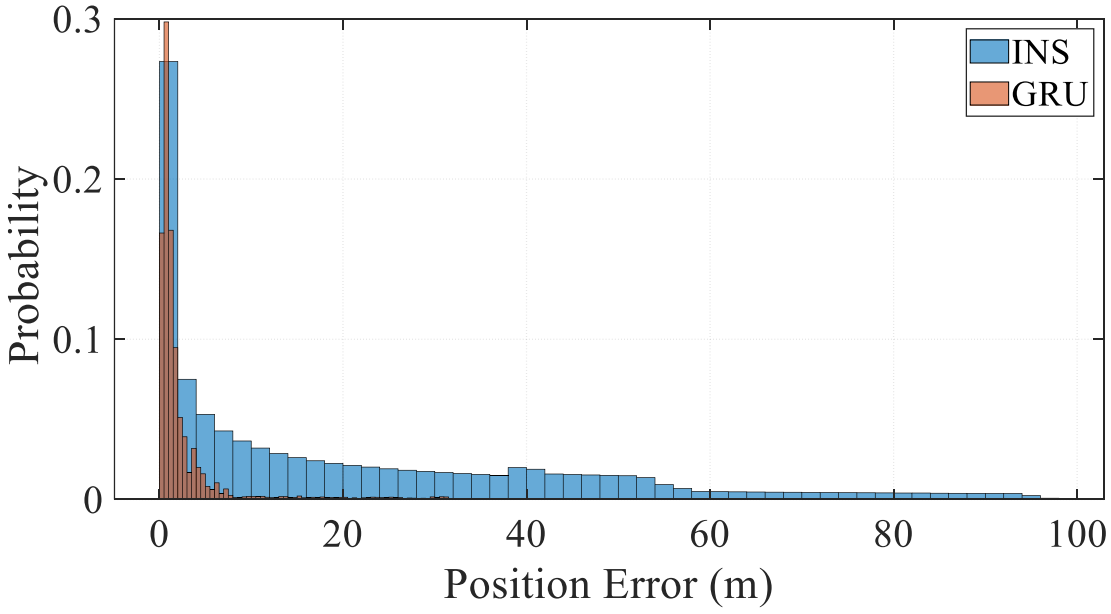


Figure 5-5 Horizontal Position Error Box Plot Comparison between Sensor A INS, Sensor A Decision Tree, Sensor A RNN and Sensor A GRU

Additionally, Figure 5-6a and Figure 5-6b illustrate the distribution of absolute horizontal position errors for the INS and the proposed GRU. In both training and testing scenarios, the GRU model exhibits a notable reduction in position errors, signifying its ability to rectify errors introduced by the INS's reliance on continuous sensor data integration for obtaining velocity and position data. Furthermore, the absence of overfitting to specific trajectories or segments of the trajectory highlights the generalisation capacity of the proposed GRU architecture.



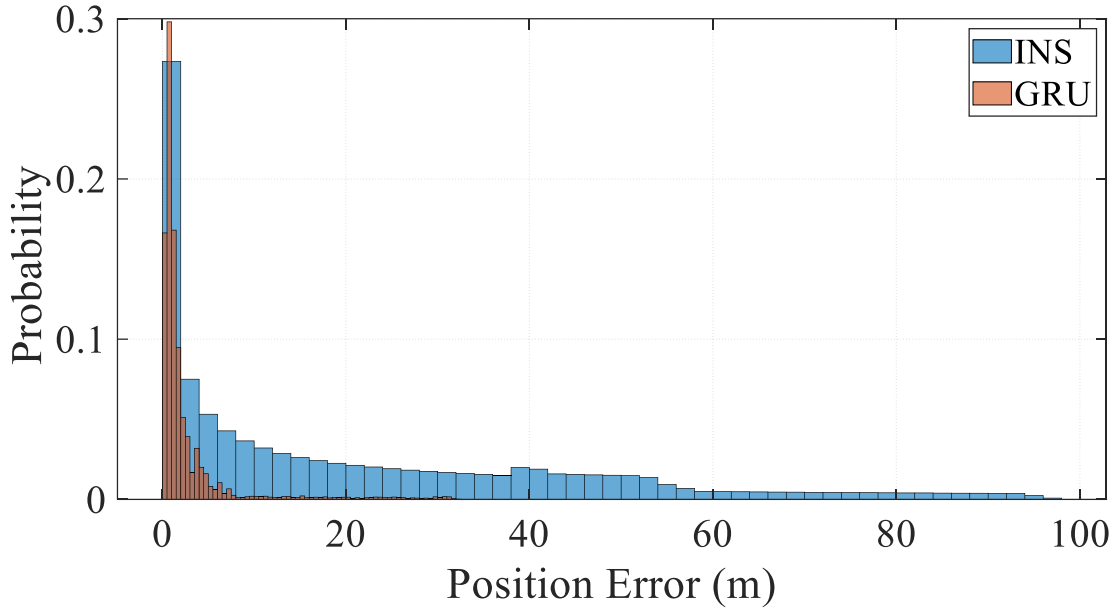
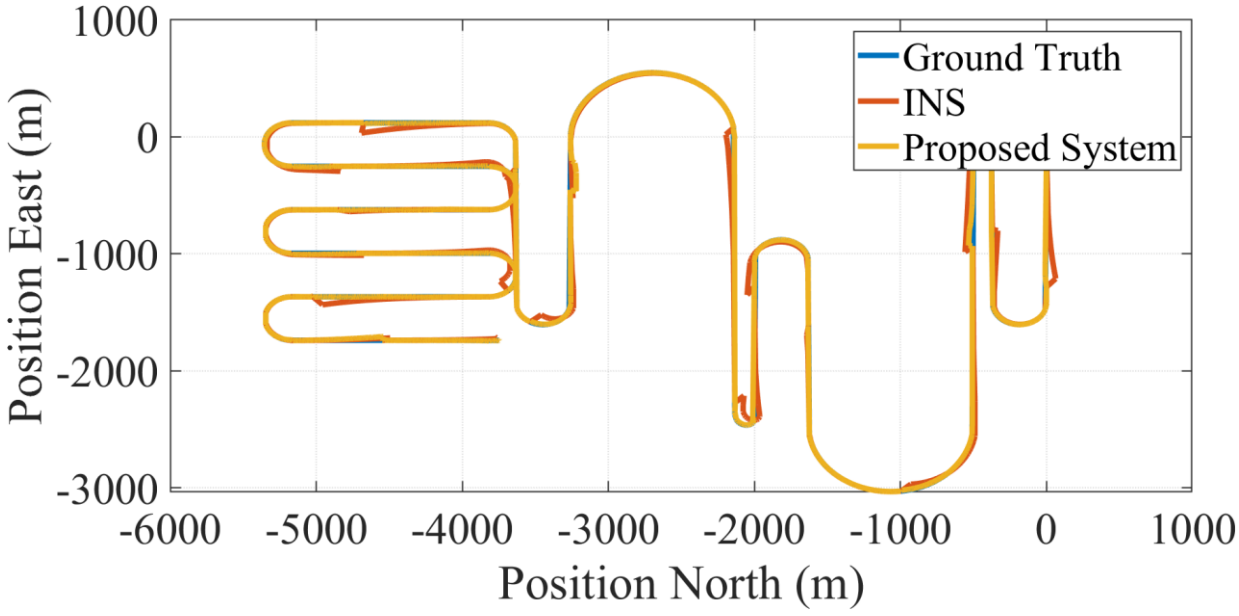


Figure 5-6 Position Error Distribution (a) (b) for Training and Testing Data with Sensor A INS and Sensor A GRU

Figure 4-11 compares the position performance of both the Inertial Navigation System (INS) and the proposed Gated Recurrent Unit (GRU) system. The proposed position estimate in the GRU system is obtained by subtracting the predicted position error from the INS output. The results demonstrate that the proposed system exhibits improved ground truth tracking compared to the INS.



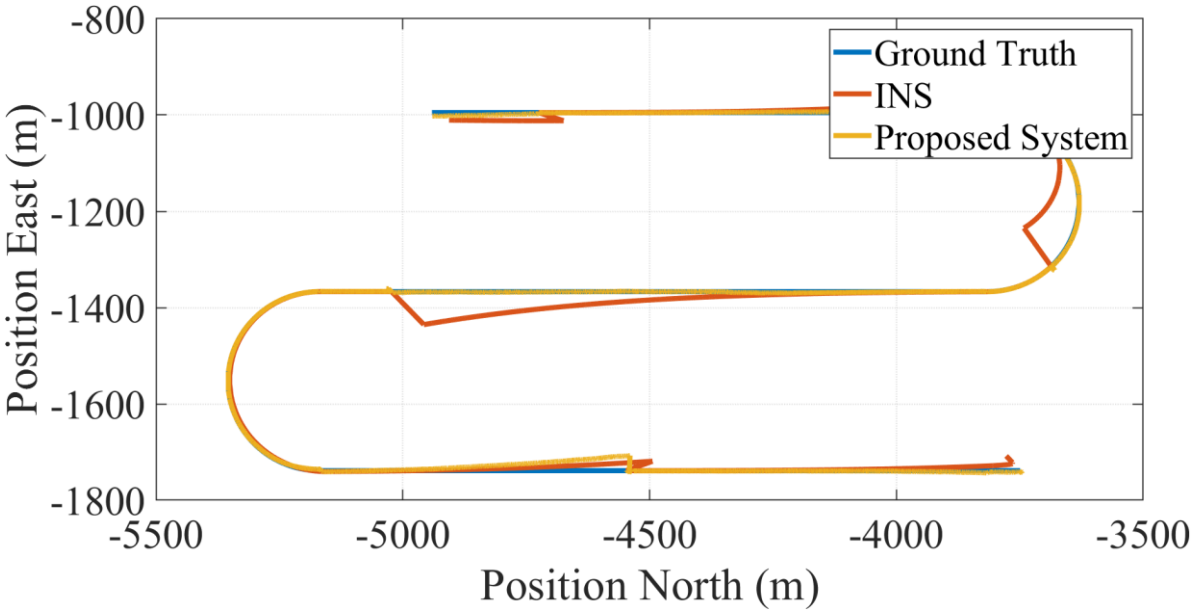


Figure 5-7 Trajectory Comparison between Ground Truth, Sensor A INS and Sensor A GRU for Training (a) and Testing Data (b)

5.5.2 Training and Inference Speed

As mentioned previously, training the GRU aims to provide a data-based model for predicting INS errors. Therefore, the network weights and biases must be trained to improve its prediction performance. The training was set up to iterate until no further convergence occurs on the testing dataset. The training time and inference speed is compared against the network mentioned in publication [72]. The training speeds include the time until no further convergence was observed for 100 epochs for the testing dataset, and the inference speed is based on the output of one iteration ($Pred_t$).

	Training Speed (min/sec)	Inference Speed (ms)
GRU (proposed)	14 min 32 seconds	3.01
RNN [72]	11 min 21 seconds	2.32

The RNN's training speed was observed to be faster than the GRU's, and its inference speed was faster than the sensors' update rates for both GRU and RNN. This indicates that despite the GRU's improved architecture, the inference speed penalty does not significantly implement its real-time use for error prediction.

5.5.3 Perturbation Analysis

Table 5-VI This paper compares various error metrics in the North, East, and Vertical directions for different systems: Sensor B INS, GRU trained on Sensor A and used for predicting Sensor B INS errors, Sensor C INS, and GRU trained on Sensor A and used for predicting Sensor C INS errors. The columns highlighted in green indicate the best performance within their respective comparisons.

When comparing the performance of INS and GRU, it is evident that GRU consistently outperforms INS across all evaluated metrics, as shown in the analysis above, with an error reduction of up to 93%. This finding suggests that the GRU model can accurately predict the errors associated with INS caused by the iterative integration of acceleration and angular rate data. This can be attributed to the GRU's ability to leverage past error predictions within the system, enabling it to understand and mitigate these errors effectively and can be visually seen in 11 when the INS position deviated from the ground truth.

Furthermore, it is worth noting that even though the GRU model is not specifically trained on Sensor B and Sensor C errors, it still demonstrates a reduction in position errors comparable to the performance achieved during training. To better understand the effects of variations in sensor errors, we examine the differences in error magnitudes between Sensor A and Sensor B and Sensor A and Sensor C and assess how the algorithm handles and accounts for these error disparities.

Table 5-VI Comparison of Position Error in the North, East, Vertical and Horizontal Direction with INS Sensor B, INS Sensor C, GRU Testing Data Sensor B and GRU Testing Data Sensor C

	σ error(m)			**μ(m)**			**95th Error (m)**		
	N	*E*	*D*	*N*	*E*	*D*	*N*	*E*	*D*
INS Sensor B	13.3	13.9	0.75	16.4	17.8	1.10	62.4	67.4	4.06
Test GRU B	1.44	1.58	0.14	4	2.70	0.22	13.4	9.67	0.80
INS Sensor C	13.8	14.4	0.78	17	18.6	1.15	64.9	70.1	4.22
Test GRU C	1.65	1.89	0.16	5.18	2.94	0.25	17.2	10.7	0.91

Table 5-VII compares the performance improvements in the North, East, and Vertical directions, with the INS data from Sensors B and C as the reference point. The data analysis reveals a performance gain for both Sensor B and Sensor C when using the GRU to predict INS errors, surpassing the performance of INS by up to 86%.

Additionally, the table provides insights into the Mean Horizontal Error, Standard Deviation (SD), and 95th percentile error, along with the corresponding performance improvements. Notably, the testing data for Sensor B and C exhibits improvements of up to 82%, similar to the testing data from earlier results which demonstrated an improvement of 80%. This consistency in performance suggests no overfitting in the model, indicating its ability to generalize well beyond the training dataset.

Table 5-VII Comparison of Mean, Standard Deviation and 95th Percentile of Position Error for INS Sensor B, INS Sensor C, GRU Testing Data Sensor B and GRU Testing Data Sensor C

	Improvement (%)				**Position Error (m)**		
	N	*E*	*D*	*H*	σ	μ	95^{th}
INS Sensor B	N/A	N/A	N/A	N/A	20.4	23.2	90
Test GRU B	78	86	80	82	2.37	4.71	16.5
INS Sensor C	N/A	N/A	N/A	N/A	21.2	24.1	93.6
Test GRU C	74	85	78	78	2.84	5.81	20.2

To investigate how the GRU architecture handles fluctuations in noise and bias, a comparative analysis is conducted, finding the error differences between INS Sensor A and INS Sensor B as well as INS Sensor A and INS Sensor C. By calculating the difference between these error distributions, the specific impact of sensor bias and noise on position performance can be isolated. Similarly, the error distribution differences of GRU A and GRU B and GRU A and GRU C, which are employed to mitigate sensor errors, are compared. This comparison enables an assessment of whether the GRU effectively mitigates the errors or ignores them.

Mathematically, the comparison of error distributions can be represented as follows:

Let E_A^{INS} and E_B^{INS} represent the horizontal errors of INS Sensor A and INS Sensor B, respectively. Let E_A^{GRU} and E_B^{GRU} denote the horizontal error of GRU A and GRU B, respectively. The difference between the errors can be calculated as:

$$\delta^{INS} = E_B^{INS} - E_A^{INS} \tag{5.6}$$

$$\delta^{GRU} = E_B^{GRU} - E_A^{GRU} \tag{5.7}$$

Then, a distribution is formed from the calculation above. By analysing the similarity or dissimilarity between δ^{INS} and δ^{GRU}, insights can be gained regarding the GRU's handling of sensor errors. If δ^{GRU} closely matches δ^{INS}, it indicates that the GRU architecture may ignore the errors rather than effectively mitigate them. Conversely, if δ^{GRU} shows a reduced error compared to δ^{INS}, it suggests successful mitigation of errors. An increase in errors in δGRU implies potential overfitting of the GRU to INS Sensor A data.

The comparative analysis and mathematical representation outlined above provide a systematic framework for evaluating the GRU's performance in mitigating sensor bias and noise. By investigating the differences between error distributions, valuable insights can be gained into the GRU's robustness and generalisation capabilities in handling various sensor setups.

Figure 5-8 presents the error distribution analyses as previously described. In Figure 5-8a, the differences in error distributions between INS A and INS B and between GRU A and GRU B are examined. Likewise, Figure 5-8b focuses on the distinctions in error distributions between INS A and INS C and between GRU A and GRU C. Notably, in both cases, the observed error distributions were inconsequential. This suggests that the architecture neglects the specific impact of sensor bias and noise during fluctuations. It is worth noting that, for the relatively modest fluctuations investigated in this study, these factors did not significantly influence prediction accuracy.

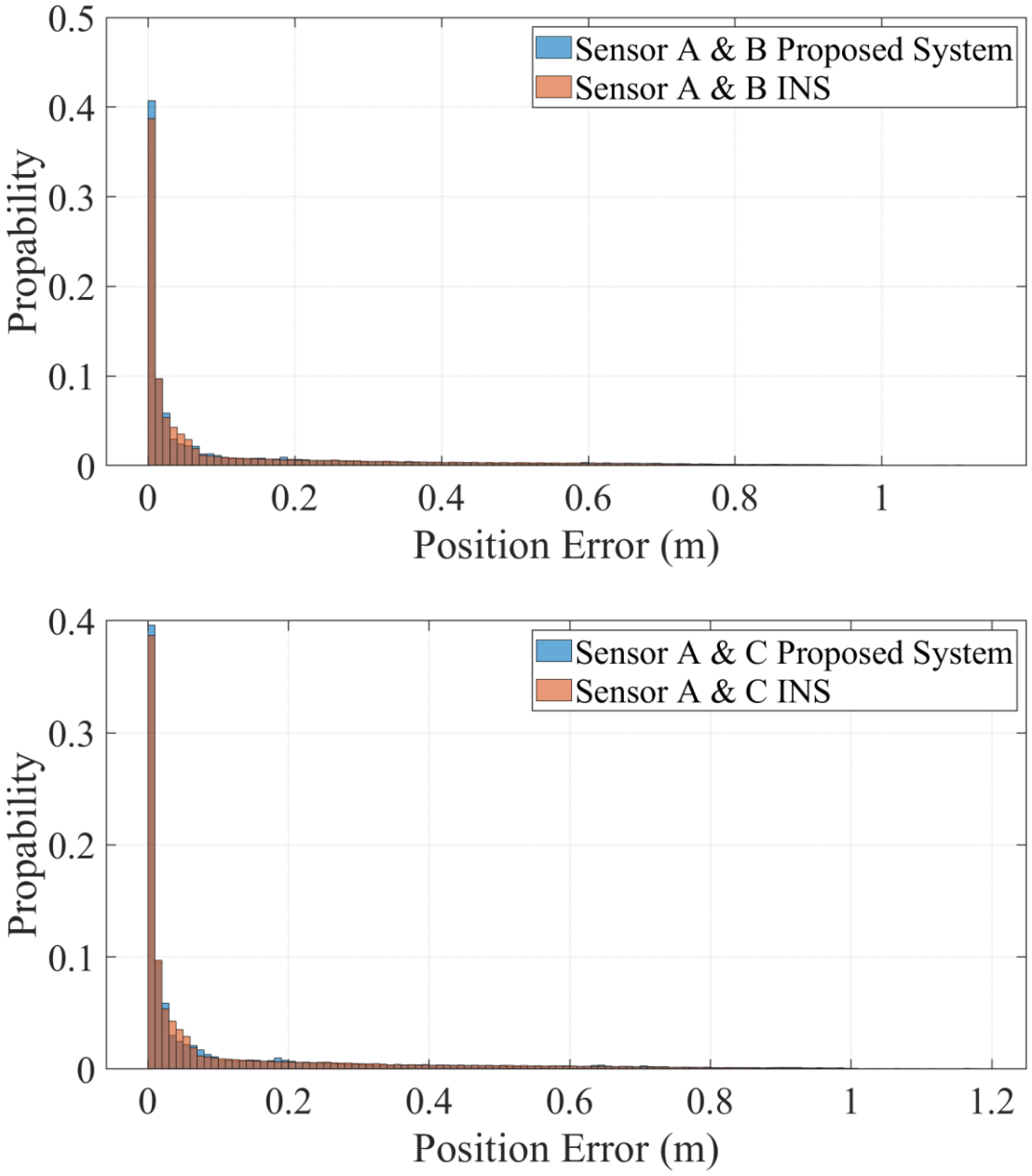


Figure 5-8 Comparative Analysis of Sensor Perturbation Error Performance for Sensor B GRU (a) and Sensor C GRU (b) with the Respective Difference in INS error

To further investigate the similarities in the difference in distribution between the 3 different Sensors, a Kullback–Leibler (KL) divergence analysis is carried out. Here, we will compare the differences between the error distribution of INS and GRU for the respective sensors. Then, we can compare the 3 KL values globally to understand how close or far each KL divergence is from each sensor. If error distributions of INS and GRU on each sensor $i = A, B, C$ are denoted by $P_i(x)$ and $Q_i(x)$ respectively, then mathematically, this is represented as:

$$D(P_i||Q_i) = \Sigma P_i(x) log\left(\frac{P_i(x)}{Q_i(x)}\right) \quad (5.8)$$

where $D(P_i||Q_i)$ is the KL divergence of $P_i(x)$ and $Q_i(x)$, x represents the events in X, the set of all possible events and the sum extends over all x in X. Therefore, the KL for each sensor between the two models can be calculated. The results of the KL study are shown in Table 5-VIII, where we see a KL for Sensor A of 0.5328, 0.5583 for Sensor B, and 0.5263 for Sensor C. This outcome corroborates the analysis earlier and shows that the GRU neglects these fluctuations.

Table 5-VIII Comparison of KL divergence between INS and GRU for Sensor A, Sensor B and Sensor C

	KL divergence
Sensor A	0.533
Sensor B	0.558
Sensor C	0.526

Figure 5-9 illustrates a direct comparison between the Inertial Navigation System (INS) and the proposed Gated Recurrent Unit (GRU) architecture, explicitly focusing on the horizontal 95th percentile position errors. The analysis takes into account the time elapsed since the most recent Global Navigation Satellite System (GNSS) update, with Figure 5-9a corresponding to Sensor B and Figure 5-9b corresponding to Sensor C.

The results reveal consistent improvements in position accuracy achieved by the GRU architecture compared to the INS across all examined time intervals. Notably, the GRU system demonstrates enhanced performance despite being trained exclusively on Sensor A data. This finding suggests the GRU's robustness and reliability in mitigating position errors during extended GNSS outages, particularly in cases where minor disturbances may affect sensors.

However, it is essential to note a slight reduction in the duration for which the GRU system can be relied upon, specifically observed in the case of Sensor C. For instance, at a 10-meter threshold, the duration decreases from 42 seconds to 38 seconds, indicating a slightly diminished performance under more challenging conditions.

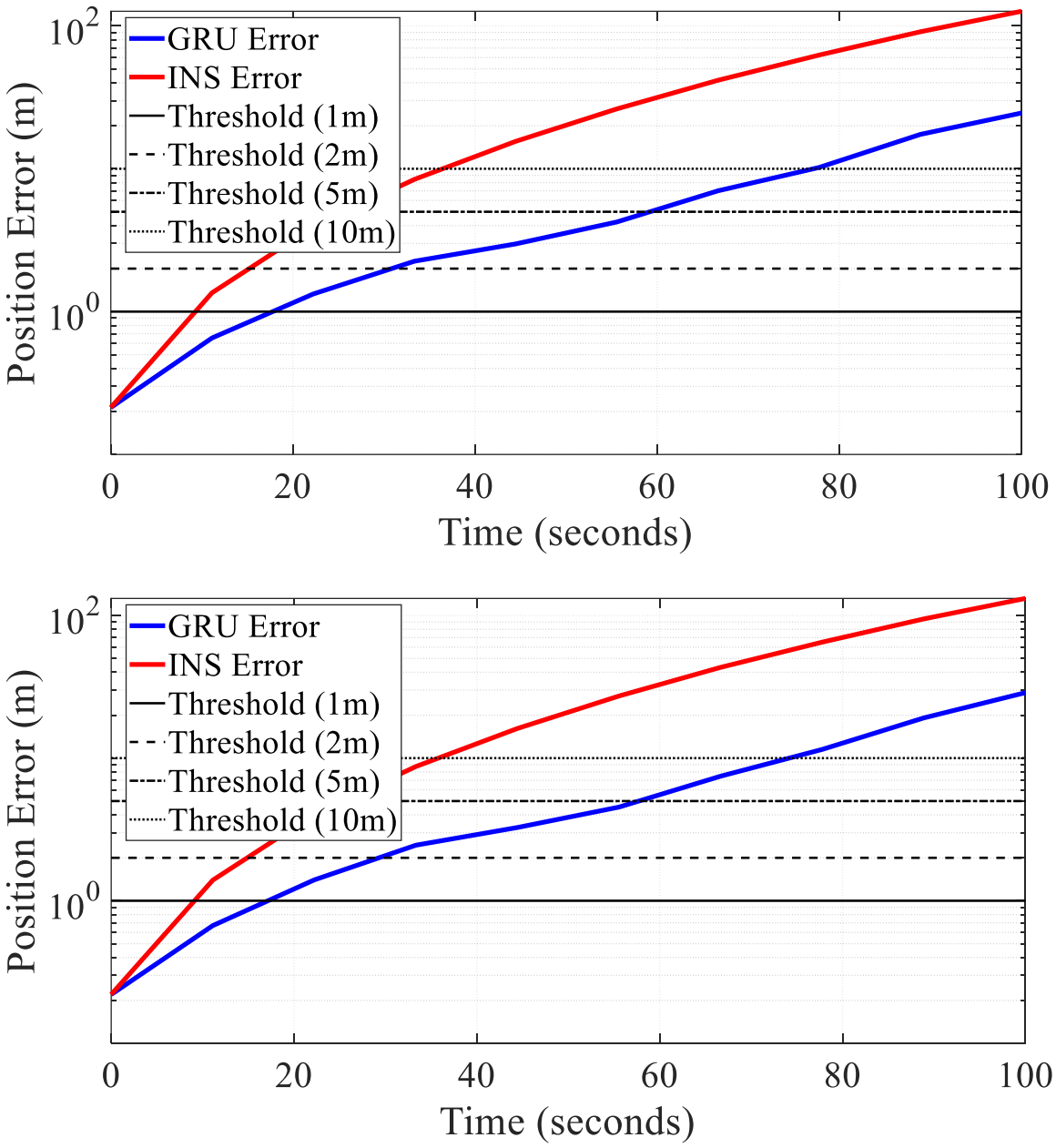


Figure 5-9 Absolute Horizontal Position Error at Each Time Interval with No GNSS Update comparison with Sensor B INS/ Sensor B GRU (a) and Sensor C INS/ Sensor C GRU

5.6 Conclusion

In conclusion, the present study has analysed the performance of GRU architecture compared to INS for error prediction. A notable improvement was observed in the GRU's ability to predict INS errors, thereby successfully mitigating them.

The statistical comparison demonstrated significant improvements in error prediction by the GRU. In the North (N), East (E), and Down (D) directions, the mean errors for the INS, GRU training, and GRU testing were respectively (13.6, 14.2, 0.77), (1.46, 1.70, 0.14), and (0.92, 2.29, 0.15) meters. The corresponding 95th percentile errors were (63.6, 68.8, 4.14), (13.7, 10, 0.82), and (4.76, 16.1, 0.83) meters. These results indicate that the GRU outperformed the INS by up to 93% in position error improvement.

Moreover, the study evidenced horizontal improvements of up to 82% and 80% in training and testing data respectively, demonstrating a consistent performance that indicates no overfitting. In a 100-second GNSS outage period, the GRU achieved a lower error of 25.73 meters compared to INS's 129.16 meters, highlighting a reduction of 103.43 meters in error.

A comparative evaluation with other techniques, including Decision Tree and Recurrent Neural Network (RNN), further affirmed the superiority of the GRU architecture. Specifically, GRU's gate implementation enhanced memory properties' longevity, outperforming other techniques.

The study's figures illustrate the absolute horizontal position error, trajectory comparison, and position error distribution, substantiating the GRU model's efficacy in mitigating position errors, even in highly non-linear scenarios. However, certain challenges persist, such as the fluctuating gradients and significant peaks correlating with heading changes.

