

**DEEP LEARNING-ENHANCED AUTONOMOUS AERIAL AND GROUND
ROBOTICS USING UWB AND LIDAR IN GNSS-DENIED ENVIRONMENTS**

ZAHRA ARJMANDI

A DISSERTATION SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN EARTH AND SPACE SCIENCE

YORK UNIVERSITY
TORONTO, ONTARIO

AUGUST 2024

© Zahra Arjmandi, 2024

Abstract

Over the last decade, advancements in Unmanned Aerial Vehicle (UAV) technology and Artificial Intelligence (AI) have offered promising solutions for enhancing navigation and positioning capabilities under challenging conditions. Despite significant progress, widespread adoption in the field has been limited due to the complexities of integrating various technologies and ensuring real-time, reliable data processing. This thesis addresses these challenges by developing a comprehensive framework that combines advanced data collection platforms, deep learning algorithms, and novel fusion methods to improve UAV positioning accuracy and reliability.

The core of this research is the development of the Quality Driven Drone (Q-Drone) Ultra-Wideband (UWB) benchmark dataset, which includes data collected from a UAV equipped with five UWB sensors across five diverse sites, covering indoor, outdoor, and semi-outdoor environments over a distance of 4 km. This dataset provides a standardized benchmark for testing UAV positioning systems, allowing researchers to develop and validate their algorithms under varied conditions. The publicly available dataset supports advancements in UAV navigation and positioning research.

Building on this foundation, the thesis introduces an incremental smoothing approach that integrates high-rate and low-rate UWB measurements with inertial data within a unified pose graph framework. This method, featuring an "add-after-eliminating" strategy, reduces Mean Absolute Error (MAE) by approximately 0.2 meters compared to baseline multilateration methods and achieves a 0.3-meter MAE reduction compared to two-factor pose graph methods.

The research further advances positioning accuracy with the development of the Deep-Learning-based Dynamic Covariance Prediction in Pose Graphs (DeepCovPG) framework,

which combines a Variational Autoencoder (VAE) with a Long Short-Term Memory (LSTM) network to predict and incorporate dynamic covariances into the pose graph. This deep learning-based approach results in a 48% reduction in The Root Mean Squared Error (RMSE) and a 51% reduction in Range Covariance RMSE, with significant improvements of 0.41 meters in tunnels and 0.23 meters in fields. Additionally, the framework achieves a 26% reduction in multilateration RMSE and a 32% reduction in multilateration Covariance RMSE, demonstrating substantial improvements over baseline methods.

The thesis also explores Light Detection and Ranging (LiDAR)-based positioning and mapping and proposes the Inferred Attention Fusion (INAF) fusion method, which dynamically selects relevant information from geometric and AI-based odometry techniques. This method excels in managing large angle rotations and motion distortions, achieving a 3.90% improvement over direct fusion methods and a 0.25% improvement over attention-based fusion methods. The INAF fusion method's adaptability to various driving conditions enhances accuracy in both straight and dynamic environments, showcasing the effectiveness of integrating LiDAR technology with advanced fusion techniques.

The research conducted in this thesis addresses several scientific challenges, each aimed at improving navigation and positioning systems. By developing a comprehensive dataset, refining multilateration techniques, and integrating advanced deep-learning models, this work makes substantial contributions to UAV positioning. Additionally, the novel INAF fusion method enhances vehicular positioning using LiDAR and AI-based odometry techniques.

To the dream of a just and equal world!

Acknowledgements

The achievements of the past five years would not have been possible without the exceptional guidance and mentorship of Dr. Gunho Sohn and Dr. Costas Armenakis. Their inspiring ideas and challenges have shaped my journey and growth, both academically and personally. I am profoundly grateful for their unwavering support, patience, and encouragement, which have meant more to me than words can express. There is no way I can fully thank them for all they have done for me.

I extend my deepest gratitude to my supervisory committee, Dr. Ping Wang and Dr. Liam Butler, for their reliable and persistent support throughout these five years. Their expert guidance and constructive feedback have been invaluable in shaping my research and academic growth. I also appreciate the valuable time and insights contributed by my examination committee, whose thoughtful evaluations and recommendations have significantly enhanced the quality of my work.

A special thanks goes to Dr. Jungwon Kang. His hard work, remarkable personality, and professional dedication have greatly influenced me professionally and personally. His encouragement and support have been a constant source of inspiration. I am deeply grateful for his mentorship and the invaluable guidance he has provided throughout my journey.

I deeply thank Dr. Yeonjeong Jeong, Dr. Mozhdeh Shahbazi, Dr. Yujia Zhang, and Dr. Connie Ko for their invaluable professional guidance. Their support has been instrumental in my academic journey, and they have been inspiring role models whose dedication and passion for their work have greatly influenced me.

I want to express my deepest gratitude to my husband, Ali, my greatest and most dependable friend and a pillar of support. His love, patience, and encouragement have

been pivotal in my life. I am also deeply grateful to my parents, whose endless love and unwavering support have been a constant source of comfort and strength; to my dear sister and brother—Tahereh and Mohammad—whose encouragement has been invaluable; and to my beloved nieces and nephews—Parsa, Hosna, Hossein, and Salma—whose joy and affection have been a cherished part of this journey. Their combined support has made every challenge easier and every achievement more meaningful.

I am also blessed with incredible friends—Bahar, Dr. Fatemeh, Dr. Salimeh, Dr. Prinaz, Dr. Sanaz, Dr. Maryam, Aleena, and Mahya—who stood by me during my lowest moments and provided tireless support. I especially remember Raha and Atefeh, whose memories and influences continue to inspire me even though they are no longer with us. Their friendship has been a source of great strength and comfort.

My lab mates Seyed Mostafa Ahmadi, Kunwoo Park, Mohammad Moein Sheikholeslami, Afnan Ahmad, Jacob Yoo, Hyungju Lee, Kivanc Babacan, and Ali Hassan played a crucial role in my journey. Their constant support and friendship have been invaluable.

Finally, I want to thank Dana, my furry companion, whose gentle purrs brought me joy and comfort during the toughest times.

Table of Contents

Abstract	ii
Acknowledgements	v
Table of Contents	vii
List of Tables	xii
List of Figures	xiv
Abbreviations	xx
1 Introduction	1
1.1 Motivation	1
1.2 Research Objectives	10
1.3 Overview of the Thesis	12
2 Literature Review	15
2.1 UAV Systems and Benchmark Dataset	15
2.2 Filtering and Incremental Smoothing Positioing	18
2.3 Deep-Learning-based UAV Positioing	19
2.4 Integrating AI and Geometric Methods for LiDAR SLAM	21
3 Q-Drone System Design and Benchmark Dataset	25
3.1 Q-Drone System Overview	27

3.1.1	Q-Drone system	28
3.1.2	System Calibration	31
3.1.3	Baseline Positioning Method	35
3.1.4	Data Acquisition Results	38
3.2	Q-Drone Benchmark Dataset: Comprehensive Online Resource	47
3.3	Summary	50
4	Incremental Smoothing with Ranges and Multilateration	51
4.1	Overview	52
4.2	System Overview	53
4.3	Filtering out Outliers in UWB Range Data	54
4.4	Multilateration for Direct Measurement of UAV Position	55
4.5	UAV Positioning Using Factor Graph	56
4.5.1	Factor Graph Framework	57
4.5.2	Factors	58
4.5.3	Online Operation of UAV Positioning	60
4.6	Experimental Results	60
4.6.1	Experimental Setup	61
4.7	Summary	67
5	DeepCovPG: Dynamic Covariance Prediction for UAV Positioning	68
5.1	Overview	69
5.2	System Overview	70
5.2.1	UWB Anchor Configuration	70
5.2.2	Integrated Range and IMU Data Collection	71
5.2.3	Ground Truth Data Collection and Covariance Calculation	72
5.2.4	DeepCovPG’s Three-Step Positioning Framework	74
5.3	Problem Formulation	74
5.4	VAE-LSTM Multitask Network	75

5.4.1	Formulation of the Likelihood Loss Function	76
5.5	Dynamic Covariance Modeling	77
5.5.1	Multilateration Positioning Algorithm	77
5.5.2	Dynamic Covariance Modeling	78
5.6	Dynamic Covariance Pose Graph	80
5.7	Experimental Results	83
5.7.1	UAV System and UWB Sensor	83
5.7.2	UAV Experimental Setup	83
5.7.3	Network Implementation and Training	84
5.7.4	Results and Discussion: Neural Network Performance	84
5.7.5	Results and Discussion: Positioning Accuracy	90
5.8	Summary	90
6	Integrating AI and Geometric Methods for LiDAR SLAM	93
6.1	Overview	94
6.2	Methodology	95
6.2.1	Odometry Pipelines	95
6.2.2	Ensemble Learning vs. Fusion Networks	96
6.2.3	Advantages of Inferred Attention Over Incremental and Attention Based Approaches	98
6.2.4	Proposed Inferred Attention Fusion	100
6.2.5	Backend Optimization	100
6.3	Explaining System Modules	101
6.3.1	ICP-QE: ICP Quaternion Encoder	103
6.3.2	DL-PE: DL Point Cloud Encoder	104
6.3.3	Odometry Prediction Block	105
6.3.4	Loss Function	106
6.4	Dual Quaternions	107
6.5	Fusion Modules	109

6.5.1	Direct Fusion	110
6.5.2	Attention Based Fusion	110
6.5.3	Inferred Attention Fusion(Propose Method)	111
6.6	Model Training	114
6.6.1	Data Preparation	114
6.6.2	Data Transformation	115
6.6.3	Model Architecture	117
6.6.4	Training Procedure	120
6.6.5	Hyperparameter Tuning	122
6.7	Backend Loop Closure and Optimization	124
6.7.1	Loop Closure	124
6.7.2	Graph Optimization	125
6.8	Results	127
6.8.1	Performance Evaluation of the Proposed Method	127
6.8.2	Comparison with Classical SLAM	127
6.8.3	Comparison with AI SLAM	129
6.8.4	Comparison with State-of-the-Art Methods	129
6.8.5	Qualitative Analysis of INAF Fusion Technique	131
6.9	Summary	138
7	Conclusions and Future Work	140
7.1	Conclusions	140
7.1.1	Q-Drone UWB Benchmark Dataset	140
7.1.2	Incremental Smoothing-Based Method	141
7.1.3	DeepCovPG Framework	141
7.1.4	Integrating AI and Geometric Methods for LiDAR SLAM	141
7.2	Future Work	142
7.2.1	Enhancing the Q-Drone UWB Benchmark Dataset	142
7.2.2	Improving the Incremental Smoothing-Based Method	143

7.2.3	Advancing the DeepCovPG Framework	143
7.2.4	Refining the INAF Fusion Method	143
	Bibliography	145
	Appendices	161
A	UWB Positioning Experimental Results Figures	162
A.1	Benchmark Dataset	163
A.2	MLAT Results	168
A.3	Incremental Smoothing Result	173
A.4	DeepCovPG Result	178
B	LiDAR Positioning Experimental Results Figures	182

List of Tables

1.1	Comparative Analysis of Wireless Sensors for Positioning. Adapted from sources on wireless sensor technology [26].	6
1.2	Summary of positioning systems and contributions studied in this thesis. . .	11
3.1	Details of the Q-Drone benchmark dataset acquired for this research, including environment types, locations, number of data and covered area.	38
3.2	Q-Drone Benchmark and UWB Range Accuracy	41
3.3	Comparison of UWB Datasets.	47
4.1	The MAE of multilateration outcomes (Unit: meter).	61
5.1	Comparison of RMSE between the Refined Range calculated by DeepCovPG and its Raw Value (left), and the MLAT Pose calculated using the Refined Range, along with its value using the raw ranges (right).	85
5.2	Comparison of RMSE between the Predicted Covariance Matrix of range calculated by DeepCovPG and its static value (left) and the Covariance Matrix of MLAT Pose calculated by Covariance Propagation, along with its static value (right).	88
5.3	The MAE of DeepCovPG outcomes (UNIT: METER).	91
6.1	Statistics of KITTI Lidar Odometry Dataset Sequences 00 to 10.	114
6.2	Positioning results comparing ICP, a geometric method, and ICP-QE, a deep learning network.	129

6.3	Positioning results comparing ICP, a geometric method, and ICP-QE, and DL-PE two deep learning networks.	130
6.4	Performance Comparison of Various Methods on KITTI Sets. The table presents translational and rotational errors for deep learning-based and geometry -based methods.	130
6.5	Positioning results comparing ICP-QE + DL-PE, with different Fusion methods.	135

List of Figures

1.1	Key Modules in an Autonomous Vehicle System.	2
1.2	Frequency vs. Power Consumption for Different Wireless Technologies. . . .	7
3.1	Q-Drone Data Acquisition System in Different Environments.	27
3.2	Comparison of Payload Systems on Q-Drone X and Q-Drone 360. (a) depicts the payload system of Q-Drone X, while (b) illustrates the payload system of Q-Drone 360.	29
3.3	Robotic total station tracks UAV-mounted prism for accurate positioning. Leica Nova MS60 and Trimble VX provide high-precision reference data for diverse environments. The setup includes four UWB anchors, the ground control system and UAV.	30
3.4	Customized communication: DJI Matrice 100's SDK allows flexibility. Intel NUC computer communicates in real-time via Micro-USB, linking the flight controller, sensors, and UWB antenna. GRZ101 prism measures XYZ of UAV.	31
3.5	UWB Coordinate System: (A) UWB network frame, (B) UAV body frame, and (C) Total Station frame.	33
3.6	UAV Body Frame with UWB tag.	34
3.7	The MLAT baseline positioning method uses a multilateration algorithm that processes UWB data and IMU height data. After preprocessing and multilateration, the data undergoes Levenberg-Marquardt optimization and further outlier removal to determine the final position.	35

3.8	UWB Anchors configurations of the Q-Drone benchmark dataset in various environments: a) Indoor, b) Field, c) Building, d) Bridge, and e) Tunnel. . .	39
3.9	Range difference between UWB and ground truth.	42
3.10	Multilateration results and ground truth for various datasets: a) indoor, b) field (square pattern), and c) field (zigzag pattern).	44
3.10	Multilateration results and ground truth for various datasets: d) building, e) bridge, and f) tunnel.	45
3.11	Data Structure.	49
4.1	Description of the proposed method for UAV positioning based on incremental smoothing, which seamlessly integrates two types of UWB-based measurements and inertial measurement into a unified factor graph framework. Here, a node indicates a state to be estimated, i.e., UAV position.	52
4.2	An overview of the proposed Incremental Smoothing System system for Q-Drone.	54
4.3	Acquired range over time between a UWB tag and a certain UWB anchor, and the result of the proposed outlier filtering algorithm, where the result is represented by a colour (green: normal data, red: outlier) in each data. The set of data in the dashed circle indicates true outliers trapped in NLOS cases.	54
4.4	Factor graph representation of the proposed algorithm. The proposed algorithm has obtained the state variables over time for UAV position. $\mathbf{x}_{\text{pos}}^k$, orientation $\mathbf{x}_{\text{ori}}^k$, velocity $\mathbf{x}_{\text{v}}^k$ of the UAV, and biases $\mathbf{x}_{\text{b}}^k$ for an accelerometer and a gyroscope of the IMU. The measurements and the state variables are associated with factors. Here, three factors were used to handle multilateration-based position, single UWB range, and inertial measurements, respectively.	57
4.5	Description of the <i>Add-After-Eliminating</i> strategy for adding a multilateration factor to the factor graph.	59
4.6	Comparison between estimated positions by proposed pose graph (blue) and ground truth (red) across various environments: (a) indoor, (b) field, (c) building, (d) bridge, and (e) tunnel.	63

4.7	Box plot illustrating the MAE and quartiles of incremental pose graph results across diverse environmental sets, including indoor(red), field(yellow), building(green), bridge(pink), and tunnel(purple).	65
4.8	Example case trajectories. The colour indicates the XYZ error. (a) An example of evenly distributed error is typical of field and open indoor spaces. (b) A building dataset with high error near the building walls. (c) The bridge dataset has a medium error close to the bridge columns. (d) Tunnels have high errors near the walls and during take-off moments due to high speed.	66
5.1	Overview of the DeepCovPG framework for UAV position estimation. The DeepCovPG framework dynamically predicts the covariance of the factors, enabling adaptive uncertainty handling within a unified DCPG.	71
5.2	Architecture of the VAE-LSTM multitask network for UAV trajectory estimation. The network combines VAE units with an LSTM structure to refine sensor data and dynamic covariance matrices, facilitating a novel approach to trajectory refinement by accurately modelling sensor uncertainties.	76
5.3	Illustration of incremental smoothing pose graph representing the proposed DCPG with dynamic covariance.	80
5.4	Illustrating the network’s capability in rectifying raw range data corrupted by multipath interference for indoor environment set 1.	86
5.5	Covariances depicted in a 4x4 matrix for indoor dataset set 1, showcasing ground truth (cyan), static covariance (red), and predicted covariance by VAE-LSTM network (black), illustrating the alignment between network outcome and ground truth compared with static covariance.	89
5.6	Comparison between estimated positions by DeepCovPG (blue) and ground truth (green) across various environments: (a) indoor, (b) field, (c) building, (d) bridge, and (e) tunnel. The complete trajectories across all datasets and environments are available in the appendix.	92

6.1	Pipelines illustrating three configurations: two single-branch setups (ICP-QE and DL-PE) and one fused setup combining both ICP-QE and DL-PE branches, followed by the Odometry Prediction Block and the Loop Closure and Optimization Block.	96
6.2	a) Ensemble Learning vs. b) Network Fusion. In ensemble learning, multiple networks are trained independently and remain frozen during combination. In contrast, network fusion involves training multiple networks together, allowing for integrated learning and mutual adaptation.	97
6.3	Adaptation of networks to new data across different methods: (a) Incremental learning, where the network is updated with new data over time; (b) Attention-based networks, which utilize attention mechanisms to focus on relevant parts of the input; and (c) Inferred Attention (Ours), where network parameters are continuously adapted based on feedback from the environment.	99
6.4	Overview of the three phases in my network: (a) Training phase, where the network learns from data; (b) Prediction phase, where the trained network is frozen (indicated by a snowflake symbol) and used for inference, and (c) Prediction phase with Inferred Attention Fusion (INAF), where some network parameters continue to adapt based on feedback from the environment. . . .	101
6.5	The proposed SLAM system, illustrating three configurations: (a) Fusion of ICP-QE and DL-PE, (b) ICP-QE and (c) DL-PE followed by the Odometry Prediction and Loop Optimization Blocks	102
6.6	Illustration of three fusion methods: (a) Direct Fusion, where inputs are concatenated directly; (b) Attention Fusion, which leverages multihead self-attention mechanisms; and (c) Inferred Attention Fusion (INAF) (Ours), which adapts network parameters based on environmental feedback for improved performance.	109

6.7	Illustration of the chain rule in calculating error gradients for the INAF fusion method, tracing back from the output error to the features derived from the branches $\theta_{\text{ICP-QE}}$ and $\theta_{\text{DL-PE}}$	112
6.8	The left side displays unprocessed point clouds with the central local axis (RGB representing the XYZ axis). On the right side, the input vertices are presented and colour-coded based on their distance from the origin to enhance visualization[138].	116
6.9	Scan Context overview: A point cloud from a single 3D scan is first encoded into scan context. Next, an N_r -dimensional vector is derived from the scan context to retrieve the nearest candidates and construct the KD tree. Finally, the retrieved candidates are compared to the query scan context, with the loop being identified as the candidate meeting the acceptance threshold and closest to the query [130].	125
6.10	In a 2D pose graph problem example, pose variables are depicted as white circles, a prior is represented by a blue factor, black factors denote four odometry measurements, and a red factor indicates a single loop closure measurement [131].	126
6.11	Histogram of all KITTI sequences showing the rotation angle in degrees. . .	132
6.12	Examples of four sequences from the KITTI dataset (00, 05, 07, and 09). The trajectories illustrate situations where the rotation angle exceeds 1 degree, highlighted in red.	133
6.13	Visualization of the learned weights under different conditions for INAF fusion. In scenario a, the vehicle drives straight, while in scenario b, a moving object (a car) is present. Scenario d depicts a turning maneuver. The weights of selected features in the geometric branch and LiDAR branch illustrate the increasing importance of different branches in various scenarios, highlighting the adaptive nature of the fusion process.	137

A.1	Indoor datasets collected using the Q-Drone system, with trajectories generated based on ground truth.	163
A.2	Field datasets collected using the Q-Drone system, with trajectories generated based on ground truth.	164
A.3	Building datasets collected using the Q-Drone system, with trajectories generated based on ground truth.	165
A.4	Bridge datasets collected using the Q-Drone system, with trajectories generated based on ground truth.	166
A.5	Tunnel datasets collected using the Q-Drone system, with trajectories generated based on ground truth.	167
A.6	Indoor datasets collected using the Q-Drone system, with trajectories generated based on MLAT.	168
A.7	Field datasets collected using the Q-Drone system, with trajectories generated based on MLAT.	169
A.8	Building datasets collected using the Q-Drone system, with trajectories generated based on MLAT.	170
A.9	Bridge datasets collected using the Q-Drone system, with trajectories generated based on MLAT.	171
A.10	Tunnel datasets collected using the Q-Drone system, with trajectories generated based on ground truth.	172
A.11	Indoor datasets collected using the Q-Drone system, with trajectories generated using incremental smoothing method.	173
A.12	Field datasets collected using the Q-Drone system, with trajectories generated using incremental smoothing method.	174
A.13	Building datasets collected using the Q-Drone system, with trajectories generated using incremental smoothing method.	175
A.14	Bridge datasets collected using the Q-Drone system, with trajectories generated using incremental smoothing method.	176

A.15 Tunnel datasets collected using the Q-Drone system, with trajectories generated using incremental smoothing method.	177
A.16 Field datasets collected using the Q-Drone system, with trajectories generated using the DeepCovPG method.	178
A.17 Building datasets collected using the Q-Drone system, with trajectories generated using the DeepCovPG method.	179
A.18 Bridge datasets collected using the Q-Drone system, with trajectories generated using the DeepCovPG method.	180
A.19 Tunnel datasets collected using the Q-Drone system, with trajectories generated using the DeepCovPG method.	181
B.1 The figure compares the results of ICP-QE odometry, and the ground truth poses from sequences 00 to 09 of the KITTI dataset.	183
B.2 The figure compares the results of DL-PE odometry and the ground truth poses from sequences 00 to 09 of the KITTI dataset.	184
B.3 The figure compares the results of INAF odometry, and the ground truth poses from sequences 00 to 09 of the KITTI dataset. Loop Closure included. . . .	185

List of Abbreviations

AI Artificial Intelligence

BLE Bluetooth Low Energy

CNN Convolutional Neural Network

DCM Dynamic Covariance Modeling

DCPG Dynamic Covariance Pose Graph

DeepCovPG Deep-Learning-based Dynamic Covariance Prediction in Pose Graphs

DL Deep Learning

DNN Deep Neural Networks

DOP Dilution of Precision

EKF Extended Kalman Filter

FCC Federal Communications Commission

GNSS Global Navigation Satellite System

ICP Iterative Closest Point

IMU Inertial Measurement Unit

INAF Inferred Attention Fusion

LEO Low Earth Orbit

LiDAR Light Detection and Ranging

LOAM Lidar Odometry and Mapping

LSTM Long Short-Term Memory

MAE Mean Absolute Error

MAP Maximum A Posteriori

MDS Multidimensional Scaling

MLAT Multilateration

MLP Multi-Layer Perceptrons

MSE Mean Squared Error

NLOS Non-Line-of-Sight

Q-Drone Quality Driven Drone

RFID Radio Frequency Identification

RMSE The Root Mean Squared Error

ROS Robot Operating System

SDK Software Development Kit

SLAM Simultaneous Localization and Mapping

TD_{oA} Time Difference of Arrival

ToF Time of Flight

TWR Two-Way Ranging

UAV Unmanned Aerial Vehicle

UGV Unmanned Ground Vehicle

UWB Ultra-Wideband

VAE Variational Autoencoder

VSLAM Visual SLAM

Chapter 1

Introduction

In this thesis, I explore the challenges and solutions related to positioning in autonomous vehicles and their navigation systems. I discuss the importance of accurate localization and mapping and introduce novel techniques to enhance these systems in environments where traditional Global Navigation Satellite System (GNSS) are limited or unavailable. This includes examining advanced technologies like UWB, LiDAR and their integration with other sensors to improve performance and reliability. The goal is to provide a comprehensive understanding of achieving precise navigation and mapping in complex and demanding conditions.

1.1 Motivation

Positioning is a foundational aspect of autonomous vehicles, essential for ensuring safe and efficient navigation. Autonomous systems face risks such as collisions, operational inefficiencies, and failures without accurate positioning. As illustrated in Figure 1.1, positioning is one of the key modules in an autonomous vehicle system, working in conjunction with other critical components to maintain vehicle integrity and safety.

In this thesis, the focus is specifically on the localization and mapping of autonomous vehicles. This includes developing and improving algorithms for mobile mapping systems to accurately and efficiently navigate and map unknown environments. By concentrating on these aspects, the research aims to enhance the performance and reliability of autonomous

navigation systems.



Figure 1.1 Key Modules in an Autonomous Vehicle System.

The Growing Role of Unmanned Aerial Vehicles (UAVs)

UAVs, known as drones, have quickly become essential tools across many industries, revolutionizing tasks ranging from inspection, reality capture, and tracking to safety and delivery. UAVs can navigate diverse environments using methods unattainable through traditional means, launching new productivity and safety levels.

In the inspection field, UAVs allow for close-up views and assessments of infrastructure like bridges, power lines, and wind turbines, significantly reducing the need for hazardous manual inspections [1]. Additionally, drones are automating flights over railroad tracks, using technology to detect rails and follow track lines in real-time, resulting in more frequent and safer inspections than traditional methods [2]. In agriculture, drones provide precise monitoring and analysis of crop health, leading to optimized resource usage and improved products [3]. Similarly, drones capture reality on-site in construction, using automated drones and ground robots to gather data, thereby improving worker safety and operational efficiency [4]. UAVs have also significantly influenced logistic services, where drones fly through cities to make deliveries and access areas unreachable by ordinary vehicles, thus shortening lead times [5]. Furthermore, drones are invaluable in emergency and disaster response, providing real-time situation awareness and delivering supplies to areas inaccessible by road [6]. These capabilities improve operational efficiency across various sectors, enhance safety, and reduce costs, highlighting the truly transformational impact of UAVs.

Advancements in drone tracking, such as the virtual-reality training system developed by MIT researchers, allow drones to visualize complex environments while flying in empty

spaces. This system can potentially reduce crashes during training sessions, improving safety and proficiency [7]. Furthermore, drones are vital in ensuring the success and safety of infrastructure projects, offering cost-effective inspection solutions and delivering high-quality imaging and positioning data [8]. Additionally, in response to the COVID-19 pandemic, drone companies have explored delivery applications, particularly in delivering medical supplies to frontline workers. This proactive response underscores drones' potential to meet critical needs during emergencies and streamline delivery operations [9].

One of the main reasons for using UAVs is their efficiency in maneuvering and functioning in areas where conventional procedures have difficulty. Conventional positioning systems relying on GNSS signals often encounter difficulties in the event of signal obstructions or unavailability, hence the need for alternative solutions. This provides a platform on which the introduction of Q-Drone system hardware and software components becomes essential.

GNSS Challenges in Various Environments

Localization forms the core of UAV operation, and several techniques are adopted for accurately locating the vehicle in space. Until now, GNSS-aided positioning has been the core of UAV navigation. GNSS have revolutionized positioning and navigation by providing universal coverage and high accuracy. The current state-of-the-art GNSS technology, including systems like GPS, GLONASS, Galileo, and BeiDou, offers precise positioning in various applications ranging from autonomous vehicles to personal navigation devices [10].

While highly effective in many applications, GNSS technology faces significant challenges in specific environments that can severely impact performance. In urban canyons and dense forests, GNSS signals often encounter multipath effects, where signals reflect off buildings, trees, and other surfaces, leading to inaccuracies in position estimation. This occurs because the receiver struggles to distinguish between direct and reflected signals. Additionally, GNSS signals are susceptible to electromagnetic interference from various electronic devices, such as cellular phones and radio transmitters, which can distort or weaken the signals, further complicating accurate positioning [11] [12].

Atmospheric conditions also impact GNSS accuracy. Variations in the ionosphere and

troposphere can cause delays in signal propagation, leading to position calculation errors [13]. Moreover, GNSS signals are vulnerable to jamming, whether intentional or unintentional, which can block or disrupt signals and pose operational risks [14]. Spoofing, where false GNSS signals are transmitted to deceive the receiver, further complicates the integrity and security of GNSS systems [15].

Emerging technologies such as Low Earth Orbit (LEO) satellite constellations are being explored to address these limitations. LEO satellites promise enhanced positioning accuracy and reduced latency due to their proximity to Earth, which could mitigate some of the current GNSS challenges and improve performance in challenging environments [16]. While focusing on other technologies for positioning and mapping, this research recognizes the evolving landscape of GNSS. It considers future advancements like LEO satellites as potential improvements in autonomous navigation.

Given these limitations, relying solely on GNSS for positioning in challenging environments is often inadequate. Alternative positioning methods are essential to ensure reliable operations, especially in GNSS-denied settings. Techniques such as trilateration (e.g. UWB) technology, combined with Inertial Measurement Unit (IMU), LiDAR, and visual odometry, provide robust solutions. By integrating data from multiple sensors, these methods enhance the accuracy and reliability of positioning systems in complex environments where GNSS signals may be obstructed or unreliable [17, 18, 19].

Alternative Wireless Sensors and the Choice of UWB

In GNSS-denied environments, alternative positioning methods are essential to ensure reliable and accurate navigation. Various wireless sensors can serve as replacements or supplements to GNSS, including UWB, Wi-Fi, Bluetooth Low Energy (BLE), and Radio Frequency Identification (RFID). Each technology has distinct advantages and limitations that influence its suitability for different applications.

UWB technology is distinguished by its high precision in three-dimensional positioning. The technology operates across a broad frequency spectrum, which enables acceptable spatial resolution. UWB utilizes Time of Flight (ToF) techniques such as Time Difference of

Arrival (TDoA) and Two-Way Ranging (TWR) to achieve positioning accuracy typically in the range of 10 to 30 centimetres [20]. Studies have demonstrated that UWB systems can achieve centimetre-level accuracy, which is critical for applications requiring detailed spatial awareness, such as autonomous navigation and industrial automation.

UWB technology exhibits superior resilience to interference compared to other wireless communication methods. The wide frequency range and low energy per bit of UWB signals allow it to handle interference and multipath effects effectively (Figure 1.2). UWB signals are less susceptible to degradation caused by reflections and noise, owing to their short pulse duration and low power spectral density. Research highlights that UWB maintains robust performance even in environments with significant multipath interference and Non-Line-of-Sight (NLOS) conditions. The versatility of UWB makes it suitable for both indoor and outdoor environments. Its ability to penetrate obstacles and perform well in challenging conditions is particularly advantageous for complex environments such as indoor settings with dense structures and outdoor areas with potential signal blockage. This adaptability ensures that UWB can provide reliable positioning in various scenarios where other technologies might fail [21, 22, 23].

One of the critical benefits of UWB technology is its low power consumption, which is crucial for battery-operated devices like UAVs. UWB systems are designed to operate with minimal power usage, thus extending the operational time of UAVs without significantly depleting their battery resources. This characteristic is significant for maintaining long-duration missions and reducing the frequency of battery replacements or recharges [24].

UWB provides high temporal resolution, which allows for precise tracking of rapid movements and fast changes in position. This capability is essential for UAVs that often need to maneuver through dynamic and rapidly changing environments. The ability to detect and respond to quick movements ensures that the positioning system remains accurate and responsive during complex flight operations [25].

The theoretical upper bound for UWB-based positioning accuracy is often influenced by the signal's time-of-flight measurement, with ideal conditions allowing for accuracies

around 1-10 cm. However, practical factors such as multipath effects, signal attenuation, and calibration errors typically result in a 10-30 cm real-world accuracy range. My work achieved approximately 20 cm accuracy in 3D positioning, highlighting the impact of these real-world challenges on achievable accuracy.

Combining UWB with IMUs enhances continuous positioning even during rapid movements or sudden changes. While UWB provides accurate distance measurements, IMUs offer detailed data on motion dynamics, such as acceleration and rotation. This integration enables consistent and reliable navigation by addressing the limitations of each technology when used alone.

UWB Corrects IMU Drift: UWB technology helps correct the drift that commonly affects IMUs. Over time, IMUs can accumulate errors due to biases and noise, leading to positional drift. The precise measurements from UWB assist in recalibrating the IMU, thus minimizing drift and ensuring the long-term accuracy of the positioning system.

IMU Captures Detailed Motion Dynamics: IMUs provide detailed insights into motion dynamics, such as rapid accelerations and rotations. While UWB excels at distance measurement, combining it with IMU data offers a comprehensive view of the UAV's movement. This integration enhances positioning accuracy and better handling of dynamic flight conditions.

Table 1.1 Comparative Analysis of Wireless Sensors for Positioning. Adapted from sources on wireless sensor technology [26].

Technology	Accuracy	Range (m)	Battery Lifetime
UWB	<10 centimeters	10-50 meters	Moderate to Long
Wi-Fi	<1.5 m	30-100 meters	Moderate
BLE	<30 cm	10-50 meters	Long
RFID	<1 meter (typical)	1-10 meters	Long
LEO Satellites	0.1-1 m	Global Coverage	Moderate
5G (Fifth Generation)	0.1-1 m	Several kilometers	Moderate

While this research acknowledges the potential of advanced positioning technologies such as 5G and LEO satellites, the focus remains on UWB technology due to its specific advantages in the context of GNSS-denied environments.

5G technology, known for its high data rates and low latency, is primarily designed to

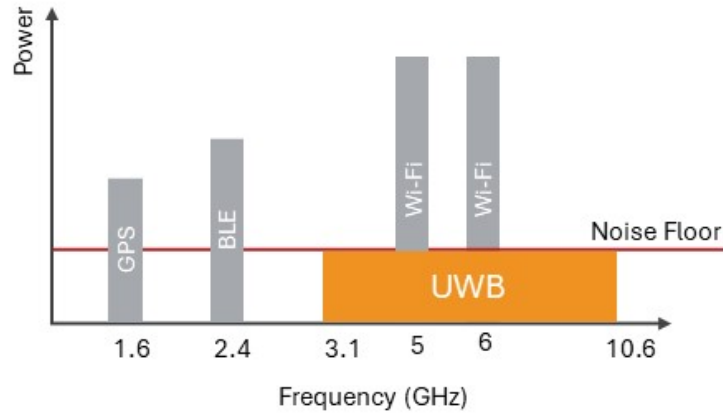


Figure 1.2 Frequency vs. Power Consumption for Different Wireless Technologies.

enhance mobile communications rather than specialized positioning tasks. While 5G can improve positioning accuracy in urban environments with dense infrastructure, its reliance on terrestrial base stations limits its effectiveness in remote or GNSS-denied areas. Additionally, the deployment and operational costs of 5G infrastructure can be prohibitive, especially in less developed or rugged environments.

LEO satellites offer advanced capabilities for global coverage and high-precision positioning. However, they are not yet widely deployed or integrated into mainstream positioning systems. The operational complexity, data transmission challenges, and reliance on satellite networks make LEO satellites less practical for real-time, high-frequency applications in GNSS-denied environments. Consequently, the study focused on UWB technology due to its proven effectiveness in such scenarios. It offers high accuracy and reliability without the limitations associated with terrestrial infrastructure or satellite networks.

Use Cases for UWB-Based Positioning

Indoor Navigation and Asset Tracking: UWB technology is particularly effective for indoor navigation and asset tracking due to its high precision and ability to manage challenges like multipath interference. In settings such as warehouses and hospitals, UWB allows for accurate real-time tracking of assets and personnel, which enhances operational efficiency and safety. For instance, in healthcare facilities, UWB systems are used to monitor equipment

and staff, which helps streamline workflows and ensure better management of resources [27].

Autonomous Vehicles: In the field of autonomous vehicles, UWB improves localization and navigation, especially in environments where other positioning systems might not perform well. UWB is beneficial for autonomous drones and ground vehicles as it provides precise location data in indoor and cluttered settings. It works well alongside other sensors, such as LiDAR and IMU, to enhance overall navigation accuracy and reliability [28].

Smart Building Systems: UWB is also useful in smart building systems by offering precise indoor positioning for various applications. It supports real-time location-based services, such as smart lighting and HVAC (heating, ventilation, and air conditioning) systems, by adapting to changes in occupancy. This capability improves energy efficiency and user comfort within buildings [29].

Industrial Automation: In industrial automation, UWB plays a key role in accurately positioning machinery and robotic systems. It is essential for operations involving automated guided vehicles (AGVs) and robotic arms, where precise localization and real-time adjustments are critical for both efficiency and safety. UWB helps ensure that these systems operate smoothly and accurately within industrial environments [30].

Augmented Reality (AR): For augmented reality (AR) applications, UWB enhances spatial accuracy and interaction quality. It precisely tracks users and objects within AR environments, improving virtual experiences' realism and responsiveness. This accurate tracking is crucial for creating immersive and interactive AR applications [31].

Pose-Graph Optimization and Covariance Management Pose-graph optimization is crucial in improving the accuracy of UWB-aided UAV positioning by converting sensor data into actionable positioning information [32, 33, 34]. It represents the UAV's trajectory with nodes on a graph and their spatial and temporal relationships with edges, aiming to minimize location estimate uncertainty. However, managing the dynamic uncertainties present in the sensor data poses a significant challenge [35, 36]. Conventional methods, which often treat these uncertainties as static, overlook the fluctuating nature of environmental factors—such as signal noise and multipath reflections—that continuously impact sensor readings [37].

Variability in UWB range measurements requires an advanced pose-graph optimization that addresses dynamic uncertainties, including signal noise, environmental changes, and clock drift that affects UWB timing [38, 39, 40].

UGV Positioning and Mapping with Deep Neural Networks: Leveraging LiDAR and Innovative Fusion Techniques

LiDAR technology plays a crucial role in autonomous vehicle systems, particularly in positioning and mapping. Its high-resolution, three-dimensional data provides detailed information about the environment, essential for creating accurate maps and enabling precise navigation. For autonomous vehicles, accurate mapping is vital as it supports the development of detailed environmental models that aid in obstacle detection, path planning, and safe navigation. LiDAR’s ability to capture fine details of the surroundings, even in complex or dynamic environments, complements other sensor data such as UWB and IMU, enhancing overall system performance. While my research initially aimed at UAV positioning and mapping, the lack of an available LiDAR dataset for UAVs led me to utilize the KITTI dataset for Unmanned Ground Vehicle (UGV)s. This allowed me to explore and validate the integration of LiDAR data in positioning and mapping tasks, thereby extending the benefits of LiDAR to different vehicle platforms.

The "Self-supervised Learning of LiDAR Odometry for Robotic Applications" [41] research has significantly inspired my work, particularly in the context of my INAF fusion method. This pioneering study demonstrated the potential of self-supervised learning to achieve high accuracy in LiDAR odometry by leveraging the inherent structure of LiDAR data without the need for extensive labelled datasets. The approach taken in this research to derive a loss function from the self-supervised learning framework resonated with my objectives of enhancing positioning accuracy in GNSS-denied environments. Inspired by this, I integrated their innovative loss function into my supervised network to serve as feedback for parameter optimization. This integration allowed me to effectively quantify and minimize the error in positioning parameters, thereby improving the robustness and accuracy of my proposed

INAF fusion method.

Additionally, my work drew inspiration from the research "Selective Sensor Fusion for Neural Visual-Inertial Odometry," [42], which explores fusing different sensors in various ways to optimize odometry. This approach allowed me to fuse different methods rather than just sensor data. Building on this concept, I designed the INAF fusion method, which employs a novel way of integrating diverse sources of information. This selective and dynamic fusion approach enabled me to achieve superior performance in terms of accuracy and robustness, particularly in environments where traditional methods struggle.

1.2 Research Objectives

This research delves into various aspects of UAV technology, particularly enhancing positioning capabilities in challenging environments. The primary objectives of this study are as follows:

1. Create a unique UWB benchmark dataset to facilitate robust testing of UAV positioning systems across diverse environmental conditions.
2. Advance multilateration techniques to improve UAV positioning accuracy in GNSS-denied environments.
3. Integrate deep learning with traditional pose graph optimization to enhance positional estimations.
4. Develop a novel fusion method that combines geometric and AI-based techniques for improved odometry.

A summary of the positioning systems studied in this thesis is provided in Table 1.2 for quick reference.

Development of a Comprehensive UWB Benchmark Dataset

The first goal of this research is to establish the Q-Drone UWB benchmark dataset. This standardized dataset is designed to test UAV positioning systems robustly across diverse

Table 1.2 Summary of positioning systems and contributions studied in this thesis.

Chapter	Key Topics	Objectives	Methods	Contribution	Sensors
3	UWB benchmark dataset	Establish a dataset across various conditions	Multilateration	Provided a comprehensive dataset across five distinct sites.	UWB/IMU 
	Z. Arjmandi, J. Kang, K. Park and G. Sohn, <i>"Benchmark Dataset of Ultra-Wideband Radio Based UAV Positioning,"</i> in ITSC, 2020 (Published) K. Park, J. Kang, Z. Arjmandi, M. Shahbazi and G. Sohn, <i>"Multilateration Under Flip Ambiguity For Uav Positioning,"</i> in ISPRS, 2020 (Published)				
4	Incremental Smoothing with Range and Multilateration	Refine positioning techniques to enhance UAV positioning accuracy in GNSS-denied environments.	Pose Graph Smoothing, Multilateration	Improved positioning accuracy through an innovative smoothing technique.	UWB/IMU 
	J. Kang, K. Park, Z. Arjmandi, G. Sohn, M. Shahbazi and P. Menard, <i>"Ultra-Wideband Aided UAV Positioning Using Incremental Smoothing with Ranges and Multilateration,"</i> in IROS, 2020. (Published)				
5	Dynamic covariance prediction	Enhance the adaptability and accuracy of UAV positioning in real-world scenarios.	Deep Learning (VAE, LSTM), Covariance	Enhanced the adaptability of positioning in challenging scenarios.	UWB/IMU 
	Z. Arjmandi, J. Kang, G. Sohn and C. Armenakis, M. Shahbazi, <i>"DeepCovPG: Deep-Learning-based Dynamic Covariance Prediction in Pose Graphs for Ultra-Wideband-Aided UAV Positioning"</i> in CASE, 2024. (Published) Z. Arjmandi, J. Kang, G. Sohn and C. Armenakis, M. Shahbazi, <i>"Dynamic Covariance Prediction in Pose Graphs using Deep Learning for Ultra-Wideband-Aided UAV Positioning"</i> in IJRR Journal, 2024 (Ready to be submitted)				
6	AI and geometric methods integration, LiDAR SLAM	Synergize geometric and AI-based odometry methods to enhance positioning accuracy in challenging environments.	Deep Learning (ResNet, LSTM), Fusion	Enhanced UGV positioning accuracy, proving highly effective in diverse and challenging environments.	LiDAR 
	Z. Arjmandi, G. Sohn and C. Armenakis, <i>"Dynamic Recalibration in LiDAR SLAM: Integrating AI and Geometric Methods with Real-Time Feedback Using INAF Fusion"</i> in RA-L Journal, 2024. (Drafted)				

environmental conditions. Utilizing the Q-Drone system equipped with five UWB sensors, the dataset is developed over five distinct sites, including indoor areas and various outdoor

settings. Additionally, the benchmark includes IMU and ground truth position data. This comprehensive dataset aims to provide a realistic benchmark for researchers to enhance the development and testing of algorithms under varied conditions.