The study also focused on analysing the fluctuation in sensor error performance on GRU, trained explicitly on Sensor A and tested on Sensors B and C to represent fluctuations due to thermal and environmental effects on IMUs. The GRU model consistently outperformed INS, reducing errors by up to 93% when applied to Sensors B and C, which differ from the training data; significant reductions in position errors were observed. For instance, errors were lessened from 13.3 to 1.44 in the North direction for Sensor B. The performance gains for both Sensors B and C surpassed INS by up to 86%, and improvements in Mean Horizontal Error and other statistics were up to 82%. However, a Kullback-Leibler (KL) divergence analysis between INS and GRU resulted in values of 0.533, 0.558, and 0.526 for Sensors A, B, and C, respectively, suggesting that fluctuations due to environmental effects were not being mitigated in the GRU architecture. Additionally, there was a slight reduction in performance duration for Sensor C at a 10-

meter threshold, from 42 seconds to 38 seconds, signalling a minor decline in performance under more challenging conditions.

Notwithstanding the GRU's inability to specifically detect fluctuations in the data attributed to thermal and environmental effects on the sensors, these variations did not present a substantial challenge to the system. The GRU was still able to enhance position accuracy, reflecting its resilience and efficacy even in the presence of potential inconsistencies that might affect traditional IMUs.

Overall, the findings underscore the promising potential of the GRU architecture in enhancing positioning performance and tracking the ground truth more closely. These results pave the way for further research into the generalizability of the proposed architecture, particularly in more complex and variable navigation environments, to fully explore its capabilities and limitations.

Chapter 6 Machine Learning-Enhanced GNSS in Urban Canyons: A Novel Multipath Detection and Signal Ranking Approach

Abstract

Environments that produce rich multipath present a range estimation accuracy challenge for techniques involving Global Navigation Satellite Systems (GNSS) signals. Often, air or ground vehicles transition through a range of multipath environments. Existing techniques to mitigate the presence of multipath exist using sensor fusion and interference cancellation techniques. Yet, most of these techniques do not exploit the temporal nature of GNSS datasets to improve multipath conditions. The challenge is to develop novel approaches for mitigating multipath errors in vehicular flying scenarios. Here, we first propose a Gated Recurrent Unit (GRU)-based machine learning model to enhance signal classification between Line-Of-Sight (LOS), multipath, and None Line-Of-Sight (NLOS). After filtering out NLOS, we rank the remaining multipath and direct signals based on their GDOP contribution values to triage quality. The end-to-end system was tested using industrial standard Hardware-In-The-Loop (HIL) capabilities (e.g., realistic GNSS signal generator using Spirent GSS7000 and signal ray tracing using OKTAL-SE Sim3D). Results demonstrate improvements in classifying GNSS signals surpass other methods with a classification accuracy of 93.81%, which translates to a positioning accuracy improvement of up to 67%. This study demonstrates the potential of GRU classification and ranking in real-world GNSS applications for multipath-rich navigation by autonomous vehicle systems.

6.1 Introduction

It is essential not only to provide GNSS positioning consistently but to do so with improved accuracy and precision as the risk to life is severely increased in dense living conditions [79]. Consequently, it is of interest to maintain or improve GNSS position accuracy and availability in these surroundings to provide safe navigation [80]. Whilst the previous chapter focused on reducing errors from INS systems when GNSS is unavailable, it is important to still include GNSS in the longer-term strategy of navigation. Therefore, evaluating techniques to mitigate GNSS errors is crucial.

However, urban canyons provide a challenging GNSS signal propagation and accurate positioning environment. Not only do transition, transmission and receiver errors need to be accounted for when calculating the pseudorange between the GNSS satellites and receivers [81], but multipath of GNSS signals are also factors that increase GNSS positioning errors [82]. As discussed in earlier chapters, NLOS and Multipath can cause severe degradation to GNSS positioning performance. Hardware/Software-level solutions include resolving signal ambiguity through receiver correlator design. The basic idea of the correlator design is to compare three parts: the early, prompt, and late channels in the code tracking loop. It checks for the difference between the direct signal and the reflected one. However, this design is ineffective in reducing the problems caused by NLOS because no direct signal exists for this satellite.

Existing techniques vary in their approach to this issue, which can be categorised as antenna and receiver design, signal processing, integrated systems, visual-aid, and statistical approaches [28]. In this chapter, we will focus on GNSS modules available to consumers and limit the depth of this chapter to methods that can be applied after signal processing.

Given the above need to find a solution to GNSS multipath conditions in urban canyons, this chapter proposes a two-step process for increasing position accuracy by classifying signals into three distinct categories by utilising a Gated Recurrent Unit (GRU) architecture and ranking these signals based on their Geometric Dilution of Precision (GDOP) individual contribution γ_i. The choice of GRU allows for further use of temporal dependencies to improve classification accuracy. These categories are LOS, Multipath

and NLOS. The proposed feature input to the neural network is the observable data available to commercial customers, such as pseudorange, elevation, satellite position, Doppler shift, and C/N0. Once signals are classified for that timestep, NLOS signals will be removed from the list of available satellites. Additionally, all signals are ranked based on their $\gamma_i$values for each satellite. Those with the least γ_i will be used to calculate user position using the least-squares method. This reduces the number of signals that affect the accuracy from considering NLOS signals and precision by reducing the overall GDOP.

The remainder of this chapter is organised as follows: Section two will discuss a literature review on existing techniques. Section three introduces the proposed architecture. Section four focuses on experiment setup and data collection methodology. Section five presents and discusses the results of the experiment and any relevant implications from the analysis. Lastly, section six presents the chapter's concluding remarks and future work.

6.2 Literature Review

6.2.1 Signal Classification Techniques

One of the most common techniques used to check GNSS integrity is Receiver Autonomous Integrity Monitoring (RAIM). RAIM supplies integrity monitoring for GNSS Fault Detection (FD) [83]. It requires the pseudorange measurements of at least five satellites for FD or six satellites for Fault Detection and Exclusion (FDE). The range and position errors modelled for RAIM are carried out in an open sky environment. Therefore, the performance is degraded in urban canyons, with the majority of pseudorange measurements either not being available or experiencing multipath effects [84] .

Another existing method includes 3D environment model mapping, which estimates the possible magnitude of pseudorange errors or helps provide alternative navigation routes to autonomous systems. This method requires a 3D map of the environment to examine the areas, depending on the time of day, for the number of available satellites with LOS access to the receiver of the vehicle [85]. However, these 3D environments may need to be updated compared to the real world. Additionally, this may not aid in detecting NLOS signals when large vehicles are on the ground, blocking the direct path to the receiver.

Lastly, alternative navigation routes may only sometimes be possible as the destination might be in an area where most satellites are not in LOS.

Recently, statistical approaches have been researched as a possible solution in FDE for removing multipath pseudorange measurements. Paper [86] uses a k-means cluster approach for multipath detection. The algorithm uses a set of features such as carrier phase, pseudorange, and carrier-to-noise ratio to categorise into two groups. These groups are ‘direct signal’ and ‘indirect signal’. From simulations, it has been shown to provide improvements over RAIM when multiple pseudorange measurements are affected by multipath. However, the author notes that more analysis is required on selecting parameters that affect the performance of the suggested algorithm. Furthermore, the carrier-to-noise ratio for detecting LOS signals may change depending on the receiver's location and the satellite's elevation.

Another method from the paper [87] uses Support Vector Machines (SVM) to classify the pseudorange measurements into three distinct categories: clean, multipath, and NLOS. The classification is carried out using four features: Received Signal Strength (RSS), change of rate of RSS, pseudorange residual and pseudorange rate. The experiment results show a classification accuracy of 75%. An advantage of using SVM is the capability to handle non-linear datasets. The disadvantage is that those past error dependencies should be considered. Moreover, large datasets slow down SVM solutions.

Paper [88] investigates the development of the random sample consensus (RANSAC) algorithm for fault detection and exclusion in the tightly coupled integration of GNSS/INS, presenting a methodology that leverages a voting mechanism to distinguish between inliers and outliers for enhanced navigation reliability. This approach relies on iterative sampling and model consistency checks to identify and exclude erroneous GNSS data, thereby improving positioning accuracy. However, integration of INS introduces complexities, notably in the synchronisation and fusion of data from additional INS sensors, which can complicate system design and introduce noise and bias terms. It can be challenging to balance the algorithm's sensitivity to its parameters—such as subset sample size, inlier threshold, and iteration count—against the dynamic and often unpredictable nature of GNSS signal degradation. Additionally, the temporal

dependencies are not factored in the algorithm, nor does it allow for adaptability when more data is available, which could aid in improving positioning performance.

6.2.2 Signal Ranking Techniques

Paper [89] proposes a novel approach for GNSS signal ranking. The authors introduce a machine learning-based architecture that classifies GNSS signals as Line-of-Sight (LOS) or Non-Line-of-Sight (NLOS) using quality indicators. The classified signals are then assigned weights, ranking them based on reliability. This weighting scheme incorporates factors such as the Carrier-to-Noise ratio (C/N0), elevation angle, and the probability of NLOS. The study reports a classification accuracy of 93.43% with the random forest algorithm, significantly improving positioning accuracy in all three dimensions. This research highlights the potential of machine learning in enhancing GNSS signal ranking systems.

Another paper that focuses on ranking is [90] proposes multipath detection by continuously monitoring the Carrier-To-Noise Ratio (C/N0) of incoming signals to detect NLOS. Using elevation to determine the C/N0 threshold, the author's algorithm could judge whether the signal experienced a Multipath. Furthermore, to overcome the issues of the signal fluctuating between the threshold, continuous time-series C/N0 were investigated to put a time requirement for the signal to be above the threshold before it can be used again for positioning. The test results show a decrease in the 90th percentile of horizontal positioning error from 37.8 m to 4.31 m. The tests were conducted in multiple different environments with different urban landscapes. Horizontal Root Mean Squared Errors (RMSE) also decreased positioning errors from 12.47 m to 2.23 m. However, the tests conducted in this chapter are all based on static antennas. Therefore, the thresholds are not re-evaluated and may not be suitable for moving vehicles. Furthermore, whilst it does use a continuous-time series to provide a minimum time required for the C/N0 to be above the threshold, it does not consider past error dependencies. Therefore, a minimum time set for all scenarios might not be suitable everywhere.

One paper that uses observables to determine the rank of the signals is paper [91] which evaluates the contribution of individual satellites to the overall Geometric Dilution of Precision (GDOP) in a satellite navigation system. It introduces an observation matrix

H_m for m satellites and a modified matrix H_j^m where one satellite (j^{th}) is removed. The relationship between the full matrix and its modified version is used to calculate the GDOP changes when excluding a satellite. By applying the Sherman-Morrison formula, the document presents a way to calculate the inverse of the modified matrix (G_i^{m-1}) from the original matrix inverse (G_m). This calculation helps in determining the contribution (δG) of the j^{th} satellite to the GDOP value, where a larger δG indicates a higher impact of the satellite on the system's accuracy. This method quantitatively measures each of the satellite's importance to the precision of the navigation solution . This score is then combined using a weighted scoring function cap with the pseudorange prediction to provide the final rank of the signals. Whilst this method introduces a novel approach to signal ranking and using the available GNSS observables, the weighted score is calculated empirically for one location. It needs to recalculate when changing to a different position. Furthermore, the algorithm heavily weighs the contribution from the pseudorange prediction.

6.3 Proposed Architecture

6.3.1 High Level Overview

The proposed architecture, depicted in Figure 6-1, outlines a multi-stage pipeline for processing GNSS signals to enhance position estimation accuracy in challenging environments. The system integrates machine learning-based classification, signal filtering, and geometric optimisation to ensure that only the most reliable satellite signals are used in the position calculation. The stages of the architecture include GRU-based classification, non-line-of-sight (NLOS) signal removal, ranking of signals based on their dilution of precision contribution, and the final computation of position estimates using a least-squares method.

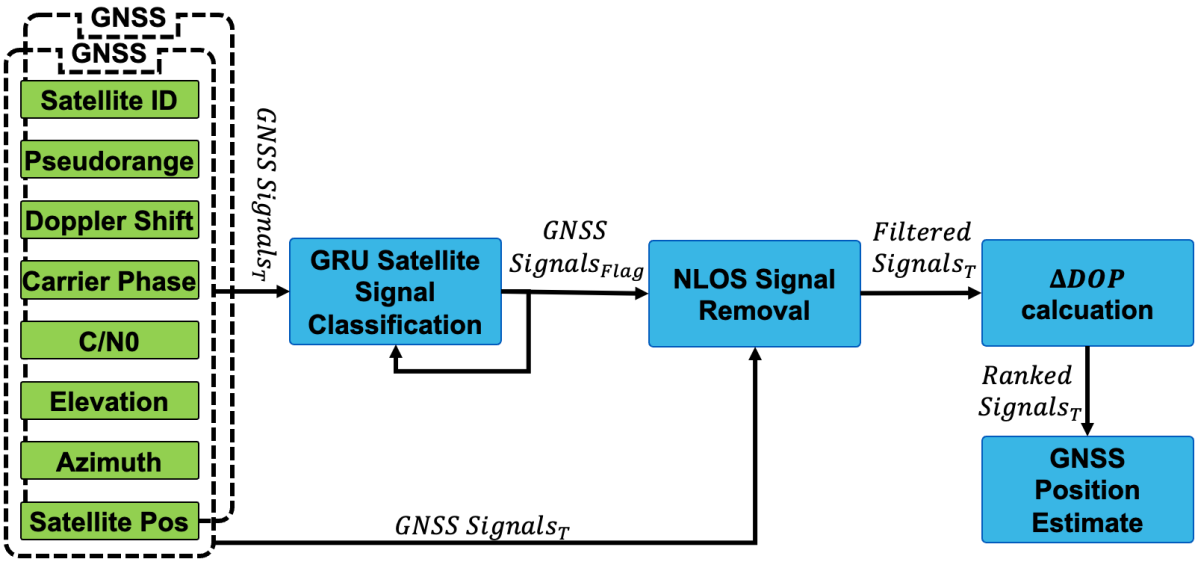


Figure 6-1 High level proposed architecture overview

At each timestep T, the GNSS receiver collects observable data for all visible satellites. These observables include essential signal parameters such as satellite ID, pseudorange, Doppler shift, carrier phase, signal-to-noise ratio (C/N0), elevation, azimuth, and satellite position. This data provides the foundation for subsequent processing and is first passed to a gated recurrent unit (GRU) neural network for classification.

The GRU is designed to analyse the temporal behaviour of each satellite's signal, using time-series data spanning the last 20 timesteps for each satellite. Based on this analysis, the GRU classifies the signals as either line-of-sight (LOS), multipath, or NLOS. This classification takes advantage of the GRU's ability to learn temporal dependencies and recognise patterns indicative of signal impairments, such as the characteristic fluctuations caused by multipath or NLOS conditions. The classification results for all satellites at timestep T are stored in an exclusion list, which is the basis for filtering out unreliable signals.

Once all satellite signals for timestep T have been classified, the system proceeds to the NLOS signal removal stage. Signals identified as NLOS are excluded from further processing, as they are typically unreliable and contribute significant errors to the position calculation. The remaining LOS and multipath signals are then ranked based on their geometric dilution of precision contribution, denoted as γ_i. The γ_i metric quantifies the

contribution of each satellite to the overall geometry of the GNSS constellation, with lower γ_i values indicating a more favorable contribution to positioning accuracy.

To optimise the position estimation process, only the seven signals with the lowest γ_iValues are retained. These signals provide the best geometric configuration for minimising the dilution of precision, ensuring that the position and velocity estimates are as accurate as possible. Finally, the retained signals are inputs to the least-squares algorithm, which computes the GNSS position and velocity estimates for the current timestep.

This architecture integrates temporal analysis, classification, and geometric optimisation to provide a robust framework for GNSS signal processing. By filtering out unreliable signals and leveraging only the most geometrically advantageous observations, the system mitigates the effects of multipath and NLOS conditions, resulting in more accurate and reliable position estimates. The GRU-based classification and iDOP-based ranking are particularly critical in environments with significant signal impairments, such as urban canyons or dense foliage, where traditional GNSS processing methods often struggle.

6.3.2 GRU-based Satellite Signal Classification

GNSS receivers' observations will be recorded to collect feature inputs to the network. These include satellite ID, pseudorange, doppler shift, carrier phase, C/N0, elevation, azimuth, and satellite position in Earth-Centered Earth Fixed (ECEF) coordinates. The data is then processed using weights, biases, and non-linear functions to link the input features and the output decision. The inputs were chosen based on the current literature systems investigated in the state-of-the-art review. C/N0 was effectively used in [86] to classify NLOS. Elevation-based C/N0 was also reviewed in [90] as an influential aspect in deciding whether multipath was present. Furthermore, Doppler shift and pseudorange were also used in the paper [87] to improve the classification performance in SVM. Using these features, the GRU architecture will be trained to classify each signal at the timestep T as either LOS, Multipath, or NLOS. The predicted outputs will then be compared against the label for each signal. The training will continue until no further improvement in accuracy is observed. The output from the GRU will be used in the next stage.

The classification for each signal at the timestep T will then be used to exclude NLOS flags. Therefore, only Multipath and LOS signals will remain after the classification process. This will then be carried forward to the next process, which is signal ranking.

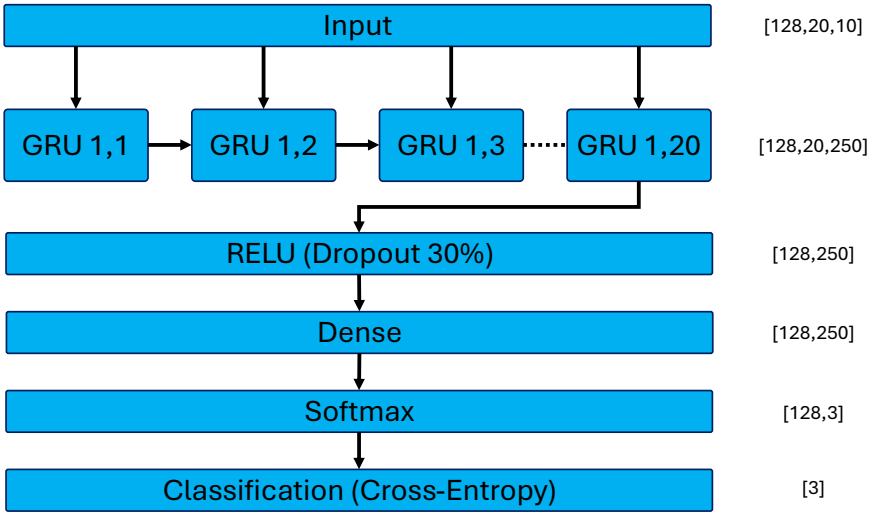


Figure 6-2 Machine learning architecture with optimized parameters

The GRU architecture is shown in Figure 6-2. Hyperparameter optimization using random search from the Keras library was applied to optimize learning parameters, number of fully connected layers, activation function, regularization strength, and if standardization will affect convergence speed.

The data is standardized using a z-score where the mean equals zero and a standard deviation of one. One layer with 250 hidden units and a dropout rate of 30% is used. A SoftMax layer is added, and a cross-entropy loss function is used to train the classification algorithm and provide an output of which signal is NLOS, multipath or LOS. The labels are provided using the ground truth data from the GSS7000 simulator. The trained weights and biases were saved in a .h5 file for testing. Testing is carried out with a train/test split of 80/20. The output of test data is recorded and is compared against the true labels of the data. The training was conducted on an ML platform consisting of a i7-12700K and a Nvidia RTX3070Ti. The response time for testing was recorded as 6ms per output.

6.3.3 Discussion on using GRU for classification and regression simultaneously

In the previous chapter, a GRU was employed to decrease INS errors, demonstrating the model's effectiveness in leveraging temporal patterns for regression tasks in navigation systems. This finding prompts consideration of extending the GRU's functionality to simultaneously handle the classification of GNSS signals as NLOS, Multipath, or LOS. While combining these tasks into a single GRU may initially seem advantageous for reducing model complexity, practical considerations suggest maintaining distinct GRUs.

A key challenge lies in the input data for each task. The GRU for INS error prediction relies solely on inertial measurements designed for GNSS-denied environments. In contrast, the GNSS classification task depends exclusively on GNSS signal features, such as signal-to-noise ratio or satellite geometry, and does not require inertial data. Since the two tasks do not share input features, a unified GRU would not benefit from any meaningful overlap in the input space, making the combination less effective.

Furthermore, the temporal characteristics of the input data streams differ. INS data is sampled at high frequencies (100–1000 Hz), providing dense temporal sequences for error modelling. On the other hand, GNSS data is received at much lower frequencies (1–10 Hz), which aligns with the needs of classification tasks. Using a single GRU for these two tasks would require it to process asynchronous and temporally disjoint data streams, adding unnecessary complexity and potentially undermining its performance.

Given these factors—distinct input domains and differing temporal characteristics—it is recommended to maintain separate GRUs for INS error prediction and GNSS signal classification. This separation ensures that each GRU can be optimized for its specific task, leveraging the temporal data in a way that aligns with the respective task's unique requirements.

6.3.4 Signal Ranking Using γ_i

For ranking the multipath and LOS signals, the chapter proposes a method to calculate their γ_i using the azimuth and the elevation angle for each satellite in respect to the pose of the vehicle. γ_i is the impact of the individual satellite's geometry on the overall GDOP. It is an indication of how the removal or addition of a particular satellite affects

positioning accuracy. First, the geometry matrix $\boldsymbol{H}$ for each timestep needs to be calculated.

$$H_r = [N_r, E_r, U_r, 1] \tag{6.1}$$

Where, N, E, U are as follows:

$$U = \sin(\theta) \tag{6.2}$$

$$N = co\,s(\theta)\,co\,s(\phi) \tag{6.3}$$

$$E = \cos(\theta)\sin(\phi) \tag{6.4}$$

therefore, matrix $\boldsymbol{H}$ at timestep t is:

$$\boldsymbol{H_t} = \begin{bmatrix} N_1 & E_1 & U_1 & 1 \\ N_2 & E_2 & U_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ N_n & E_n & U_n & 1 \end{bmatrix} \tag{6.5}$$

For each satellite, the ΔDOP is calculated by projecting the contribution of that satellite's geometry to the overall precision matrix Q.

$$Q = H^T H \tag{6.6}$$

If H_r is represents a row from the geometry matrix for a specific satellite, then the γ_i for that satellite can be given by [92]:

$$\gamma_i = h_r Q h_r^T \tag{6.7}$$

γ_i must be calculated for each satellite at each timestep. The algorithm employed to do so and the ranking of these signals are shown in Algorithm 1. Once the signals have been ranked, with the lowest value having the highest standing, the seven best-performing satellites will then be used to calculate the final GNSS positioning using the Least Squares approach.

Algorithm 1: Computation of GDOP and γ_i for Each Timestamp

1: Input: result (cell array containing satellite information for each timestamp)
2: Output: GDOP_values (vector of GDOP values), result (updated with delta_DOP)
3: num_timestamps ← length(result)
4: GDOP_values ← array of zeros with size num_timestamps

```
5: γ_i _cells ← cell array with size num_timestamps
6: for t ← 1 to num_timestamps do
7:    Info ← result{t}(:, 3:4) // Extract relevant satellite      information
8:    n ← length(Info)
9:    H ← matrix of zeros with size n x 4
10:   γ_i  ← array of zeros with size n
11:   Info ← convert degrees to radians (Info)
12:   for i ← 1 to n do

13:      East ← cos(Info(i,1)) * sin(Info(i,2))
14:      North ← cos(Info(i,1)) * cos(Info(i,2))
15:      Up ← sin(Info(i,1))
16:      H(i,:) ← [North East Up 1]
17:   end for
18:   Q ← inverse(transpose(H) * H)
19:   GDOP ← sqrt(trace(Q))
20:   GDOP_values(t) ← GDOP // Store the computed GDOP
21:   for i ← 1 to n do
22:      γ_i (i) ← H(i,:) * Q * transpose(H(i,:))
23:   end for
24:   result{t} ← append γ_i to result{t}
25:   result{t} ← sort rows of result{t} based on the last column
26: end for
```

6.3.5 Position Estimation Using Least-Squares

Estimating the position of a GNSS receiver involves minimizing the discrepancies between the observed and computed pseudorange from a constellation of satellites. The least squares method is employed for this optimization problem, yielding the receiver's position and clock bias. This is a modification of mentioned least-squares approach in previous chapter. The pseudorange R_i to the i-th satellite can be represented as:

$$R_i = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} + c(b - b_i) + e_i \tag{6.8}$$

Where (x, y, z) denotes the receiver coordinates, (x_i, y_i, z_i) represents the known coordinates of the i-th satellite, c is the speed of light, b is the receiver clock bias, b_i is the i-th satellite clock bias, and e_i is the measurement noise. Due to the nonlinear nature of the observation equations, a linearization is required around an initial approximation (x_0, y_0, z_0) with an estimated clock bias b_0. The linearized form of the observation equation is:

$$R_i + \delta R_i = \rho_i + (x - x_0)\frac{\partial \rho_i}{\partial x} + (y - y_0)\frac{\partial \rho_i}{\partial y} + (z - z_0)\frac{\partial \rho_i}{\partial z} + c\delta b + e_i \tag{6.9}$$

Where p_i is the geometric range from the approximate position to the i-th satellite, and δR_i is the range correction. The design matrix A, which includes the partial derivatives of the geometric ranges with respect to the position and clock bias, is constructed as follows:

$$A = \begin{bmatrix} \frac{\partial R_1}{\partial x} & \frac{\partial R_1}{\partial y} & \frac{\partial R_1}{\partial z} & c \\ \frac{\partial R_2}{\partial x} & \frac{\partial R_2}{\partial y} & \frac{\partial R_2}{\partial z} & c \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial R_n}{\partial x} & \frac{\partial n}{\partial x} & \frac{\partial R_n}{\partial x} & c \end{bmatrix} \tag{6.10}$$

The least squares solution entails computing the vector of corrections δ to minimize the sum of squared residuals. The correction vector δ is determined by solving the normal equations:

$$(A^T A)\delta = A^T L \tag{6.11}$$

Where L is the vector of observed minus computed ranges, representing the residuals. The correction is given by:

$$\delta = (A^T A)^{-1} A^T L \tag{6.12}$$

The corrected position and clock bias are subsequently updated as:

$$\begin{bmatrix} x \\ y \\ z \\ b \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ b_0 \end{bmatrix} + \delta \tag{6.13}$$

This iterative process refines the position and clock bias estimates until the improvements become negligible, thereby achieving an optimal solution as per the least squares criterion. For this chapter's specific deployment of the architecture, MATLAB receiver position function is utilized to improve processing performance.

6.4 Simulation Setup & Methodology

This chapter utilises the Spirent GSS 7000 simulator to simulate GNSS Radio Frequency (RF) L-band signals in a Hardware-In-a-Loop (HIL) setup as described in the previous chapter. This enables realistic simulation of GNSS constellation and signal characteristics when transmitting between the satellites and the receiver. The scenario setup and route for the vehicle will be used as the input parameters. Furthermore, simulating GNSS errors such as tropospheric, satellite clock bias and more will also be set. To account for multipath errors in an urban canyons setting, OKTAL-SE Sim3D software introduces realistic reflection simulation of signals. The location chosen for this setup is in Toulouse, France. An illustration of the 3D environment from Sim3D is shown below.

Figure 6-3 Sim3D environment of Toulouse, France

These signal reflections are calculated using state-of-the-art ray tracing to improve GNSS realism in complex landscapes. Furthermore, the reflection and refraction characteristics of the materials in these environments will also be considered. The route proposed for the simulation test is comprised of an infinity-shaped path with constant velocity for a duration of twenty-four hours. This is shown in Figure 6-4. The twenty-four hour period consists of two global passes for the GNSS satellites and therefore allows for various satellites with varying elevation angles for robust training and testing purposes. The velocity chosen for this simulation is at a constant 10 m/s. This is a representative speed of most urban vehicles. The vehicle's height is 20 m above sea level, and a complete circuit takes 200 seconds. The number of loops completed is 345 loops.