Advancement of Multilateration for UAV Positioning

The second goal involves refining multilateration techniques to enhance UAV positioning accuracy in GNSS-denied environments. This includes developing an innovative smoothing technique integrating high-rate and low-rate UWB range measurements with inertial data within a unified factor graph. This method significantly improves positioning accuracy, as demonstrated through extensive experimental validation.

DeepCovPG: Deep-Learning-based Dynamic Covariance Prediction in Pose Graphs for Ultra-Wideband Aided UAV Positioning

Building upon previous work in pose graph optimization, the third goal introduces DeepCovPG, a framework that integrates a VAE with an LSTM network to predict dynamic covariances. This shift from static to dynamic models significantly enhances the adaptability and accuracy of UAV positioning in real-world scenarios, as evidenced by a 51% reduction in RMSE and improvements in MAE through extensive testing.

INAF - Inferred Attention Fusion: A Novel Approach Integrating AI and Geometric Methods for LiDAR SLAM

The final goal introduces INAF, a novel technique that synergizes geometric and AI-based odometry methods. Unlike the rest of the research, which primarily utilizes UWB, INAF employs LiDAR sensors to enhance localization accuracy by dynamically selecting relevant data. This approach proves highly effective in diverse and challenging environments, showcasing the potential for significant advancements in UAV positioning technology.

1.3 Overview of the Thesis

This thesis comprises six chapters, each contributing to enhancing UAV positioning and SLAM technologies.

Chapter One outlines the thesis’s background, research objectives, contributions, and scope. It provides an orientation to the document’s structure and content, detailing the focus of each subsequent chapter.

Chapter Two focuses on a literature review of previous studies and research relevant to the thesis topic. This chapter identifies key findings and gaps in the existing knowledge, forming the foundation for the current study.

Chapter Three provides a detailed description of the Q-Drone system based on UWB technology for positioning. It explains the architecture, components, and features of the Q-Drone system and discusses the development of a benchmark dataset for a UAV equipped with UWB sensors. The details of this work are discussed in the paper "*Benchmark Dataset of Ultra-Wideband Radio Based UAV Positioning*," published in ITSC, 2020.

Chapter Four introduces a new smoothing technique combined with UWB technology to achieve better positioning accuracy for UAVs. Traditional positioning methods that update UAV location estimates based on single UWB range measurements often fail under challenging conditions. This chapter describes a method integrating high-rate, weak-constraint single 1D UWB ranges and low-rate, strong-constraint multilateration results with inertial measurements in an incremental smoothing-based framework, forming a unified factor graph. Extensive experiments demonstrate significant improvements in positioning accuracy over conventional techniques. The findings are detailed in the paper "*Ultra-Wideband Aided UAV Positioning Using Incremental Smoothing with Ranges and Multilateration*," published in IROS, 2020.

Chapter Five advances UAV navigation research by introducing the DeepCovPG framework, which integrates deep learning techniques with the previously developed incremental smoothing method to address dynamic challenges in UAV navigation. This chapter represents a significant advance by adopting dynamic models that adjust to environmental variations and measurement inaccuracies, using a combination of a VAE and an LSTM network to enhance UWB range data quality. Experimental results show a 51% reduction in RMSE and notable improvements in MAE, demonstrating the practical applications and effectiveness

of DeepCovPG in real-world scenarios. The details are covered in the paper "*DeepCovPG: Deep-Learning-based Dynamic Covariance Prediction in Pose Graphs for Ultra-Wideband-Aided UAV Positioning*," published in CASE, 2024. An extended analysis is forthcoming in the paper "*Dynamic Covariance Prediction in Pose Graphs using Deep Learning for Ultra-Wideband-Aided UAV Positioning*," ready to be submitted to IJRR Journal.

Chapter Six introduces INAF, a novel fusion method that merges geometric and deep learning odometry techniques to improve localization capabilities. Unlike previous chapters focusing on UWB positioning systems, this chapter emphasizes LiDAR sensors. INAF leverages the robustness of geometric methods and the adaptability of deep learning techniques to dynamically select relevant data, enhancing localization accuracy in challenging environments. This work is discussed in the draft paper "*Dynamic Recalibration in LiDAR SLAM: Integrating AI and Geometric Methods with Real-Time Feedback Using INAF Fusion*," planned for publication in RA-L Journal, 2025.

The final chapter concludes the thesis by summarizing the significant findings and contributions. It discusses the broader implications of the research and suggests future directions for UAV positioning and navigation technology. This chapter highlights the impact and potential applications of the developed methodologies.

Chapter 2

Literature Review

2.1 UAV Systems and Benchmark Dataset

Since 2005, the Federal Communications Commission (FCC) has allowed and regulated the applications of UWB systems for unlicensed purposes [43]. This has created huge interest in utilizing UWB sensors to locate moving vehicles, which has been explored in numerous research studies. This application, commonly referred to as 3D position estimation through UWB ranges, is used to locate the accurate position of an object within three-dimensional space. Fusing UWB with other sensors, for example, inertial and visual or LiDAR, can further improve the positioning accuracy.

Methods for solving the estimation problems of the position include non-linear least squares, geometric algorithms, filtering-based methods such as the Extended Kalman Filter (EKF), and smoothing-based ones. The current work presented in these papers has focused on refining these well-established methodologies. For instance, in our work [44], multilateration was applied to resolve geometric ambiguity, leading to error reduction. This can be done by utilizing the space information from several anchors of UWB to remove ambiguities while estimating the position.

On the other hand, [45] proposed an EKF algorithm to incorporate UWB and Inertial Measurement Units with IMU indoors in experiments, showing improvement. This uses the benefits that UWB and IMU bring together to track position continuously, therefore not

suffering from the limitations of each individual sensor on its own. Using data fusion from multiple sensors, such as UWB and IMU, or visual sensors, one can develop a more reliable and precise positioning system for various applications ranging from robotics and autonomous vehicles to indoor navigation systems.

Other research, such as [46], has demonstrated the effectiveness of UWB technology in indoor positioning, achieving impressive accuracies of 10 cm in 2D and 20 cm in 3D. Such precision enhances various applications, from warehouse inventory management to supporting navigation for augmented reality setups. The capability to find locations accurately in indoor settings, where GNSS signals may be unreliable or unavailable, is essential for advancing autonomous technologies and evolving intelligent indoor environments.

In outdoor experiments, [47] integrated gyroscope data with UWB to estimate the dynamic state of a quadcopter, allowing precise control and navigation in a challenging outdoor environment. Likewise, [48] integrated radar, IMU, and UWB sensors to compensate for the absence of GNSS signals that would provide an all-weather robust solution to outdoor positioning in places where conventional satellite-based systems are probably covered or not reliable. These sensor fusion advances highlight the importance of combining multiple sensing modalities to achieve better positioning accuracy in a broader class of operating environments, laying the groundwork for the general acceptance of UWB technology in indoor and outdoor applications.

similarly my paper [49] explained in Chapter 4 introduces a novel method of integrating UWB and IMU within a unified factor graph framework for UAV positioning. This method significantly enhances accuracy and robustness, particularly in environments where traditional positioning systems often fail.

Employing a factor graph framework significantly enriches UAV positioning strategies by facilitating the integration of diverse sensor modalities and environmental constraints. For example, [50] explores the integration of UWB and IMU within indoor navigation environments, enhancing positioning accuracy through tightly coupled methods. Similarly, [51] demonstrates the effectiveness of multi-sensor fusion using factor graphs to optimize

navigation solutions in real-time. Furthermore, [52] work on UAV localization in GNSS-denied environments shows the potential of factor graphs to integrate multiple data sources like IMU, GNSS, and visual odometry to produce robust and accurate positioning outcomes. These studies confirm that a holistic approach, incorporating various sensors and dynamic environmental data into a unified factor graph model, substantially improves the precision and reliability of UAV navigation in complex settings.

Significant effort has been dedicated to generating UWB sensor data available for the scientific community, enabling signal processing capabilities and facilitating the development of advanced positioning algorithms. For example, the datasets made publicly available by the Institute for Dynamic Systems and Control at ETH Zurich [53, 54] have proved a priceless asset for all researchers in the area. These datasets include very diversified kinds of UWB signals within many different environments and setups, making it possible to test and adapt the algorithms under such conditions. Additionally, datasets like the LocURa4IoT testbed [55] have been important in advancing UWB research by providing datasets collected in various indoor scenarios. These datasets serve as benchmarks for evaluating the performance of positioning algorithms and sensor fusion techniques in real-world settings.

Although different datasets for UWB serve different purposes, for instance, indoor positioning with sports postures [56], and studies involving the effects of wall materials on non-line-of-sight communication [57], there exists a large gap in the existence of benchmark datasets purposely designed for research focusing on UAV positioning within both indoor and outdoor environments. The absence of such general data sources creates big challenges for researchers aiming to develop and validate innovative and new positioning methods specifically tailored for UAVs. Diverse datasets and environmental conditions, flight scenarios, and sensor configurations are necessary to make the performance assessments reasonable.

2.2 Filtering and Incremental Smoothing Positioning

Numerous studies have used EKF in estimating the 3D positions of UAVs, leveraging UWB sensors and supplementary inertial measurements for positioning. This approach, widely recognized in the research community, applies EKF to integrate and refine data from multiple sources, enhancing the accuracy and reliability of UAV navigation [58, 59, 60, 61].

Besides the traditional filtering-based approaches such as EKF, recent interest in research of UAV positioning includes smoothing-based techniques. These include methods based on graph optimization, refining the estimate of positions using the whole trajectory of the UAV instead of only measurements. By formulating the positioning problem as a graph optimization task, smoothing-based approaches can exploit temporal and spatial correlations in sensor data to good advantage, resulting in improvements in the accuracy and consistency of the estimated UAV trajectory [62, 63, 64, 65].

Despite their effectiveness, the currently available methods based only on individual low-dimensional UWB range measurements often have significant challenges with estimating UAV positions, especially in hard operational conditions like obstacles, multipath effects, and signal attenuation. In such scenarios, a single range measurement is likely to give a weak constraint to the solution space, which generates inaccuracies and uncertainties in the estimated UAV position. These errors are further magnified in dynamic environments because of the complexities added through UAV motion, such as changes in relative positioning and variations in signal propagation characteristics during motion. Studies, for instance, those in [66], [67], and [68], have shown that the integration of UWB with other sensors, such as IMU, and employment of developed algorithms, such as factor graphs, significantly improve the robustness and reliability of positioning systems under such hostile conditions.

Signal processing advancements, such as machine learning approaches and Bayesian estimation techniques, offer potent solutions for extracting useful information from raw sensor measurements to improve the positioning accuracy of UAV in challenging or uncertain environments. Notably, machine learning has been applied to enhance UAV navigation by

optimizing real-time applications [69], and deep learning techniques have been specifically utilized to maximize UAV positioning accuracy, demonstrating high levels of accuracy in various scenarios [70]. Additionally, Bayesian optimization has effectively fine-tuned EKF parameters, ensuring more precise attitude estimations in UAVs [71]. Together, these integrated efforts are developing innovative solutions for UAV positioning capabilities.

This thesis builds upon the concept of multilateration, a technique similar to that utilized in GNSS. Multilateration involves directly calculating the three-dimensional position of a UAV by triangulating range data from multiple anchors. I propose leveraging the multilateration outcome as a robust constraint in smoothing approaches. In this framework, each individual range measurement functions as a high-rate measurement with a relatively weak constraint. In contrast, the multilateration outcome acts as a low-rate measurement with a strong constraint, enhancing the overall accuracy and reliability of the UAV positioning system. By integrating multilateration-derived constraints into smoothing algorithms, I aim to exploit the spatial information provided by multiple anchors more effectively, resulting in improved performance, especially in challenging environments characterized by obstacles and signal interference.

2.3 Deep-Learning-based UAV Positioning

Deep Neural Networks (DNN)s have changed the domain of sensor data quality improvement, excelling at tasks such as denoising, reconstruction, and prediction of sensor data [72]. These networks are particularly adept at learning complex relationships within the data, which has been leveraged to enhance the quality of measurements from sensors like LiDAR and UWB, crucial for UAV positioning. For instance, [69] discuss the application of machine learning techniques to improve UAV navigation, [67] address the reduction of NLOS errors in UWB sensors using deep learning, and [73] demonstrate the use of DNNs in processing LiDAR data for accurate UAV positioning. These advancements indicate a significant shift towards more reliable and precise UAV positioning capabilities, paving the way for their expanded use in complex operational environments.

Pose graph optimization provides a unified mathematical framework to solve various estimation problems and is particularly robust when handling uncertainty [74]. It offers a systematic approach for integrating multiple sensor measurements and constraints, thus presenting an ideal solution for complex sensor fusion tasks. Pose graphs are especially useful for systems requiring high accuracy and confidence in their estimates, like UAV positioning systems. [75, 76, 77]

Incorporating DNN techniques into pose graph optimization frameworks is a growing trend. For instance, [78] utilize unsupervised learning to enhance lidar features for probabilistic trajectory estimation, demonstrating the ability of DNNs to refine sensor data processing in autonomous systems. Similarly, [79] developed the DPC-Net, which applies deep learning to correct pose estimations, effectively integrating DNNs with existing geometric localization methods to increase the accuracy of pose corrections. However, these advancements have not yet fully addressed the unique challenges posed by UWB-based UAV positioning systems, particularly in environments with significant signal interference and physical obstructions. This study introduces a novel approach that merges the data processing power of DNNs with the mathematical thoroughness used in pose graph optimization, aiming to enhance the precision and reliability of UAV positioning in such challenging scenarios.

The optimization of the pose graph is extremely important to enhance the accuracy of the positioning of the UWB-aided UAV by converting sensor data into actionable information about the position of the UAV [32, 33, 34]. In the method, a graph representing the trajectory of a UAV with nodes and their inter-spatial and temporal relations by edges is designed to reduce uncertainty in the location estimate. It is, however, quite challenging to handle dynamic uncertainties in sensor data [35, 36]. However, conventional methods regard such uncertainties as static and do not capture the dynamic nature of environmental factors, for example, signal noise and multipath reflections, which vary dynamically with time but affect sensor measurements [37]. The UWB range measurement variability demands advanced pose-graph optimization for dynamic uncertainties, including signal noise, changes of environmental states, and clock drift that affects UWB timing [38, 39, 40].

Despite progress in separate technologies, there is still no complete and dependable method for accurately determining the position of UAVs in complicated environments, particularly those with issues like signal interference and obstacles blocking direct sight. This study addresses this need by presenting a new approach that effectively merges the data processing power of DNNs with the mathematical methods used in pose graph optimization. This combination aims to improve the precision and dependability of UAV positioning in challenging situations.

2.4 Integrating AI and Geometric Methods for LiDAR SLAM

Simultaneous Localization and Mapping (SLAM) is a general problem in robotics where an environment map is built with robot localization on that map. Significant milestones have marked its evolution, the first concept being suggested by Randall Smith in 1990 for the incremental estimation of a robot’s posture and landmarks positions [80], using EKF. In 2006, Durrant-Whyte and Bailey proposed the word SLAM, where they defined a complete probabilistic theoretical formulation for SLAM problems [80].

Recent progress in SLAM has been accelerated with sensor technology, methodological developments, and architectural improvements. Today, SLAM systems use a variety of sensors, including LiDAR, cameras, and IMUs [80]. The state estimation methods have developed from the initial filter-based approaches to graph optimization [80]. Moving from single-threaded systems to multi-threaded ones constitutes an architectural improvement providing efficiency and scalability [80]. Integrating multiple sensors has transformed SLAM from a prototype into a technology now essential for many robotic applications.

GNSS-based SLAM, which relies on the GNSS, is widely adopted for its ability to provide accurate and globally consistent position data. However, this leads to faults or failures in determining the position when obstacles shadow or reflect the signals from the GNSS. GNSS-based SLAM often combines with other sensors or technologies to tackle these challenges and

achieve reliable and precise localization and mapping [81, 82].

Visual SLAM (VSLAM) has taken off quickly because it is low-cost and could easily be fused with other sensors to understand more about the environment [83]. Traditional VSLAM has covered a lot; still, some hard challenges remain in some environments. Synergies between deep learning and SLAM have attracted attention for the novel solutions to be achieved over the challenges mentioned above [83]. Semantic SLAM, leveraging high-level environmental information, promises enhanced environmental understanding for robots [83]. The combination of deep learning and SLAM in semantic VSLAM represents a promising road for future research, enabling robots to provide more sophisticated assistance in diverse scenarios [83].

LiDAR-based SLAM is a mainstream research method for autonomous navigation and positioning in large-scale environments [81]. It has been extensively developed in automated driving, mobile robots, forestry surveying, and urban surveying [84]. Yet, LiDAR-based SLAM systems can experience performance degradation impacting both localization and mapping capabilities in challenging environments characterized by high dynamic motion or limited features [81]. A multitude of LiDAR-based multi-sensor fusion SLAM approaches could solve these problems, and research works are emerging [81]. Moreover, the research works related to the development process of LiDAR-based multi-sensor fusion SLAM and the latest research are highlighted [81].

IMU-based SLAM, a technology that combines Inertial Measurement Unit sensors for motion tracking with Simultaneous Localization and Mapping techniques, is vital across various fields, especially for mapping indoor environments [85]. Unlike GNSS, which may not work indoors, SLAM can accurately determine the position and surroundings of a device. IMU-based SLAM has advantages over other methods like Visual and LiDAR-based SLAM: it's less affected by textureless surfaces and certain orientations that can challenge those systems [85]. IMU data helps establish scale in monocular SLAM setups and provides crucial pitch and roll measurements for global positioning [86].

DNN based SLAM has emerged as a significant advancement in the field of computer

vision, with applications in autonomous driving, indoor robotics, mixed reality, and more [87]. DNN-based SLAM systems, such as DNS SLAM[87] and DNN-SLAM [88], have demonstrated superior performance in rendering quality and scene reconstruction for static environments compared to traditional dense SLAM [88]. These systems leverage the inherent smoothness and coherence encoded in Multi-Layer Perceptrons (MLP) weights, making them suitable for sequential tracking and mapping tasks[87]. However, they encounter tracking drift and mapping errors in real-world scenarios with dynamic interferences [88]. To address these issues, DNN-SLAM integrates semantic features and proposes a feature point segmentation method that combines semantic features with a mixed Gaussian distribution model [88]. Experimental results demonstrate that DNN-SLAM can robustly track and produce high-quality reconstructions in dynamic environments while appropriately preserving potential dynamic objects [88]. Therefore, DNN-based SLAM systems are paving the way for truly autonomous robots able to operate within real-world environments [89].

Conventional geometry-based methods for performing odometry as part of the SLAM problem using LiDAR data often involve minimizing the differences between consecutive point clouds by aligning them. One widely used alignment method is the Iterative Closest Point (ICP) algorithm [90], which iteratively minimizes the distance between the points in the source scan and the corresponding points in the reference point cloud. Another promising geometry-based method is Lidar Odometry and Mapping (LOAM) [91], which employs feature-based techniques to evaluate the smoothness of local surfaces, detect edges and surfaces, and utilize them to improve alignment accuracy.

However, geometric-based methods have their limitations. They typically assume a static world where only the robot is in motion, and they are sensitive to imperfect sensor measurements, requiring careful tuning of each module. Additionally, while deep learning methods [92] show promise in overcoming these assumptions and potentially improving performance, they have not yet surpassed the accuracy of geometric algorithms when constructing rich and dense maps. Deep learning approaches often struggle with generalization to unobserved motions and heavily rely on accurate motion initialization. Recent research has explored

hybrid approaches that combine classical geometric estimators with learned components to address these challenges. For instance, LO-Net [93] leverages deep learning methods to provide initial estimates before the mapping stage. At the same time, DMLO [94] proposes a hybrid strategy that learns to construct direct and non-iterative registration between consecutive scans. Additionally, deep learning techniques can be applied for pose correction, augmenting the performance of traditional odometry algorithms such as visual odometry [95, 96]. These advancements promise to enhance the accuracy and robustness of SLAM systems, particularly in challenging and dynamic environments.

Transitioning into more integrated solutions, ensemble learning, and fusion networks are two advanced strategies that leverage the strengths of multiple systems to enhance overall model performance. While ensemble learning employs various models to improve decision accuracy and robustness, fusion networks take a more integrated approach by combining data from multiple sources within a neural network’s architecture. This allows for deeper and more comprehensive feature extraction and learning. For instance, fusion networks have been shown to outperform traditional ensemble methods in complex tasks that involve integrating different types of data, such as image and text classification [97, 12]. Studies like those presented in [98] and [99] demonstrate that fusion networks not only enhance the capabilities of individual models but also offer a more unified approach to processing and synthesizing heterogeneous data, making them particularly suited for the dynamic requirements of SLAM systems in challenging environments.

Chapter 3

Q-Drone System Design and Benchmark Dataset

This chapter comprehensively explains the practical exploration undertaken in developing the Q-Drone system and compiling a benchmark dataset. These foundational activities are crucial for establishing the framework for the subsequent theoretical discussions.