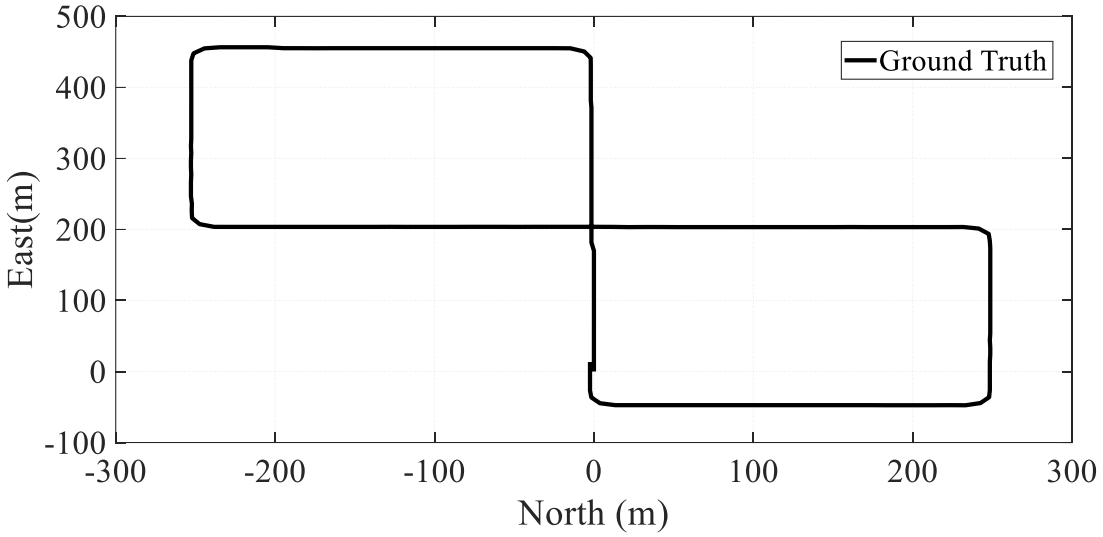


Figure 6-4 Proposed trajectory for data capture and architecture testing

A Ublox F9P receiver receives the generated RF from the GSS7000. The receiver will then generate observables that are required for classifying and ranking GNSS signals. For this chapter, the GNSS observables are saved in .ubx format. The data is then fed into RTKLIB to convert .ubx files into a standardized RINEX format. This provides us with two files. One is the observable data that contains the satellite observations collected by the GNSS receiver as shown in the figure below.

```
> 2022  1 23  0  0 32.0000000  0  4
G 1  20534987.964 7 107912089.921 7     -2771.624          46.000
G 3  21730212.708 7 114193037.49617     -1517.063          45.000
G19  21122670.544 7 111000382.160 7       360.521          45.000
G20  19907009.742 7 104612042.047 7     -1501.942          46.000
> 2022  1 23  0  0 33.0000000  0  5
G 1  20535515.400 7 107914861.840 7     -2772.547          45.000
G 3  21730501.539 7 114194554.262 7     -1517.152          46.000
G11  22858983.717 6 120124769.42026      -403.996          37.000
G19  21122601.980 7 111000021.499 7       360.323          43.000
G20  19907295.517 7 104613544.012 7     -1502.210          43.000
```

Figure 6-5 Example observable output in RINEX format

These are the measurements the receiver has made from the signals broadcast by the satellites. These measurements include pseudorange, carrier phase, Doppler, signal-to-noise ratios, and more. The other file is the navigation data that contains the orbit parameters or ephemerides data for the satellites, which allows users to compute the

precise location of the satellites at any given time. This information is necessary for processing the observation data to determine an accurate position.

The RINEX files are imported into MATLAB to correct pseudoranges with the navigation file. Then, each signal will be classified as LOS, multipath or NLOS. NLOS signals will be removed, and the rest of the signals will be ranked depending on their γ_i as described in the earlier section. The top 7 signals are calculated using the receiver position function in MATLAB. Comparisons of techniques will include methods such as decision trees [93], SVM [87] and Random Forest [94]. Their classification and overall positioning performances will be contrasted.

6.5 Results & Discussion

6.5.1 Classification Performance

The classification performance of all four networks was compared in this work. These include decision tree, SVM, Random Forest and GRU. The three classes include LOS (Class: 1), Multipath (Class: 2) and NLOS (Class: 3). For all confusion matrix results presented in this chapter, the testing outcomes are considered only with the diagonal boxes representing scenarios where the correct classification was provided for the given input features.

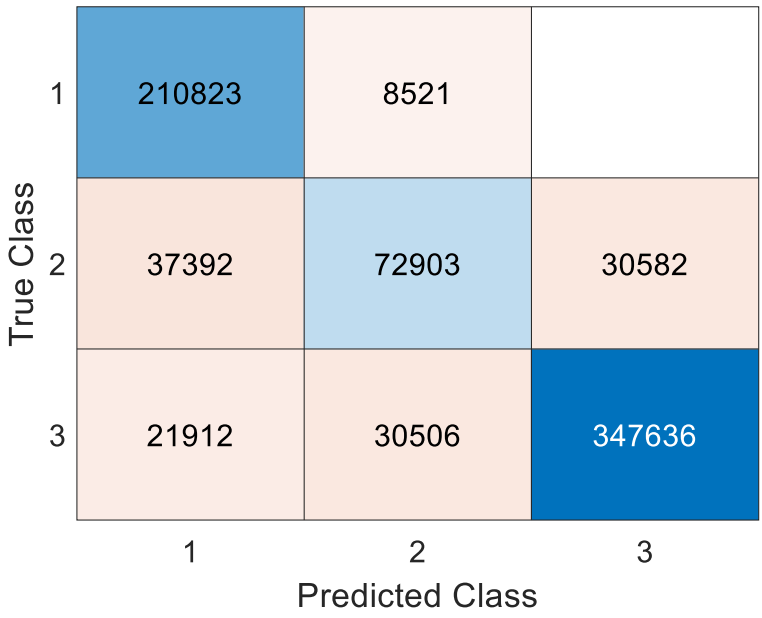


Figure 6-6 Confusion matrix for test results of decision tree classification performance

Figure 6-6 shows the confusion matrix results from the Decision Tree Classification. Here, the overall accuracy observed is 79.84%, with its best classification performance being class 3 and the least being class 2, where more than double the classification errors occur with the confusion between class 1 and class 3.

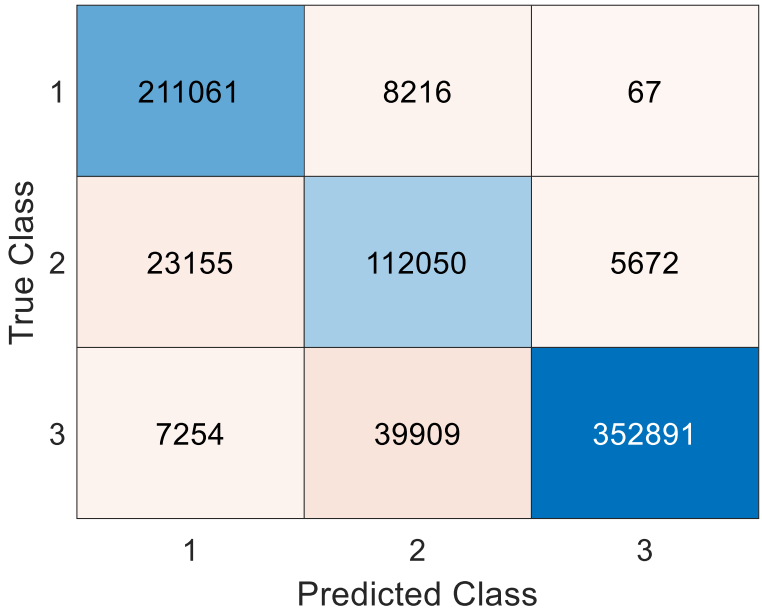


Figure 6-7 Confusion matrix for test results of SVM performance

Figure 6-7 shows the confusion matrix results from SVM. Here, the overall classification accuracy observed is 88.92%. This is an improvement compared to the Decision Tree network, which has reduced classification errors for all classes. This is especially true for class 2. With the use of kernel tricks, SVMs can efficiently handle high-dimensional data and are capable of creating non-linear decision boundaries. Decision Trees, on the other hand, inherently construct linear boundaries (for the feature space), which might need to be revised for complex, non-linear problems. Also, SVMs can be more robust to outliers than decision trees. This is because SVMs focus on the points closest to the decision boundary (support vectors). This is supported in Table 6-I where SVM consistently outperform Decision Trees across multiple classes. In contrast, Decision Trees can be significantly influenced by outliers, leading them to create skewed splits that are not representative of the overall data set.

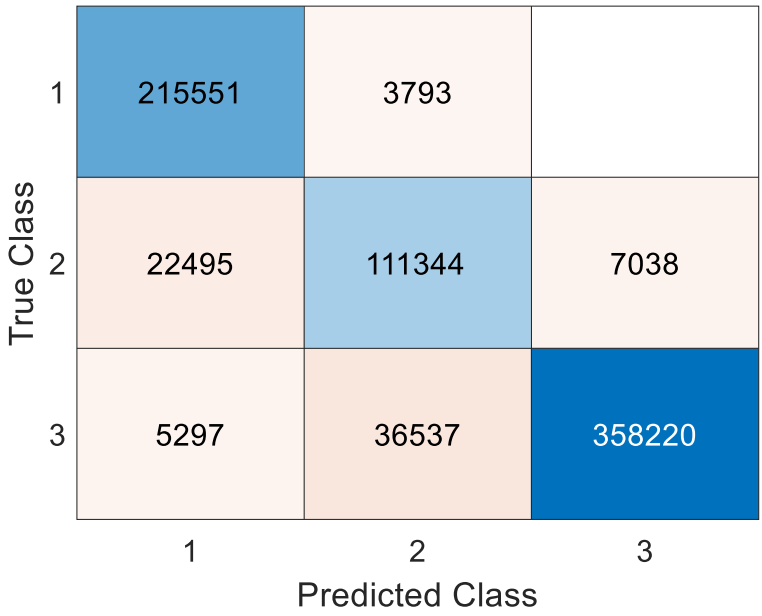


Figure 6-8 Confusion matrix for test results of random forest classification performance

Figure 6-8 shows the confusion matrix results from Random Forest. Here, the overall classification accuracy observed is 85.988%. This is a reduction in accuracy over SVM but an increase compared to the Decision Tree. This is expected as Random Forests use multiple Decision trees and sample a random subset of features to classify. Then, a majority vote is taken to find the most suitable output classification. This technique, therefore, is better at handling outliers and reduces overfitting. This can be seen in the F1 Score across all classes, which suggests that it reduces overfitting, a common problem with single decision trees. However, SVM has a higher precision of 98.40% compared to RF 89.55%. This indicates that when SVM predicts an instance as NLOS, it is correct more often than Random Forest.

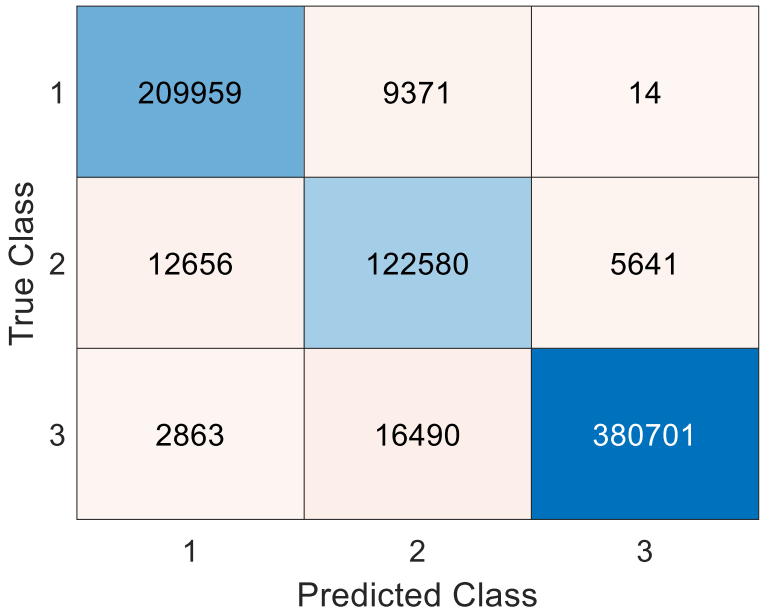


Figure 6-9 Confusion matrix for test results of GRU (proposed) classification performance

Figure 6-9 shows the confusion matrix results from the GRU. The GRU model, with an accuracy of 93.81%, outperforms SVM (88.92%), RF (85.98%), and DT (79.84%) in classifying GNSS signals. This high accuracy indicates that the GRU model correctly identifies a more significant proportion of LOS and NLOS signals. The GRU's recurrent architecture, capable of capturing temporal dependencies in the sequential GNSS data, likely contributes to this enhanced accuracy. The GRU can make more informed predictions on the sequential observables inherent in GNSS signals by effectively utilising past information through its gating mechanisms.

Regarding precision, the GRU leads with 98.54% for NLOS signals, slightly higher than the SVM's 98.40%, and significantly surpassing RF's 89.55%. Precision measures the proportion of true positives among all positives identified by the model. In the context of GNSS, high precision is significant as it minimises false positives — erroneous NLOS classifications that could disrupt signal integrity. The GRU's gating mechanisms allow it to retain only pertinent historical information, likely leading to fewer false positives and increasing precision.

Lastly, the F1 score, the harmonic mean of precision and recall, stands at 96.82% for the GRU. This is significantly higher than both SVM (93.03%) and RF (89.35%), reflecting a solid balance between precision and recall. The high F1 score for the GRU suggests that

it is consistent in its signal classification and maintains robustness against the differing challenges presented by the NLOS and LOS classifications.

Table 6-I Network performance comparison of decision tree (DT), SVM, random forest (RF) and GRU (proposed) in regard to accuracy, class precision, recall and F1 Score

		Networks			
Class	**Metrics**	**DT (%)**	**SVM (%)**	**RF (%)**	**GRU (%)**
All	**Accuracy**	79.84	88.92	85.98	93.81
LOS	**Precision**	75.62	87.41	87.95	93.12
	Recall	87.39	96.22	95.96	95.72
	F1 Score	81.08	91.61	91.78	94.4
MP	**Precision**	65.13	69.95	73.41	82.58
	Recall	51.78	79.54	65.35	87.01
	F1 Score	57.69	74.44	69.15	84.74
NLOS	**Precision**	86.90	98.40	89.55	98.54
	Recall	85.05	88.21	89.16	95.16
	F1 Score	85.97	93.03	89.35	96.82

Several key factors emerge after examining the performance disparities between the GRU, SVM, Random Forest, and Decision Tree models. The GRU model's superior performance can be attributed to its advanced capability to process temporal sequences, a critical feature given the sequential nature of GNSS signals. This is contrasted with SVM and Decision Trees, which, despite their robustness in static classification tasks, may need more dynamic temporal processing capabilities essential for accurately classifying sequential GNSS data. While mitigating some of the Decision Tree's overfitting issues through ensemble learning, Random Forests still falls short of the GRU's nuanced temporal analysis. This analysis underscores the importance of model selection in GNSS signal processing, where temporal dynamics play a pivotal role.

6.5.2 Position Performance Comparison

Table 6-II compares the mean error in terms of North, East, Horizontal and Vertical direction for the different approaches to positioning, including the receiver-based position, no NLOS signal, which represents signals that are only reflections being removed, ranking signals only, LOS only as reference which are signals with no reflections and our proposed architecture of using GRU to classify signals and ranking based on their ΔDOP score. The labels are used to determine which signals are LOS, Multipath, and NLOS to obtain the above categories and are removed accordingly. As an additional point of comparison, SVM, Random Forest and Decision Tree output classification results were used to calculate the position performances. GRU outperforms

all other techniques for all directions with a horizontal error of 7.04 m. This showcases the advantage of GRU's temporal abilities and the ranking of signals to work in conjunction, therefore reducing the mean error.

Table 6-II Comparison of mean error for various positioning technique in the north, east, horizontal and vertical directions

	Mean (m)			
	North	**East**	**Horizontal**	**Vertical**
LOS only (ref)	2.52	3.76	4.53	14.35
Rank + GRU	5.55	4.33	7.04	22.20
SVM	6.41	5.55	8.48	22.22
Random Forest	6.73	7.45	10.04	40.45
Decision Tree	8.02	4.44	9.17	30.24
Rank Only	9.00	8.63	12.47	44.77
No NLOS	9.59	9.22	13.31	45.89
Receiver	10.36	13.83	17.28	75.40

Table 6-III is a similar comparison to Table 6-II with standard deviation being evaluated. In a majority of cases, the Rank + GRU architecture outperforms all other algorithms except in the east direction, where the error standard deviation of the proposed architecture is higher than SVM at 4.34 m compared to 3.33m This indicates that there are greater variations in the performance in the east direction.

Table 6-III comparison of error standard deviation for various positioning technique in the north, east, horizontal and vertical directions

	Standard Deviation (m)			
	North	**East**	**Horizontal**	**Vertical**
LOS only (ref)	3.05	3.34	4.52	16.80
Rank + GRU	5.45	4.34	6.97	22.40
SVM	7.76	3.33	8.44	34.40
Random Forest	6.55	8.52	10.75	33.21
Decision Tree	8.52	8.55	12.07	30.33
Rank Only	7.89	7.30	10.74	33.10
No NLOS	8.68	7.82	11.69	34.94
Receiver	8.59	12.09	14.83	63.65

Table 6-IV compares the 95th percentile performance of the listed methods. We see a similar situation as in Table 5-II, where the GRU performs best in the north, east, and Vertical directions but worse in the east direction compared to SVM. Therefore, the error distribution is more extensive for the GRU in the east direction. However, in terms of horizontal performance, the increased performance in the north direction contributes to better performance. This is to be expected as GRU can identify temporal dependencies that SVM cannot access.

Table 6-IV Comparison of 95th percentile error for various positioning technique in the north, east, horizontal and vertical directions

	95th Percentile Error (m)			
	North	**East**	**Horizontal**	**Vertical**
LOS only (ref)	8.62	10.43	13.57	47.95
Rank + GRU	16.45	13.01	20.97	67.00
SVM	21.93	12.21	25.37	91.02
Random Forest	19.83	24.49	31.53	106.87
Decision Tree	25.06	21.54	33.31	90.90
Rank Only	24.77	23.23	33.96	110.98
No NLOS	26.96	24.87	36.68	115.77
Receiver	27.54	38.00	46.94	202.71

Table 6-V is the percentage improvement when compared to the receiver. Rank + GRU shows an improvement of 55% in the horizontal direction compared to the receiver and an improvement of up to 42.25% compared to existing techniques.

Table 6-V Comparison of percentage position error improvement for various positioning technique and VDOP/HDOP in the north, east, horizontal and vertical directions

	Improvement (%)			
	North	**East**	**Horizontal**	**Vertical**
LOS Only (ref)	69%	73%	71%	76%
Rank + GRU	40%	66%	55%	67%
Random Forest	20%	68%	46%	55%
SVM	28%	36%	33%	47%
Decision Tree	9%	43%	29%	55%
Rank Only	10%	39%	28%	45%
No NLOS	2%	35%	22%	43%
Receiver	Reference			

Figure 6-10 shows a position plot comparison between the Ground Truth and GNSS receiver in the horizontal direction. The plot shows significant variations between each other. The receiver calculates the position based on LOS, Multipath or NLOS signals. Therefore, there is a reduction in positioning accuracy and precision. For urban canyons with tighter safety tolerances, relying only on the GNSS receiver without any aid from other techniques or fusion of sensors would generate a hazardous environment.

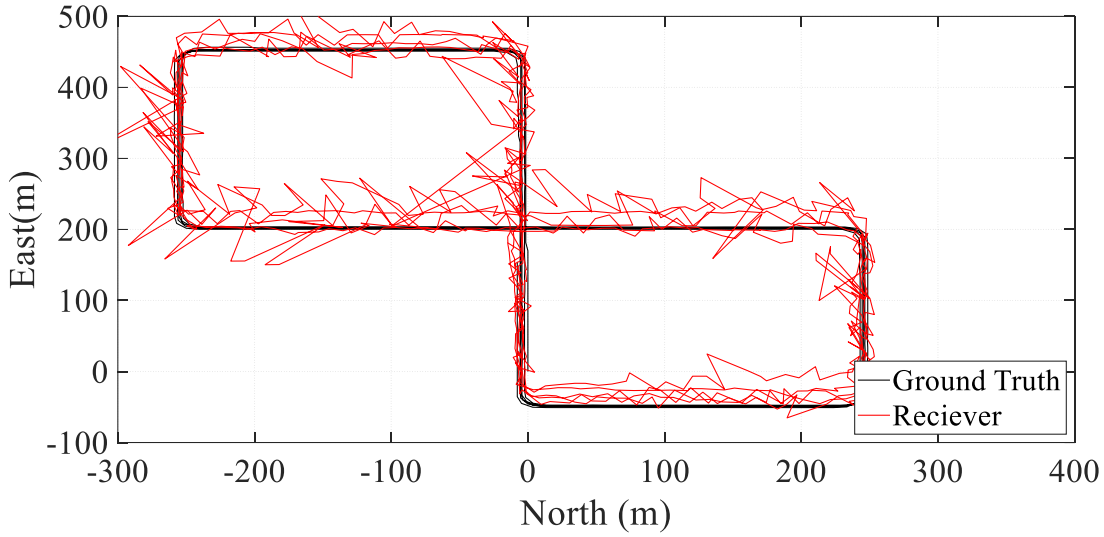


Figure 6-10 Comparison of position plot performance between ground truth and receiver in the horizontal direction

Figure 6-11 compares the performance of our proposed Rank + GRU architecture to SVM from the current literature. The proposed architecture more accurately follows the ground truth than the SVM. Some of the SVM variations in position can be attributed to earlier classification results, where a greater number of NLOS signals are being misclassified as LOS or multipath signals, reducing the system's positioning accuracy. In contrast, the improved performance of the GRU due to its gates and its ability to explore temporal dependencies reduces the position errors observed.

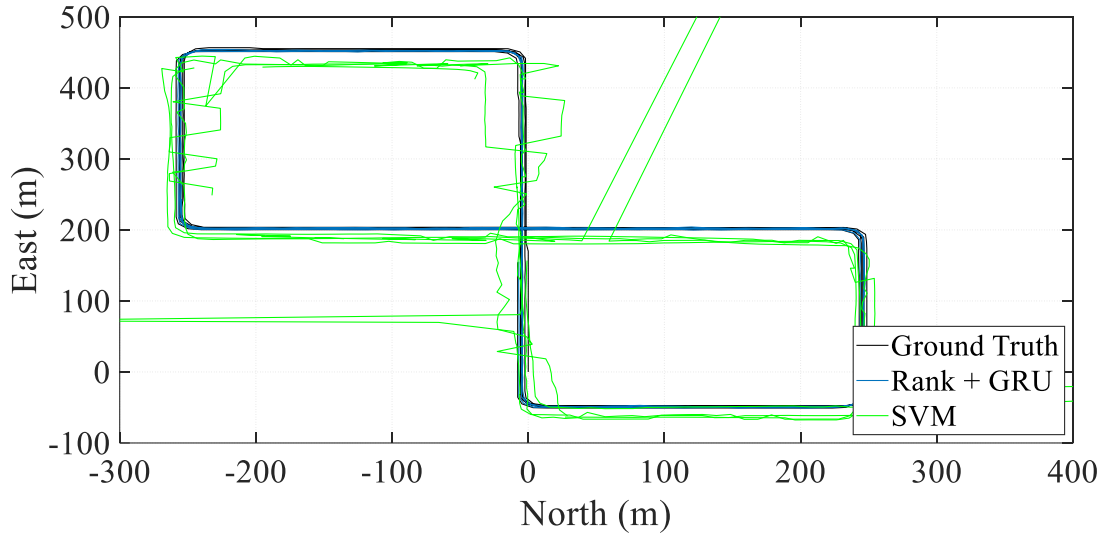


Figure 6-11 Comparison of position plot performance between ground truth, proposed solution (Rank + GRU) and SVM in the horizontal direction

Figure 6-12 is a comparison of absolute horizontal position error distribution between the proposed architecture, receiver and SVM. The receiver has the largest position error distribution with peaks of up to 60m. SVM shows a reduction in the distribution due to the removal of NLOS signals from the GNSS position calculation. Furthermore, our proposed solution reduces this further and shows increased precision and accuracy in its position estimates. This indicates that using GRU for classification and ranking for using the best signals to remove perturbation errors improves overall position performance.

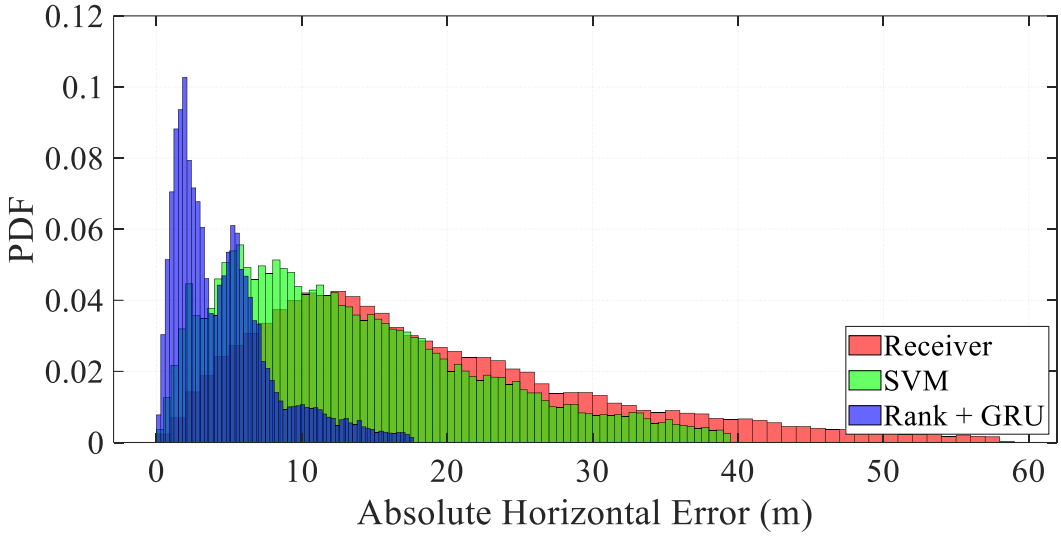


Figure 6-12 Comparison of position error distribution between receiver, proposed solution (Rank + GRU) and SVM in the horizontal direction

6.6 Conclusion & Future Work

The results show that combining a GRU model for signal classification with GDOP contribution ranking improves the accuracy of GNSS signal classification and positioning. The GRU model achieved 93.81% accuracy, outperforming other models such as SVM (88.92%), Random Forest (85.98%), and Decision Tree (79.84%) in classifying GNSS signals.

The GRU model consistently outperformed other techniques regarding mean error across north, east, horizontal, and vertical directions. Notably, it achieved a horizontal error of 7.04 m, showcasing the effective combination of its temporal abilities and signal ranking in reducing mean error. A detailed comparison of mean error in various directions further

underscores the GRU model's efficacy: in the North direction, the mean error was 5.55 m, and in the East Horizontal direction, it was 4.53 m.

The GRU model also demonstrated superior performance when evaluating the standard deviation of errors. In the North direction, the error standard deviation was 5.45 m, and in the East Horizontal and Vertical directions, it was 4.34 m and 6.97 m, respectively. The 95th percentile error measurements further reinforced the model's efficacy, with the GRU model showing significantly lower errors than other methods across all directions.

Importantly, when compared to the baseline receiver, the GRU model showed an improvement of 55% in the horizontal direction and up to 42.25% compared to existing techniques. This highlights the practical implications of the GRU model in real-world GNSS applications, particularly in challenging environments like urban canyons, where accurate and reliable signal classification and positioning are crucial for applications such as autonomous vehicle navigation.

In summary, the two-step approach significantly advances GNSS signal processing by removing only NLOS signals and ranking the remaining signals, offering marked improvements in signal classification accuracy and positioning precision. This has profound implications for enhancing the reliability and efficacy of GNSS systems, particularly in urban settings where multipath errors are a significant challenge for intelligent vehicles.