My contributions to this research were multifaceted, encompassing collaborative and individual efforts. Key aspects of my involvement included:

System Design and Development: I played a central role in designing and developing the Q-Drone 360 system. This task involved complex engineering and integration of various components.

Experimental Setup: The experimental setup was extensive and methodologically rigorous. This phase involved configuring the Q-Drone systems for diverse operational scenarios and carefully planning the test environments. The experiments were conducted across various indoor, outdoor, and semi-outdoor locations to comprehensively evaluate the system's performance.

Data Collection: Data collection was done with high precision. The Q-Drone, equipped with five UWB sensors, gathered a substantial dataset under varied conditions. This process involved detailed flight operations and careful data recording to ensure the dataset's accuracy and reliability.

Post-Processing and Analysis: I processed and analyzed data after post-collection.

This included data refinement, sensor calibration, and quality assurance, ensuring the data was robust and suitable for subsequent analysis.

Development of Positioning Methods: I developed the positioning method utilized in this research, integrating various algorithms and refining the approach based on the collected data. This development was critical for enhancing the accuracy and effectiveness of the positioning system.

As a postdoctoral researcher in our lab, Dr. Jungwon Kang was instrumental in the initial design of the Q-Drone and in establishing the experimental framework. His expertise and dedication were crucial in setting up the experiments and ensuring the successful collection of high-quality data.

Kunwoo Park Managed the drone flight operations and developed an alternative positioning method using the Extended Kalman Filter (EKF), detailed in:

K. Park, J. Kang, Z. Arjmandi, M. Shahbazi, and G. Sohn, "*Multilateration Under Flip Ambiguity For UAV Positioning*," in ISPRS, 2020 (Published).

Dr. Mozhdeh Shahbazi and the Centre de géomatique du Québec (CGQ) Contributed significantly to the engineering and calibration of the Q-Drone 360 system. Dr. Shahbazi's expertise, alongside the support and resources provided by CGQ, was vital in optimizing the system's performance and ensuring precise calibration of the sensors.

The collaborative efforts and extensive experimental work have significantly advanced the practical application of UWB sensor-based positioning methods and provided a robust benchmark dataset for evaluating navigation systems in GNSS-denied environments. The key findings based on the research presented in this thesis were published in my paper:

Z. Arjmandi, J. Kang, K. Park, and G. Sohn, "*Benchmark Dataset of Ultra-Wideband Radio Based UAV Positioning*," in ITSC, 2020.

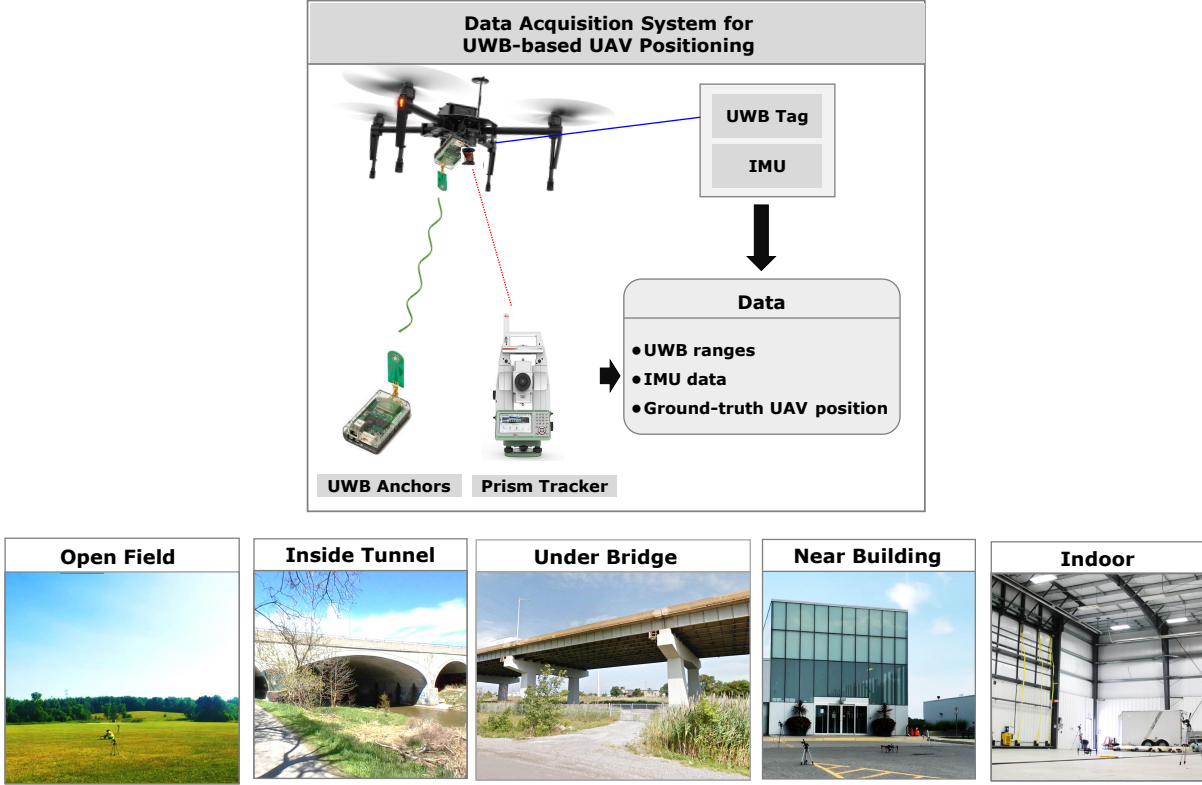


Figure 3.1 Q-Drone Data Acquisition System in Different Environments.

3.1 Q-Drone System Overview

This section overviews my methodology for constructing a comprehensive UAV positioning dataset. As depicted in Figure 3.1, the system comprises several components, including the UAV platform with onboard sensors, UWB anchors, a robotic total station serving as a prism tracker, and a communication system facilitating interaction among these modules. I introduce the Q-Drone system, equipped with five UWB sensors, highlighting its design and deployment in various environments. The data acquisition process, including the systematic flight patterns executed by the Q-Drone, is then outlined. The baseline positioning method is also introduced, serving as a standardized reference. This section lays the groundwork for a detailed exploration of the steps taken to ensure the dataset’s richness, diversity, and reliability, which is crucial for subsequent analysis and experimentation. Furthermore,

this comprehensive dataset will serve as a foundation for the theoretical discussions and developments presented in the following chapters.

3.1.1 Q-Drone system

UAV Platforms and Onboard Sensors

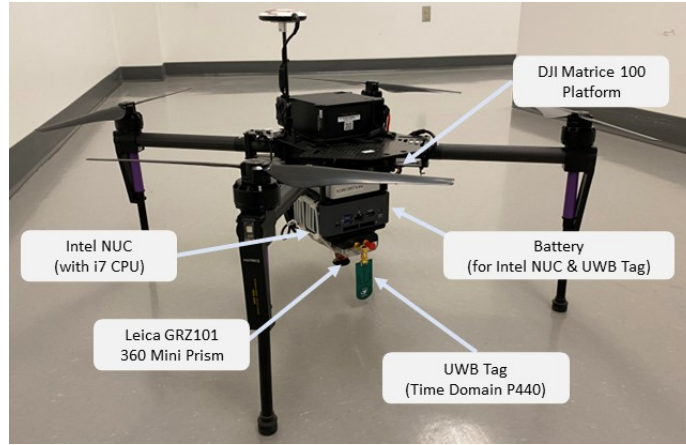
We developed two drone systems: Q-Drone X and Q-Drone 360. Their naming reflects their respective system capabilities and sensor configurations. Q-Drone X served as my primary system for data acquisition in this study. Simultaneously, during the study, we conceptualized Q-Drone 360 as an enhanced version. It's important to note that Q-Drone 360 was not employed in this research. Given my exclusive use of Q-Drone X throughout the study, any reference to "Q-Drone" hereafter specifically denotes Q-Drone X.

- Q-Drone X

To obtain the UWB benchmark, the Q-Drone UAV system was developed and customized for engineering inspections to create 3D asset environments and anomaly detection in GNSS-denied settings. The Q-Drone system is engineered for UWB-based positioning, dynamic path planning, automated quality assessment during inspection missions, and computer vision-based anomaly detection. The current Q-Drone configuration, deployed in this study, achieves UWB-based positioning through pose graph optimization and multi-trilateration techniques, leveraging the DJI Matrice 100 platform with a payload capacity of 2.4 kg. The payload includes a UWB tag (112 grams), an Intel NUC computer (728 grams), and a battery (542 grams), as illustrated in figure 3.2a.

- Q-Drone 360

Integrating advanced technology into the DJI Matrice 600 platform, the Q-Drone 360 is an adaptable and high-performance UAV system. This platform has been configured to accommodate an array of passive sensors, enhancing its capabilities for a broad spectrum of applications. The sensor suite includes the Sony a7 III camera, weighing 650 grams, a thermal camera at 325 grams, and the Velodyne LiDAR PUCK, weighing 590



(a)



(b)

Figure 3.2 Comparison of Payload Systems on Q-Drone X and Q-Drone 360. (a) depicts the payload system of Q-Drone X, while (b) illustrates the payload system of Q-Drone 360.

grams. Alongside these, the existing UWB system ensures precise positioning, while the Jetson TX2, a lightweight computing unit at 144 grams, manages the data processing tasks. Mounted on a gimbal, this ensemble of sensors is strategically affixed to the DJI Matrice 600, providing a seamless and integrated solution for various applications. Figure 3.2b illustrates the Q-Drone 360, showcasing its capacity to capture detailed visual, thermal, and LiDAR data. With the DJI Matrice 600 as its foundation, the Q-Drone 360 delivers remarkable performance for advanced inspections, environmental monitoring, and beyond.

UWB Anchor System

The UWB Anchor System employs PulsON 440 modules, part of the P400 family developed by Time Domain. In this experiment, five PulsON 440 modules were utilized, each equipped with four UWB tags mounted on tripods in a square arrangement on the ground. Using ToF method, the UWB anchors provide distance measurements with a remarkable accuracy of 2 cm and a frequency of up to 125 Hz. Operating in temperatures between 40°C and 85°C , these UWB anchors excel in high shock and vibration environments.

Robotic Total Station

A robotic total station with a discontinued prism tracking feature was employed to validate positioning methods precisely. Using a modulated infrared light wave, the total station actively tracked a GRZ101, 360° mini prism mounted on the UAV. Utilizing two different total stations, the Leica Nova MS60 MultiStation and the Trimble VX Spatial Station, provided reference positioning data for indoor, bridge, tunnel, and outdoor datasets, offering a measurement range of up to 1000m and 5500m, respectively, with high accuracy of $1''(0.00027^{\circ})$ angle.

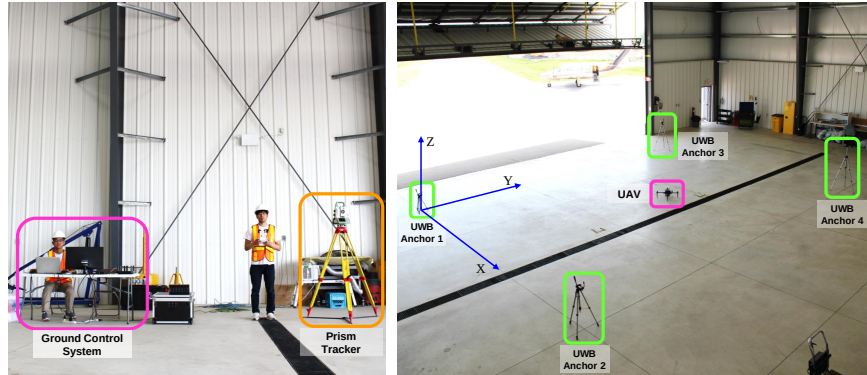


Figure 3.3 Robotic total station tracks UAV-mounted prism for accurate positioning. Leica Nova MS60 and Trimble VX provide high-precision reference data for diverse environments. The setup includes four UWB anchors, the ground control system and UAV.

Communication System

The communication system integrates the DJI Matrice 100's onboard Software Development Kit (SDK), enabling customization. The mounted Intel NUC computer connects to

the UAV through the Micro-USB port, facilitating real-time communication with the flight controller and onboard sensors. The SDK ensures seamless data exchange, allowing the NUC computer to receive onboard IMU data and UWB ranging data from the UWB antenna tag. Additionally, the GRZ101 prism on the UAV measures XYZ using the total station.

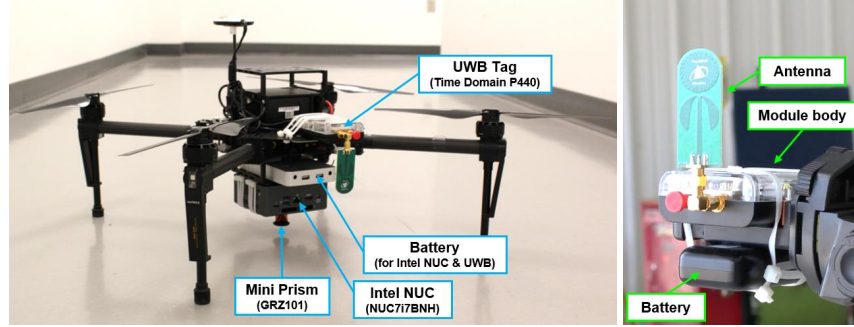


Figure 3.4 Customized communication: DJI Matrice 100's SDK allows flexibility. Intel NUC computer communicates in real-time via Micro-USB, linking the flight controller, sensors, and UWB antenna. GRZ101 prism measures XYZ of UAV.

3.1.2 System Calibration

This section details essential calibration procedures crucial for precise positioning. Beginning with time synchronization, the time reference clock synchronizes data from various sensors via the communication layer. Displacement calibration ensures spatial reference coherence across frames, addressing both Total Station-UWB and IMU-UWB displacements. Height Calibration moderates height differences between IMU and total station measurements through translation. Lastly, Range Calibration refines range measurements to reflect true data accurately, minimizing differences across diverse flight environments. These calibration procedures establish a solid foundation for accurate positioning in various operational scenarios.

Time Synchronization

The NUC time as the reference clock was employed to synchronize data from various sensors, achieving unified time coordination. A data communication layer based on DJI's

SDK transferred IMU and GNSS time references to the NUC. The UWB ranging data, directly connected to the NUC via USB, ensured synchronization with the NUC clock. Time-synchronization between onboard sensors and robotic total stations was accomplished through a systematic translation method.

Displacement Calibration

In Figure 3.5, there are three frames: one for the UAV body, one for the UWB network, and one for the total station. I chose the UWB network frame as the main coordinate system.

Total Station-UWB Displacement Calibration: In Figure 3.5, four UWB anchors were set up, named UWB_0 , UWB_1 , UWB_2 and UWB_3 , on the ground and attached one UWB tag to the UAV. Before flying, the positions of all the UWB anchors were measured. A coordinate system for the UWB network on a flat surface was created: UWB_0 's position became the starting point, a line between UWB_0 and UWB_1 became the x-axis, and the height of the total station defined the z-axis. The gathered data from the total station was adjusted to fit this UWB network frame.

IMU-UWB Tag Displacement Calibration: The IMU provides information about the UAV's movement, like acceleration, rotation, height, and orientation. the distance between the IMU and the UWB tag needed to be calibrated, particularly for the height. The IMU sits in the middle of the UAV, where the control unit is placed. The positions of the sensors on the UAV were fixed, and the offset between the UWB tag and the UAV's center of gravity was measured. The center of gravity is where the propeller beams intersect at the same level as the plane of rotation. The components of the offset vector are illustrated in Figure 3.6 and are defined in Equation 3.1.

$$V = 0.0092i - 0.186k \quad (3.1)$$

Height Calibration

The height data from the IMU and the total station must be adjusted to ensure accurate height measurements. Since we have set up the UWB network frame to match the total station's vertical axis, their height data aligns. However, the IMU's height data is relative to

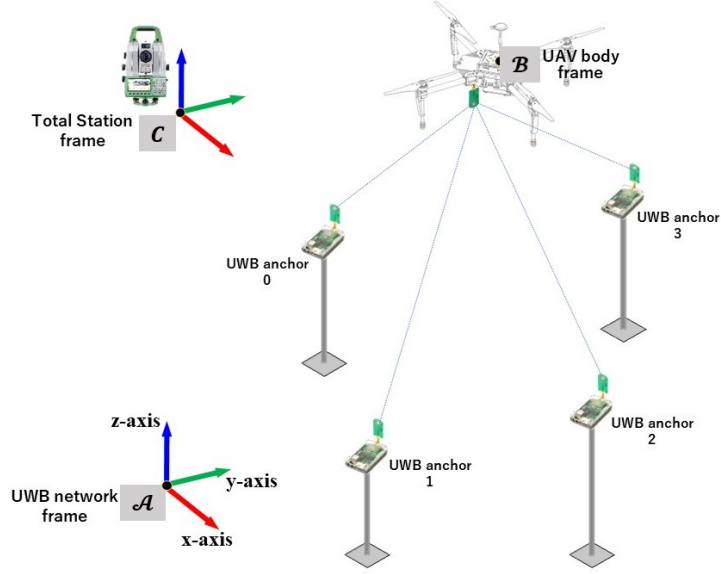


Figure 3.5 UWB Coordinate System: (A) UWB network frame, (B) UAV body frame, and (C) Total Station frame.

its starting point, where the UAV begins its flight. A translation between the IMU's height data and the total station's vertical measurement was performed to reconcile this. This translation, depicted in Equation 3.2, involves adding the initial total station height data (h_t) to the IMU's height data ($h_{IMU,t}$). This adjustment ensures that both sources of height data are aligned properly.

$$h_t = h_{IMU,t} + h_0 \quad (3.2)$$

Range Calibration

A range calibration needs to be conducted to ensure that the raw range measurements from the UWB sensors accurately reflect the true range data obtained by the robotic total stations. Despite both sets of measurements being taken in the same local coordinate system, there may be slight differences between them. These discrepancies could derive from factors such as differences between sensor systems, flight environment variations, or sensor configuration changes.

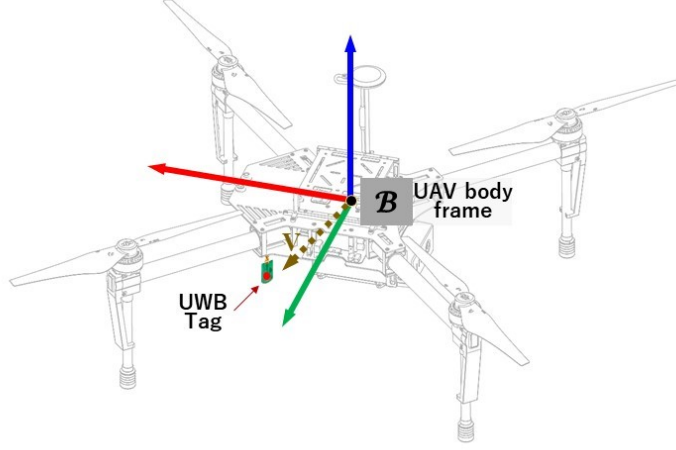


Figure 3.6 UAV Body Frame with UWB tag.

To address these discrepancies, correction coefficients are applied, which are described in Equation 3.3 and estimated in Equation 3.4. Through the use of a linear regression method, the difference between the adjusted range and the actual ground truth range was minimized. The calibrated range is calculated using the formula:

$$R'_{a_i} = a_{a_i} R_{a_i} + \bar{b}_{a_i} \quad (3.3)$$

$$S_i = \sum_{t=1}^N (a_{a_i} R_{t,a_i} + \bar{b}_{a_i} - \tilde{R}_{t,a_i})^2 \quad (3.4)$$

Where R'_{a_1} represents the calibrated range, R_{a_i} is the raw range data acquired by the UWB sensor, $\tilde{R}_{t,a_i}$ is the ground truth range data, a_{a_i} : Estimated range scale factor of i th UWB anchor, $\bar{b}_{a_i}$: Range bias coefficient of i th UWB anchor. The total squared difference between the calibrated range data and the ground truth data of i th UWB anchor is computed

in Equation 3.4, denoted by S_i . All these computations are performed within the NUC time frame.

3.1.3 Baseline Positioning Method

The baseline positioning method utilizes a multilateration algorithm, as depicted in Figure 3.7. This algorithm processes two streams of data with varying frequencies. Initially, the UWB data undergoes preprocessing to remove outliers, after which multilateration is applied. Together, the height from the IMU data and the calculated pose run through Levenberg-Marquardt optimization. The resulting output then undergoes another round of outlier removal to generate the final position.