Chapter 7 Robust Federated Multi-Sensor Scheme: Deep learning Approach for Integrity-based Positioning in Urban Canyons

Abstract

The necessity for accurate Unmanned Aerial Vehicle (UAV) navigation in urban canyons, where Global Navigation Satellite Systems (GNSS) signals are often obstructed, cause multipath conditions or when there is denial from interference, highlighting the limitations of current technologies explored in literature. Conventional machine learning approaches provide deterministic results but need to improve in evaluating the integrity of sensor data. This is an important element for safely deploying autonomous systems in complex environments. Additionally, the centralised architecture that is proposed in literature, is less resilient to individual sensor failures. This would reduce positioning accuracy and, consequently, threaten the surrounding environments. This chapter proposes a robust federated multi-sensor architecture with a multi-layered approach to sensor fusion to address the challenges of multipath conditions. The fusion architecture consists of the following sensors: GNSS and Inertial Navigation System (INS) positioning that is corrected using Gated Recurrent Units (GRU) to improve position accuracy; Monocular Visual Odometry (VO), in combination with a Barometer, that are used to enhance accuracy of positioning. The chapter also proposes a Bayesian Long Short-Term Memory (LSTM) architecture that provides stochastic output by implementing Monte Carlo dropouts. The distribution of the predicted position is then used to estimate the system's integrity by evaluating the estimated Protection Level (PL). The hardware setup includes realistic simulation of operating scenarios in Urban Canyon using Spirent GSS7000 and OKTAL-SE Sim3D for GNSS Hardware-In-the-Loop (HIL) simulation and ray-tracing quality multipath replication. To obtain VO sensor data, AirSim and Unreal Engine are employed to increase visual realism for Urban Canyon

simulation. The system's robustness is tested by including variations in GNSS quality and weather conditions. Results show a 22.1% reduction in the 95th percentile horizontal error compared to state-of-art literature. A federated approach fuses GNSS/INS/Vo/Barometer, showcasing improved navigation accuracy and precision. Additionally, when the system's integrity was compared to an EKF, a 98% reduction in misleading integrity information was observed.

7.1 Introduction

UAVs are at the verge of reforming a variety of industries, offering unique capabilities in various applications [95]. Their potential in urban environments is particularly notable. In these settings, drones can be utilised for a wide range of purposes, including rapid delivery services [96], where they could significantly reduce delivery times [97]. Similarly, in emergency response scenarios, UAVs can quickly reach areas that might be inaccessible to traditional vehicles due to traffic congestion or natural disasters [98], thereby facilitating faster response times and potentially saving lives [99]. Moreover, UAVs hold the potential to improve the depth and frequency of infrastructure inspection, where they can access hard-to-reach areas of buildings, bridges, and other structures, providing detailed inspections without the need for costly and risky human intervention [100]. However, the realisation of these applications hinges on overcoming significant challenges, particularly in ensuring robust autonomous navigation in densely populated urban environments [101]. The complexity of navigating these environments is heightened by the presence of numerous obstacles, varying weather conditions, and the need for precise manoeuvring in confined spaces [102]. This demands highly advanced Position, Navigation, and Timing (PNT) solutions to be accurate, precise, and robust. The challenge is further compounded by the necessity for these systems to operate with high integrity, particularly in urban low-altitude areas where the tolerance for error is much lower compared to more open environments [103]. In this context, integrity refers to the trustworthiness of the navigation system's information, a critical factor for safe and efficient UAV operation [104] [105].

This chapter aims to consolidate what has been learned from previous chapters to improve navigation accuracy, integrity, and robustness for UAVs in urban settings. By applying machine learning techniques that provide a stochastic output, the proposed system evaluates the system integrity every timestep, ensuring accurate position estimates and robustness amidst fluctuating environment conditions. The effectiveness of this approach will be validated through simulations using the data collection system mentioned in the previous chapter. The novel contribution of this chapter is the proposal and testing of a federated robust multi-sensor fusion scheme which takes a three-step approach, that is, sensor error correction, localised filtering and sensor fusion, providing improved

positioning and uncertainty in the data. Furthermore, the proposed architecture is tested using Hardware-in-the-Loop (HIL) simulation within realistic 3D environments, incorporating variability in GNSS and weather conditions to replicate potential urban canyon scenarios accurately.

The remainder of this chapter is organised as follows: Section two will discuss a literature review on existing techniques. Section three explains the background of some of the techniques used in the chapter. Section four introduces the proposed architecture and describes the individual subdivisions of the system in detail. Section five focuses on experiment setup and data collection methodology. In section six, we present and discuss the experiment results and compare them against existing work and any relevant implications from the analysis. Lastly, section seven presents the concluding remarks and future work.

7.2 Related Work

The state-of-the-art review is structured to cover various techniques and sensors employed. This includes investigating developed GNSS error detection and mitigation approaches, sensor integration/fusion, and integrity monitoring for multi-sensor-based systems.

7.2.1 Multi-Sensor Fusion

Multi-sensor systems provide additional positioning capabilities to either improve accuracy and/or bridge gaps between GNSS outages [106]. For example, sensors used in a multi-sensor system include IMU, barometer, VO, LiDAR, RADAR, etc. In some cases, especially for indoor navigation [107], GNSS is wholly omitted in the navigation process. This reduces the reliance on external systems aiding navigation as they do not require global positioning knowledge. However, for navigation, especially in outdoor situations, global positioning awareness is required in most cases. Therefore, it is essential to fuse/integrate multiple sensors to mitigate sensor-specific errors and improve positioning accuracy and precision.

Recent literature has introduced many sensor integration and fusion approaches that allow for the continuous position estimation of vehicles, especially in safety-critical environments. These include the use of Unscented Kalam Filters [108] for integration of

IMU and GNSS to more complex multi-sensor system fusion architectures. Sensor fusion/integration methods for positioning can be broadly categorised based on processing levels (data, feature, and decision level), fusion algorithms (statistical methods and machine learning), and application areas (indoor, outdoor, and automotive). [109] This classification facilitates a structured approach to selecting appropriate fusion techniques, ensuring enhanced accuracy, reliability, and efficiency in diverse positioning systems. For this chapter, we focus on data and feature based approaches for outdoor use. Additionally, sensor fusion architectures can vary in terms of their processing methods, including centralised or federated systems [110]. Centralised sensor fusion systems aggregate and process data from all sensors simultaneously for navigation decisions, while federated systems distribute the data processing across multiple subsystems before combining the results. An illustration of this is shown below.

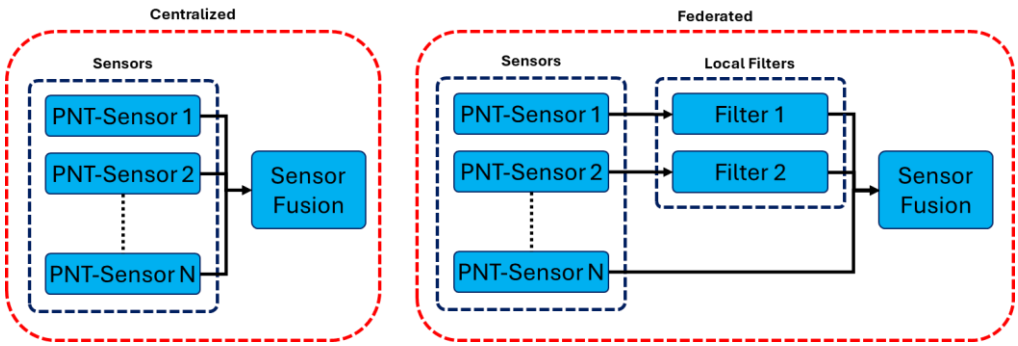


Figure 7-1 Illustration of centralized (left) and federated(right) fusion architectures.

Paper [111] introduces a fast point cloud registration method using Voxel-SIFT for efficient LiDAR registration, overcoming the limitations of traditional federated filters in adapting to dynamic information distribution. It also proposes a hybrid federated filter that dynamically updates information distribution factors for real-time global optimal estimates, utilizing a weighted least square method for improved accuracy and efficiency. Paper [112] proposes an improved adaptive federated Kalman filtering algorithm integrating INS, GNSS, and VNS, with INS as the reference. This algorithm increases fault tolerance through residual chi-square and sliding-window averaging for fault detection and introduces an adaptive sharing factor to dynamically adjust the sub-filters accuracy.

Both papers, however, may not efficiently handle the temporal dynamics and complex dependencies in navigation data, limiting their ability to predict future states or adapt to new environments.

7.2.2 Integrity Monitoring

The authors of the paper [113] introduce a multi-sensor fusion approach combining low-cost GPS, MEMS IMU, and digital compass data, starting with a noise reduction step using an Empirical Mode Decomposition-based filter for more accurate inertial sensor data. It then proposes a hybrid strategy integrating a Least-Squares Support Vector Machine-based NARX model with a Kalman filter for improved modelling and prediction of INS position errors during GPS outages. However, some disadvantages of this system include vulnerability to cumulative errors from each sensor type, especially in challenging environments where GPS signals are weak or obstructed, leading to reduced reliability of the fused data. Additionally, the complexity of the fusion algorithm, particularly when integrating models like LSSVM-NARX with Kalman filters, can introduce difficulties in accurately assessing and maintaining the integrity of the navigation solution across diverse operational scenarios. Additionally, no protection-level estimate is provided for the whole system.

To address this issue, the author of paper [114] presents a multi-level autonomous integrity monitoring method for a resilient fusion navigation system utilising multiple PNT sources. It features a navigation source fault detection model for identifying and isolating faulty sources and a system-level verification model for real-time validation of isolated sources before reintegration. It enhances sensor-level fault detection with a dimension-expanding matrix to address detection delays in multiple faulty sensors, proposes a dynamic monitoring structure for integrity risk allocation, and calculates the system's protection level, demonstrating improved fault detection and recovery capabilities through simulation tests with INS/GNSS/LBL/USBL sources.

However, the multi-level autonomous integrity monitoring method needs the neural network's ability to learn from data and adapt to new patterns, making it less flexible in dynamic environments. Unlike ANNs, this method does not inherently improve with exposure to more data, potentially limiting its effectiveness in complex navigation scenarios. Therefore, it would be beneficial to implement machine learning networks.

However, typical ANNs cannot predict the uncertainty in data. Additionally, simple ANNs do not take advantage of temporal dependencies in the data critical to navigation. Therefore, a federated system with several steps, including correcting individual sensor errors and taking advantage of temporal machine learning capabilities with the ability to predict uncertainty in the data, would be ideal for an integrity-based fusion system.

7.2.3 Uncertainty-based deep learning approaches

Recent advancements in machine learning have introduced uncertainty-aware models, such as Bayesian LSTM, which extend traditional LSTM by incorporating a probabilistic framework for uncertainty estimation [115]. Unlike conventional LSTM models, which output a single deterministic value, Bayesian LSTM can provide both a position estimate and an uncertainty interval. This feature can be particularly valuable for sensor fusion in navigation systems, where uncertainty quantification is critical for tasks such as integrity monitoring and PL calculation. To the knowledge of the author, Bayesian-LSTM has not been utilised in PNT for multi-sensor fusion.

Traditional approaches, such as KF and its variants (EKF and UKF), rely on linear or approximated non-linear models to fuse GNSS and INS data as stated in earlier chapters. Bayesian LSTM, in contrast, directly learns these relationships from data, without the need for explicit modelling of system dynamics. This makes it particularly effective in scenarios with non-linear error patterns, such as multipath interference or GNSS outages.

Several other deep learning models have also been proposed in the literature for probabilistic modelling, including Bayesian Neural Networks (BNNs) [116] [117] and Gaussian Processes (GPs) [118]. Bayesian Neural Networks apply Bayesian inference to standard feedforward neural networks, enabling them to model uncertainty in their predictions. However, BNNs can be computationally expensive to train, due to the need to sample from the posterior distribution over weights. The same is true for Bayesian-LSTM. Methods like variational inference and Monte-Carlo dropouts can aid with this issue. However, Bayesian-LSTM can utilise model uncertainty in time-series data with temporal dependencies whilst BNNs are not suited for this application. Gaussian Processes offer a non-parametric approach to probabilistic modelling and are known for their ability to provide uncertainty estimates. While effective for smaller datasets, GPs

typically struggle with scalability, especially when applied to large-scale sensor fusion tasks.

The unique combination of temporal modelling, uncertainty quantification, and data-driven learning makes Bayesian LSTM an ideal choice for multi-sensor fusion in complex environments.

7.3 Proposed Architecture

7.3.1 Overview

The system design of the proposed scheme is illustrated in Figure 7-2. The structures begin by dividing the sensor data into three distinct processing streams. Initially, GNSS data, encompassing positional estimates and observables, is directed towards the GNSS Non-Line-of-Sight (NLOS) classification and removal phase. In this phase, GNSS observables are employed to categorise signals into three classifications: Line-of-Sight (LOS), multipath (simultaneous presence of LOS and NLOS signals), or purely NLOS. Following signal classification, NLOS signals are then excluded. The subsequent phase involves prioritising signals based on their contribution to Geometric Dilution of Precision (γ_i) values. Afterwards, the GNSS position is calculated using the least-squares method, producing North-East-Down (NED) coordinates.

In parallel, Inertial Measurement Unit (IMU) data, including acceleration and angular rotation data, are utilised in conjunction with the Direction Cosine Matrix (DCM) to calculate the NED position. However, as previously noted, this approach results in positional drift over time. To mitigate this, a GRU-based Inertial Navigation System (INS) error correction algorithm is utilised to estimate the position error. The prediction is then used to correct the INS position estimate to yield a corrected position prediction. Corrected INS position and GNSS data are combined with a local EKF to provide continuous position estimates, even in GNSS outages.

Furthermore, Visual Odometry (VO) position data is generated, as explained in the following sections, and then combined with barometric data to derive a three-dimensional position estimate for the VO system.

Finally, the data from both streams (VO/Barometer and EKF) are fused using a Bayesian Long Short-Term Memory (LSTM) algorithm. This algorithm not only yields the fused position estimate but also quantifies the uncertainty in the data, represented as a distribution of potential position estimates for each NED position timestep. This uncertainty is critical in calculating the PL, which is then assessed against the AL set by the Mission Requirements. Such an assessment aids controllers in understanding mission dynamics and, if necessary, altering operational modes. However, the specifics of different mode selections are beyond the scope of this chapter.

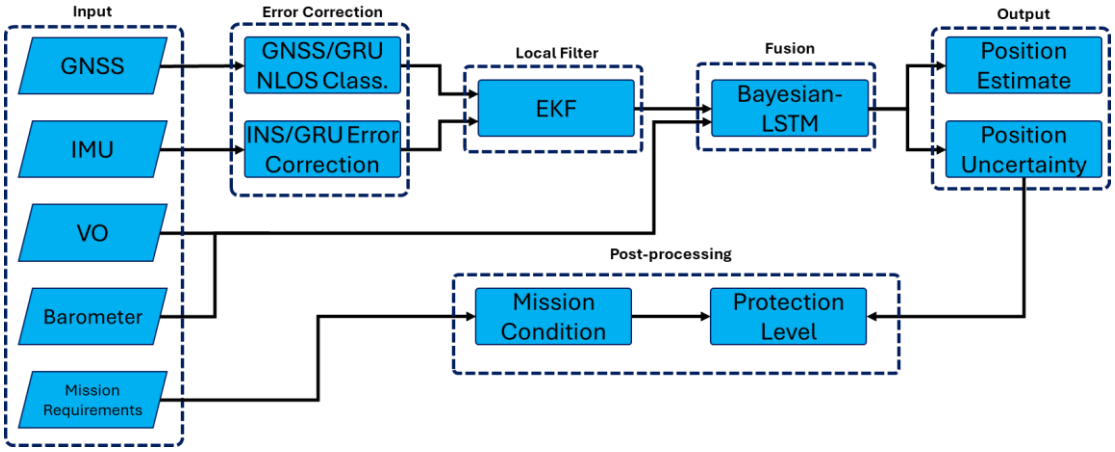


Figure 7-2 Proposed high-level overview of the robust multi-sensor federated scheme.

In the proposed architecture, the federated processing system ensures robustness by integrating independent data streams, each contributing independently to the overall position estimate. By being able to reduce errors using dedicated processes for each stream, it improves the overall system accuracy when these streams are combined. This separation also allows each subsystem to function independently, reducing the risk of total system failure due to the degradation of any single data source. This, however, does come at the cost of increased complexity.

VO is particularly valuable in this federated setup because it operates independently of GNSS signals. It uses image-based motion estimation, making it immune to GNSS-specific errors such as multipath interference or complete signal loss. In environments with unreliable or obstructed GNSS signals, VO can aid in providing positional estimate, ensuring continuity in section with reduced GNSS performance.

The barometer adds another layer of independence by providing altitude information that is unaffected by GNSS or VO conditions. This makes it a critical component for accurate vertical positioning, particularly in GNSS-denied environments or situations where changes in elevation are significant. By keeping the processing of barometric data distinct, the federated system can incorporate altitude readings without relying on GNSS measurements for correction, ensuring greater resilience to signal degradation.

Therefore, by incorporating these features, with stochastic machine learning provided by the Bayesian-LSTM, the proposed system should aid in improving integrity and robustness when navigating in conditions such as urban canyons.

7.3.2 Feature-based Monocular Visual Odometry with ORB

The integration of positioning data in this study is achieved through the conjunction of the Visual Odometry (VO) algorithm and a MEMS barometer sensor. The VO algorithm is a visual-based technique extensively used in robotics to estimate user motion from a sequence of images, particularly in scenarios where a GNSS solution is not feasible. VO algorithms are categorised into two main types: appearance-based and feature-based. A discussion of the systems can be found in paper [119], which the following methodology for NED positioning is based on and a recap is provided here. The Intensity Centroid (IC) approach is used to find the orientation vector, where $m_{pq}$is the moment of a patch, C the IC, and θ the orientation:

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \tag{7.1}$$

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right) \tag{7.2}$$

$$\theta = atan2(m_{01}, m_{10}) \tag{7.3}$$

Subsequently, a steered BRIEF [120] (Binary Robust Independent Elementary Feature) descriptor is employed for matching descriptors between two frames as defined as:

$$g_n(p,\theta) = f_n(p) | (x_i, y_i) \, \epsilon S_0 \tag{7.4}$$

For this purpose, the FLANN [121] (Fast Library for Approximate Nearest Neighbours) the algorithm is utilised due to its real-time feature-matching performance in scenarios

with numerous features, in contrast to the BFmatcher algorithm. The next phase involves estimating the camera's motion, rigidly attached to the UAV, through one of three methods: 2D to 2D, 3D to 3D, or 3D to 2D. Given a single monocular camera, a 2D-2D method is adopted. This method requires the essential matrix to extract the camera's ego motion, which is defined as follows:

$$E = [t]_x R \tag{7.5}$$

Here, R is the rotational matrix and t the translational vector. The estimated essential matrix enables the extraction of the rotational matrix and the translation vector. Triangulation is then applied to isolate a viable solution for the camera's ego-motion. A RANSAC (Random Sample Consensus) algorithm is implemented to enhance the estimated trajectory's accuracy. Finally, for precise utilisation of the VO data, a conversion from the camera frame to the navigation frame is executed, as detailed in references [119] and represented in the following equation:

$$\begin{bmatrix} p_{t,N}^n \\ p_{t,E}^n \\ 1 \end{bmatrix} = \lambda K^{-1} [T_{camera}^{NED}; \quad T_{camera}^{NED} r_{nc}^n]^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{7.6}$$

In this equation, T_{camera}^{NED} represents the transformation from the camera to the navigation frame, K is the intrinsic camera matrix, r_{nc}^n is the position of the camera in the navigation frame, and λ is the scale factor. The barometer is used in addition to the VO algorithm to provide altitude information, addressing the z-axis instability issues encountered with the monocular camera. Consequently, the N and E positioning coordinates are derived from the monocular camera using the VO algorithm, and the D coordinate from the barometer. The output from the VO and barometer is then fused with the NED positioning output from the GRU model, thereby enhancing the INS output.

7.3.3 Loosely-coupled local filter EKF for GNSS/INS integration

The Extended Kalman Filter (EKF) in this architecture is designed to integrate the corrected INS/GRU data with GNSS/GRU position data. A loosely coupled approach is used to achieve the architecture's modularity and the implementation's straightforwardness. The state vector for the EKF, denoted as x_k, encapsulates the UAV's

estimated position in the North-East-Down (NED) coordinate frame. This is expressed as:

$$x_k = \begin{bmatrix} P_{xN} \\ P_{yE} \\ P_{zD} \end{bmatrix} \tag{7.7}$$

Where P_xN, P_xE and P_xD represent the positions derived by both GNSS and INS. The EKF begins with an initialisation step, setting the UAV's base position in the NED frame alongside the initial covariance matrix. This step is crucial for establishing a baseline for subsequent predictions and corrections. The prediction step of the EKF, aimed at forecasting future states, involves two main equations:

$$\hat{x}_k^- = g(\hat{x}_{k-1}^-, u_{k-1}) \tag{7.8}$$

$$\hat{x}_k^- = F_{k-1}\hat{x}_{k-1}^- \tag{7.9}$$

Where $\hat{x}_{k-1}^-$represents the prior estimated UAV position, u_{k-1} is the control input and F_{k-1} is the dynamic covariance matrix. The priori covariance matrix P_k^- is updates as:

$$P_k^- = F_{k-1}P_{k-1}F_{k-1}^T + G_{k-1}Q_{k-1}G_{k-1}^T \tag{7.10}$$

incorporating the noise covariance matrix G_{k-1} and the process noise covariance matrix Q_{k-1}.

In the update or correction step, the dimensions of the measurement matrix are determined by the number of states and measurements. The measurement residual, or innovation, is then calculated using:

$$y_k = z - h(\hat{x}_k^-) \tag{7.11}$$

where $y_k$is the measurement residual, z the observation vector, and $h(\hat{x}_k^-)$ the measurement function based on predicted states. The Kalman gain, K_k, is calculated to assess the accuracy of the predictions. It is defined as:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_{k-1})^{-1} \tag{7.12}$$

Where H_k is the measurement matrix and R_{k-1} is the measurement covariance matrix. A low value of K_k indicates high confidence in the predicted states, while a higher value

suggests less confidence. Finally, the predicted state vector and covariance matrix are updated as:

$$x_k = \hat{x}_k^- + K_k\left(z - h(\hat{x}_k^-)\right) \tag{7.13}$$

$$P_k = (I - K_k H_k)P_k^- \tag{7.14}$$

7.3.4 Bayesian-LSTM implementation for multi-sensor fusion

Bayesian Long Short-Term Memory (LSTM) networks integrate the principles from Bayesian statistics into the LSTM architecture [122]. Bayesian LSTMs differ from traditional LSTMs by incorporating uncertainty in their predictions. There are various methods to implement Bayesian inference in LSTMs, including variational inference. [123], Markov Chain Monte Carlo (MCMC) methods [124], and Monte Carlo (MC) Dropout [125]. Each method offers a different approach to quantifying and managing uncertainty in the model's predictions.

Among these methods, Monte Carlo Dropout is a prevalent and practical technique for implementing Bayesian principles in LSTM networks. MC Dropout involves applying dropout not only during the training phase but also during the prediction phase. MC Dropout effectively creates a Bayesian approximation by enabling dropouts during prediction, allowing the network to generate a distribution of outputs for a given input.

The process involves running the same input through the network multiple times, each time with a different dropout pattern, and then aggregating these results to estimate the predictive distribution. This method is computationally less intensive than other Bayesian approaches like MCMC, making it more feasible for practical applications. Additionally, MC Dropout seamlessly integrates with existing network architectures, requiring minimal modifications to the standard LSTM structure [126]. The process of randomized dropouts during predictions and their effects are illustrated in Figure 7-3.

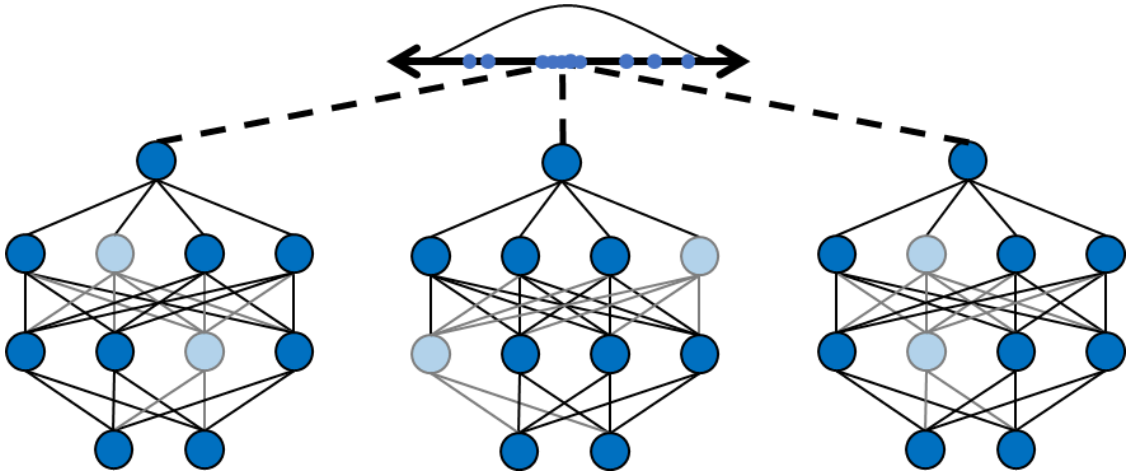


Figure 7-3 Illustration of Monte Carlo dropouts to produce observation distribution.

Ideally, it would be possible to compute the posterior distribution of the weights given the data, denoted as $P(W|X,Y)$ where Wrepresents the weights, and X and Y represent the input and output data, respectively. This is typically intractable, so we approximate it using a variational distribution $q(W)$. Therefore, sampling sets of weights $\{W_i\}_{i=1}^{N}$ from the approximate posterior $q(W)$ is an approach that is used. Each set of weights W_i represents a possible realisation of the network parameters. For a given input x, the network's output with the i-th set of sampled weights is computed as $y_i = LSTM(x; W_i)$. The final predictive distribution for the output is approximated by aggregating the outputs from all sampled models, which can be represented as:

$$P(y|x,X,Y) \approx \frac{1}{N}\sum_{i=1}^{N} P(y|x,W_i) \tag{7.15}$$

The study employs a Bayesian Long Short-Term Memory (LSTM) network, underpinned by Monte Carlo Dropout techniques, to generate a combined position estimate and its corresponding uncertainty to integrate data from multiple enhanced processing streams. The network input comprises INS/GNSS EKF NED estimate, and the VO/Barometer NED. The output is characterised by the fused mean NED position and the estimate's standard deviation. The inclusion of Monte Carlo Dropouts facilitates the generation of a distribution comprising 100 potential position estimate points, from which the mean, standard deviation, and covariance are extracted. The temporal mean of these outputs serves as the optimal position estimate. Concurrently, the standard deviation, in conjunction with the covariance, is utilised to compute the Protection Level (PL) as per

the methodology described in section X. The PL is used in the assessment against the mission's Alert Limit (AL). Should the PL surpass the AL, this anomaly will be flagged to the mission controller. Using Bayesian-LSTM provides a way to improve the integrity aspect for positioning, which is critical in urban environment conditions.

7.3.5 Integrity Analysis

The International Civil Aviation Organization (ICAO) provides standardised definitions for key parameters in aviation and satellite-based navigation systems, such as the Protection Level (PL) and Alert Limit (AL), encompassing the Horizontal Protection Level (HPL) and Vertical Protection Level (VPL). These measures are essential for ensuring the safety and reliability of navigation systems. The Horizontal Protection Level, as defined by the ICAO, is a statistical measure representing the radius within which the true position of an aircraft is expected to be, with a certain probability. This measure is crucial in assessing the integrity of a navigation system. The HPL is calculated using the formula:

$$HPL = K \times \sigma_{HPE} \tag{7.16}$$

where K is a scaling factor derived from the required level of integrity, and σ_{HPE} is the standard deviation of the Horizontal Position Error (HPE). The HPE measures the discrepancy between the estimated and the actual horizontal position of the aircraft. σ_{HPE} is calculated as follows [127]:

$$\sigma_{HPE} = \sqrt{\frac{\sigma_E + \sigma_N}{2} + \sqrt{\left(\frac{\sigma_E - \sigma_N}{2}\right)^2 + P_{NE}^2}} \tag{7.17}$$

Where σ_E is the standard deviation in the East direction (m), σ_N is the standard deviation in the North direction and P_{NE} is the North and East covariance.

The Vertical Protection Level, as per ICAO's definition, provides a bound within which the true vertical position of the aircraft (altitude) is expected to lie, with a specified probability. This level is key to ensuring vertical navigational integrity. The VPL is computed as:

$$VPL = K \times \sigma_{VPE} \tag{7.18}$$

In this formula, K is again a scaling factor related to integrity, and σ_{VPE} is the standard deviation of the Vertical Position Error (VPE). The Alert Limit is a predefined threshold that determines the acceptability of the position estimates provided by the navigation system. If the PL (either HPL or VPL) exceeds the Alert Limit, it indicates a potential compromise in the system's integrity, and the position information should not be used for critical navigation decisions. The AL is set based on operational requirements and represented as a constant value.