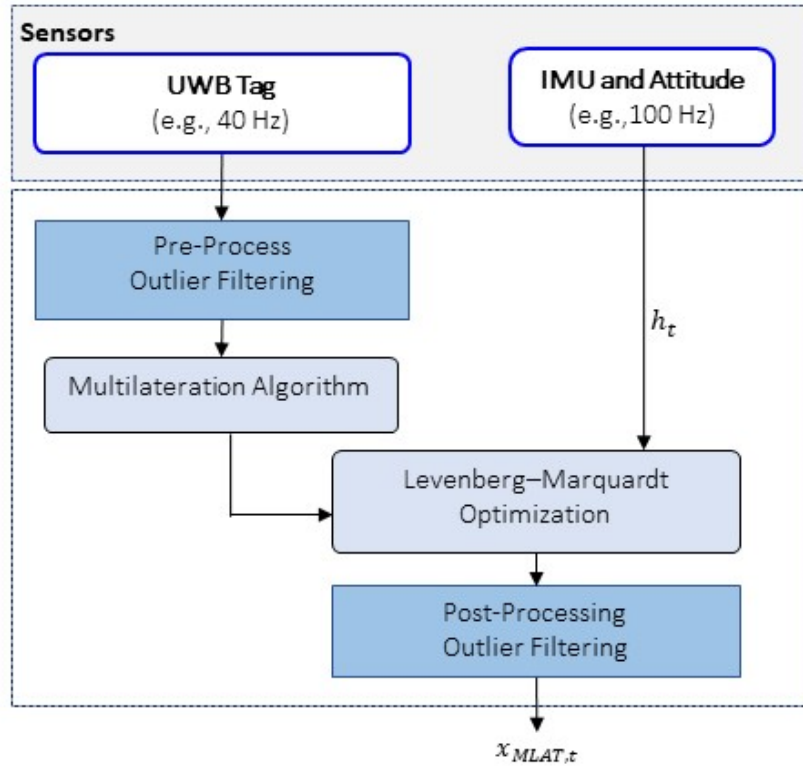


Figure 3.7 The MLAT baseline positioning method uses a multilateration algorithm that processes UWB data and IMU height data. After preprocessing and multilateration, the data undergoes Levenberg-Marquardt optimization and further outlier removal to determine the final position.

Outlier Removal

To validate the quality of the benchmark data, a data processing pipeline was implemented to estimate the UAV's position using raw UWB data from the UWB tag. The positioning pipeline includes a two-stage outlier filtering module: a pre-processing stage for eliminating outliers from the raw range data and a post-processing stage for removing outliers from the resulting positions.

Following [100], a theoretical model of UWB outliers can be parameterized using an optimizer like the Kalman filter. A predicted noise state coupled with observed range data forms the basis for a theoretical outlier assessment within this framework. However, the effectiveness of this probabilistic method greatly depends on the accuracy of earlier assumptions regarding noise characteristics and the selected positioning algorithms. In this investigation, we implemented a regional consensus-based thresholding technique to eliminate outlier range data [49]. Given a sequence of N consecutive range data points, denoted as $R_n = \{R_1, R_2, \dots, R_N\}$, the proximity of range values between two adjacent data points, represented as D_T , is determined heuristically. This norm of outlier data serves as a threshold value for conducting a theoretical outlier assessment of R_i by comparing the difference in range values between R_i and R_{i-1} with D_T . However, rather than determining the outlier status of R_i solely based on a comparison with its adjacent data points, the outlier determination of R_i is established only if any member of a range kernel ($N = 8$) is identified as an outlier. Conversely, R_i is categorized as an inlier only if all members of the range kernel are identified as inliers. This regional consensus-based thresholding approach is outlined in Equation 3.5:

$$\begin{cases} \text{If } \Delta R_{a_i,t}, \Delta R_{a_i,t_1}, \dots, \Delta R_{a_i,t-N} < D_T, \rightarrow R_{a_i,t} \text{ is an inlier} \\ \text{If } \Delta R_{a_i,t} \text{ or } \Delta R_{a_i,t_1} \text{ or } \dots \text{ or } \Delta R_{a_i,t-N} > D_T, \rightarrow R_n \text{ is an outlier} \end{cases} \quad (3.5)$$

Where the term $\Delta R_{a_i,t} = R_{a_i,t} - R_{a_i,t-1}$ represents the difference between the ranging measurements at time t and the previous time $t - 1$ for anchor a_i , denoted as $\Delta R_{a_i,t}$, and N represents the range kernel and n ranges from 1 to N .

The same logic of this thresholding technique is applied to eliminate outliers in the

range observations, not based on regional consecutive range datasets but using consecutive position datasets. This position-based consensus thresholding algorithm is employed as a post-processing stage.

Baseline Algorithm

Many UAV positioning algorithms rely on computer vision-based odometry or GNSS signals, but these methods struggle with poor lighting, featureless environments, indoor signal blockage, and multipath interference. Range-based techniques, such as the Extended Kalman Filter [101], Multidimensional Scaling (MDS) [102], triangulation [103], and trilateration [104], provide effective alternatives. In this chapter, I implement a Multilateration positioning algorithm to validate the benchmark dataset based on TDoA [105] as a baseline positioning model.

Given a set of ranges and positions of anchors, the approximate position of the UWB tag is determined by solving equation 3.6, where n is the number of anchors (minimum of three anchors for 3D positioning, $n = 3$). In this experiment, four anchors were used. In cases where range data from one anchor is not received, the algorithm solves the equation with three range data points. In equation 3.6, X_{a_i} is the position of the i th anchor defined by the total station.

$$\begin{bmatrix} 1 & -2X_{a_1} & -2Y_{a_1} & -2Z_{a_1} \\ \vdots & \vdots & \vdots & \\ 1 & -2X_{a_n} & -2Y_{a_n} & -2Z_{a_n} \end{bmatrix} \begin{bmatrix} x^2 + y^2 + z^2 \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} R_{a_1}^2 - x^2 - y^2 - z^2 \\ \vdots \\ R_{a_n}^2 - x^2 - y^2 - z^2 \end{bmatrix} \quad (3.6)$$

To address issues like multipath and NLOS signals and enhance the Multilateration solution, the Levenberg–Marquardt optimization algorithm [105] is employed. The Multilateration outcome serves as an initial point for the optimization problem. The objective function includes the geometric distance between the UWB tag and anchors, with the IMU height as an additional constraint.

$$F = \lambda(h_t - z_{\text{MLAT},t})^2 + \sum_{i=0}^N (R_{a_i,t}^2 - \|(x_{\text{MLAT},t} - X_{a_i})\|^2) \quad (3.7)$$

This optimization problem is used to refine the Multilateration solution. The objective function F includes the geometric distance between the UWB tag and anchors. The IMU height is added as another constraint to overcome negative height results.

3.1.4 Data Acquisition Results

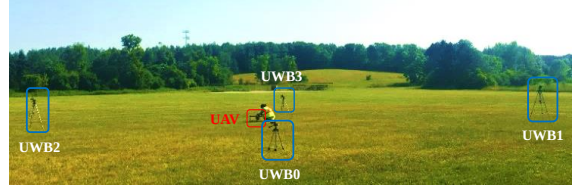
The Q-Drone system, detailed in the previous section, served as the platform for acquiring the UWB benchmark dataset across five distinct sites, each exhibiting diverse environmental characteristics. These benchmark sites encompassed indoor, open field, semi-open tunnel, under a bridge, and near-building scenarios, offering a comprehensive evaluation ground for testing positioning algorithms (Table 3.1). During the field data acquisition, configurations of payload sensors, anchor networks, and the Q-Drone platform were adjusted based on the specific site. This section delves into the specific configurations adopted for each site and provides insights into the measurement data formats and calibration processes.

Table 3.1 Details of the Q-Drone benchmark dataset acquired for this research, including environment types, locations, number of data and covered area.

Names	Environments	Locations	# of data	Area (m^2)
Indoor	Indoor	Oshawa, Ontario, Canada	5	26×33
Field	Outdoor, open sport field	Uxbridge, Ontario, Canada	5	12×13
Building	Outdoor, close to building	Newmarket, Ontario, Canada	3	7×7
Bridge	Semi-open outdoor, under a concrete-metal bridge	Niagara, Ontario, Canada	4	10×23
Tunnel	Semi-open outdoor, under a metal bridge	Oshawa, Ontario, Canada	4	8×30



(a)



(b)



(c)



(d)



(e)

Figure 3.8 UWB Anchors configurations of the Q-Drone benchmark dataset in various environments: a) Indoor, b) Field, c) Building, d) Bridge, and e) Tunnel.

UWB Configuration

The UWB system contains two key components: the UWB antenna mounted on the UAV platform and four anchors placed on the ground. The spatial arrangement of these components significantly influences UWB-based positioning performance. In the experimentation, two distinct configurations were adopted based on the prism’s spatial relationship with the UWB antenna:

- **Side Antenna Configuration for Indoor Dataset:** The UWB antenna tag was attached to the side of the platform body. At the same time, the battery and computer were mounted beneath the UAV platform. This site placement minimized body occlusions, enhancing visibility. The prism was installed at the platform’s bottom to maximize visibility for the total station.
- **Central Antenna Configuration for Outdoor Datasets:** Similar to the central configuration, this design involved placing the UWB antenna tag at the center of the UAV platform. This configuration aimed to minimize UWB signal blockage by the UAV platform while optimizing mobility. The central antenna configuration was employed for all Outdoor datasets (Field, Building, Bridge, and Tunnel).

Four UWB anchors were deployed on the ground with rectangular spatial configurations for each benchmark site. The dimensions varied based on site accessibility limitations, resulting in different physical dimensions. The UWB anchor configurations for each benchmark site are summarized in Table 3.1.

Benchmark Data Acquisition Result

1. **Results:** Table II provides a snapshot of the data collected during the benchmark tests conducted by the Q-Drone system. Across five test sites covering an area of 789.4 square meters, Q-Drone completed a total of 23 flights, covering a distance of approximately 4.3 kilometres over 1 hour, 50 minutes, and 28 seconds. Throughout these flights, a vast amount of data was gathered, including 253,832 UWB data points, 302,647 IMU heights, and 605,311 IMU data instances. Additionally, ground truth range data was

measured using robotic total stations, yielding 20,827 data points. The prism used for these measurements was small, with dimensions of about 30 mm, providing a pointing accuracy of 1.5 mm. However, due to their size, there were instances where the robotic total stations failed to track the prisms, resulting in missing data. To address this issue, missing information in the ground truth data was filled using spline interpolation. The benchmark datasets comprised various scenarios, including free motion indoors, square and zigzag patterns in outdoor fields, scanning building walls, sweeping under a bridge ceiling, and tunnel wall scans.

2. **Range accuracy:** As indicated in Table 3.2, I evaluated the accuracy of the UWB range measurements by comparing them to the ground truth obtained from the robotic total stations. The evaluation resulted in a MAE of 0.902 meters across the five sites. Notably, larger MAE were observed in the Building and Tunnel sites, likely due to multipath interference caused by the complex structures in these locations.

Table 3.2 Q-Drone Benchmark and UWB Range Accuracy

Sites	# of datasets	Duration (s)	Travelled distance (m)	Area (m ²)	UWB range MAE (m)
Indoor	5	1,687	1,260.88	91.96	0.37
Field	5	2,313	1,780.52	158.74	0.17
Building	3	823	388.92	50.10	2.02
Bridge	4	726	330.02	246.80	0.34
Tunnel	6	1,079	504.47	241.80	1.61
Total	23	6,628	4,264.81	789.4	0.902

3. **Range calibration:** The range accuracy was measured after rectifying the observed ranges with range correction coefficients, as explained in Section 3.1.2. To avoid overfitting, these coefficients were estimated using a sample range and position data acquired during a test flight and then applied to the remaining data for further analysis and position estimation. Figure 3.9 illustrates an example of the range data measured with four anchors from one of the benchmark datasets, showing the raw, calibrated, and ground truth ranges for each anchor. The average error between the ground truth

and UWB ranges was calculated to be less than 25 cm for three anchors and less than 50 cm for the remaining anchor, considering the raw data without outlier removal.

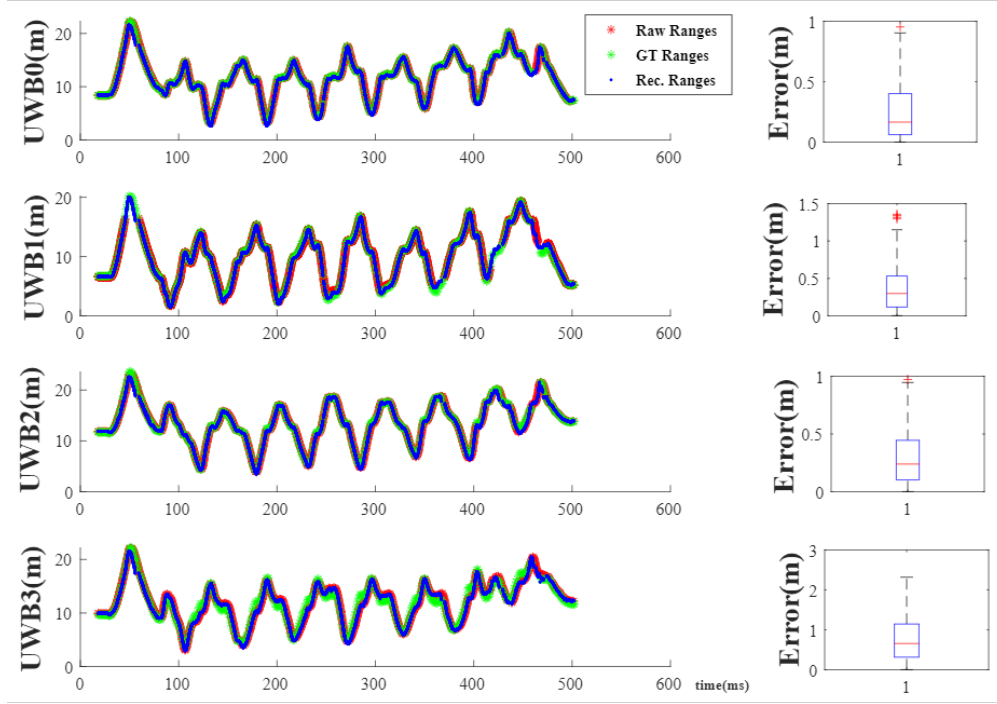


Figure 3.9 Range difference between UWB and ground truth.

4. **Missing range data:** The UWB data stream from the four anchors arrives in a random sequence. Occasionally, one anchor's data may not be received due to signal blockage, sensor failure, or certain areas where two sensors do not face each other. The algorithm addresses the issue by utilizing three range data points.

Acquired Trajectories

Figure 3.10 illustrates UAV trajectories generated by the MLAT baseline positioning algorithm compared to ground truth positions. Notably, instances may occur where the UWB tag fails to receive the UWB range signal from one of the four anchors, leading to a solution derived from three ranging data (three-anchor data). A representative trajectory from each benchmark site is depicted, encompassing scenarios such as free motion in an indoor environment, square and zigzag patterns in an outdoor field, scanning the walls of an

outdoor building, sweeping the ceiling under a bridge, and scanning tunnel walls.

Flight Patterns for Various Environments

In designing flight patterns for UAVs, it is crucial to account for Dilution of Precision (DOP), a key factor in determining the accuracy of the positioning system. DOP quantifies how the geometric arrangement of anchors or satellites relative to the UAV affects the precision of the estimated position. Specifically, DOP is a measure of the uncertainty in position estimates caused by the relative positions of the anchors or satellites used for localization. A lower DOP value indicates a better geometric configuration, leading to more accurate position estimates, while a higher DOP value suggests a less optimal arrangement, resulting in increased positioning errors.

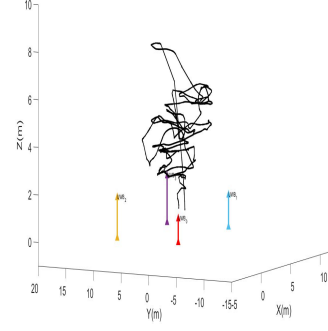
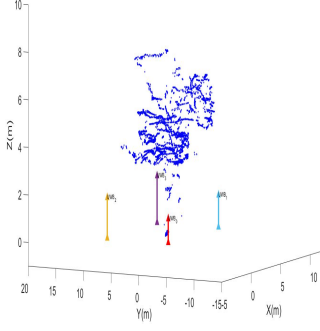
To optimize positioning accuracy, it is essential to capture a range of scenarios that produce varying DOP values. This can be achieved by employing different flight patterns with a fixed set of anchors. For example, where anchors are relatively stationary, varying the UAV's position through free motion allows the system to experience a range of DOP values, from low to high, depending on the UAV's relative distance and angle to the anchors. This variability helps identify and mitigate potential accuracy issues by providing a comprehensive assessment of positioning performance under different geometric conditions.

Free Motion in Indoor Environments: Free motion flight patterns in indoor settings are selected for their ability to provide thorough coverage of the available space. This pattern allows the UAV to navigate around obstacles and capture detailed data within complex indoor environments. The UAV can adapt to dynamic and cluttered spaces by avoiding rigid paths.

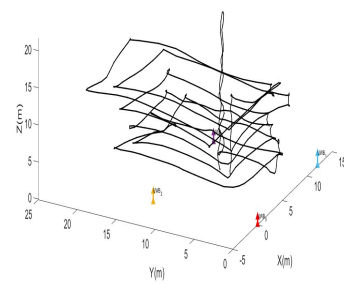
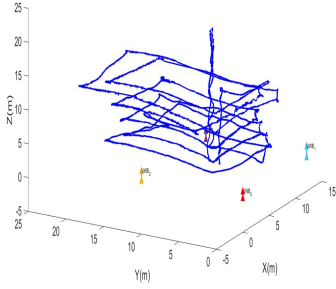
Square and Zigzag Patterns in Outdoor Fields: Square and zigzag patterns are employed in outdoor fields to ensure systematic coverage of larger areas. The square pattern effectively creates a grid-like coverage, while the zigzag pattern helps scan the entire field efficiently by minimizing overlap and maximizing data capture. These patterns help achieve good terrain coverage, which is essential for applications like agricultural monitoring and environmental surveys. Additionally, these patterns can improve the DOP by ensuring that the UAV maintains a diverse set of measurements relative to the anchors, thus reducing positioning

MLAT output

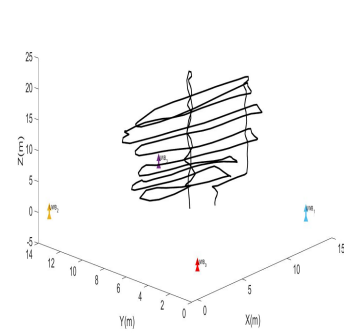
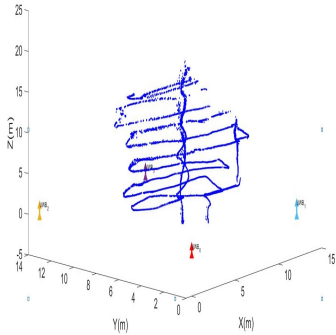
Ground truth



a) Indoor dataset



b) Field dataset (square pattern)

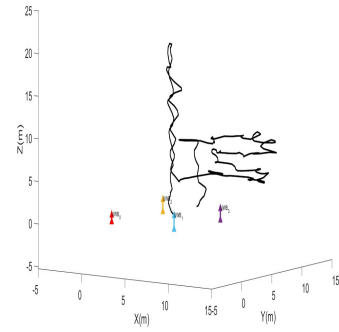
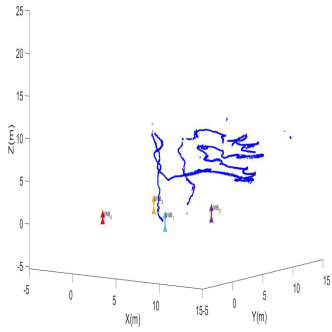


c) Field dataset (zigzag pattern)

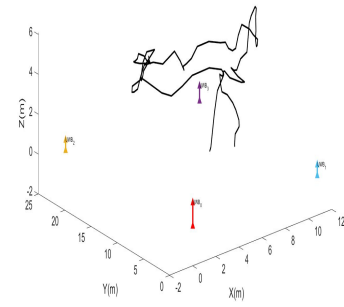
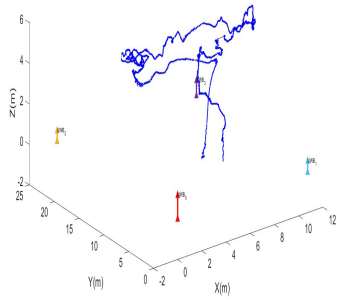
Figure 3.10 Multilateration results and ground truth for various datasets: a) indoor, b) field (square pattern), and c) field (zigzag pattern).

MLAT output

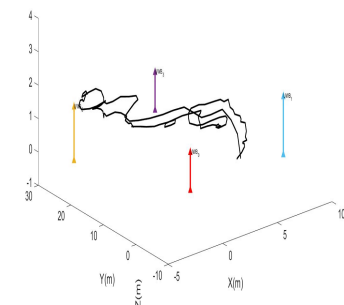
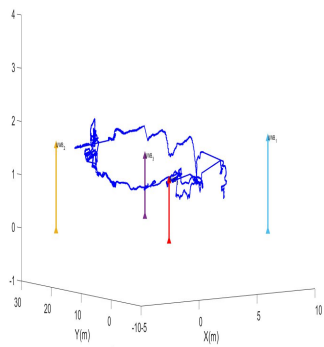
Ground truth



d) Building dataset



e) Bridge dataset



f) Tunnel dataset

Figure 3.10 Multilateration results and ground truth for various datasets: d) building, e) bridge, and f) tunnel.

errors.

Scanning the Walls of Outdoor Buildings: When scanning the walls of outdoor buildings, the UAV follows a pattern that ensures complete coverage of vertical surfaces. This approach is particularly useful for infrastructure inspection and maintenance, where detailed information about building facades is required. By systematically covering the walls, the UAV can detect structural issues and gather data necessary for repair and analysis.

Sweeping the Ceiling Under a Bridge: For scenarios such as sweeping the ceiling under a bridge, the UAV uses a sweeping motion to cover the curved and often inaccessible surfaces beneath structures. This pattern is designed to gather data on areas challenging to inspect from the ground or with stationary sensors. The sweeping motion helps capture continuous and comprehensive data critical for structural health monitoring and inspection.

Scanning Tunnel Walls: Scanning tunnel walls involves a pattern that allows the UAV to fly along the length of the tunnel while maintaining proximity to the walls. This pattern is essential for tunnel inspection and mapping, where high precision is needed to identify defects or anomalies in the tunnel structure. By following the walls closely, the UAV can minimize the effects of multipath interference and ensure accurate data collection.