These parameters are typically evaluated using the Stanford diagram, a graphical representation that plots the PL (either HPL or VPL) against the corresponding position error (HPE or VPE) as shown in Figure 7-4. This diagram depicts the AL as a horizontal line. The Stanford diagram facilitates a quantitative system integrity assessment by visually mapping the relationships. The distribution of data points across the diagram quadrants provides empirical evidence of the system's operational status.

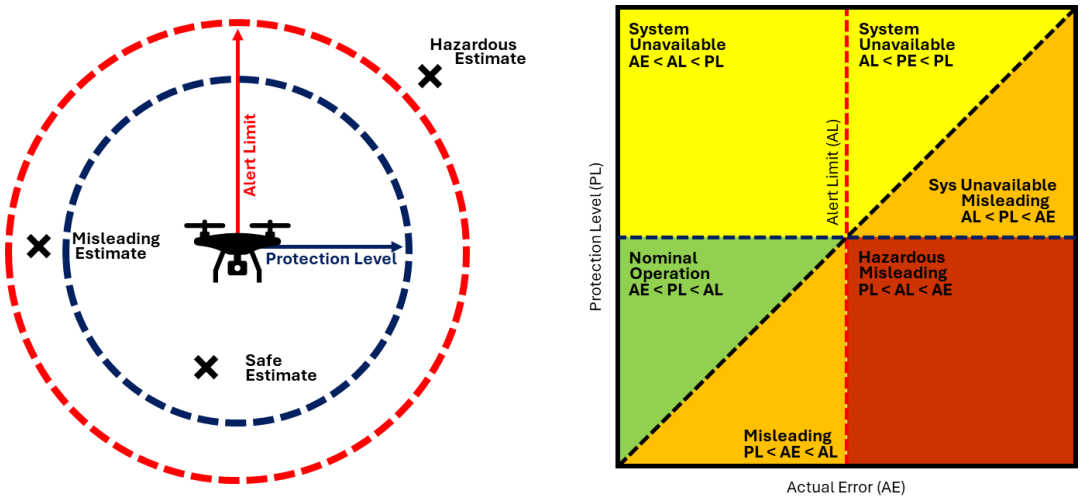


Figure 7-4 Illustration of integrity regarding a) comparison of protection level and alert limit and b) Stanford diagram.

7.4 Simulation Overview and data collection

7.4.1 Scenario Overview

Unreal Engine was utilised in combination with AirSim to evaluate the proposed architecture. This specific simulation integrated real-world photogrammetry data from the city centre of Toulouse into Unreal Engine to create a realistic urban environment.

The trajectory chosen to train and test the fusion algorithm is shown in Figure 7-5. It consists of a 150-meter square area survey trajectory that will equally test left turns, right turns and straight segments in each direction and diagonally. The simulation aimed to test the Visual Odometry (VO) algorithm while also examining the effects of multipath on the Global Navigation Satellite System (GNSS) and the drift caused by the MEMS Inertial Measurement Unit (IMU) over time.

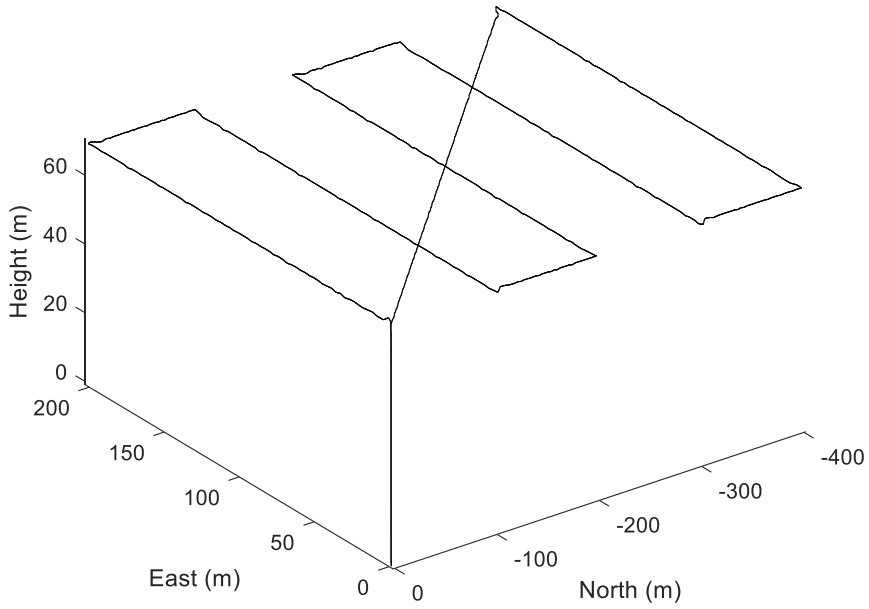


Figure 7-5 Simulation ground truth trajectory

7.4.2 GNSS Condition (GNSS Receiver)

To simulate GNSS conditions in urban canyons, OKTAL-SE Sim3D tool is used in conjunction with Spirent GSS7000. Sim3D uses ray-tracing and material properties of buildings to estimate signal reflection characteristics accurately and therefore helps simulate effects like multipath conditions. The ray-tracing paths are illustrated in Figure 7-6. Adding this layer of complexity to simulations enables a thorough approach to testing the algorithm in a simulated environment and better tests changing conditions which is important from a robustness perspective.

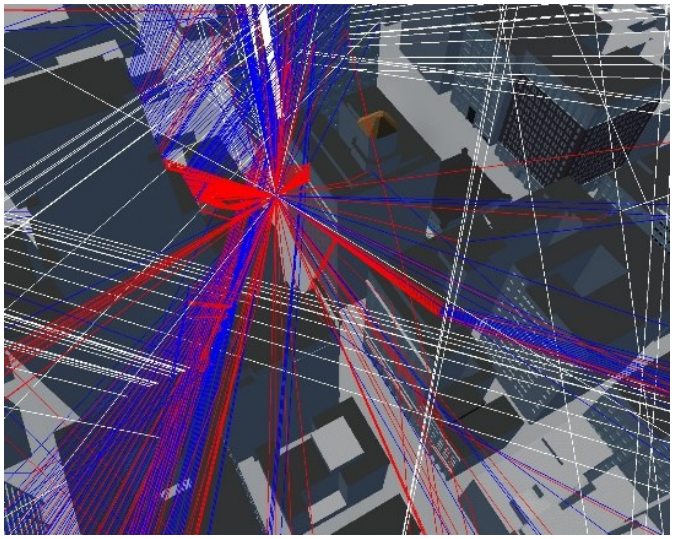


Figure 7-6 Ray-tracing GNSS signals in urban canyon using OKTAL-SE Sim3D

The modelling of urban canyons and their multipath conditions will lead to areas with GNSS outages. This adds a further challenge for the Bayesian-LSTM algorithm to understand conditions better and mitigate for the increased position error. Furthermore, ideally the Bayesian-LSTM would indicate the uncertainty increase in these conditions by increasing the PL estimate. The data generated from the combination of Sim3D and GSS7000 is transmitted as a RF signal to the receiver for generating observables and GNSS position.

7.4.3 Visual Odometry Weather Condition and Camera Calibration

The first scenario involved flying the Unmanned Aerial Vehicle (UAV) during normal daylight at 14:51 local time in Unreal Engine, and the second involved a flight at 18:00. Additionally, two other conditions were considered, performing the trajectories under fog and dust conditions, as depicted in Figure 7-7.

Figure 7-7 Scenario weather condition setup in Toulouse, France with a) afternoon b) evening c) dust d) fog [119]

Before testing the fusion architectures, the monocular camera mounted on the UAV and pointing downwards was configured in AirSim with an image width of 752 pixels and an image height of 480 pixels. To calibrate, a chessboard was introduced in Unreal Engine for calibration purposes. This involved finding the intrinsic matrix defined as K, composed of focal lengths (f_x and f_y) and optical centers (c_x and c_y). The chessboard featured 10 rows and 10 columns of alternating white and black squares, as shown in Figure 7-8. Each square measured 2 meters in both length and width. The UAV then performed a small square trajectory over the chessboard at a cruise altitude of 70 m, recording all camera frames. These 220 frames, extracted from Unreal Engine, were processed using MATLAB's 'Camera Calibrator' tool to obtain the intrinsic matrix. This matrix was then used to enhance the realism of the VO algorithm.

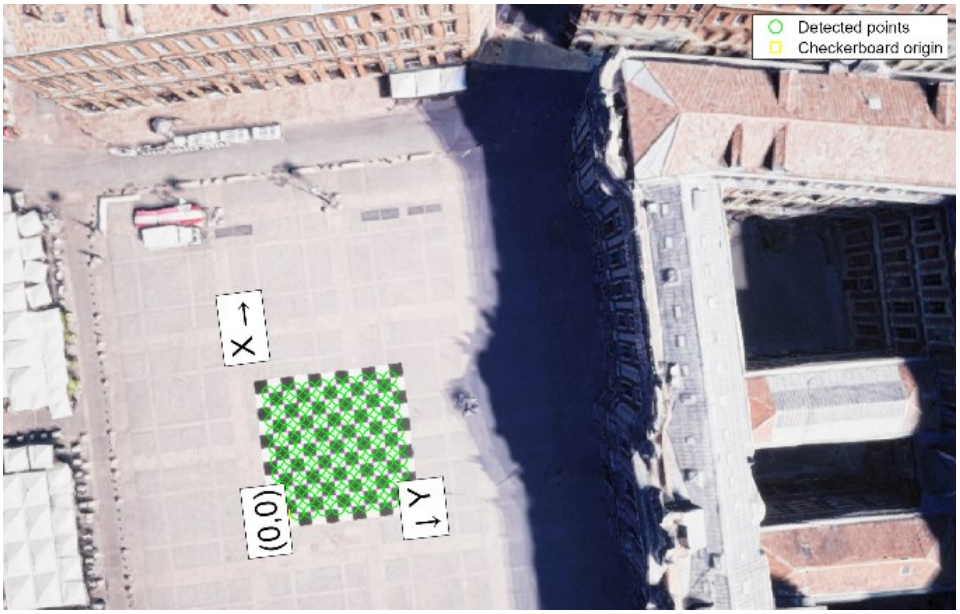


Figure 7-8 Visual sensor calibration step

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 378.6062 & 0 & 376.2584 \\ 0 & 378.6560 & 240.4231 \\ 0 & 0 & 1 \end{bmatrix} \tag{7.19}$$

7.4.4 Data Collection

Data from a range of sensors, including an accelerometer, gyroscope, GNSS, Visual Odometry (VO), and barometer, are collected and formatted for our UAV study. The accelerometer captures specific force data in the UAV's body frame, and the gyroscope tracks the angular velocity. Both sets of data are stored in CSV format.

The GNSS data, generated from the C009-F9P Ublox board and accessed via u-center software, includes Satellite ID, Pseudorange, Doppler Shift, Carrier Phase, C/N0, Elevation, Azimuth, and Satellite Position. This is recorded for each timestep and for every satellite visible to the receiver. The obtains data from multiple constellations - GPS, Galileo, and GLONASS - enhancing the positioning accuracy. Each timestep's GNSS data is enriched with a receiver-based position estimate.

VO data is converted to a navigation frame and combined with barometric measurements to achieve 3D positioning. This integration allows for tracking the UAV's vertical movement, with all data referenced to the navigational frame and stored in CSV format.

7.4.5 Machine Learning Training Phase

Each machine learning model within our system was trained sequentially, starting with independent training of individual models using raw data inputs aligned with their target outputs derived from ground truth data. This sequential training process ensures that each model is optimised to effectively process and respond to its specific input data type. After achieving convergence—where no further significant improvements were noted—the models, except the Bayesian LSTM, completed their training phases. Following this, the outputs of these trained models served as inputs to the Bayesian LSTM. This model then underwent its own training phase, incorporating outputs from the pre-trained models to enrich its training data. This hierarchical training strategy, complemented by thorough hyperparameter optimisation, ensures that each subsequent model in the sequence leverages the distilled knowledge and outputs from its predecessors, thus enhancing the overall system's predictive capabilities and accuracy.

7.4.5.1 INS error correction regression GRU

The training and optimisation of the machine learning algorithm were achieved by employing random searches to determine the best configuration, including the ideal number of hidden units, GRU layers, and the most effective activation function. This optimisation method, which only used a subset of the entire dataset, aimed to balance reducing computational demands with enhancing predictive accuracy. The architecture includes an input layer receiving features such as accelerometer data, angular rate, and orientation in matrix columns, with time steps and sensor data in the rows. The model comprises two hidden layers with 256 and 128 units, respectively, each layer having an unrolled length of 5 to capture sequence length input from each batch and includes a dropout rate of 15%. A tanh activation function was employed to improve non-linear correlation further. A dense layer connects all nodes, yielding the neural network's regressed output. ADAMAX optimiser was utilised for tuning weights and biases, starting with a learning rate of 0.001. The mean squared error loss function was applied to train the algorithm. The trained weights and biases were saved in a .h5 file for testing.

The data is first normalised during testing using the training parameters and then processed through the algorithm. The output is recorded and compared with the actual position error, calculated from the difference between ground truth and INS position

predictions. The training and testing split used for the INS/GRU architecture are 80/20, where the batch sequence was randomised but not the sequence in each batch to maintain the temporal relationship.

7.4.5.2 GNSS multipath classification GRU

Like the above, the architecture was first optimised using random search to determine the ideal number of hidden units, GRU layers, and the most effective activation function. However, unlike the regression architecture above, a classification structure was used to classify signals. Here, the input includes the satellite ID, pseudorange, Doppler shift, carrier phase, C/N0, elevation, azimuth, and satellite position as the feature input. The data is standardised using a z-score where the mean equals zero and a standard deviation of one. One layer with 250 hidden units and a dropout rate of 30% is used. A softmax layer is added and a cross-entropy loss function is used to train the classification algorithm and provide an output of which signal is NLOS, Multipath or LOS. The labels are supplied by using the ground truth data from the GSS7000 simulator.

Similar to INS/GRU, testing is carried out with a train/test split of 80/20. The output of test data is recorded and is compared to against the true labels of the data.

7.4.5.3 Bayesian-LSTM fusion and uncertainty prediction

Following the individual processing of each sensor's data, the information is fused using a Bayesian-LSTM network. The data fed into this system is organised as a matrix, including input features such as GNSS Position NED, INS Position NED, and INS/Barometer/VO EKF Position NED. Each feature is characterised by a sequence length of 20 and is processed in batches of 64. The Bayesian-LSTM structure comprises two layers with 128 and 64 cells, respectively, each followed by a RELU activation function layer. A fully connected layer is integrated at the end, outputting a three-dimensional vector representing NED frame positioning. Additionally, to account for uncertainty in predictions, a Monte Carlo dropout layer is incorporated post-LSTM layer, facilitating randomised dropouts during the algorithm's prediction phase. This architecture was also optimised using random search to determine the ideal number of hidden units, GRU layers, and the most effective activation function.

After the training phase, the model generates 100 predictions for each time point which are then utilized to compute the mean and standard deviation of each prediction. Additionally, the covariance for each batch of predictions is also calculated for later use in determining the PL.

7.5 Results & Discussion

7.5.1 Training and inference speed

The training of the Bayesian LSTM system is designed to optimise its predictive capabilities for uncertainties in navigation systems. This training consists of multiple machine learning algorithms (as discussed above), and therefore, the time reported is the cumulative training time of all systems. Inference speed is gauged based on the time it takes to produce one prediction cycle ($Pred_t$).

	Training Speed (min/sec)	Inference Speed (ms)
Proposed	224 min 12 seconds	12.06
RNN [72]	11 min 21 seconds	2.32

The results indicate that the proposed system requires significantly longer training times than the RNN model referenced in publication [36]. Although its inference speed is slower, it remains suitable for real-time applications. This slower speed reflects a deliberate trade-off, prioritizing detailed uncertainty prediction and robust system performance over rapid processing. These characteristics justify using the Bayesian LSTM in applications where accuracy and depth of predictive insight are paramount.

7.5.2 Sensor Fusion Performance

Table 7-I presents results from several stages in the positioning algorithm, each showcasing varied levels of accuracy in different directions and under diverse environmental conditions. The GNSS receiver output is used as a comparison point for the other data points. It exhibits mean errors in the North, East, Horizontal, and Vertical directions at 22.32 m, 30.48 m, 37.78 m, and 23.34 m, respectively. The table shows the inherent limitations of standalone GNSS systems, particularly in complex environments where signal degradation and multipath cause an increase in position ambiguity. Implementing the GNSS/GRU algorithm ,by incorporating NLOS classification and ranking, effectively reduces mean errors to 9.48 m, 6.54 m, 11.52 m, and 4.83 m in the respective directions. The GRU plays an important role here, particularly in handling

time-series data where the temporal patterns in GNSS signals improve the classification accuracy. The EKF integration, which combines GNSS/GRU with INS/GRU data, further improves positioning accuracy, leading to mean errors of 7.84 m, 4.43 m, 9.01 m, and 3.50 m. The integration of INS data, corrected using GRU insights, provides a more reliable and dynamic positional fix, particularly in environments where GNSS signals are compromised. The robustness of the VO combined with barometer readings is tested under various weather scenarios—Afternoon, Evening, Fog, and Dust. The consistency of vertical errors at 1.03 m across all conditions is expected due to the use of a barometer as a way to estimate altitude.

The culmination of individual systems is represented by the Bayesian-LSTM fusion, achieving improved positioning performance. With horizontal errors not exceeding 0.62 m and vertical errors around 0.30 m, the Bayesian-LSTM model demonstrates its importance in improving accuracy. This method's resilience to changing conditions, both in multipath urban environments and different weather conditions, showcases its ability to effectively integrate and analyse the time-series data from both the EKF and VO/barometer systems. The Bayesian approach provides a probabilistic framework that estimates the uncertainties, while the LSTM component provides the capability to learn from sequential data, making the fusion method ideal for cases in critical safety conditions.

Table 7-I. Comparison of absolute mean position error for various sensor/machine learning configurations and weather conditions

	Absolute Mean Error (m)			
	North	East	Horizontal	Vertical
Receiver	22.32	30.48	37.78	23.34
GNSS/GRU	9.48	6.54	11.52	4.83
EKF	7.84	4.43	9.01	3.50
EKF with VO (afternoon)	3.28	4.24	5.36	3.50
VO (Afternoon)/Barometer	4.44	5.59	7.14	1.03
VO (Evening)/Barometer	4.48	6.03	7.51	1.03
VO (Fog)/Barometer	4.46	6.05	7.52	1.03
VO (Dust)/Barometer	5.48	6.65	8.62	1.03
Bayesian-LSTM (Afternoon)	0.45	0.23	0.50	0.30
Bayesian-LSTM (Evening)	0.45	0.24	0.52	0.34
Bayesian-LSTM (Fog)	0.45	0.24	0.51	0.29
Bayesian-LSTM (Dust)	0.56	0.27	0.62	0.31

Table 7-II provides an analysis of the standard deviation in various positioning methods, reflecting their consistency and precision under different conditions. The GNSS receiver shows standard deviations of 20.34 m in North, 19.93 m in East, 28.48 m in Horizontal, and 15.53 m in Vertical. These higher values suggest a notable variability in the GNSS receiver's measurements, indicative of its susceptibility to environmental factors and signal degradation. A marked improvement in consistency is observed with the GNSS/GRU algorithm. The standard deviations are significantly reduced, indicating enhanced stability in the measurements due to the algorithm's ability to effectively handle NLOS signals and improve signal quality.

The EKF method exhibits further reduction in standard deviations, upholding its capability in providing a more consistent and stable estimation of position by fusing GNSS/GRU and INS/GRU data. This suggests improved handling of data inaccuracies and an increase in performance even in challenging conditions. The VO combined with barometer readings maintains a consistent vertical standard deviation of 2.06 m across different weather conditions, highlighting its reliability in altitude measurement. However, variations in horizontal standard deviations under different conditions (Afternoon, Evening, Fog, Dust) indicate a dependency of the VO system's performance on environmental factors. The reduced standard deviations in the Bayesian-LSTM fusion method across all conditions present the case for the fusion method.

Table 7-II. Comparison of absolute standard deviation position error for various sensor/machine learning configurations and weather conditions

	Standard Deviation Error (m)			
	North	East	Horizontal	Vertical
Receiver	20.34	19.93	28.48	15.53
GNSS/GRU	15.26	9.68	18.07	4.43
EKF	10.28	4.48	11.21	1.95
EKF with VO (Afternoon)	8.28	5.01	10.96	1.98
VO (Afternoon)/Barometer	6.98	5.97	9.18	2.06
VO (Evening)/Barometer	7.04	6.02	9.26	2.06
VO (Fog)/Barometer	7.01	6.40	9.49	2.06
VO (Dust)/Barometer	7.98	6.43	10.25	2.06
Bayesian-LSTM (Afternoon)	0.38	0.20	0.42	0.21
Bayesian-LSTM (Evening)	0.38	0.20	0.43	0.22
Bayesian-LSTM (Fog)	0.38	0.21	0.43	0.20
Bayesian-LSTM (Dust)	0.43	0.21	0.48	0.24

Table 7-III analyses the 95th percentile errors from several stages in the positioning algorithm, offering insights into the upper bounds of error likely to be encountered under

different operational scenarios. The GNSS receiver exhibits the highest 95th percentile errors across all categories, with 63.00 m in North, 70.34 m in East, 94.73 m in Horizontal, and 54.40 m in Vertical. These figures indicate a significant potential for large errors in extreme cases, reflecting the limitations of standalone GNSS in challenging conditions. There is a notable improvement with the GNSS/GRU algorithm, with the 95th percentile errors dropping to 40.00 m, 25.90 m, 47.66 m, and 13.69 m, respectively. This demonstrates the algorithm's effectiveness in reducing the likelihood of extreme errors, likely due to better handling of signal obstructions and multipath effects.

The EKF fusion method shows further reductions in the 95th percentile errors, suggesting enhanced accuracy and reliability. The errors are reduced to 28.40 m, 13.39 m, 31.43 m, and 7.40 m, indicating the method's ability to effectively combine GNSS/GRU and INS/GRU data for improved positional estimation. The VO system combined with a barometer demonstrates reasonable upper bounds of error under different environmental conditions. The figures are relatively consistent, especially in the vertical measurement (5.15 m), indicating the barometer's reliability. However, variations in the horizontal measurements under different conditions (Afternoon, Evening, Fog, Dust) reflect the influence of environmental factors on the VO system's accuracy.

The Bayesian-LSTM fusion method exhibits the lowest 95th percentile errors in all conditions, indicating a high level of precision and reliability. The errors are remarkably low, with the highest horizontal error at just 1.57 m and the vertical error at 0.79 m. These low values suggest that even in the worst 5% of cases, the Bayesian-LSTM method maintains high accuracy, underscoring its effectiveness in diverse conditions.

Table 7-III. Comparison of position accuracy for various sensor/machine learning configurations and weather conditions

	95th Percentile Error (m)			
	North	East	Horizontal	Vertical
Receiver	63.00	70.34	94.73	54.40
GNSS/GRU	40.00	25.90	47.66	13.69
EKF	28.40	13.39	31.43	7.40
EKF with VO (Afternoon)	19.84	14.26	24.43	7.46
VO (Afternoon)/Barometer	18.40	17.53	25.51	5.15
VO (Evening)/Barometer	18.56	18.07	26.04	5.15
VO (Fog)/Barometer	18.48	18.85	26.50	5.15
VO (Dust)/Barometer	21.44	19.51	29.11	5.15
Bayesian-LSTM (Afternoon)	1.20	0.62	1.35	0.72
Bayesian-LSTM (Evening)	1.21	0.64	1.37	0.78

Bayesian-LSTM (Fog)	1.21	0.66	1.38	0.69
Bayesian-LSTM (Dust)	1.42	0.69	1.57	0.79

Table 7-IV quantifies the improvement in positional accuracy of various methods compared to the standard GNSS receiver, expressed as a percentage. The GNSS/GRU algorithm shows improvements of 37% in North, 63% in East, 50% in Horizontal, and 75% in Vertical directions. This indicates a substantial enhancement over the baseline GNSS receiver, particularly in the East and Vertical measurements. The EKF method exhibits further improvements, with 55% in North, 81% in East, 67% in Horizontal, and 86% in Vertical. These figures suggest a significant increase in accuracy, especially in the East and Vertical directions, compared to the baseline.

The VO system combined with a barometer demonstrates improvements in the range of 66% to 71% in North, 72% to 75% in East, 69% to 73% in Horizontal, and consistently 91% in Vertical across different environmental conditions. These numbers indicate a reliable enhancement in accuracy, with the most significant improvement observed in vertical positioning.

The Bayesian-LSTM fusion method displays the highest improvements across all metrics and conditions, with 98% to 99% in North, East, and Horizontal, and consistently 99% in Vertical. This reflects a near-total enhancement in accuracy compared to the baseline receiver.

Table 7-IV. Relative comparison to receiver for performance improvement of various sensor/machine learning configurations and weather conditions

	Improvement (%)			
	North	East	Horizontal	Vertical
Receiver		Ref		
GNSS/GRU	37%	63%	50%	75%
EKF	55%	81%	67%	86%
EKF with VO (Afternoon)	68%	80%	74%	86%
VO (Afternoon)/Barometer	71%	75%	73%	91%
VO (Evening)/Barometer	71%	74%	73%	91%
VO (Fog)/Barometer	71%	73%	72%	91%
VO (Dust)/Barometer	66%	72%	69%	91%
Bayesian-LSTM (Afternoon)	98%	99%	99%	99%
Bayesian-LSTM (Evening)	98%	99%	99%	99%
Bayesian-LSTM (Fog)	98%	99%	99%	99%
Bayesian-LSTM (Dust)	98%	99%	98%	99%

Figure 7-9 compares the trajectory estimated by a Bayesian-LSTM model against the ground truth, with axes representing the North and East coordinates in meters. The ground

truth path delineates the actual movement, while the Bayesian-LSTM path shows the estimated trajectory based on the model's predictions.

The Bayesian-LSTM model's trajectory closely aligns with the ground truth for the majority of the path, indicating a high level of accuracy in the model's positional estimates. The similarity between the two paths suggests that the Bayesian-LSTM model is effectively capturing the movement dynamics in both the North and East directions.

At turning points in the ground truth trajectory, the Bayesian-LSTM model exhibits slight deviations. These deviations are more pronounced at segments where the trajectory changes direction sharply. Such discrepancies are attributable to the non-linear nature of the movements, which are particularly challenging for models to predict due to the complex dynamics involved. The accelerometer and gyroscope measurements at the IMU level are affected by these non-linearities, leading to the observed deviations in the estimated path.

Despite these challenges, the model maintains a consistent trajectory estimation throughout the plot. The absence of erratic behaviour or extreme deviation from the ground truth path despite changing GNSS conditions in Dust conditions indicate that the Bayesian-LSTM model provides robustness to the estimated positions. The model's performance, especially its ability to closely track the actual trajectory, underscores its potential utility in precise navigation and location tracking applications.

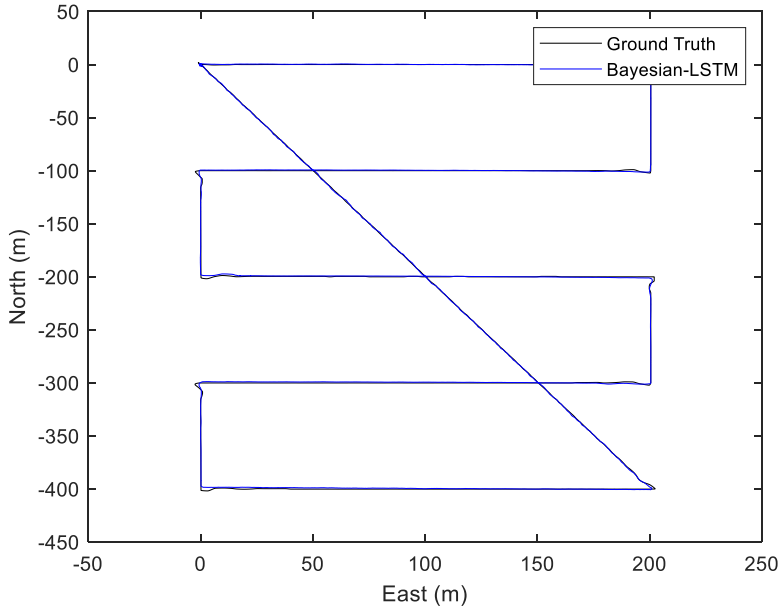


Figure 7-9 Visual position accuracy comparison of horizontal positioning performance between Bayesian-LSTM and ground truth

Figure 7-10 compares the vertical movement over time, contrasting the ground truth height with the height estimated by a Bayesian-LSTM model.

In the graph, both the ground truth and the Bayesian-LSTM estimated heights are plotted against time in seconds. For the vast majority of the timeline, the Bayesian-LSTM estimated height closely matches the ground truth, maintaining a consistent altitude with negligible deviation, as indicated by the overlap of the two lines.

There are two distinct events where a change in height is observed. Initially, both the ground truth and the Bayesian-LSTM model show a sharp increase in height, after which there is a plateau that persists for the majority of the time series. Following this period of stability, a sharp decline in height is registered by both the ground truth and the model at the end of the time series. The model accurately captures the timing and magnitude of these changes, suggesting that the Bayesian-LSTM has effectively learned the pattern of ascent and descent.

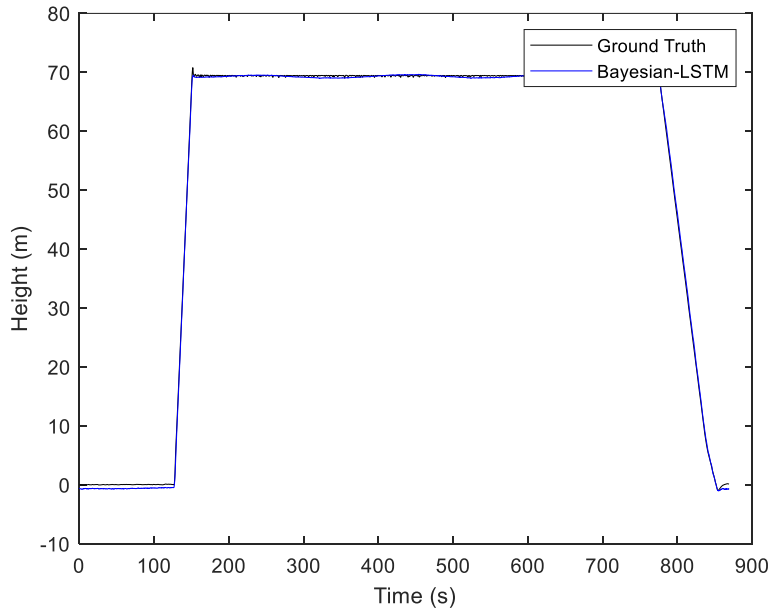


Figure 7-10 Visual position accuracy comparison of vertical positioning performance between Bayesian-LSTM and ground truth

Figure 7-11 displays the Probability Density Function (PDF) of position errors for three estimation methods: the Extended Kalman Filter (EKF), Visual Odometry (VO) in dusty

conditions when used with a barometer, and the Bayesian-LSTM fusion of EKF and VO/Barometer.

In the plot, the EKF method's PDF indicates the widest spread of position errors, with a relatively flat distribution that extends towards larger error values. This reflects a greater variance in the position errors produced by the EKF method, suggesting a lower precision compared to the other methods. This is to be expected as the varying GNSS conditions for the UAV as it navigates these urban canyons reduced overall precision that can be achieved.

The VO (Dust) method exhibits a narrower distribution than the EKF, with a peak that is higher and closer to zero error. This indicates that while the VO method under dusty conditions is prone to errors, it typically maintains a smaller range of error values than the EKF.

The Bayesian-LSTM's PDF shows the narrowest and highest peak, concentrated near the zero-error mark. This demonstrates that the Bayesian-LSTM fusion method achieves a high probability of minimal position errors, outperforming the individual EKF and VO (Dust) methods. The sharp peak suggests that the majority of its position estimates are very close to the actual values, indicating a high level of accuracy and precision.

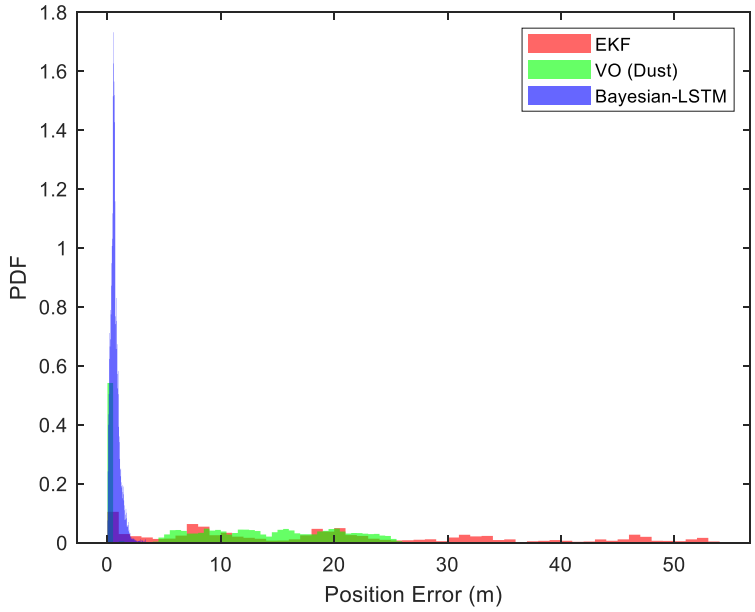


Figure 7-11 Comparison of absolute horizontal position error distribution between EKF, VO and Bayesian-LSTM in presence of dust

Figure 7-12 analysis of horizontal position errors for the three estimation techniques. The EKF method shows a relatively high degree of variability in horizontal error over time, with several spikes going above 50 m indicating moments of significant positional inaccuracy. This could be reflective of the challenges faced by the EKF when dealing with the integration of INS/GRU and GNSS/GRU data, especially when the errors in the inertial measurement unit (IMU) data accumulate over time.

For the VO (Dust) method, the horizontal error demonstrates a steady increase over time. This trend can be attributed to the nature of VO systems, where errors are cumulative and tend to grow as a function of distance travelled, particularly in adverse conditions such as dust, which can obscure visual markers and reduce the accuracy of visual feature tracking. The barometer provides vertical guidance, which does not correct the horizontal drift inherent in the VO method.

The Bayesian-LSTM fusion method displays a distinct pattern, with horizontal errors remaining consistently low throughout the time series. This indicates that the fusion approach effectively compensates for the individual limitations of the EKF and VO methods, leveraging the strengths of both to produce a more accurate and stable estimation of horizontal position.

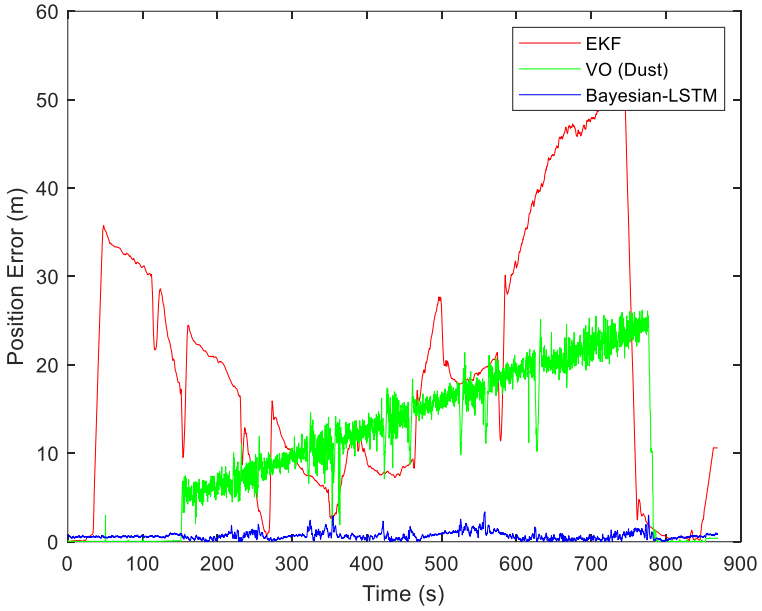


Figure 7-12 Comparison of absolute horizontal position error vs time between EKF, VO and Bayesian-LSTM in presence of dust

Figure 7-13 illustrates vertical position errors over time for the Extended Kalman Filter (EKF), Visual Odometry (VO) in dusty conditions with barometer data, and the Bayesian-LSTM fusion method.

The EKF displays an initial spike in vertical error, which rapidly diminishes and remains relatively low for the duration of the time series. This spike may represent the EKF's initial adjustment period or a response to a specific event that temporarily affected the vertical accuracy.

The VO (Dust) method shows more variability in vertical error compared to the EKF. Despite the incorporation of barometer data, which is generally reliable for vertical measurements, the VO method's error exhibits several peaks throughout the time series. This could be due to the integration challenges of visual data with barometric pressure changes, particularly in dusty conditions that may affect sensor readings.

The Bayesian-LSTM fusion maintains the lowest vertical error throughout the majority of the time series. It demonstrates a significant reduction in error magnitude compared to both the EKF and VO (Dust) methods, with minimal fluctuations. This indicates the effectiveness of the fusion method in leveraging the strengths of both the EKF and VO/barometer data to achieve more accurate vertical positioning.

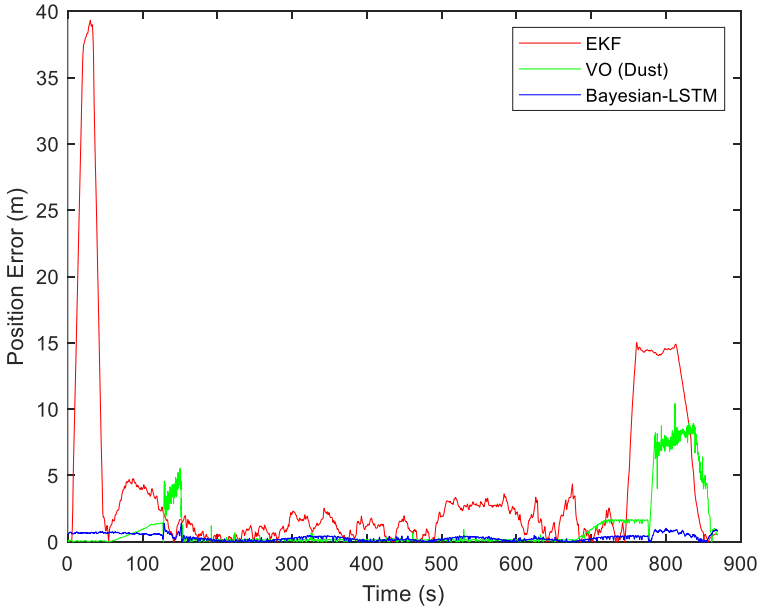


Figure 7-13 Comparison of absolute vertical position error vs time between local EKF, VO and Bayesian-LSTM in presence of dust

7.5.3 Protection Level Estimation Performance

Figure 7-14 is a Stanford diagram, which is used to assess the integrity of a navigation system by plotting the Horizontal Protection Level (HPL) against the Horizontal Position Error (HPE). The diagram is divided into four quadrants, representing different operational states: Nominal Operation, System Unavailable Alarm, Hazardous Misleading Information, and the combination of System Unavailable and Hazardous Misleading Information.

Nominal Operation is indicated by data points lying below the threshold line, where the HPL is greater than the HPE. In this analysis, a substantial majority of the instances (86,385 or 99.38%) fall into this quadrant, indicating that the system is operating normally, and the integrity of the position information is maintained. The predominance of data points in the Nominal Operation quadrant validates the Bayesian-LSTM system's ability to accurately estimate the protection level, ensuring that the HPL consistently encompasses the actual position error. This operational characteristic is crucial for UAV navigation in environments where GNSS signals are compromised and leads to system integrity for vehicles operating in urban canyons.

System Unavailable Alarm is marked by instances where the system correctly identifies that the HPL exceeds an acceptable level, which may compromise position integrity, and thus an alarm is raised to indicate that the system is unavailable. Here, there are 535 instances (0.62%) where the system has correctly alerted that the HPL is too high, suggesting that the system is appropriately self-monitoring and alerting when it cannot guarantee the position's integrity. The majority of System Unavailable alarms are during heading changes or changes in angular velocities recorded by the IMU. Additionally, in these conditions the multipath pattern is changing due to the change in the urban canyon environment. Therefore, this nonlinear change contributes to the increased complexity of the positioning system, leading to an increase in errors and also an increase in the protection level estimate, exceeding the horizontal alert limit threshold. The correlation between these alarms and specific urban canyon conditions, such as heading changes or angular velocity shifts, suggests an area for further investigation. Understanding the

model's response to these dynamic conditions could lead to refinement in its predictive algorithms, enhancing its reliability in even more challenging urban environments.

Hazardous Misleading Information would be indicated by points above the threshold line, where the HPE is greater than the HPL and no alarm is raised. This is a critical situation as it means the system is providing incorrect position information without any warning. Importantly, there are 0 instances (0.00%) in this quadrant, demonstrating that the system did not produce any misleading information that went undetected.

The quadrant representing the combination of System Unavailable and Hazardous Misleading Information would show instances where the system is unavailable and provide hazardous misleading information. There is 0 instance (0.00%) in this quadrant, which means that there were no occurrences of simultaneous system unavailability and misleading information. The absence of points in the hazardous misleading information quadrants is a positive indication of the system's integrity, suggesting that the Bayesian-LSTM is highly reliable. The diagram suggests that when the system predicts that it cannot maintain integrity, it appropriately declares itself as unavailable, which is a critical feature for safety-critical applications.

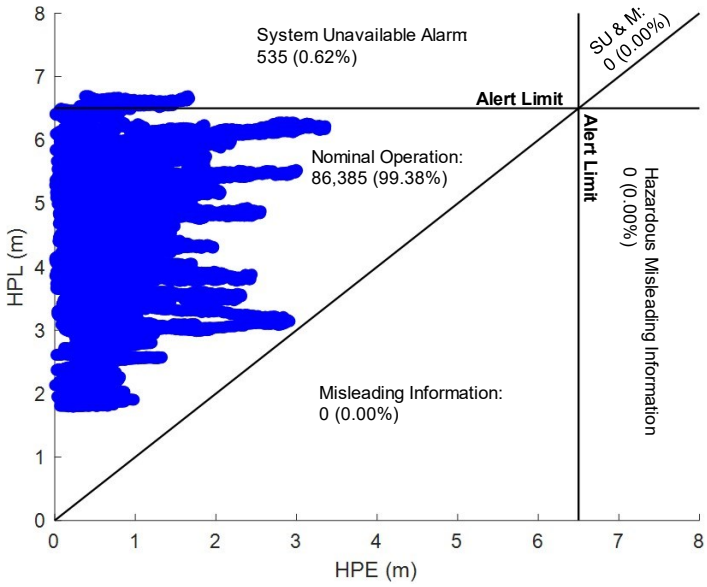


Figure 7-14 Categorizing horizontal protection level estimate from Bayesian-LSTM using Stanford diagram

Figure 7-15 represents a Stanford diagram for the integrity analysis of the Bayesian-LSTM system, illustrating the distribution of Vertical Protection Level (VPL) against Vertical Position Error (VPE). In the diagram, the majority of data points fall within the Nominal Operation quadrant, where the VPL exceeds the VPE, indicating reliable vertical position information. With 86,816 occurrences (99.88%), the system predominantly operates within acceptable bounds, maintaining a high integrity level. A single data point (0.001%) is present in the System Unavailable Alarm quadrant, reflecting an instance where the system correctly identified an inability to guarantee the integrity of the position information and thus signalled an alarm. However, there are 103 instances (0.12%) in the Misleading Information quadrant, where the VPE exceeds the VPL without triggering an alarm. This suggests that there have been a small number of occurrences where the system provided incorrect vertical position information without detection. There are no instances in the combined quadrant of System Unavailable and Hazardous Misleading Information, indicating no concurrent failures of system availability and undetected misleading information. The presence of misleading information cases, albeit small, underscores the need for continued enhancement of the integrity monitoring capabilities of the system. The figure demonstrates that while the Bayesian-LSTM system offers a high degree of reliability, attention to the instances of misleading information is crucial for safety-critical applications where vertical integrity is paramount.

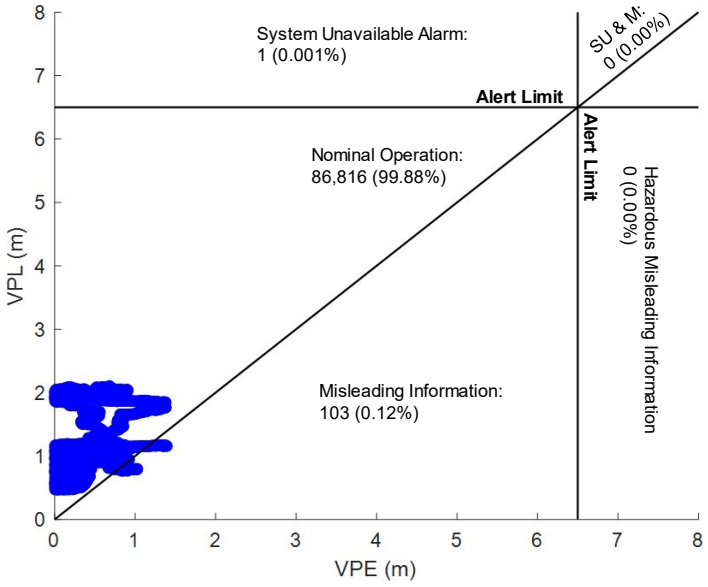


Figure 7-15 Categorizing vertical protection level estimate from Bayesian-LSTM using Stanford diagram

In Figure 7-16, the Bayesian-LSTM Horizontal Protection Level (HPL) exhibits a pattern that mirrors the peaks of the position error, albeit at a higher amplitude. This mirroring effect indicates that the HPL is responsive to changes in the position error, dynamically adjusting to maintain a protective margin that encompasses the error peaks. The responsiveness of the HPL to the error peaks is an essential characteristic of an effective integrity monitoring system. It ensures that as the position error increases, so does the HPL, maintaining a safety buffer that accounts for the increased uncertainty in the position estimate. The position error plot shows its highest peaks are well within the bounds set by the HPL, demonstrating the system's robustness in accounting for error spikes. The correlation between the error and the HPL suggests that the integrity monitoring system of the Bayesian-LSTM is not only functioning correctly but is also well-tuned to the dynamics of the position error behaviour. The ability of the HPL to follow the position error peaks without being breached is indicative of the system's reliability. It suggests that users of the system can have confidence in the positional information provided, with the assurance that the HPL will provide an accurate reflection of the current error state, even at times when the error magnitude is at its greatest.

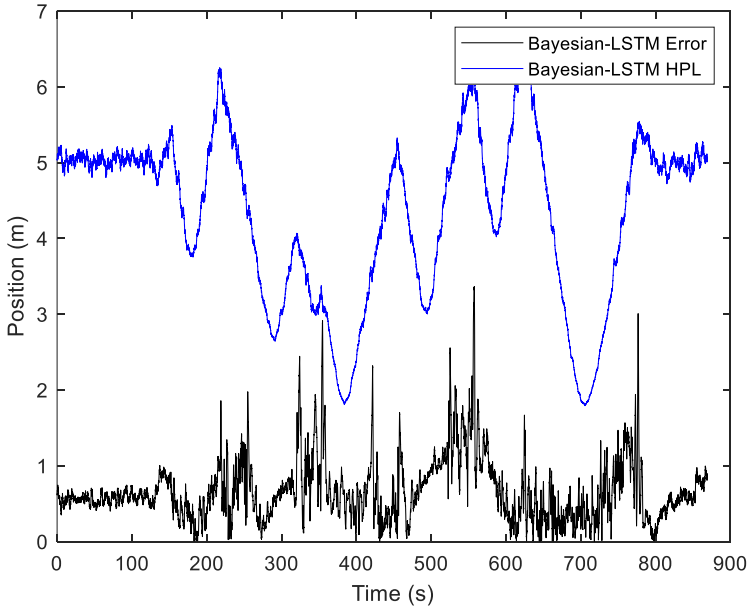


Figure 7-16 Comparison of horizontal protection level estimate with actual absolute position error vs time.

7.5.4 Comparative Performance Analysis

7.5.4.1 EKF Position and Protection Level Performance

Figure 7-17 compares the integrity performance between two navigation system architectures: a conventional Extended Kalman Filter (EKF) utilizing GNSS from the receiver and IMU data without any post-processing and the Bayesian-LSTM system proposed in this chapter.

The EKF points, depicted in pink, display a significant spread across the HPL spectrum, with many points exceeding the threshold line that delineates acceptable performance. The presence of these points above the threshold suggests instances where the EKF system's horizontal protection level may not be sufficient to cover the position error, indicating a potential risk of misleading information. Furthermore, the wide dispersion of the EKF points signifies a variability in performance, which could translate to inconsistent reliability in applications where consistent positioning accuracy is critical. There are a significant number of points that provide misleading information, which indicate timesteps where no alert will be given to the mission controller. There are several points in the hazardous misleading information section, which indicate timesteps where not only does the position error exceed the alert limit, but also the system still believes the protection level to be below the alert limit, therefore no indication to the situation is provided to the mission controller.

The Bayesian-LSTM system, represented by the blue points, shows a more concentrated clustering of data, predominantly in the nominal operation space. This concentration suggests that the Bayesian-LSTM system not only provides a more precise estimation of position but also offers a more robust measure of protection, as indicated by the lower HPL values relative to the corresponding HPE. The instances where Bayesian-LSTM points do cross the threshold are notably fewer compared to the EKF system.

The implications of choosing one system over the other are substantial. Opting for the Bayesian-LSTM system could potentially lead to a higher level of safety and reliability in navigation-critical applications, such as autonomous vehicle navigation, aircraft landing systems, or any other application where integrity and precision are paramount. The lower and tighter distribution of HPL values in the Bayesian-LSTM system indicates

a higher confidence level that the true position error will remain within the predicted bounds.

Conversely, while the EKF system may be simpler and more established, its performance, as depicted by the wider spread and higher instances of crossing the threshold, suggests that it may not be as reliable for safety-critical applications where high integrity is required. The EKF system may be more susceptible to scenarios where the actual position error could exceed the predicted protection level without adequate warning, leading to a higher chance of undetected misleading information.

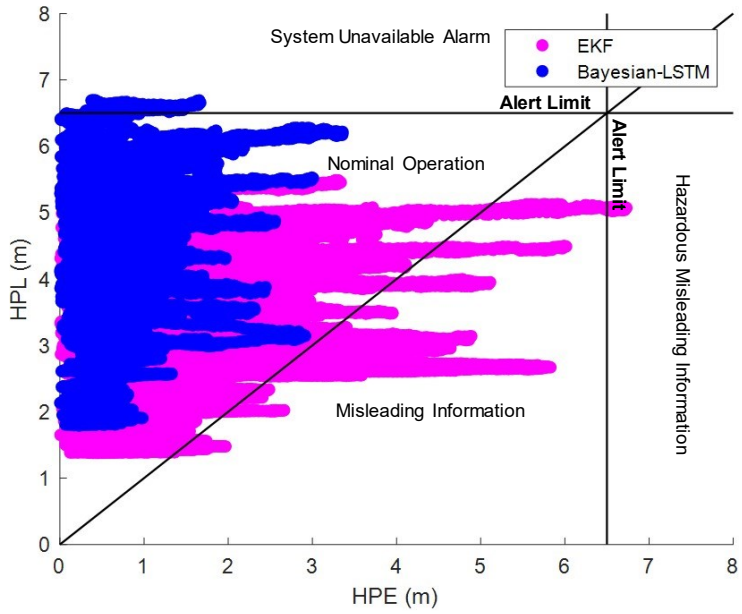


Figure 7-17 Comparison of horizontal protection level estimate performance between Bayesian-LSTM and simple EKF integration

Figure 7-18 presents a trajectory plot comparing the performance of a traditional Extended Kalman Filter (EKF) and the Bayesian-LSTM system against the known ground truth. This trajectory plot is particularly illustrative when considering the integrity analysis discussed previously. The EKF trajectory shows deviations from the ground truth at various points, particularly noticeable where there are sharp turns or changes in direction. These deviations could be attributed to the accumulation of small errors that are magnified during such manoeuvres. The Bayesian-LSTM system's trajectory closely follows the ground truth with much smaller deviations. The improved performance of the Bayesian-LSTM system can be linked to its ability to learn from sequential data and make

more accurate predictions, especially in handling the non-linearities of the motion which seem to challenge the EKF. The deviation of the EKF from the ground truth in this figure correlates with the higher spread of EKF points observed in the integrity analysis Stanford diagram. It provides a visual confirmation that the Bayesian-LSTM system not only maintains a higher level of integrity, as previously discussed, but also translates this into a more accurate and reliable trajectory estimation. This figure reinforces the implications of using the Bayesian-LSTM over the EKF. In applications where the integrity and accuracy of positional information are critical, the Bayesian-LSTM system's ability to closely mirror the ground truth and maintain a consistent trajectory with minimal deviation is a substantial advantage.

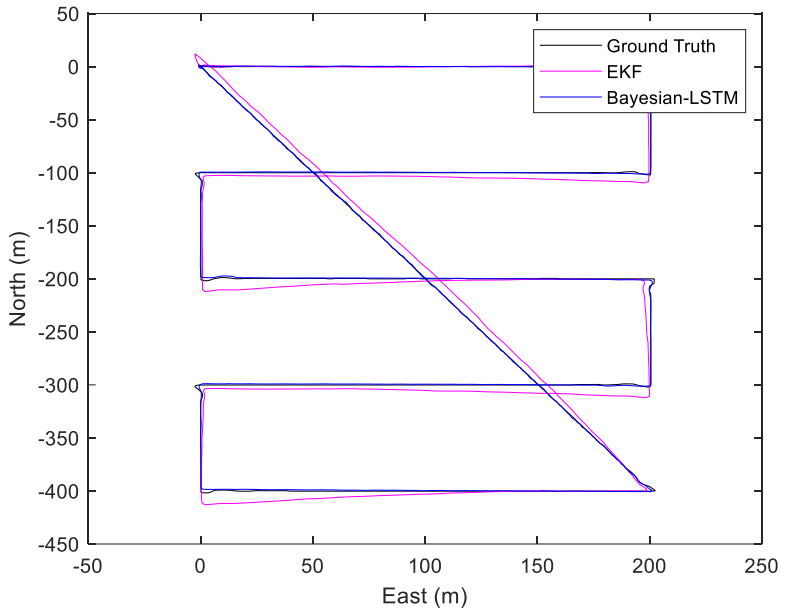


Figure 7-18 Comparison of horizontal positioning accuracy between Bayesian-LSTM, EKF and ground truth

7.5.4.2 Literature Comparison

Figure 7-19 displays a bar chart comparing the positional errors of the Bayesian-LSTM system against another architecture from a referenced paper [128]. Errors are analysed across North, East, and Down (NED) coordinates, and include the 95th percentile errors for horizontal and vertical positions. The Bayesian-LSTM system exhibits lower errors in the North direction compared to the architecture from the referenced study. In contrast, the Bayesian-LSTM system shows a slightly higher error in the Down (vertical) position, suggesting its horizontal positioning is more accurate while its vertical accuracy is less

so compared to the other architecture. The 95th percentile errors, which indicate the worst-case scenario performance, are lower for the Bayesian-LSTM system, suggesting a potentially more reliable performance in positional accuracy. Despite a higher vertical error, the overall performance of the Bayesian-LSTM system, particularly in terms of the 95th percentile errors, indicates an improvement over the architecture from the referenced study.