The chosen flight patterns offer several benefits, including:

Good Coverage of Space: Each pattern is designed to cover different types of environments effectively, ensuring comprehensive data collection across various spatial configurations.

Optimization of DOP: By varying the flight patterns, the UAV can maintain a diverse set of measurements relative to anchors, which helps minimize the DOP. Lower DOP values indicate better positioning accuracy, as the system's precision is less diluted by the geometry of the anchors.

Addressing Multipath Effects: Different patterns help mitigate multipath interference by changing the UAV's trajectory relative to reflective surfaces and obstacles. This approach reduces the likelihood of erroneous measurements caused by signal reflections and ensures more reliable data. By employing these patterns, the UAV can effectively navigate and collect accurate data in diverse environments, addressing the challenges associated with indoor and

outdoor settings.

3.2 Q-Drone Benchmark Dataset: Comprehensive Online Resource

The Q-Drone dataset is unique compared to other available UWB datasets used for positioning (Table 3.3). Unlike other datasets, it covers indoor, outdoor, and semi-outdoor environments, whereas other datasets are limited to indoor or outdoor settings. This comprehensive environmental coverage is crucial for developing robust positioning algorithms that function effectively across different scenarios.

Table 3.3 Comparison of UWB Datasets.

Dataset Name	Agent	Environment type	Duration (min)	Area (m^2)	Sensors	# Nodes	Challenges
Q-Drone	UAV	Indoor Outdoor Semi Outdoor	110	50-246	UWB IMU	4	Multipath
UTIL [106]	UAV	Indoor	150	100	UWB IMU	4	NLOS
UWB Relative Localization [107]	UGV	Outdoor	NR	64	UWB	2-8	Aligning nodes
Reza et al. [108]	UGV	Indoor	23	16	UWB	4	Cross-technology
Vey et al. [109]	Mobile node	Indoor	NR	NR	UWB	6	NLOS Multipath

The Q-Drone dataset spans 110 minutes of data collection, which is more extensive than most available datasets. Although UTIL [106] offers a longer duration of 150 minutes, it is confined to indoor settings. The Q-Drone dataset also covers a larger area (50-246 square meters) than all other datasets, providing better spatial variability and a more realistic testing ground for positioning systems.

Equipped with both UWB and IMU sensors, similar to UTIL, the Q-Drone dataset also addresses the critical multipath challenge, enhancing its practical relevance. Other datasets, such as those by Vey et al. [109] and UWB Relative Localization, also address challenges like NLOS and node alignment but lack the comprehensive environmental and sensor integration

found in the Q-Drone dataset.

The Q-Drone dataset’s extensive duration, rich area coverage (Table 3.3), and diverse environmental settings make it a more comprehensive and superior resource for advancing UWB-based localization research. The Q-Drone benchmark is precisely designed to reflect real-world challenges by encompassing various environmental complexities. The benchmark includes indoor environments with confined spaces, outdoor fields with varying terrain, and complex urban and industrial settings. These scenarios are based on actual locations where infrastructure inspection is required, ensuring that the challenges represent real-world conditions. Specific challenges such as multipath effects, GNSS signal degradation, and varying geometric configurations are addressed. The multipath effect is analyzed and fixed in Chapter 5 using a VAE-LSTM network. Although GNSS data was not used in the Q-Drone benchmark, the assumption was that GNSS was unavailable. The test sites were specifically chosen to reflect typical locations where GNSS struggles, thereby showcasing the effectiveness of UWB in environments where GNSS is not feasible.

In alignment with the objectives of this thesis, which include evaluating UWB technology’s performance in GNSS-denied environments, the selection of test sites was strategically designed. Although GNSS data was not used in the Q-Drone benchmark, the assumption was that GNSS is unavailable. The test sites were specifically chosen to reflect typical locations where GNSS struggles, thereby showcasing the effectiveness of UWB in environments where GNSS is not feasible. The benchmark encompasses a range of settings, such as indoor environments with confined spaces and outdoor fields with varying terrain. These sites were selected to represent scenarios where GNSS encounters significant challenges, such as signal obstruction and degradation.

The Q-Drone benchmark datasets collected from five different sites are accessible to the public through a dedicated benchmark website, which can be found at <https://benchmark.qdrone.ausmlab.com>. This website offers resources: Sensor datasets, including UWB ranging data, IMU and height data as shown in Figure 3.11, their ground truth data measured using robotic total station and calibration parameters, which are required for accurate positioning.

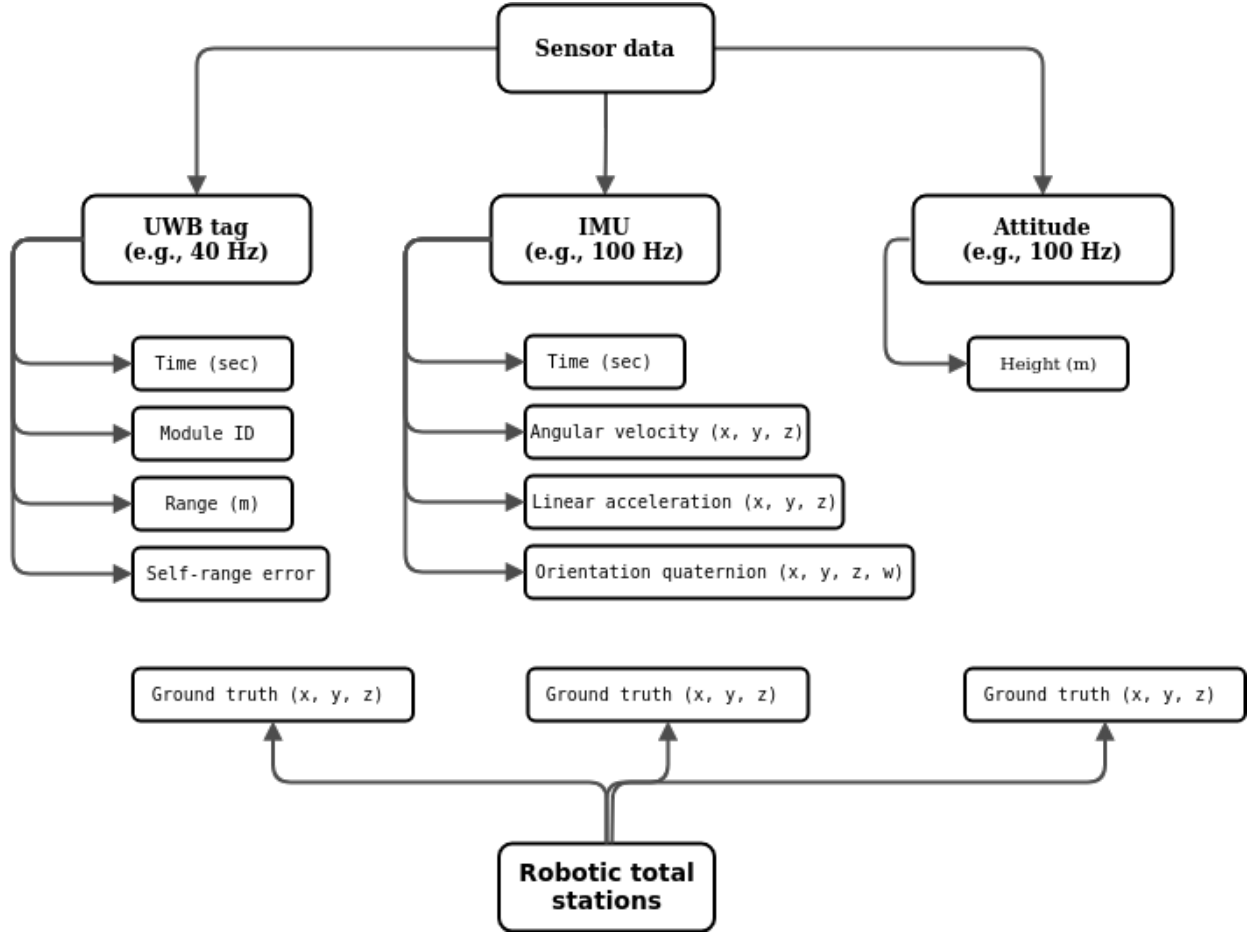


Figure 3.11 Data Structure.

The benchmark users can upload the positioning results generated using the benchmark dataset. The website then assesses the performance of these submitted results by comparing them to the reference positioning data obtained from UAVs tracked with robotic total stations. The evaluation outcomes are subsequently communicated to the users. Additionally, part of the ground truth data is released to the public to facilitate the training of learning-based methods, enabling researchers to utilize this information for further analysis and improvement of positioning algorithms.

3.3 Summary

In this chapter of the thesis, we introduced the Q-Drone UWB benchmark dataset, a comprehensive resource aimed at facilitating research in UAV positioning. The Q-Drone system was purposefully designed for accurate UAV platform positioning using UWB sensors across a range of challenging GNSS-denied environments, encompassing indoor spaces, open fields, close to buildings, under bridges, and semi-open tunnels. The dataset captured synchronized IMU and UWB data aligned with the onboard computer clock. Ground truth UAV positioning was established by precisely tracking a mini prism affixed to the UAV platform using a robotic total station. Calibration procedures were implemented to refine the relationships between the IMU and UWB tag, as well as between UWB anchors and the robotic total position. Additionally, correction coefficients were estimated to enhance the accuracy of raw range data concerning ground truth measurements. Rigorous validation measures were applied, employing the Multilateration positioning algorithm with Levenberg–Marquardt optimization and consensus-based filtering algorithms to eliminate outliers in range data. The Q-Drone benchmark dataset is set to be made publicly accessible through a dedicated benchmark website, enabling advancements in UAV positioning research utilizing UWB range data. This benchmark dataset is anticipated to serve as a valuable resource for explaining the limitations of developing positioning methods and validating their performance across diverse real-world environments.

Chapter 4

Incremental Smoothing with Ranges and Multilateration

This chapter presents an innovative approach to enhancing UAV positioning accuracy using the dataset collected as described in Chapter 3 through a novel smoothing technique. Traditional methods, such as MLAT discussed in Chapter 3, often rely on updating UAV position estimates based on individual low-dimensional UWB range measurements. However, these methods can be sensitive to inaccuracies, especially in challenging conditions. The proposed method integrates two types of UWB-based measurements to address this limitation: high-rate, weak-constraint single 1D ranges and low-rate, strong-constraint multilateration outcomes. This chapter introduces a unified factor graph seamlessly incorporating these measurements and inertial data within an incremental smoothing-based framework. The effectiveness of the proposed approach in significantly improving UAV positioning accuracy is demonstrated through extensive experiments across various scenarios.

The findings of this chapter are published in the following paper:

J. Kang, K. Park, Z. Arjmandi, G. Sohn, M. Shahbazi, and P. Menard, "*Ultra-Wideband Aided UAV Positioning Using Incremental Smoothing with Ranges and Multilateration*," in IROS, 2020. (Published).

The work builds upon Dr. Jungown Kang's initial design for indoor datasets. I collaborated with Dr. Kang on developing the incremental smoothing-based method and extended this approach to additional environments, including outdoor and semi-outdoor settings. My

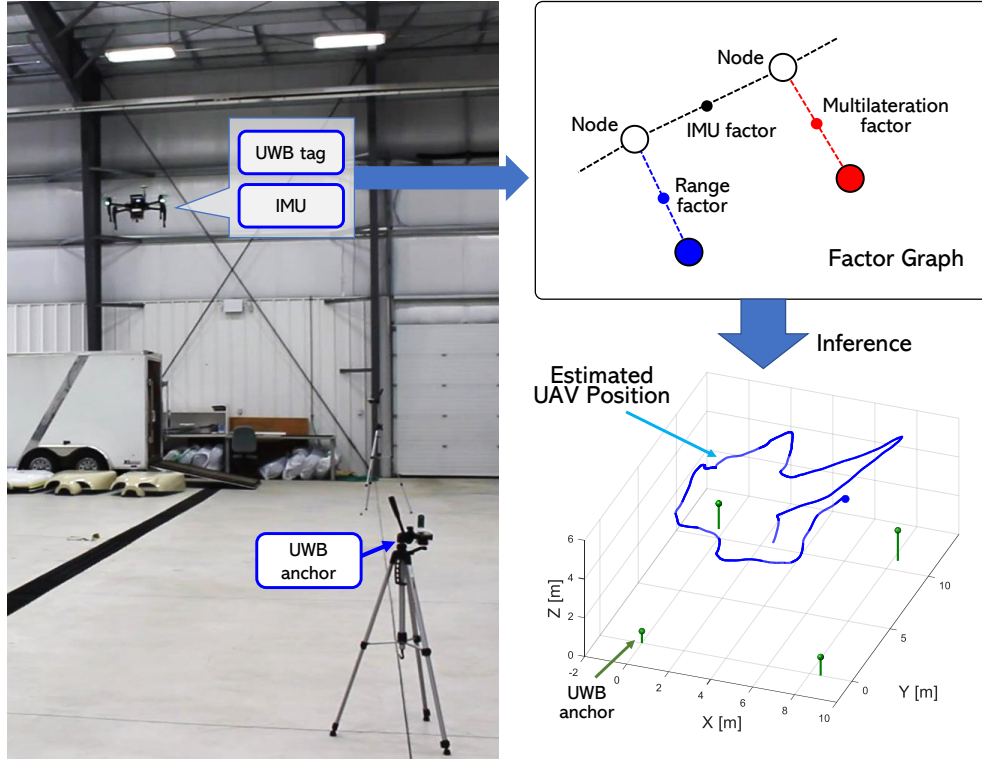


Figure 4.1 Description of the proposed method for UAV positioning based on incremental smoothing, which seamlessly integrates two types of UWB-based measurements and inertial measurement into a unified factor graph framework. Here, a node indicates a state to be estimated, i.e., UAV position.

contributions included implementing and adapting the smoothing technique for these varied environments and conducting comprehensive experiments to validate the effectiveness of the proposed approach across different scenarios.

4.1 Overview

Multilateration, like other anchor-based systems such as GNSS, can be applied to UWB systems [110, 111, 112, 113]. Although multilateration only applies to a range data set that meets certain conditions, it directly produces the 3D position of the UAV. With a set of range data between a tag and some anchors, multilateration yields the 3D position of the tag. This chapter was inspired by the idea explained in Chapter 3 that the multilateration

outcome can be utilized as a measurement providing the 3D position of the UAV, i.e., a strong constraint for the position, in filtering or smoothing approaches, compared to a weak constraint from a single range measurement. Hence, every single 1D range can be used as a high-rate measurement with a weak constraint, and the multilateration outcome can be utilized as a low-rate measurement with a strong constraint.

In this chapter, a method for UAV positioning based on incremental smoothing, which seamlessly integrates two types of UWB-based measurements and inertial measurement into a unified factor graph framework, is presented, as described in Fig. 4.1. The major contributions are as follows:

- A multilateration method that computes an algebraic solution (similar to Chapter 3) is presented, followed by a non-linear optimization.
- proposing an incremental smoothing approach integrating each single range measurement, multilateration-based 3D position of UAV, and inertial measurement into a unified framework using a factor graph.
- Furthermore, it proposes a novel strategy for adding a multilateration factor into a graph for online operations.
- Conducted experiments under various scenarios in GNSS-denied environments (dataset collected in Chapter 3) to demonstrate the effectiveness of the proposed method.

4.2 System Overview

The system overview, depicted in Fig. 4.2, includes UWB anchors, sensor data detailed in the previous chapter 3.1.1, sensor data management, measurement management and the UAV incremental smoothing algorithm.

The system requires a minimum of four UWB anchors positioned strategically throughout the area, within the frame $\mathfrak{R}_{\text{uwb0}}$. Anchor positions, represented by $\mathbf{X}_a = [X_{a_i}, Y_{a_i}, Z_{a_i}]$, are predetermined.

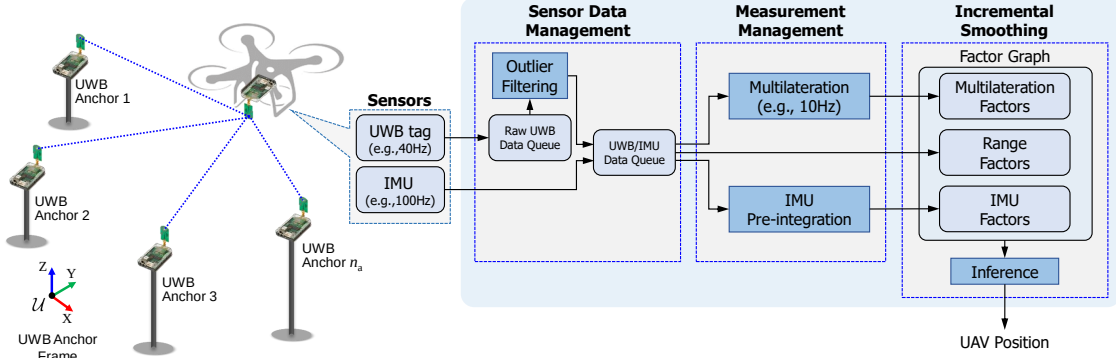


Figure 4.2 An overview of the proposed Incremental Smoothing System system for Q-Drone.

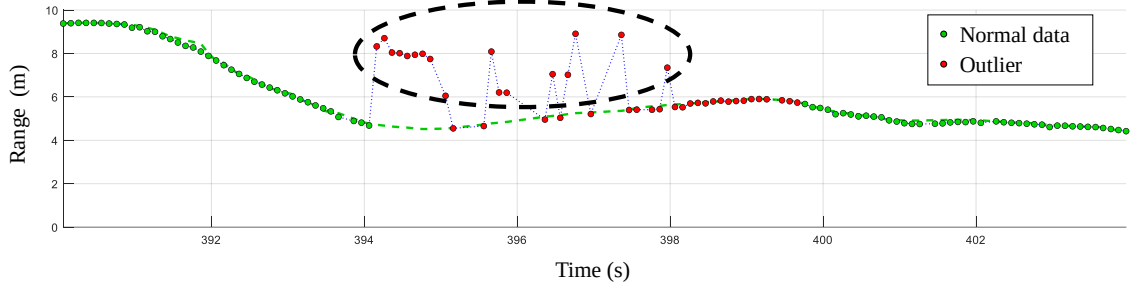


Figure 4.3 Acquired range over time between a UWB tag and a certain UWB anchor, and the result of the proposed outlier filtering algorithm, where the result is represented by a colour (green: normal data, red: outlier) in each data. The set of data in the dashed circle indicates true outliers trapped in NLOS cases.

Sensor data is collected from the UWB tag and IMU. UWB range data between the tag and anchors $\mathbf{R}_t = [R_{t,a_0}, R_{t,a_1}, R_{t,a_2}, R_{t,a_3}]$ is acquired alternately in a round-robin manner. IMU data, including linear acceleration (α_t), angular velocity (ω_t), and height (h_t), is also collected.

4.3 Filtering out Outliers in UWB Range Data

All incoming UWB range data from a UWB tag is first stored in the raw UWB data queue as shown in Fig. 4.2. After that, an outlier detection process is started to check if the data point greatly differs from the expected range. Traditional outlier detection methods, as described in [59, 64], rely heavily on the expected vehicle position to compute the expected range,

which may be erroneous in case of an inaccurate expected position. Hence, the intention is to develop an outlier detection mechanism independent of the expected vehicle position.

Upon analysis, as shown in Fig. 4.3, substantial fluctuations around outlier data points over time were observed. Leveraging this observation, a method was devised to classify range data into normal or outlier categories based on the temporal variation of range values. Specifically, when a new range R_{t,a_i} is received from anchor i at time step t , the range variation between consecutive data points within a temporal sliding window of size S_w is computed. If all range variation values fall below a predefined threshold R_o , indicating minimal fluctuation, the data is classified as normal and stored in the UWB/IMU data queue for further processing. However, if any range variation value exceeds R_o , signifying significant fluctuation, the data is considered an outlier and discarded. An example of this outlier filtering algorithm, employing $S_w = 15$ and $R_o = 0.7$, m, is illustrated in Fig. 4.3. Utilizing a temporal sliding window for classification ensures a high recall rate for outlier detection.

4.4 Multilateration for Direct Measurement of UAV Position

An algorithm for multilateration is presented in this section. Utilizing a set of range data, ideally acquired simultaneously, and known positions of anchors, multilateration calculates the 3D position of the UWB tag in frame $\mathfrak{R}_{\text{uwb0}}$. However, in my scenario, the data is collected within a short time interval rather than simultaneously.

From the UWB/IMU data queue, m data points ($m \geq 4$), each representing the latest data for an anchor, are selected. The time difference between the newest and oldest data points is then computed. Multilateration is executed using these m data points only if the time difference is below T_d , a small threshold value (e.g., $T_d = 0.1$ s).

The multilateration algorithm proposed here first involves obtaining an algebraic solution and a subsequent non-linear optimization step.:

Obtaining an initial position by recursive least squares

The approach outlined in [111] was employed to obtain an initial position solution, which utilizes recursive least squares based on algebraic constraints. The 3D position of the UWB tag is denoted as $\mathbf{x}_{\text{mlat}} = [x_{\text{mlat}}, y_{\text{mlat}}, z_{\text{mlat}}]$. Utilizing the equation $\|\mathbf{x}_{\text{mlat}} - \mathbf{X}_{a_i}\|_2^2 = (R_{a_i})^2$, it can be formulated as $\mathbf{A}\mathbf{x}_{\text{mlat}} = \mathbf{b}$, where each row of $\mathbf{A}$ is defined by $[1, -2X_{a_i}, -2Y_{a_i}, -2Z_{a_i}]$, and the corresponding row of $\mathbf{b}$ is $[(R_{a_i})^2 - (X_{a_i})^2 - (Y_{a_i})^2 - (Z_{a_i})^2]$. Solving this equation yields the 3D position $\mathbf{x}_{\text{mlat}}$ of the UWB tag in the $\mathfrak{R}_{\text{uwb0}}$ frame.