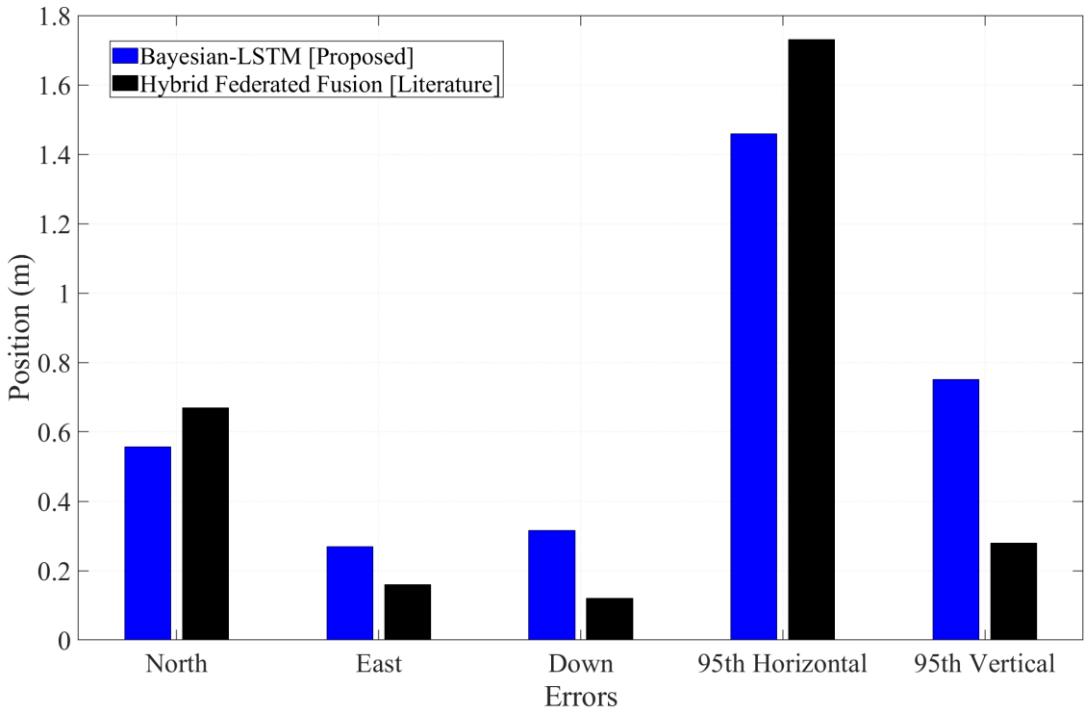


Figure 7-19 Comparison of mean position errors (North, East and Down) and 95th percentile error in the horizontal and vertical direction

7.6 Conclusion and Future Work

In this chapter, we proposed and assessed a federated multi-sensor fusion architecture leveraging deep learning techniques designed to enhance positioning integrity in urban canyon environments. The system integrates GNSS, IMU, monocular-VO, and barometric sensors through a Bayesian-LSTM algorithm aimed at addressing the complexities of urban navigation.

Our findings reveal improvements in positioning accuracy and reliability. The baseline GNSS receiver demonstrated initial mean positional errors of 22.32 m (North), 30.48 m (East), 37.78 m (Horizontal), and 23.34 m (Vertical), with 95th percentile errors reaching

63.00 m, 70.34 m, 94.73 m, and 54.40 m, respectively. The application of the GNSS/GRU algorithm reduced these errors to 9.48 m, 6.54 m, 11.52 m, and 4.83 m, respectively, showcasing the method's efficiency in signal discrimination and error reduction.

The integration of EKF, combining GNSS/GRU and INS/GRU data, further decreased mean errors to 7.84m, 4.43 m, 9.01 m, and 3.50 m, respectively. The addition of the VO system and barometer readings maintained vertical accuracy effectively, while the comprehensive Bayesian-LSTM fusion algorithm achieved horizontal errors below 0.62m and vertical errors around 0.30m under various environmental conditions.

Integrity analysis, conducted through Horizontal Protection Level (HPL) and Vertical Protection Level (VPL) assessments, indicated that 99.38% and 99.88% of data points fell within the Nominal Operation quadrant for HPL and VPL, respectively. This demonstrates the system's effectiveness and reliability, particularly in challenging urban scenarios where conventional systems falter.

The potential applications span across autonomous vehicles, drones, mobile mapping, and emergency services, where precise, reliable navigation is crucial. This architecture not only addresses current urban navigation challenges but also sets a new benchmark for future positioning systems.

Chapter 8 General Discussion

This research is situated within the broader domain of autonomous navigation and UAV operations in urban environments, addressing critical challenges such as GNSS dependency and sensor inaccuracies. By focusing on machine learning for integrity-based sensor fusion, the work contributes to the urgent global demand for reliable, autonomous systems capable of navigating complex urban landscapes. This is particularly relevant in the context of increasing urbanization and the need for innovative solutions to support urban mobility, emergency response, and smart city initiatives.

Concerning the initial research objectives, four main goals were set to explore the role of machine learning integrity-based sensor fusion and develop and evaluate PNT navigation solutions for UAVs in urban environments. These solutions aimed to achieve high accuracy, reduce dependence on GNSS, and improve robustness using machine learning and alternative PNT methods. The details of these objectives and their achievement are as follows:

- **O1**: Identify and discuss candidate PNT and fusion technologies for assured PNT in safety-critical missions. The performance characteristics of these technologies, along with their complementary aspects, will be analysed to support the development of an effective fusion solution and identify mechanisms for further improvements/adaptation of individual technologies.
    - Literature reviews in Chapters Two and Three addressed issues with traditional sensors like GNSS and IMU and discussed filtering/fusion techniques such as EKF. These chapters also explored advanced techniques to tackle IMU drift and GNSS multipath errors, enhancing GNSS performance and positioning accuracy.
    - Chapter three reviewed current sensor fusion technologies, comparing different approaches and architectures, including centralized and federated systems.
    - Chapter seven introduced additional sensors like camera and barometer for improved positioning performance

- **O2**: To propose a robust sensor fusion architecture and underlying technologies for UAVs in urban canyons using deep learning approaches based on mission specifications: environment, integrity, and accuracy requirements. This method will inform the development of performance-based navigation concepts for autonomous systems.
    - A federated fusion system using Bayesian-LSTM architecture for sensor fusion was introduced in Chapter Seven. This setup included various sensors and used an EKF for integrating corrected INS and GNSS data, while fusion with VO/Barometer data was achieved using Bayesian-LSTM. This approach allowed for uncertainty estimation, enhancing the system's integrity.
- **O3**: Develop a sensor simulation environment for data collection and post-processing. This will enable testing of the proposed architecture in a controlled manner for a robust analysis.
    - The simulation architecture described from Chapters Four used Spirent GSS7000 for GNSS signal simulation, OKTAL-SE Sim3D for creating multipath effects, and AirSim along with Unreal Engine for generating VO data. This setup allowed for comprehensive testing and analysis of the fusion system under different conditions.
- **O4**: Evaluate the performance of the developed architecture using hardware in the loop simulations and existing performance metrics for comparative analysis.
    - The evaluation covered essential metrics for positioning and machine learning performance, comparing our system against other state-of-the-art architectures to demonstrate improvements in performance and integrity.

In summary, all objectives have been successfully implemented to achieve the stated aim. However, it is also important to discuss the findings in a broader context than has been explored in each chapter. Here, we will discuss observations regarding implementation and how the proposed architecture solves the stated issues in the introduction (Chapter 1), how the work presented here has a worldwide implication and future innovation that is required for the successful implementation.

8.1 Implementation and Identified Issues

The introductory chapter highlighted significant challenges within urban development, particularly the impact of rising traffic volumes on the transportation of goods and services as populations grow. These challenges, if not addressed, could have considerable financial and social repercussions. To mitigate these issues, the improvement of robust UAM positioning accuracy was proposed as an urgent challenge to tackle, particularly regarding UAVs and addressing significant safety concerns. Notably, the safety issues identified include the insufficient accuracy, integrity, and robustness necessary for the autonomous operation of UAVs.

In response, this dissertation proposed a novel federated multi-sensor deep learning framework designed to enhance positioning accuracy through error correction techniques and integration of multiple sensors. This approach also aims to improve the system's integrity by quantifying the uncertainty in the predictions of the deep learning model.

The results, discussed in Chapters 5 to 7, demonstrated that the aforementioned system improved positioning accuracy compared to state-of-the-art research, achieving 95th percentile horizontal errors of less than 1.6 meters (Chapter 4). Such accuracy is particularly effective in urban contexts, where the average half-street width in cities like New York is approximately 6.5 meters, thereby facilitating reliable navigation and efficient service delivery in densely populated areas.

Furthermore, the system's integrity was enhanced by its ability to estimate the PL from the uncertainty distributions derived by a Bayesian-LSTM and comparing it against the AL. The ability to understand the system integrity performance is essential for maintaining shared safety in urban environments.

The robustness of the system was also tested under various challenging signal conditions and weather scenarios typical in urban canyons, where conditions change rapidly. The results confirmed the method's capability to withstand fluctuations, thereby reinforcing its suitability for such environments and affirming the direction of this research.

Regarding the practical implementation of this system, several key factors need consideration. The transition from simulated data to real-world scenarios introduces complexities that may have been previously overlooked, such as system behaviour with

other vehicles present in the environment. Additionally, the deployment of machine-learning algorithms in real-world contexts presents further challenges. Mitigation strategies could include establishing safety standards and benchmarks specifically for machine learning applications. Hardware considerations are also crucial, as the deployment of such systems requires robust and continuously available components that can support advanced predictive models and integrating multiple sensor inputs.

8.2 Worldwide Context of the Research

The research presented in the preceding chapters have demonstrated several achievable advancements in UAM, particularly through the development of accurate and integrity-based navigation systems. These improvements address the critical safety and regulatory concerns outlined in the introduction of this thesis. By resolving the navigation challenges associated with UAM, UAVs can navigate safely through urban canyons, thereby alleviating the congestion currently experienced by road vehicles. Moreover, the reduction in position uncertainty enables an increase in the density of UAM traffic, enhancing the utilization of airspace in urban environments.

Beyond the developments in the positioning system, the integration of existing and emerging technologies could foster a range of improvements in utilization, safety, and the development of innovative use cases. For instance, incorporating communication elements between vehicles, as well as a general awareness of environmental conditions and the positions of other vehicles, could significantly enhance operational safety and efficiency. Future technologies such as 6G Integrated Sensing and Communication (ISAC) [129] and Low Earth Orbit (LEO) [130] Positioning, PNT are expected to improve environmental awareness and provide additional positioning resources through the concept of Position as a Service.

The global applicability and scalability of the system are also crucial considerations. While adaptability can be equated with the system's demonstrated robustness to changing conditions, as detailed in Chapter 7, scalability poses a more substantial challenge. Although the proposed solution operates independently of other vehicles in the environment, it requires refined and potentially costly hardware for widespread implementation across a fleet of UAVs. Nevertheless, the decreasing costs of hardware,

coupled with enhancements in efficiency and processing power, suggest that these financial barriers may diminish in the future.

8.3 Future Innovation Requirements

As UAM evolves, the integration of advanced technologies such as quantum sensing, transformers, and Vehicle-to-Everything (V2X) communication is poised to revolutionize UAV navigation. These technologies address critical challenges in navigating complex urban environments, yet their readiness for practical deployment varies and must be carefully evaluated.

Quantum sensing offers groundbreaking opportunities for UAV navigation through its unparalleled ability to measure environmental variables with precision and stability. Technologies such as quantum gyroscopes and accelerometers could replace GNSS systems, providing robust navigation solutions in GNSS-denied or interference-heavy environments, such as urban canyons or tunnels. Beyond navigation, quantum sensing could allow UAVs to detect subtle changes in atmospheric conditions, enabling proactive adjustments in flight paths to optimize safety and energy efficiency [131]. While still in the research and development phase, advancements in miniaturization and cost reduction suggest that quantum sensors could see pilot implementations within the next decade. Current UAV systems can begin to adopt hybrid approaches, combining traditional sensors with early-stage quantum devices to enhance navigation capabilities incrementally.

Transformers, known for their exceptional ability to model complex data dependencies, offer transformative potential in sensor fusion and decision-making for UAVs. By processing multi-sensor data streams in real time, transformers can enhance situational awareness and operational safety. Recent studies have employed a graph transformer neural network to predict satellite visibility, effectively learning environmental representations from irregular GNSS measurements to mitigate multipath interference and enhance generalisation [132]. Other use case examples include transformers that could enable UAVs to prioritize collision-avoidance data in high-risk scenarios while simultaneously analysing visual inputs to identify optimal landing zones. However, their integration into UAV systems is currently limited by computational constraints, particularly for smaller UAVs with resource limitations. Advances in edge computing

and hardware accelerators are addressing these challenges, making transformers increasingly feasible for high-value applications, such as urban delivery or search-and-rescue missions.

V2X communication technologies complement these advancements by enabling UAVs to communicate seamlessly with vehicles, infrastructure, and other UAVs. In an intelligent transport system (ITS), UAVs equipped with V2X can coordinate with ground vehicles, receive updates on traffic patterns, and interact with urban infrastructure like smart traffic lights [133]. For instance, UAVs delivering goods could synchronize with autonomous delivery trucks, optimizing routes and handoffs. Similarly, UAVs monitoring urban environments could collaborate with ground sensors to respond more effectively to localized events, such as accidents or natural disasters. While V2X standards for ground vehicles are well-established, adapting these protocols for UAVs involves overcoming challenges such as ensuring interoperability across three-dimensional mobility and addressing cybersecurity risks. Despite these hurdles, V2X is among the most readily deployable technologies, with components already being tested in urban environments.

Furthermore, the development of 6G networks and Integrated Sensing and Communication (ISAC) promises to push the boundaries of navigation accuracy and system reliability even further. The potential for enhanced communication between UAVs and terrestrial vehicles through shared sensors and data opens new avenues for ensuring pedestrian safety and operational efficiency.

The interplay between these technologies could redefine UAV navigation. For example, V2X data streams could be processed by transformers to predict traffic flows and adjust UAV routes proactively. Quantum sensors could enhance the accuracy of these adjustments by providing high-resolution environmental data. Machine learning could further refine these processes by enabling UAVs to learn and adapt to dynamic conditions, creating a feedback loop that enhances performance over time.

However, the integration of these technologies into operational UAV systems is not without challenges. Quantum sensing and transformer-based processing require significant advancements in hardware miniaturization and energy efficiency to meet the constraints of UAV platforms. Machine learning and V2X technologies face regulatory

and ethical considerations, including data privacy, communication security, and public acceptance. Addressing these challenges will require collaborative efforts across industries, policymakers, and urban planners to establish robust frameworks for deployment and governance.

While full integration of these technologies may take years, their incremental adoption is already possible. Machine learning, as shown in this thesis, can enhance existing UAV systems in complex environments, future transformers can support specialized applications requiring advanced sensor fusion, and V2X can enable more efficient coordination with urban infrastructure. Quantum sensing, although still emerging, represents a long-term opportunity to revolutionize navigation. By aligning near-term applications with current readiness and addressing long-term challenges, UAV navigation systems can progressively evolve into a more adaptive, intelligent, and interconnected ecosystem, prepared to meet the complexities of future urban environments.

Chapter 9 Conclusion

In conclusion, this thesis has tackled the pressing issue of autonomous navigation for UAVs in complex urban environments, where the urban canyon effect significantly undermines traditional navigation methods. By introducing a federated multi-sensor fusion architecture, this work marks an advancement over conventional systems, integrating Bayesian-LSTM algorithms with GNSS, INS, and monocular VO. This innovative approach not only outperforms existing approaches but also mitigates the limitations posed by dense urban structures.

The architecture's capability to preprocess GNSS data, NLOS signals, combined with its GRU-based architecture for correcting INS data errors, has proven to be transformative. Quantitative evaluations showcase a notable reduction in the 95th percentile horizontal error and a 30.1% enhancement in horizontal error performance when benchmarked against GNSS/IMU/VO EKF-GRU systems. These results underline the potential to elevate the precision and reliability of UAV navigation within urban canyons, setting a new standard for future navigational technologies.

The implications of this research extend far beyond technological excellence, offering profound societal and business benefits. Enhanced UAV navigation systems, as developed in this thesis, hold the potential to revolutionise urban logistics by enabling more efficient, reliable delivery services, thereby reducing urban traffic congestion and contributing to environmental sustainability. In emergency scenarios, the ability of UAVs to navigate to critical locations quickly and accurately can be lifesaving, underlining the importance of this research in public safety and disaster response efforts. Additionally, using UAVs in urban planning and infrastructure maintenance presents innovative approaches to city management and development, promoting sustainable urban growth.

Additionally, the development of standardized protocols for UAV navigation is essential for achieving interoperability and ensuring safety across various platforms and applications. Such standardization would facilitate a cohesive framework for urban air mobility, enabling a seamless integration of UAVs into the broader ecosystem of urban transportation.

In summary, this thesis not only addresses current challenges in UAV navigation but also lays the groundwork for future advancements. The successful implementation of the federated stochastic ML architecture represents a significant leap forward in autonomous urban navigation, offering a blueprint for future research and development in this rapidly evolving field. As we continue to explore the integration of new technologies and tackle the broader implications of UAV deployment, the potential for transformative impact on urban mobility, public safety, and environmental sustainability is immense.

Reference

[1] M. R. Lowry, “Towards High Density Urban Air Mobility,” in *AIAA AVIATION Forum 2018*, Atlanta, Georgia, 2018.

[2] S. G. G. a. A. K. N. Yahi, “On-the-Fly Coordination of Maneuvers for Separation Assurance of UAM Aircraft in Congested Airspace,” *IEEE Transactions on Intelligent Transportation Systems,* vol. 25, no. 11, pp. 18714-18733, 2024.

[3] D. A. Tedjopurnomo, Z. Bao, B. Zheng, F. M. Choudhury and A. k. Qin, “A Survey on Modern Deep Neural Network for Traffic Prediction: Trends, Methods and Challenges,” *IEEE Transactions on Knowledge and Data Engineering,* pp. 1544-1561, 2020.

[4] L. Asmer, H. Pak, P. Prakasha, B. Schuchardt, P. Weiand, F. Meller, C. Torens, D. Becker, C. Zhu, K. Schweiger, A. Volkert and R. Jaksche, “Urban Air Mobility Use Cases, Missions and Technology Scenarios for the HorizonUAM Project,” in *AIAA AVIATION 2021 FORUM*, VIRTUAL, 2021.

[5] C. Roberts, N. Forrest, G. Baker and B. Evans, “Skies without limits v2.0,” pwc, 2022.

[6] Y. W. L. Z. a. Y. Z. C. Tang, “GNSS/Inertial Navigation/Wireless Station Fusion UAV 3-D Positioning Algorithm With Urban Canyon Environment,” *IEEE Sensors Journal,* vol. 22, no. 19, pp. 18771-18779, 2022.

[7] P. Xie and M. G. Petovello, “Measuring GNSS Multipath Distributions in Urban Canyon Environments,” *IEEE Transactions on Instrumentation and Measurement,* vol. 64, no. 2, pp. 366 - 377, 2015.

[8] Department for Business, Energy & Industrial Strategy and UK Research and Innovation, *World's first timing centre to protect UK from risk of satellite failure,* London, UK: GOV.UK, 2020.

[9] L. M. B. a. F. G. J. Bonenberg, “Assessing Alternative Positioning, Navigation and Timing Technologies for Potential Deployment in the EU,” Publications Office of the European Union, Luxembourg, 2023.

[10] T. Li, H. Zhang, X. Niu and Z. Gao, “Tightly-Coupled Integration of Multi-GNSS Single-Frequency RTK and MEMS-IMU for Enhanced Positioning Performance,” *MDPI Sensors,* vol. 17, no. 11, p. 2462, 2017.

[11] Q. Zhang, X. Niu and C. Shi, “Impact Assessment of Various IMU Error Sources on the Relative Accuracy of the GNSS/INS Systems,” *IEEE Sensors Journal,* vol. 20, no. 9, pp. 5026-5038, 2020.

[12] S. L. Q. F. Z. L. a. Q. Z. Y. Liu, “Analysis of Kalman Filter Innovation-Based GNSS Spoofing Detection Method for INS/GNSS Integrated Navigation System,” *IEEE Sensors Journal,* vol. 19, no. 13, pp. 5167-5178, 2019.

[13] P. S. a. A. Kumar, “Overview of architecture for GPS-INS integration,” in *017 Recent Developments in Control, Automation & Power Engineering (RDCAPE)*, Noida, India, 2017.

[14] G. L. Y. S. J. S. a. B. Z. F. Tang, “Comparison of Performance Between Loosely and Tightly Coupled GNSS/INS in Real Scenarios,” in *2023 International Conference on Microwave and Millimeter Wave Technology (ICMMT)*, Qingdao, China, 2023.

[15] F. S. J. C. P. a. J. G. A. R. Jiménez, “Indoor pedestrian navigation using an INS/EKF framework for yaw drift reduction and a foot-mounted IMU,” in *7th Workshop on Positioning, Navigation and Communication*, Dresden, Germany, 2010.

[16] A. Nessa, B. Adhikari, F. Hussain and X. N. Fernando, “A Survey of Machine Learning for Indoor Positioning,” *IEEE Access,* vol. 8, pp. 214945-214965, 2020.

[17] F. -Y. W. K. W. W. -H. L. X. X. a. C. C. J. Zhang, “Data-Driven Intelligent Transportation Systems: A Survey,” *IEEE Transactions on Intelligent Transportation Systems,* vol. 12, no. 4, pp. 1624 - 1639, 2011.

[18] A. Bos, M. Snijders, A. Zevenbergen, K. Drost, H. Zelle and B. v. d. Hoeven, “Robustness Levels of Critical Infrastructures Against Global Navigation Satellite System Signal Disturbances,” in *European Navigation Conference 2023*, Noordwijk, The Netherlands, 2023.

[19] R. Benoit and E. Chatre, “The SBAS integrity concept standardised by ICAO-application to EGNOS,” in *NAVIGATION-PARIS*, Paris, 2001.

[20] D. Egea-Roca, M. Arizabaleta-Diez, T. Pany, F. Antreich, J. A. Lopez-Salcedo, M. Paonni and G. Seco-Granados, “GNSS User Technology: State-of-the-Art and Future Trends,” *IEEE Access,* vol. 10, pp. 39939 - 39968, 2022.

[21] Q. D. Hua, M. S. Williams, J. Evans and M. A. Trippy, “Use of the NAVSTAR Global Positioning System (GPS) in the Defense Communications System (DCS),” in *1993 IEEE International Frequency Control Symposium*, Salt Lake City, USA, 1993.

[22] T. Humphreys, “GPS Spoofing and the Financial Sector,” The University of Texas at Austin, Austin, USA, 2011.

[23] P. Davies, “Great Britain's Ordnance Survey National GPS Network,” *The Journal of Navigation,* vol. 53, no. 3, pp. 397 - 402, 2000.

[24] J. V. E. S. E. J. S. G. J. E. H. E. L. N. a. J. H. T. J. G. Davies, “Radar observations of the second Russian earth satellite,” *Proc. R. Soc. Lond. A,* vol. 250, no. 1262, 1959.

[25] D. M. Galvin, “History of the GPS Space Segment From Block I to The New Millennium,” in *Proceedings of the 12th International Technical Meeting of the Satellite Division of The Institute of Navigation* , Nashville, TN, USA, 1999.

[26] P. D. Groves, GNSS, Inertial, and Multisensor Integrated Navigation Systems, 2nd Ed., 2013.

[27] M. Akram, P. Liu, Y. Wang and J. Qian, “NSS Positioning Accuracy Enhancement Based on Robust Statistical MM Estimation Theory for Ground Vehicles in Challenging Environments,” *Appl. Sci.,* vol. 8, no. 6, p. 876, 2018.

[28] P. D. Groves, Principles of GNSS, inertial, and multisensor integrated navigation systems, 2nd edition, London, UK: Artech House Publishers, 2013.

[29] J. H. a. S. Guo, “The study of dual frequency ionospheric error correction method and accuracy analysis based on GPS,” in *IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, Xi'an, China, 2011.

[30] T. Kindervatter and F. Teixeria, Tropospheric and Ionospheric Effects on Global Navigation Satellite Systems, Wiley-IEEE Press, 2022.

[31] K. Krasuski and D. Wierzbicki, “Application the SBAS/EGNOS Corrections in UAV Positioning,” *Energies,* vol. 14, no. 3, p. 739, 2021.

[32] LiXingxing, J. Haung, X. Li, Z. Shen, J. Han, L. Li and B. Wang, “Review of PPP–RTK: achievements, challenges, and opportunities,” *Satellite Navigation,* vol. 3, no. 28, 2022.

[33] I. P. D. T. A. Z. N. L. J. Alcantarilla, “The Benefits of Multi-constellation GNSS Augmentations,” in *Proceedings of the 19th International Technical Meeting of the Satellite Division of The Institute of Navigation*, Fort Worth, TX,USA, 2006.

[34] J. Zidan, E. I. Adegoke, E. Kampert, S. A. Birrell, C. R. Ford and M. D. Higgins, “GNSS Vulnerabilities and Existing Solutions: A Review of the Literature,” *IEEE Access,* vol. 9, pp. 153960 - 153976, 2020.

[35] Q. Zhang, L. Zhang, A. Sun, X. Meng, D. Zhao and C. Hancock, “GNSS Carrier-Phase Multipath Modeling and Correction: A Review and Prospect of Data Processing Methods.,” *Remote Sens.,* vol. 16, no. 1, p. 189, 2024.

[36] Z. L. L. S. a. L. F. R. Hang, “Multiple carrier correlators based carrier phase multipath mitigation technique for real time kinematic,” in *IEEE/ION Position, Location and Navigation Symposium (PLANS)*, Savannah, GA, USA, 2016.

[37] S. Bijjahalli and R. Sabatini, “A High-Integrity and Low-Cost Navigation System for Autonomous Vehicles,” *IEEE Transactions on Intelligent Transportation Systems,* vol. 22, no. 1, pp. 356 - 369, 2021.

[38] P. Geragersian, I. Petrunin, W. Guo and R. Grech, “An INS/GNSS fusion architecture in GNSS denied environment using gated recurrent unit,” in *AIAA SCITECH 2022 Foru*, San Diego, CA, USA, 2021.

[39] D. A. Grejner-Brzezinska, C. K. Toth, T. Moore, J. Raquet, M. M. Miller and A. Kealy, "Multisensor Navigation Systems: A Remedy for GNSS Vulnerabilities?," *Proceedings of the IEEE,* vol. 104, no. 6, pp. 1339 - 1353, 2016.

[40] N. Boguspayev, D. Akhmedov, A. Raskaliyev, A. Kim and A. Sukhenko, "A Comprehensive Review of GNSS/INS Integration Techniques for Land and Air Vehicle Applications," *Applied Sciences,* vol. 13, no. 8, p. 4819, 2023.

[41] D. Feng, C. Wang, C. He, Y. Zhuang and X.-G. Xia, "Kalman-Filter-Based Integration of IMU and UWB for High-Accuracy Indoor Positioning and Navigation," *IEEE Internet of Things Journal,* vol. 7, no. 4, pp. 3133 - 3146, 2020.

[42] F. Zhang, Y. Wang and Y. Gao, "A Novel Method of Fault Detection and Identification in a Tightly Coupled, INS/GNSS-Integrated System," *Sensors,* vol. 21, no. 9, p. 2922, 2021.

[43] P. Roy and C. Chowdhury, "A Survey of Machine Learning Techniques for Indoor Localization and Navigation Systems," *Journal of Intelligent & Robotic Systems volume,* vol. 101, no. 3, p. 63, 2021.

[44] E. S. Brunette, R. C. Flemmer and C. L. Flemmer, "A review of artificial intelligence," in *2009 4th International Conference on Autonomous Robots and Agents*, Wellington, New Zealand, 2009.

[45] V. S. Babu and K. Banana, "A STUDY ON NARROW ARTIFICIAL INTELLIGENCE– AN," *International Journal of Engineering Science and Advanced Technology,* vol. 24, no. 4, 2024.

[46] B. Goertzel, "Artificial General Intelligence: Concept, State of the Art, and Future Prospects," *Journal of Artificial General Intelligenc,* vol. 5, no. 1, pp. 1-46, 2014.

[47] J. S. A. H. H. Helm, "Machine Learning and Artificial Intelligence: Definitions, Applications, and Future Directions.," *urr Rev Musculoskelet Med,* vol. 13, p. 69–76 , 2020.

[48] R. S. a. P. Sujatha, "A State of Art Techniques on Machine Learning Algorithms: A Perspective of Supervised Learning Approaches in Data Classification," in *Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India, 2018.

[49] Y. Z. K. K. D. a. Q. W. H. U. Dike, "Unsupervised Learning Based On Artificial Neural Network: A Review," in *IEEE International Conference on Cyborg and Bionic Systems (CBS)*, Shenzhen, China, 2018.

[50] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems.," *ASME. J. Basic Eng,* vol. 82, no. 1, pp. 35-45, 1960.

[51] E. Dawson, M. A. Rashed, W. Abdelfatah and A. Noureldin, "Radar-Based Multisensor Fusion for Uninterrupted Reliable Positioning in GNSS-Denied Environments," *IEEE Transactions on Intelligent Transportation Systems ,* vol. 23, no. 12, pp. 23384 - 23398, 2022.

[52] Q. Li, R. Li, K. Ji and W. Dai, "Kalman Filter and Its Application," in *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, Tianjin, China, 2015.

[53] O. G. Crespillo, D. Medina, J. Skaloud and M. Meurer, "Tightly coupled GNSS/INS integration based on robust M-estimators," in *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, Monterey, CA, USA, 2018.