Refining the initial position by non-linear optimization The initial solution is refined using the Levenberg-Marquardt non-linear optimization. An objective function is constructed, incorporating geometric constraints between the UWB tag and anchors. When UWB anchors are nearly coplanar, an additional term is introduced to accommodate height measurements from the IMU.

Finally, the refined solution by solving the objective function is achieved:

$$\hat{\mathbf{x}}_{\text{mlat}} = \arg \min_{\mathbf{x}_{\text{mlat}}} \lambda \cdot e_h^2 + \sum_{i \in m} (e_{a_i})^2 \quad (4.1)$$

The parameter λ , set to 0.1 to ensure accuracy for each sensor, is utilized for weighting. The error terms $e_h = h - z_{\text{mlat}}$ and $e_{a_i} = R_{a_i} - \|\mathbf{x}_{\text{mlat}} - \mathbf{X}_{a_i}\|_2$ are defined accordingly.

4.5 UAV Positioning Using Factor Graph

This section outlines the construction of a factor graph for UAV position estimation from individual measurements. Initially, a brief overview of the factor graph framework is provided. Subsequently, factors corresponding to each measurement are discussed. Finally, a method for updating the factor graph for online operation is proposed. The estimation of the 3D position of UWB tag, representing the UAV position in the UWB anchor frame $\mathfrak{R}_{\text{uwb0}}$, is the primary focus.

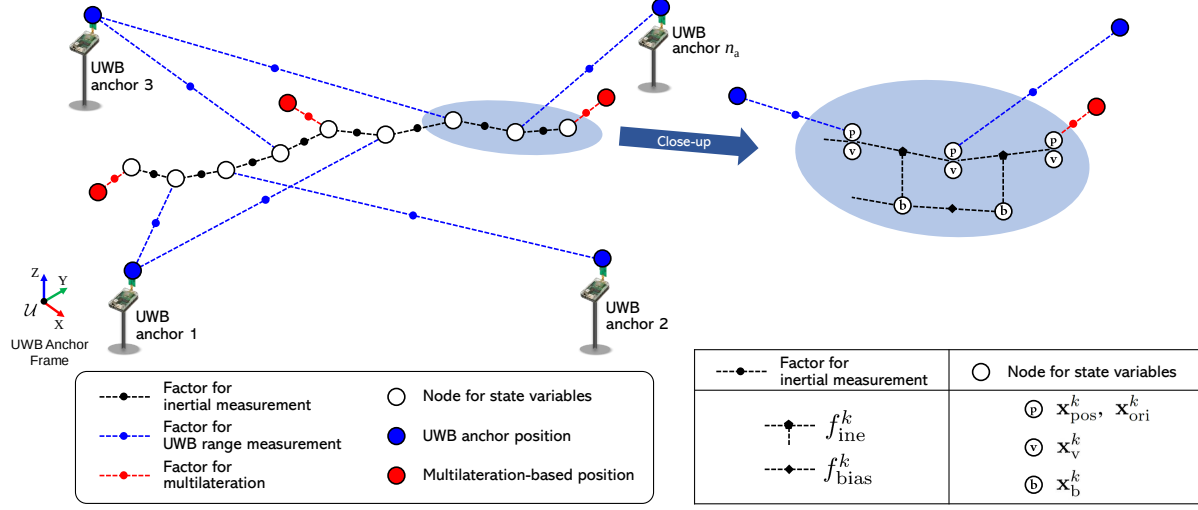


Figure 4.4 Factor graph representation of the proposed algorithm. The proposed algorithm has obtained the state variables over time for UAV position. $\mathbf{x}_{pos}^k$, orientation $\mathbf{x}_{ori}^k$, velocity $\mathbf{x}_v^k$ of the UAV, and biases $\mathbf{x}_b^k$ for an accelerometer and a gyroscope of the IMU. The measurements and the state variables are associated with factors. Here, three factors were used to handle multilateration-based position, single UWB range, and inertial measurements, respectively.

4.5.1 Factor Graph Framework

The estimation of the 3D position of the UAV was motivated by a factor graph-based incremental smoothing framework [114], as it allows handling asynchronous sensor measurements with different frequencies and enables the easy addition of new types of measurements. In this approach, a factor graph encoding the connectivity between states will be estimated, and available measurements will be constructed. Each measurement contributes a factor to the graph. The factor $f_i(\mathbf{x}_i)$ is represented by $\exp(-\frac{1}{2} \|h_i(\mathbf{x}_i) - \mathbf{z}_i\|_{\Sigma_i}^2)$, assuming a Gaussian noise model, where $\mathbf{z}_i$ is a measurement, $h_i(\cdot)$ is a measurement function, and Σ_i is the covariance matrix. The factor graph collectively represents the objective function for estimation, with each factor representing a sub-function. The 3D position of the UAV is obtained by calculating the Maximum A Posteriori (MAP) estimate of the posterior probability of the states, including position, orientation, velocity, and biases over time, given all available measurements. In this problem, the state variables at time step k include: (i) position $\mathbf{x}_{pos}^k$ in the UWB anchor frame $\mathcal{R}_{uwb0}$, (ii) orientation $\mathbf{x}_{ori}^k$ in the frame $\mathcal{R}_{uwb0}$, (iii) velocity $\mathbf{x}_v^k$ of the

Algorithm 1 Online Operation of UAV Positioning

Initialization: Factors $\mathcal{F} = \{\}$, States $\mathcal{S} = \{\}$

```

1: while measurement  $\mathbf{z}^t$  do
2:   if  $\mathbf{z}^t$  is an inertial measurement then
3:     Perform IMU pre-integration.
4:   else if  $\mathbf{z}^t$  is  $R_{t,a_i}$  then
5:     if  $\exists$  IMU pre-integrated values then
6:       Create  $f_{\text{ine}}^k, f_{\text{bias}}^k$  by (4.4) and (4.5).
7:     end if
8:     if Multilateration can be applied then
9:       Eliminate range factors and IMU factors
          in  $\mathcal{F}$  related to the multilateration.
10:      Create  $f_{\text{mlat}}^k$  by (4.3).
11:      Create  $f_{\text{ine}}^k, f_{\text{bias}}^k$ 
12:    else
13:      Create  $f_{\text{range}}^k$  by (4.2).
14:    end if
15:    Add factors to  $\mathcal{F}$ .
16:    Add states to  $\mathcal{S}$ .
17:  end if
18:  if Incremental smoothing is available then
19:    Perform an incremental inference using [115].
20:    Retrieve estimated states.
21:  end if
22: end while

```

UAV in frame $\mathfrak{R}_{\text{uwb}0}$, and (iv) biases $\mathbf{x}_{\text{b}}^k$ for an accelerometer and a gyroscope of the IMU.

4.5.2 Factors

In this algorithm, I use the following three types of factors.

4.5.2.1 Factor for single UWB range measurement

This factor is created from a single range measurement, and it is given as

$$f_{\text{range},t}(x_{\text{pos},t}) \doteq \exp\left(-\frac{1}{2} \left\| \|x_{\text{pos},t} - \mathbf{X}_{a_i}\| - R_{t,a_i} \right\|^2\right) \quad (4.2)$$



Figure 4.5 Description of the *Add-After-Eliminating* strategy for adding a multilateration factor to the factor graph.

4.5.2.2 Factor for multilateration-based measurement

This factor is generated from multilateration-based position measurement, and it is given as

$$f_{\text{mlat},t}(x_{\text{pos},t}) \doteq \exp \left(-\frac{1}{2} \|x_{\text{pos},t} - x_{\text{mlat},t}\|^2 \right) \quad (4.3)$$

where $\mathbf{x}_{\text{mlat}}^k$ is the multilateration outcome at time step t , and it is obtained by a multilateration method presented in Section 4.4.

4.5.2.3 Factor for inertial measurement

This factor is created from the incremental motion obtained from measurements of the IMU's linear acceleration and angular velocity. To handle the high-rate operation of the IMU, I adopt the IMU pre-integration technique [116] and introduce the equivalent IMU factor that represents several consecutive IMU measurements. Thus, instead of adding factors to the factor graph at IMU rate, the factors are added at a lower frequency that is determined by other sensors, e.g., UWB sensor.

$$f_{\text{ine}}^t(\mathbf{x}_{\text{pvb}}^{t-}, \mathbf{x}_{\text{pv}}^t) \doteq d(\mathbf{x}_{\text{pv}}^t - h(\mathbf{x}_{\text{pvb}}^{t-}, \Delta \mathbf{x}_{t-,t})) \quad (4.4)$$

$$f_{\text{bias}}^t(\mathbf{x}_{\text{b}}^{t-}, \mathbf{x}_{\text{b}}^t) \doteq d(\mathbf{x}_{\text{b}}^t - g(\mathbf{x}_{\text{b}}^{t-})) \quad (4.5)$$

Where $\Delta \mathbf{x}_{t-,t}$ indicates the relative motion from t^- to t that is obtained from the pre-integration of inertial measurement, and $g(\mathbf{x}_{\text{b}}^{t-})$ is a random walk model.

4.5.3 Online Operation of UAV Positioning

A novel strategy for the online operation of UAV positioning with the aforementioned factors is presented here. For a newly incoming range stored in the UWB/IMU data queue, the suitability of applying multilateration is checked based on the short time interval condition and requires a minimum of m data points ($m \geq 4$), including the latest incoming range and the preceding $m - 1$ ranges from other anchors. If multilateration is deemed unsuitable, a range factor is simply created and added to the graph. Conversely, if multilateration can be applied, a multilateration factor is generated. However, the inclusion of the multilateration factor in the graph violates the condition stipulating that all factors should be independent, given that ranges from other anchors used in multilateration have already been employed for range factors.

A method named the "add-after-eliminating" strategy is proposed to address the aforementioned issue of violating the factor graph condition when adding the multilateration factor. In this approach, the graph undergoes a process where range factors and IMU factors associated with the multilateration ranges are eliminated first, followed by adding the multilateration factor and a newly created IMU factor. This process ensures that all factors remain independent, as depicted in Fig. 4.5. It is emphasized that each range is utilized for only one factor in this proposed method.

The whole procedure for online operation is depicted in Algorithm 1, and Fig. 4.4 illustrates the created factor graph.

4.6 Experimental Results

This section describes the UAV system and experimental setup first. Then, the experimental results and a discussion of the results are described.

Table 4.1 The MAE of multilateration outcomes (Unit: meter).

-	-	Indoor	Field	Building	Bridge	Tunnel	Total
Number of Sets	-	5	5	3	4	6	23
Baseline 1: MLAT (Chapter 3)	x	0.655	0.245	1.570	0.283	1.094	0.769
	y	1.141	0.208	2.253	0.185	1.623	1.082
	z	0.566	0.292	0.305	0.302	0.393	0.372
	xyz	0.787	0.248	1.376	0.257	1.037	0.741
Baseline 2: Two-factor pose graph	x	0.055	0.068	0.077	0.084	0.037	0.064
	y	0.070	0.210	0.462	0.180	0.573	0.299
	z	0.473	0.217	0.247	0.255	0.265	0.291
	xyz	0.502	0.352	0.548	0.319	0.622	0.469
Incremental smoothing	x	0.052	0.015	0.014	0.021	0.012	0.023
	y	0.058	0.203	0.231	0.211	0.209	0.183
	z	0.186	0.094	0.097	0.189	0.188	0.151
	xyz	0.225	0.231	0.255	0.291	0.319	0.264

4.6.1 Experimental Setup

As previously detailed in Chapter 3, the UAV system utilizes a DJI M100 quadcopter with an Intel NUC, UWB tag, and battery installed at the bottom. A 360° mini prism provides ground-truth positioning. Five Time Domain PulsON 440 UWB modules offer accurate range measurements up to several hundred meters. Sensor data, including IMU readings, is acquired at 100 Hz using Robot Operating System (ROS) kinetic on Ubuntu 16.04. UWB range data is obtained through round-robin distance requests. My positioning algorithm is implemented using the GTSAM library [117].

Comparison Between Baseline 1 and Baseline 2

Baseline 2, utilizing pose graph optimization with two factors for single-range and inertial measurements, consistently outperforms Baseline 1 MLAT, which relies solely on multilateration odometry. Across various environments, Baseline 2 demonstrates an average reduction of approximately 0.2 meters in MAE values compared to Baseline 1. For instance, in indoor environments, Baseline 2 achieves an average MAE reduction of approximately 0.15 meters on the x-axis, 0.1 meters on the y-axis, and 0.2 meters on the z-axis compared to Baseline 1. This improvement can be attributed to the pose graph optimization used

in Baseline 2, which integrates additional IMU sensor measurements and constraints, and optimizing over the whole trajectory leading to more accurate position estimates than the solely multilateration-based approach of Baseline 1.

Comparison Between the Proposed Pose Graph and Baseline 2

The proposed method, which incorporates incremental smoothing and adds a multilateration factor to the pose graph optimization used in Baseline 2, exhibits further improvements in positioning accuracy compared to Baseline 2 alone. Across different environments, the proposed method consistently achieves lower MAE values compared to Baseline 2, indicating enhanced precision in position estimation. For example, in indoor environments, the proposed method demonstrates an average reduction of approximately 0.3 meters in MAE values compared to Baseline 2. This improvement can be linked to integrating two factors in the proposed method, which provides supplementary information for the optimization process, further refining position estimates. Specifically, the proposed method achieves an average reduction of approximately 0.2 meters in MAE values in the x-axis, 0.1 meters in the y-axis, and 0.3 meters in the z-axis compared to Baseline 2.

Discussion

As indicated in Table 4.1, in comparing the proposed method with the baseline, it is evident that the proposed method exhibits slightly better performance in the field and bridge datasets. However, a notable contrast between the proposed method and the baseline emerges in the indoor, building, and tunnel datasets, which pose the most significant challenges due to substantial motion changes and multipath effects.

The baseline methods face difficulty correcting estimations within sections experiencing significant motion changes. In contrast, the proposed incremental smoothing effectively operates even under these challenging conditions. However, noticeable differences in outcomes arise across different environments. To better understand this, the output mean absolute error (MAE) characteristics will be analyzed in the next discussion.

Exploring Variability across Diverse Sets

Figure 4.6 provides a graphical summary of the proposed incremental smoothing across

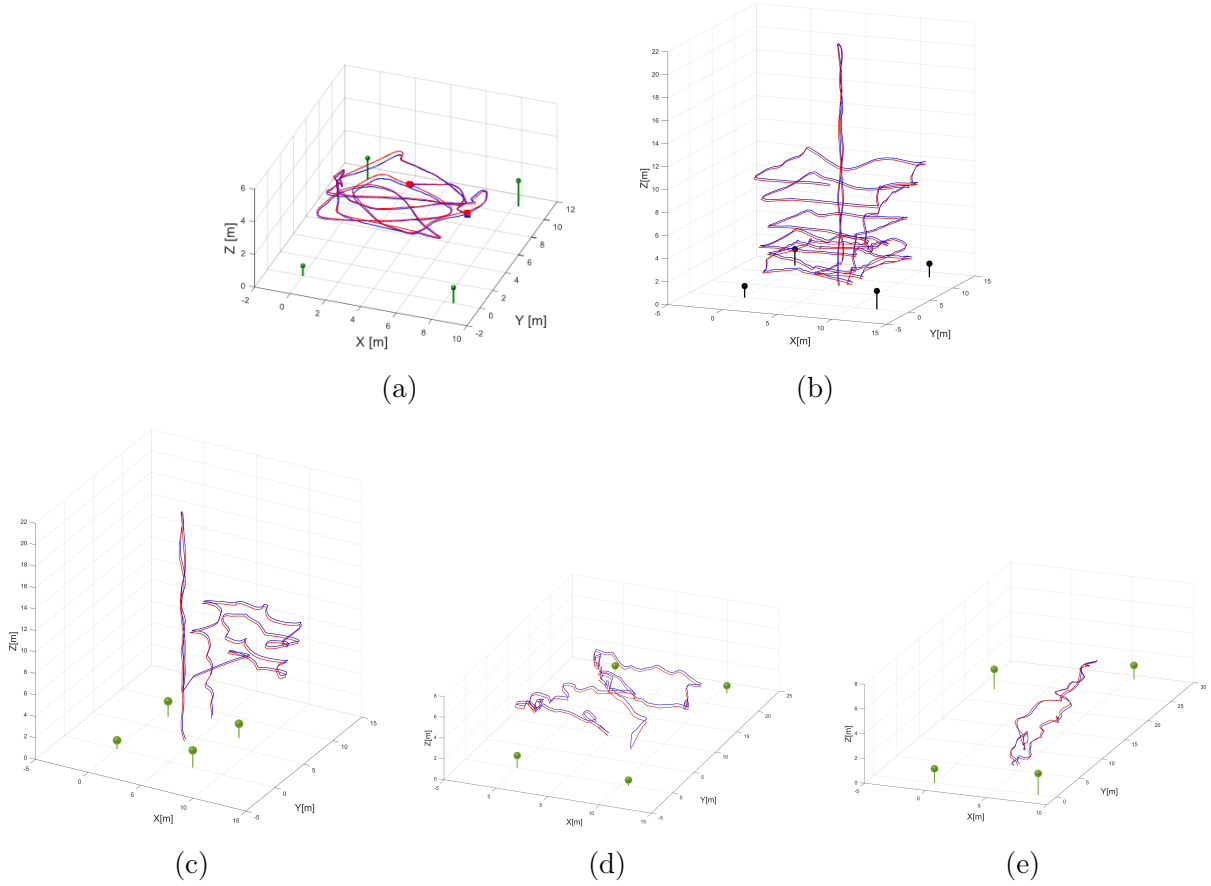


Figure 4.6 Comparison between estimated positions by proposed pose graph (blue) and ground truth (red) across various environments: (a) indoor, (b) field, (c) building, (d) bridge, and (e) tunnel.

all datasets and environments, illustrating their distribution, range, and central tendency.

First, examining the median values reveals that the indoor and field environments have lower mean MAE than other environments. Specifically, the building, bridge, and tunnel environments exhibit higher mean MAE values. The indoor experiments were conducted in a large, open area with minimal occlusion, resulting in a MAE similar to the field environments. However, the field environment shows a more uniform MAE across different sets due to the absence of occlusion, whereas the indoor environment exhibits slightly greater variance. While the building environment is generally open, the drone occasionally moves close to surfaces such as walls and windows as it sweeps to cover the area (Figure 4.8). This proximity results in a higher mean MAE compared to the field dataset. The bridge and tunnel environments

show higher mean MAEs due to increased occlusions encountered during the experiments. The tunnel environment, in particular, had a more confined space, reflected in its mean values. Notably, one set within the tunnel dataset showed a significant jump in mean value, which, upon analysis, was attributed to the flight pattern bringing the drone closer to the tunnel lining.

The second parameter to examine from this graph is the box size, representing the distance between the first quartile (Q1) and the third quartile (Q3). This range encompasses the middle 50% of the data and indicates the variability or dispersion within the dataset. The field dataset has a smaller box size, indicating low variability and, consequently, more consistent positioning results. The building dataset also shows small and similar box sizes to the field dataset in cases where the drone does not fly close to building surfaces. The indoor and bridge environments have similar box sizes, reflecting the absence of close-to-wall flight scenarios in these datasets. Conversely, the tunnel dataset exhibits noticeably larger box sizes, indicating high variability. This increased variability is due to instances where the drone flies close to the tunnel linings versus instances where it does not, resulting in a high variation in the error of the range data and measurements and, thus, greater error in the positioning results.

The third parameter to consider is the presence of outliers, which are data points that fall significantly outside the range of the majority of the data. In a box plot, outliers are displayed as individual points beyond the "whiskers." The field environment has the smallest range of outliers among all the datasets due to its more uniform error distribution. Similarly, the indoor environment has a few outliers, consistently appearing across its sets. In contrast, the building environment exhibits a higher range of outliers, especially in sets where the drone flies close to building surfaces. The bridge dataset also shows a larger range of outliers compared to the field and indoor datasets, though these outliers are not extremely high. As expected, the tunnel dataset has a very large range of outliers. These outliers represent instances where the drone flies very close to the tunnel linings, resulting in data points that deviate significantly from most of the dataset.

Discussion

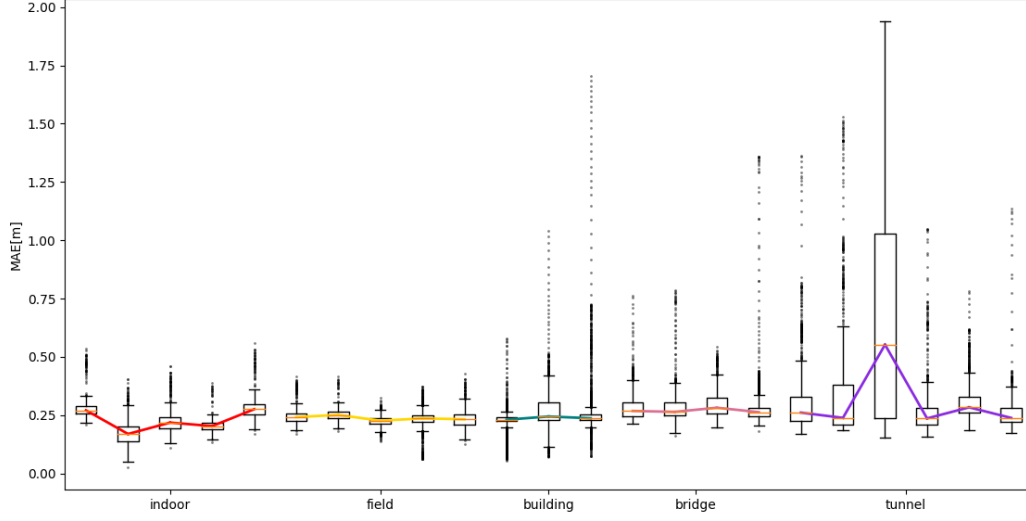


Figure 4.7 Box plot illustrating the MAE and quartiles of incremental pose graph results across diverse environmental sets, including indoor(red), field(yellow), building(green), bridge(pink), and tunnel(purple).