[54] M. Ceccato, F. Formaggio, N. Luarenti and S. Tomasin, "Generalized Likelihood Ratio Test for GNSS Spoofing Detection in Devices With IMU," *IEEE Transactions on Information Forensics and Security,* vol. 16, pp. 3496 - 3509, 2021.

[55] R. Sun, Y. Yang, K.-W. Chiang, T.-T. Duong, K.-Y. Lin and G.-J. Tsai, "Robust IMU/GPS/VO Integration for Vehicle Navigation in GNSS Degraded Urban Areas," *IEEE Sensors Journal,* vol. 20, no. 17, pp. 10110 - 10122, 2020.

[56] L. v. Stumberg, P. Wenzel, Q. Khan and D. Cremers, "GN-Net: The Gauss-Newton Loss for Multi-Weather Relocalization," *IEEE Robotics and Automation Letters,* vol. 5, no. 2, pp. 890 - 897, 2020.

[57] H. Benzerrouk and A. Nebylov, "Robust nonlinear filtering applied to integrated navigation system INS/GNSS under non Gaussian measurement noise effect," in *2012 IEEE Aerospace Conference*, Big Sky, MT, USA, 2012.

[58] Y. Liu, Q. Fu, Z. Liu and S. Li, "GNSS Spoofing Detection Ability of a Loosely Coupled INS/GNSS Integrated Navigation System for Two Integrity Monitoring Methods," in *Proceedings*

of the 2017 International Technical Meeting of The Institute of Navigation, Monterey, California, USA, 2017.

[59] M. M. V. Khodarahmi, "A Review on Kalman Filter Models," *Arch Computat Methods Eng,* vol. 30, pp. 727-747, 2023.

[60] S. K. P. a. M. J. Konatowski, "Comparison of estimation accuracy of EKF, UKF and PF filters," *Annual of Navigation,* vol. 23, pp. 69-87, 2016.

[61] S. Shah and P. P. Shinde, "A Review of Machine Learning and Deep Learning Applications," in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, Pune, India, 2018.

[62] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Udupi, India, 2017.

[63] S. Yang, L. Xing, W. Liu, W. Qian, W. Qian, H. Xue and Y. Zhu, "Robust Navigation Method for Wearable Human–Machine Interaction System Based on Deep Learning," *IEEE Sensors Journal,* vol. 20, no. 24, pp. 14950 - 14957, 2020.

[64] M. Jones and D. Johnston, "Understanding Phenomena in the Real World: The Case for Real Time Data Collection in Health Services Research," *Journal of Health Services Research & Policy,* vol. 16, no. 3, 2011.

[65] M. Bacic, "On hardware-in-the-loop simulation," in *Proceedings of the 44th IEEE Conference on Decision and Control*, Seville, Spain, 2005.

[66] "Spirent Federal," Spirent Federal System, [Online]. Available: https://spirentfederal.com/products/gss7000/. [Accessed 24 September 2924].

[67] "u-blox," u-blox, [Online]. Available: https://www.u-blox.com/en/product/zed-f9p-module. [Accessed 24 10 2024].

[68] "Spirent Sim3D," Spirent, 24 10 2024. [Online]. Available: https://www.spirent.com/products/multipath-and-obscuration-simulation-sim3d. [Accessed 24 10 2024].

[69] N. El-Sheimy and A. Youssef, "Inertial sensors technologies for navigation applications: state of the art and future trends," *Satellite Navigation,* vol. 1, no. 1, 2020.

[70] P. Geragersian, I. Petrunin, W. Guo and R. Grech, "Multipath Detection from GNSS Observables Using Gated Recurrent Unit," in *2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC)*, Portsmouth, VA, USA, 2022.

[71] N. Zhu, J. Marais, D. Betaille and M. Berbineau, "GNSS Position Integrity in Urban Environments: A Review of Literature," *IEEE Transactions on Intelligent Transportation Systems,* vol. 19, no. 9, pp. 2762 - 2778, 2018.

[72] H.-f. Dai, H.-w. Bian, R.-y. Wang and H. Ma, "An INS/GNSS integrated navigation in GNSS denied environment using recurrent neural network," *Defence Technology,* vol. 16, no. 2, pp. 334-340, 2020.

[73] H. Fei and F. Tan, "Bidirectional Grid Long Short-Term Memory (BiGridLSTM): A Method to Address Context-Sensitivity and Vanishing Gradient," *Algorithms ,* vol. 11, no. 11, p. 172, 2018.

[74] Y. Tang, J. Jiang, J. Liu, P. Yan, Y. Tao and J. Liu, "A GRU and AKF-Based Hybrid Algorithm for Improving INS/GNSS Navigation Accuracy during GNSS Outage," *Remote Sensors,* vol. 14, no. 3, p. 752, 2022.

[75] P. Groves, Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Second Edition, Artech, 2013.

[76] M. St-Pierre and D. Gingras, "Comparison between the unscented Kalman filter and the extended Kalman filter for the position estimation module of an integrated navigation information system," in *IEEE Intelligent Vehicles Symposium, 2004*, Parma, Italy, 2004.

[77] J. He, J. Xie, H. Xiaoping, L. Du and W. Zhou, "Analytical study and compensation for temperature drifts of a bulk silicon MEMS capacitive accelerometer," *Sensors and Actuators A: Physical,* vol. 239, pp. 174-184, 2016.

[78] S. Godha and M. E. Cannon, "GPS/MEMS INS integrated system for navigation in urban areas," *GPS Solutions,* vol. 11, p. 193–203, 2007.

[79] N. Zhu, J. Marais, D. Betaille and M. Berbineau, "GNSS Position Integrity in Urban Environments: A Review of Literature," *IEEE Transactions on Intelligent Transportation Systems,* vol. 19, no. 9, pp. 2762 - 2778, 2018.

[80] D. Lu and E. Schnieder, "Performance Evaluation of GNSS for Train Localization," *IEEE Transactions on Intelligent Transportation Systems,* vol. 16, no. 2, pp. 1054 - 1059, 2015.

[81] D. Tang, D. Lu, B. Cai and J. Wang, "GNSS Localization Propagation Error Estimation Considering Environmental Conditions," in *2018 16th International Conference on Intelligent Transportation Systems Telecommunications (ITST)*, Lisboa, Portugal, 2018.

[82] "Measuring GNSS Multipath Distributions in Urban Canyon Environments," *IEEE Transactions on Instrumentation and Measurement,* vol. 64, no. 2, pp. 366 - 377, 2014.

[83] S. Hewitson and J. Wang, "GNSS receiver autonomous integrity monitoring (RAIM) performance analysis," *GPS Solutions,* vol. 10, pp. 155-170, 2005.

[84] O. K. Isik, J. Hong, I. Petrunin and A. Tsourdos, "Integrity Analysis for GPS-Based Navigation of UAVs in Urban Environment," *Robotics,* vol. 9, no. 3, p. 66, 2020.

[85] K. Nagai, T. Fasoro, M. Spenko, R. Henderson and B. Pervan, "Evaluating GNSS Navigation Availability in 3-D Mapped Urban Environments," in *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, Portland, USA, 2020.

[86] C. Savas and F. Dovis, "Multipath Detection based on K-means Clustering," in *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*, Miami, USA, 2019.

[87] L.-T. Hsu, "GNSS multipath detection using a machine learning approach," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Yokohama, Japan, 2017.

[88] Q. Zhang, H. Lin, L. Ding, Q. Chen, T. Zhang and X. Niu, "RANSAC-Based Fault Detection and Exclusion Algorithm for Single-Difference Tightly Coupled GNSS/INS Integration," *IEEE Transactions on Intelligent Vehicles,* vol. Early Access, 2023.

[89] L. Li, M. Elhajj, Y. Feng and W. Y. Ochieng, "Machine learning based GNSS signal classification and weighting scheme design in the built environment: a comparative experiment.," *Satellite Navigation,* vol. 4, no. 12, 2023.

[90] N. Kubo, K. Kobayashi and R. Furukawa, "GNSS Multipath Detection Using Continuous Time-Series C/N0," *Sensors,* vol. 20, no. 14, p. 4059, 2020.

[91] T. Brun, Z. Xu, I. Petrunin, R. Wong and R. Grech, "Machine Learning Enhanced Signal Quality Assessment Leveraged with GDOP for GNSS/INS Fusion," in *2024 IEEE International Instrumentation and Measurement Technology Conference (I2MTC*, Glasgow, 2024.

[92] Y. Teng and J. Wang, "New Characteristics of Geometric Dilution of Precision (GDOP) for Multi-GNSS Constellations," *The Journal Of Navigation,* vol. 67, no. 6, 2014.

[93] Y. Wang, J. Xu, R. Yang and X. Zhan, "GNSS Multipath Detection Based on Decision Tree Algorithm in Urban Canyons," in *China Satellite Navigation Conference (CSNC 2021) Proceedings*, Singapore, 2021.

[94] J. Koiloth, D. S. Achanta and P. R. Koppireddi, "ML-based LOS/NLOS/multipath signal classifiers for GNSS in simulated multipath environment," *Aerospace Systems,* 2023.

[95] L. Abaligah, A. Diabat, P. Sumari and A. H. Gandomi, "Applications, Deployments, and Integration of Internet of Drones (IoD): A Review," *IEEE Sensors,* vol. 21, no. 22, pp. 25532 - 25546, 2021.

[96] A. P. Cohen, S. A. Shaheen and E. M. Farrar, "Urban Air Mobility: History, Ecosystem, Market Potential, and Challenge," *IEEE Transactions on Intelligent Transportation Systems,* vol. 22, no. 9, pp. 6074 - 6087, 2021.

[97] P. Geragersian, I. Petrunin, W. Guo and R. Grech, "Multipath Detection from GNSS Observables Using Gated Recurrent Unit," in *2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC)*, Portsmouth, VA, USA, 2022.

[98] "Joint Channel Allocation and Data Delivery for UAV-Assisted Cooperative Transportation Communications in Post-Disaster Networks," *IEEE Transactions on Intelligent Transportation Systems ,* vol. 23, no. 9, pp. 16676 - 16689, 2022.

[99] D. Sivabalaselvamani, D. Selvakarthi, L. Rahunathan, G. Gayathri and M. M. Baskar, "Survey on Improving Health Care System by Implementing an Air Ambulance System with the Support of Drones," in *2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, 2022.

[100] Y. Li, M. M. Karim and R. Qin, "A Virtual-Reality-Based Training and Assessment System for Bridge Inspectors With an Assistant Drone," *IEEE Transactions on Human-Machine Systems,* vol. 52, no. 4, pp. 591 - 601, 2022.

[101] Y. Jiang, S. Pan, Q. Meng, W. Gao, C. Ma, B. Yu and F. Jia, "Robust Kalman Filter Enhanced by Projection Statistic Detector for Multisensor Navigation in Urban Canyon Environment," *IEEE Sensors Journal ,* vol. 23, no. 9, pp. 9832 - 9847, 2023.

[102] C. Reiche, A. P. Cohen and C. Fernando, "An Initial Assessment of the Potential Weather Barriers of Urban Air Mobility," *IEEE Transactions on Intelligent Transportation Systems,* vol. 22, no. 9, pp. 6018 - 6027, 2021.

[103] V. Bottino, G. Alteriis, R. S. Lo Moriello and D. Accardo, "System Architecture Design of a UAV for Automated Cinematography in GNSS-Challenging Scenarios," in *2022 IEEE 9th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, Pisa, Italy, 2022.

[104] D. B. Sesso, L. F. Bismari, A. V. Silvia Neto, P. S. Cugnasca and J. B. Camargo, "Using Data Integrity as an Improvement Characteristic to Assess the Safety of ADS-B-based Systems," in *2015 IEEE International Conference on Dependable Systems and Networks Workshops*, Rio de Janeiro, Brazil, 2015.

[105] N. Zhu, J. Marais, D. Betaille and M. Berbineau, "GNSS Position Integrity in Urban Environments: A Review of Literature," *IEEE Transactions on Intelligent Transportation Systems,* vol. 19, no. 9, pp. 2762 - 2778, 2018.

[106] W. Fang, J. Jiang, S. Lu, Y. Gong, Y. Tao, Y. Tang, P. Yan, H. Luo and J. Liu, "A LSTM Algorithm Estimating Pseudo Measurements for Aiding INS during GNSS Signal Outages," *Remote Sensing,* vol. 12, no. 2, p. 256, 2020.

[107] N. El-Sheimy and Y. Li, "Indoor navigation: state of the art and future trends," *Satellite Navigation,* 2021.

[108] T. Akca and M. DemiRekler, "An Adaptive Unscented Kalman Filter for tightly coupled INS/GPS integration," in *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*, Myrtle Beach, SC, USA, 2012.

[109] M. B. Alatise and G. P. Hancke, "A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods," *IEEE Access,* pp. 39830 - 39846, 2020.

[110] M. W. Givens and C. Coopmans, "A Survey of Inertial Sensor Fusion: Applications in sUAS Navigation and Data Collection," in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, Atlanta, GA, USA, 2019.

[111] H. Li, L. Ao and X. Yan, "Indoor multi-sensor fusion positioning based on federated filtering," *Measurement,* vol. 154, 2020.

[112] X. Wu, Z. Su, L. Li and Z. Bai, "Improved Adaptive Federated Kalman Filtering for INS/GNSS/VNS Integrated Navigation Algorithm," *MDPI Applied Science,* vol. 13, no. 9, 2023.

[113] L. Xu, C. Wei, C. Chingyao, L. Bin and S. Xianghui, "Multi-sensor fusion methodology for enhanced land vehicle positioning," *Information Fusion,* vol. 46, pp. 51-62, 2019.

[114] R. Chen and L. Zhao, "Multi-level autonomous integrity monitoring method for multi-source PNT resilient fusion navigation," *Satellite Navigation,* vol. 4, no. 21, 2023.

[115] W. C. McDermott PL, "Bayesian Recurrent Neural Network Models for Forecasting and Quantifying Uncertainty in Spatial-Temporal Data.," *Entropy,* vol. 21, no. 2, p. 184, 2019.

[116] P. B. S. E. a. F. G. H. Li, "Continual Learning Using Bayesian Neural Networks," *EEE Transactions on Neural Networks and Learning Systems,* vol. 32, no. 9, pp. 4243-4252, 2021.

[117] A. Olivier, M. Shields and L. Graham-Brady, "Bayesian neural networks for uncertainty quantification in data-driven materials modeling," *Computer Methods in Applied Mechanics and Engineering,* vol. 386, 2021.

[118] X. Qiu, F. Wang, Q. Zhang, G. Tao and S. Zhou, "An improved Gaussian process for filling the missing data in GNSS position time series considering the influence of adjacent stations," *Scientific Reports,* vol. 14, 2024.

[119] S. A. Negru, P. Geragersian, I. Petrunin and W. Guo, "Resilient Multi-Sensor UAV Navigation with a Hybrid Federated Fusion Architecture," *MDPI Sensors,* vol. 24, no. 3, 2024.

[120] E. Karami, S. Prasad and M. Shehata, "Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images," in *2015 Newfoundland Electrical and Computer Engineering Conference*, St. johns, Canada, 2015.

[121] M. Muja and D. G. Lowe, "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration," in *International Conference on Computer Vision Theory and Applications*, 2009.

[122] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[123] P. Geragersian, I. Petrunin, W. Guo and G. Raphael, "Uncertainty-based Sensor Fusion Architecture using Bayesian-LSTM Neural Network," in *AIAA SCITECH 2023 Forum*, National Harbour, MD, 2023.

[124] D. Li, L. Marshall, Z. Laing, A. Sharma and Y. Zhou, "Bayesian LSTM With Stochastic Variational Inference for Estimating Model Uncertainty in Process-Based Hydrological Models," *Water Resources Research,* vol. 57, no. 9, 2021.

[125] Y. Mae, W. Kumagai and T. Kanamori, "Uncertainty propagation for dropout-based Bayesian neural networks," *Neural Networks,* vol. 144, pp. 394-406, 2021.

[126] K. Miok, D. Nguyen-Doan, D. Zaharie and M. Robnik-Sikonja, "Generating Data using Monte Carlo Dropout," in *2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)*, Cluj-Napoca, Romania, 2019.

[127] D. Racelis and M. Joerger, "Evaluating New and Current Horizontal Protection Levels in the Baseline ARAIM Algorithm," in *Proceedings of the 35th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2022)*, Denver, Colorado, 2022.

[128] S. Negru, P. Geragersian, I. Petrunin and W. Guo, "Resilient Multi-Sensor UAV Navigation with a Hybrid Federated Fusion Architecture," *MDPI Sensors,* vol. 24, no. 3, p. 981, 2024.

[129] D. K. P. Tan, J. He, Y. Li, A. Bayesteh, Y. Chen, P. Zhu and W. Tong, "Integrated Sensing and Communication in 6G: Motivations, Use Cases, Requirements, Challenges and Future Directions," in *2021 1st IEEE International Online Symposium on Joint Communications & Sensing (JC&S)*, Dresden, Germany, 2021.

[130] F. S. Prol, R. M. Ferre, Z. Saleem, P. Valisuo, C. Pinell, E. S. Lohan, M. Elsanhoury, M. Elmusrati, S. Islam, K. Celikbilek, K. Selvan, J. Yilaho, K. Rutledge, A. Ojala, L. Ferranti, J. Praks, M. Z. H. Bhuiyan, S. Kaasalainen and H. Kuusniemi, "Position, Navigation, and Timing (PNT) Through Low Earth Orbit (LEO) Satellites: A Survey on Current Status, Challenges, and Opportunities," *IEEE Access,* vol. 10, pp. 83971 - 84002, 2022.

[131] I. A. R. W. a. E. B. B. Kantsepolsky, "Exploring Quantum Sensing Potential for Systems Applications," *IEEE Access,* vol. 11, pp. 31569-31582, 2023.

[132] S. Zheng, K. Zeng, Z. Li, Q. Wang, K. Xie, M. Liu and S. Xie, "Improving the Prediction of GNSS Satellite Visibility in Urban Canyons Based on a Graph Transformer," *NAVIGATION: Journal of the Institute of Navigation,* vol. 71, no. 4, 2024.

[133] A. O. B. C. C. B. M. M. A. M. a. A. Z. A. Bazzi, "On the Design of Sidelink for Cellular V2X: A Literature Review and Outlook for Future," *IEEE Access,* vol. 9, pp. 97953-97980, 2021.

[134] K. Cho, B. v. Merrienboer, D. Bahdanau and Y. Bengio, "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches," *arxiv,* vol. 1409, no. 1259, 2014.

[135] S. Xu, I. Petrunin and A. Tsourdos, "Experimental Evaluation of GNSS and IMU Fusion Using Gated Recurrent Unit," in *2022 Integrated Communication, Navigation and Surveillance Conference (ICNS)*, Dulles, VA, USA, 2022.

[136] S. A. Negru, P. Geragersian, I. Petrunin, A. Zolotas and R. Grech, "GNSS/INS/VO fusion using Gated Recurrent Unit in GNSS denied environments," in *AIAA SCITECH 2023 Forum*, National Harbor, MD, USA, 2023.

[137] K. Cho, B. v. Merrienboer, D. Bahdanau and Y. Bengio, "On the Properties of Neural Machine Translation: Encoder–Decoder," *arxiv.org,* 2014.

[138] S. Wang, X. Zhan, Y. Zhai and B. Liu, "Fault Detection and Exclusion for Tightly Coupled GNSS/INS System Considering Fault in State Prediction," *Sensors,* vol. 20, no. 3, p. 590, 2020.

[139] A. E. Mahdi, A. Azouz, A. E. Abdalla and A. Abosekeen, "A Machine Learning Approach for an Improved Inertial Navigation System Solution," *Sensors,* vol. 22, no. 4, p. 1687, 2022.

[140] K. Premkumar and B. V. Manikandan, "Speed control of Brushless DC motor using bat algorithm optimized Adaptive Neuro-Fuzzy Inference System," *Applied Soft Computing,* vol. 32, pp. 403-419, 2015.

[141] R. Sharaf, A. Noureldin, A. Osman and N. El-Sheimy, "Online INS/GPS integration with a radial basis function neural network," *IEEE Aerospace and Electronic Systems Magazine,* vol. 20, no. 3, pp. 8-14, 2005.

[142] J. Oliveira and C. Tiberius, "Quality Control in SBAS: Protection Levels and Reliability Levels," *The Journal of Navigation,* vol. 62, no. 3, pp. 509-522, 2009.

[143] S. Khanafseh, B. Kujur, M. Joerger, T. Walter, S. Pullen, J. Blanch, K. Doherty, L. Norman, L. d. Groot and B. Pervan, "GNSS Multipath Error Modeling for Automotive Applications," in *Proceedings of the 31st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2018)*, Miami, USA, 2018.

[144] D. Rumelhart, G. Hinton and R. Williams, "Learning representations by back-propogating errors," *Nature,* vol. 323, pp. 533-536, 1986.

[145] M. Jordan, Chapter 25 - Serial Order: A Parallel Distributed Processing Approach, North-Holland, 1997, pp. 471-495.

[146] P. Geragersian, I. Petrunin, W. Guo and R. Grech, "Multipath Detection from GNSS Observables Using Gated Recurrent Unit," in *2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC)*, Portsmouth, VA, USA, 2022.

[147] G. Hu, W. Wang, Y. Zhong, B. Gao and C. Gu, "A new direct filtering approach to INS/GNSS integration," *Aerospace Science and Technology,* vol. 77, pp. 755-764, 2018.

[148] M. Elsanhoury, J. Koljonen, P. Valisuo, M. Elmusrati and H. Kuusniemi, "Survey on Recent Advances in Integrated GNSSs Towards Seamless Navigation Using Multi-Sensor Fusion Technology," in *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021)*, St. Louis, Missouri, 2021.

[149] Q. Cheng, R. Sun and J. Wang, "A new historical data based GNSS fault detection and exclusion algorithm for UAV positioning," in *2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, Weihai, China, 2020.

[150] J. Bressler, P. Reisdorf, M. Obst and G. Wanielik, "GNSS positioning in non-line-of-sight context—A survey," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Rio de Janeiro, Brazil, 2016.

[151] Y. Hu, S. Bian, B. Li and L. Zhou, "A Novel Array-Based Spoofing and Jamming Suppression Method for GNSS Receiver," *IEEE Sensors Journal,* vol. 18, no. 7, pp. 2952-2958, 2018.

[152] S. Joardar, T. A. Siddique, S. Alam and M. Hossam-E-Haider, "Analyses of different types of errors for better precision in GNSS," in *2016 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, Dhaka, Bangladesh, 2017.

[153] Y. Kamatham, "Estimation, analysis and prediction of multipath error for static GNSS applications," in *2018 Conference on Signal Processing And Communication Engineering Systems (SPACES)*, Vijayawada, India, 2018.

[154] P. D. Groves, Z. Jiang, B. Skelton, P. A. Cross, L. Lau, Y. Adane and I. Kale, "Novel Multipath Mitigation Methods using a Dual-Polarization Antenna," in *Proceedings of the 23rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2010)*, Portland, OR, 2010.

[155] M. Maqsood, S. Gao, T. W. C. Brown, M. Unwin, R. d. v. V. Steenwijk and J. D. Xu, "A Compact Multipath Mitigating Ground Plane for Multiband GNSS Antennas," *IEEE Transactions on Antennas and Propagation ,* vol. 61, no. 5, pp. 2775 - 2782, 2013.

[156] F. Fohlmeister, A. Iliopoulos, M. Sgammini, F. Antreich and J. A. Nossek, "Dual polarization beamforming algorithm for multipath mitigation in GNSS," *Signal Processing,* vol. 138, pp. 86-97, 2017.

[157] F. Graas and S. Ugazio, GNSS Signal Quality Monitoring, 2020.

[158] Manosas-Caballu, A. Lee Swindlehurst and G. Seco-Granados, "Power-based Capon beamforming: Avoiding the cancellation effects of GNSS multipath," *Signal Processing,* p. 107891, 2021.

[159] N. Kubo, K. Kobayashi and R. Furukawa, "GNSS Multipath Detection Using Continuous Time-Series C/N0," *Sensors,* p. 4059, 2020.

[160] R. Sun, L. Fu, Q. Cheng, K.-W. Chiang and W. Chen, "Resilient Pseudorange Error Prediction and Correction for GNSS Positioning in Urban Areas," *IEEE Internet of Things Journal ,* vol. 10, no. 11, pp. 9979 - 9988, 2023.

[161] S. Jeong and C. S. Sin, "GNSS Interference Signal Generation scenario for GNSS Interference Verification Platform," in *2015 15th International Conference on Control, Automation and Systems (ICCAS)*, Busan, Korea (South), 2015.

[162] J. Chung, C. Gulcehre, K. H. Cho and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS 2014 Deep Learning and Representation Learning Workshop*, Montreal, 2014.

[163] J. Hu, T. Fan, L. Han, W. Xu and J. Wu, "Research on UAS Safety and Security using System Thinking," in *2020 3rd International Conference on Unmanned Systems (ICUS)*, Harbin, China, 2020.

[164] O. K. Isik, J. Hong, I. Petrunin and A. Tsourdos, "Integrity Analysis for GPS-Based Navigation of UAVs in Urban Environment," *MDPI Robotics,* vol. 9, no. 3, 2020.

[165] P. Geragersian, I. Petrunin, W. Guo and R. Grech, "An INS/GNSS fusion architecture in GNSS denied environment using gated recurrent unit," in *AIAA SCITECH 2022 Forum*, San Diego, CA, 2022.

[166] Y. He, J. Li and J. Liu, "Research on GNSS INS & GNSS/INS Integrated Navigation Method for Autonomous Vehicles: A Survey," *IEEE Access,* vol. 11, pp. 79033 - 79055, 2023.

[167] W. S. Flenniken, J. H. Wall and D. M. Bevly, "Characterization of Various IMU Error Sources and the Effect on Navigation Performance," in *Proceedings of the 18th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2005)*, Long Beach, CA, 2005.

[168] D. Unsal and K. Demirbas, "Estimation of deterministic and stochastic IMU error parameters," in *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*, Myrtle Beach, SC, USA, 2012.

[169] S. Bhattacharyya and D. Gebre-Egziabher, "Kalman filter–based RAIM for GNSS receivers," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 51, no. 3, pp. 2444-2459, 2015.

[170] C. Zhuang, H. Zhao, S. Hu, X. Meng and W. Feng, "A Novel GNSS Fault Detection and Exclusion Method for Cooperative Positioning System," *IEEE Transactions on Vehicular Technology,* vol. 72, no. 4, pp. 4697 - 4712, 2022.

[171] B. Zheng, P. Fu, B. Li and X. Yuan, "A Robust Adaptive Unscented Kalman Filter for Nonlinear Estimation with Uncertain Noise Covariance," *MDPI Sensors,* vol. 18, no. 3, p. 808, 2018.

[172] A. Chalvatzaras, I. Pratikakis and A. A. Amanatiadis, "A Survey on Map-Based Localization Techniques for Autonomous Vehicles," *IEEE Transactions on Intelligent Vehicles,* vol. 8, no. 2, pp. 1574 - 1596, 2022.

[173] S. A. K. Tareen and Z. Saleem, "2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)," in *A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK*, Sukkur, Pakistan, 2018.

[174] H. Wang, S. Pan, W. Gao, Y. Xia and C. Ma, “Multipath/NLOS Detection Based on K-Means Clustering for GNSS/INS Tightly Coupled System in Urban Areas,” *MDPI Micromachines,* vol. 13, no. 7, 2022.

[175] X. Ye, F. Song, Z. Zhang and Q. Zeng, “A Review of Small UAV Navigation System Based on Multisource Sensor Fusion,” *IEEE Sensors,* vol. 23, no. 17, pp. 18926 - 18948, 2023.

[176] R. G. Brown, “A Baseline GPS RAIM Scheme and a Note on the Equivalence of Three RAIM Methods,” *ION Navigation Journal,* vol. 39, no. 3, pp. 301-316, 1992.

[177] Y. Alkendi, L. Seneviratne and Y. Zweiri, “State of the Art in Vision-Based Localization Techniques for Autonomous Navigation Systems,” *IEEE Access,* vol. 9, pp. 76847 - 76874, 2021.

[178] R. Kapoor, S. Ramasamy, A. Gardi and R. Sabatini, “UAV Navigation using Signals of Opportunity in Urban Environments: A Review,” *Energy Procedia,* vol. 110, pp. 377-383, 2017.