My analysis highlights the complicated correlation between environmental conditions and positioning accuracy. While environments with minimal occlusion yield more reliable positioning results, those with increased occlusions pose substantial challenges, strengthening error rates and variability. The varying error levels observed across different environmental scenarios emphasize the importance of considering environmental factors in optimization processes that rely on accurately defined error metrics. In particular, my findings emphasize the need for tailored covariance specifications in pose graph optimization algorithms. The distinct error distributions and outlier frequencies across environments suggest that a one-size-fits-all approach to covariance estimation may not be sufficient. Instead, optimization algorithms should adaptively adjust covariance parameters based on the specific characteristics of each environment and scenario. This highlights the significance of the next chapter in my research. By predicting dynamic covariance for optimization techniques that account for environmental variability, I aim to address these challenges and enhance the robustness and accuracy of localization systems in real-world applications.

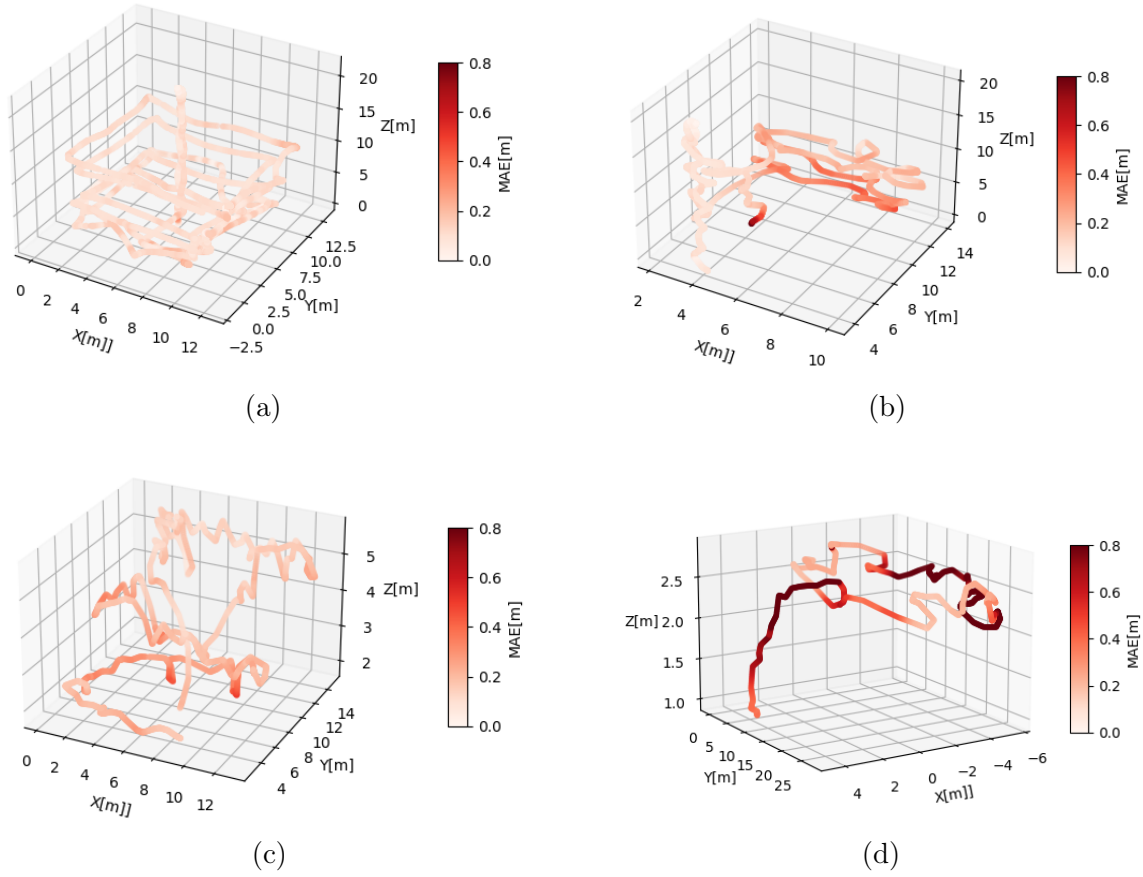


Figure 4.8 Example case trajectories. The colour indicates the XYZ error. (a) An example of evenly distributed error is typical of field and open indoor spaces. (b) A building dataset with high error near the building walls. (c) The bridge dataset has a medium error close to the bridge columns. (d) Tunnels have high errors near the walls and during take-off moments due to high speed.

4.7 Summary

In this section, an incremental smoothing-based method is proposed that seamlessly integrates these two types of UWB-based measurements and inertial measurement into a unified factor graph framework. In the proposed method, besides each single range as a high-rate measurement, I also used multilateration outcome as a low-rate measurement that is a strong constraint in a solution space. Moreover, I also presented an online operation strategy for adding a multilateration factor into a graph without violating the factor graph condition. Through experiments under various scenarios, I demonstrated the effectiveness of the proposed method.

However, there are several issues that I need to address for further development. First, the problem of properly setting uncertainty values for factors has not yet been fully addressed. In this work, I used constant covariance for the factors, derived them from the multilateration result, and used the covariance matrix for all the multilateration factors. However, a method for determining the uncertainty values more precisely needs to be further developed, considering factors (e.g., the dilution of precision [118, 119]) that can affect the uncertainty of multilateration outcomes.

Chapter 5

DeepCovPG: Dynamic Covariance Prediction for UAV Positioning

In this chapter, Building upon my previous research, which focused on pose graph optimization with static covariance, this chapter represents a natural progression toward addressing the dynamic challenges inherent in real-world UAV navigation scenarios.

Continuing my research, I introduce the integration of deep learning techniques and UWB technology within the DeepCovPG framework. Central to this new approach is the combination of a VAE with an LSTM network. The objective is to enhance UWB range data by reducing noise and predicting dynamic covariance. This shift from traditional static uncertainty models highlights my commitment to adapting to environmental changes and measurement errors, ultimately aiming to make UAV positioning more reliable and precise.

As I start this chapter, I have two goals: first, to smoothly connect with my earlier research, and second, to show how DeepCovPG can solve problems like signal interference and tricky navigation. Through rigorous testing across diverse settings, including indoor environments and urban landscapes, I anticipate demonstrating the significant improvements achieved by DeepCovPG, including a remarkable 51% reduction in RMSE and substantial enhancements in MAE compared to traditional methods.

The findings of this chapter are published in the following papers:

Z. Arjmandi, J. Kang, G. Sohn, and C. Armenakis, M. Shahbazi, “*DeepCovPG: Deep-Learning-based Dynamic Covariance Prediction in Pose Graphs for Ultra-Wideband-*

Aided UAV Positioning” in CASE, 2024. (Published).

Z. Arjmandi, J. Kang, G. Sohn, and C. Armenakis, M. Shahbazi, “*Dynamic Covariance Prediction in Pose Graphs using Deep Learning for Ultra-Wideband-Aided UAV Positioning*” in IJRR Journal, 2024 (Ready to be submitted).

The design and development of the DeepCovPG method is my original work. The conference paper provides a detailed analysis of this approach, while the journal paper presents a more comprehensive and refined examination of the method.

5.1 Overview

Pose-graph optimization plays a key role in refining the precision of UWB-aided UAV positioning, translating raw sensor data into actionable positional insights [32, 33, 34]. This method delineates the UAV’s trajectory through nodes on a graph, representing its spatial and temporal relationships with edges, with the ultimate goal of minimizing uncertainty in location estimation. However, effectively navigating the dynamic uncertainties inherent in sensor data presents a formidable challenge [35, 36]. Unlike conventional approaches, which typically treat these uncertainties as static, my prior work, as detailed in the preceding chapter, employed a pose graph framework characterized by static covariance. Yet, this methodology disregards the dynamic fluctuations of environmental variables—such as signal noise and multipath reflections—that continuously influence sensor measurements [37]. Consequently, the focus of this chapter shifts towards addressing dynamic uncertainties prevalent in sensor data. The variability inherent in UWB range measurements necessitates a sophisticated pose-graph optimization strategy that comprehensively accounts for dynamic uncertainties, containing factors such as signal noise, environmental dynamics, and clock drift, all of which impact UWB timing [38, 39, 40]. By integrating robust dynamic uncertainty management techniques into my pose-graph optimization framework, my aim is to strengthen the resilience and accuracy of UAV positioning in dynamically changing environments.

In this chapter, I introduce DeepCovPG, a novel system for UAV positioning that takes advantage of deep learning and advanced pose graph optimization. Central to my approach is the innovative integration of UWB range measurements and IMU data within a dynamic covariance framework encapsulated in a state-of-the-art VAE-LSTM multitask network. This model is highly effective at refining UWB signals for noise reduction and dynamic covariance estimation, establishing a novel pose-graph framework that enhances UAV positioning accuracy and reliability in GNSS-denied environments.

The chapter presents the following major contributions:

- A novel VAE-LSTM multitask network for UWB range data refinement and dynamic covariance prediction.
- A Dynamic Covariance Modeling (DCM) module and Dynamic IMU for integrating incremental smoothing with single and multilateral range measurements alongside IMU data.
- Extensive validation in diverse environments, demonstrating DeepCovPG’s superior performance against multipath effects and NLOS conditions.

5.2 System Overview

Figure 5.1 presents the DeepCovPG system for UAV positioning and the system setup with UWB anchors for collecting sensor measurement and ground truth labelling.

5.2.1 UWB Anchor Configuration

My experimental setup assumes the presence of four UWB anchors uniformly positioned within the test area. The global reference frame, denoted as $\mathfrak{R}_{\text{uwb}0}$, is centred on UWB anchor number 0. The spatial coordinates of the anchors, predetermined using a total station, are represented as $\mathbf{X}_a = [X_{a_i}, Y_{a_i}, Z_{a_i}]$ for each i -th anchor within the frame $\mathfrak{R}_{\text{uwb}0}$. These coordinates provide a fixed reference for computing the UAV’s position.

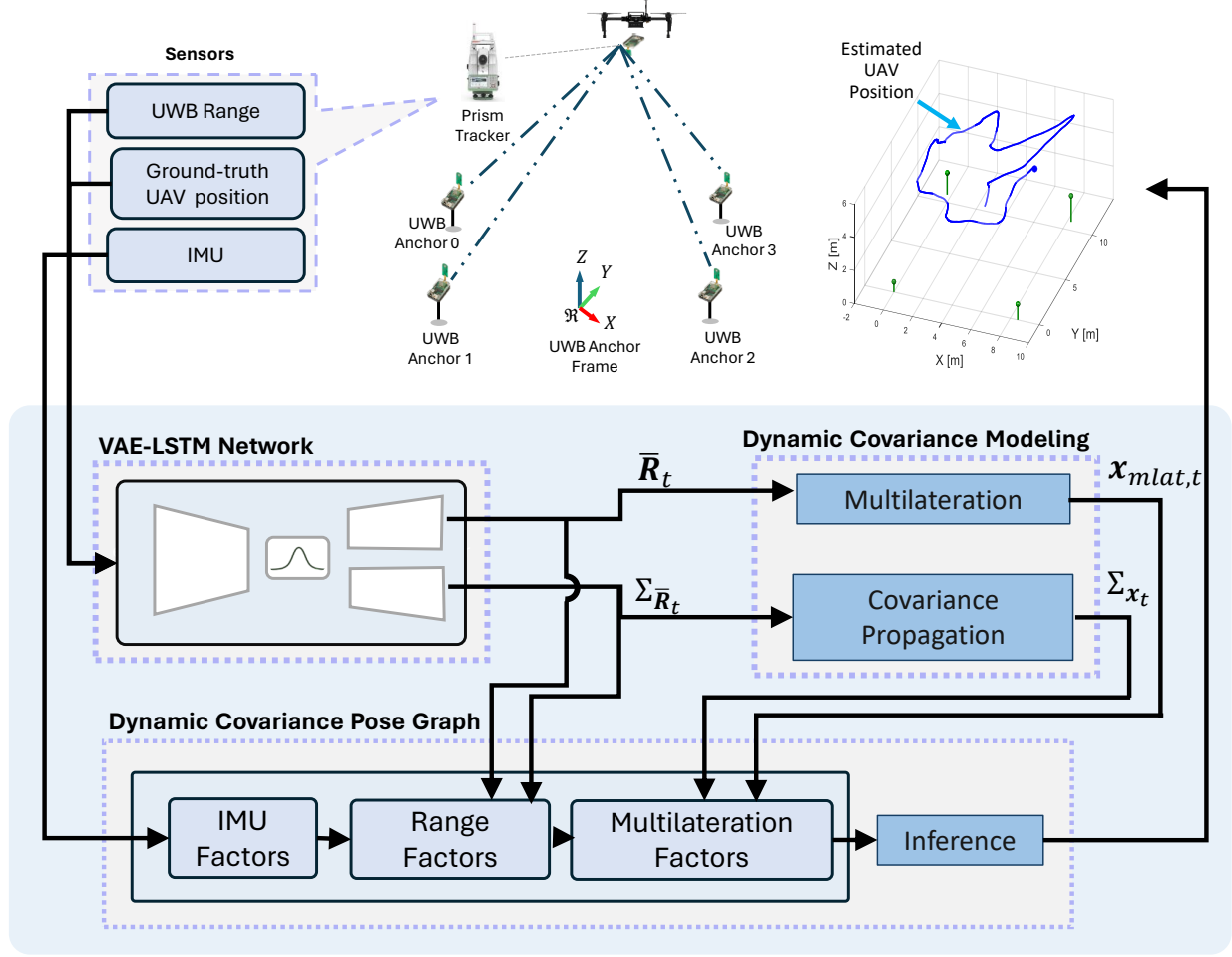


Figure 5.1 Overview of the DeepCovPG framework for UAV position estimation. The DeepCovPG framework dynamically predicts the covariance of the factors, enabling adaptive uncertainty handling within a unified DCPG.

5.2.2 Integrated Range and IMU Data Collection

In my system, the UAV is equipped with a UWB tag for precise distance measurements and an IMU for comprehensive motion tracking. The acquisition of range data from the UWB tag to each anchor follows a rapid sequential approach, which, despite the slight temporal offsets, is treated as a quasi-synchronous data set due to the minimal time difference between readings. This process results in an asynchronized quadruple range vector $\mathbf{R}_t = [R_{t,a_0}, R_{t,a_1}, R_{t,a_2}, R_{t,a_3}]$ at each time step t , effectively consolidating the distances from all

UWB anchors. Accompanying the UWB measurements, the IMU angular velocity (ω_t), and linear acceleration (α_t). To determine attitude (h_t), DJI drones employ a barometric sensor. With the altitude set to zero at the start of the flight to establish a baseline. The sampling rates of the UWB system and the IMU are chosen to ensure an optimal balance between data resolution and the operational tempo of the UAV.

5.2.3 Ground Truth Data Collection and Covariance Calculation

A robotic total station capable of prism tracking validated my high-precision positioning methodologies. The total station tracked a 360-degree GRZ101 mini prism, attached to the UAV, through modulated infrared light waves. I relied on two state-of-the-art total stations for ground truth positioning data: the Leica Nova MS60 MultiStation and the Trimble VX Spatial Station. These instruments are capable of measuring over distances up to 1000m and 5500m, respectively, and offer angular accuracy as fine as 1" (0.00027 degrees).

From the data from the total stations, I derived the precise ground truth position of the UAV at each time step, denoted $\tilde{\mathbf{x}}_t \in \mathbb{R}^3$. With the known anchor coordinates $\mathbf{a}_i$, the ground truth range vector were calculated $\tilde{\mathbf{R}}_t = [\|\tilde{\mathbf{x}}_t - \mathbf{X}_{a_0}\|_2, \|\tilde{\mathbf{x}}_t - \mathbf{X}_{a_1}\|_2, \|\tilde{\mathbf{x}}_t - \mathbf{X}_{a_2}\|_2, \|\tilde{\mathbf{x}}_t - \mathbf{X}_{a_3}\|_2]$, where $\|\cdot\|_2$ denotes the Euclidean norm. This ground truth range vector $\tilde{\mathbf{R}}_t$ serves as a crucial benchmark for assessing the accuracy of my estimated UAV positions and the effectiveness of UWB range data in challenging environments.

To accurately assess the performance of the UAV positioning system, it is essential to consider the covariance of the UWB range data. The ground truth positions, $\tilde{\mathbf{x}}_t \in \mathbb{R}^3$, are derived from precise measurements using robotic total stations, which are considered highly reliable for this purpose.

To calculate the covariance matrix for the UWB anchor range measurements, the ground truth distances between the UAV and each anchor, $\tilde{\mathbf{R}}_t$, are computed using the known anchor coordinates $\mathbf{a}_i$. These ground truth distances are a reference for comparing against the UWB range data.

Given that the range measurements from the UWB anchors are not obtained simulta-

neously, a time window w is used to align these measurements. Within this time window, the covariance matrix Σ_R is calculated based on the deviations between the UWB range measurements and the corresponding ground truth distances. Specifically, the variance of each UWB range measurement is computed, and the covariance matrix is constructed. The diagonal elements of the covariance matrix Σ_R represent the variance of each UWB range measurement as $\sigma_{R_i}^2 = (R_i - \tilde{R}_i)^2$, indicating the uncertainty of the individual measurements. The off-diagonal elements $\sigma_{R_i, R_k} = (R_i - \tilde{R}_i) \cdot (R_k - \tilde{R}_k)$ capture the covariance between different measurements, reflecting the correlation in measurement errors.

Factors Affecting Covariance Matrix Estimation The accuracy of the covariance matrix estimation relies on several critical factors. Primarily, the precision of the measurement devices plays a significant role. Due to issues like sensor calibration or signal interference, inaccuracies or noise in UWB range measurements can increase measurement variance and impact the covariance matrix. To mitigate this, sensor calibration was rigorously performed to reduce measurement noise and enhance precision.

Temporal alignment of measurements is another crucial factor. Since UWB range data are often not collected simultaneously, a carefully chosen time window w was used to align the UWB range data with ground truth distances accurately. The quality of the ground truth data also affects covariance estimation; high-precision robotic total stations were utilized to minimize errors and provide reliable reference distances.

As discussed in Section 5.7.4, the presence of multipath effects tends to increase the covariance. Specifically, when multipath interference occurs, the UWB measurements become less reliable, leading to higher variance in the estimated covariance matrix. Figure 5.5 shows that the covariance values were consistently larger during multipath conditions compared to scenarios with minimal interference, demonstrating the impact of such effects on measurement accuracy and covariance estimation.

5.2.4 DeepCovPG’s Three-Step Positioning Framework

Figure 5.1 visually summarizes DeepCovPG’s operational framework developed for determining UWB-aided UAV positioning through a structured three-step process. Initially, the VAE-LSTM network refines UWB sensor measurements to enhance their reliability for positioning. This step is crucial for setting a foundation of accurate data for further calculations. Then, the DCM module generates adaptive covariance matrices, which, together with the refined data, are incorporated into the Dynamic Covariance Pose Graph (DCPG) for UAV positioning. This integration, crucial for the system’s effectiveness, employs incremental smoothing to combine enhanced data with dynamic uncertainties, ensuring an accurate UAV positional estimate.

5.3 Problem Formulation

Accurate trajectory estimation for UAVs is critical for a myriad of applications, demanding precision in the UAV’s position states $\mathbf{x}_t \in \mathbb{R}^3$ at discrete time steps t . This estimation is derived from a quadruple range vector $\mathbf{R}_t$ obtained from UWB anchors. However, this process is inherently laden with uncertainties due to various factors such as environmental noise and signal interference. A central challenge in UAV navigation is the dynamic representation of these uncertainties, traditionally encapsulated by a covariance matrix (Σ_t).

This work tackles the optimization problem of computing a sequence of UAV poses $x_1, \dots, x_N$ that minimizes the deviation between the actual measurements and those predicted by the UAV’s sensor suite. The minimization problem, accounting for the dynamic nature of uncertainties, is given by:

$$x^* = \underset{x_1, \dots, x_N}{\operatorname{argmin}} \sum_{t=1}^N \|h(\mathbf{x}_t) - \mathbf{R}_t\|_{\Sigma_{\mathbf{R}_t}^{-1}}^2 \quad (5.1)$$

Here, $h(x_t)$ denotes the expected range measurements based on the UAV’s pose, while R_t corresponds to the actual range measurements. The inverse covariance matrix $\Sigma_{\mathbf{R}_t}^{-1}$ serves as

a weight within the optimization function, emphasizing the precision of sensor measurements.

The forthcoming section 5.7.3 will introduce a sophisticated VAE-LSTM multitask network, designed to dynamically predict the covariance matrix $\Sigma_{\mathbf{R}_t}^{-1}$. This network collaborates with a proposed pose graph called DCPG to improve the accuracy of UWB-aided UAV positioning.

5.4 VAE-LSTM Multitask Network

The DeepCovPG framework enhances UWB-aided UAV positioning with a VAE-LSTM multitask network that merges VAE’s generative modelling with LSTM’s temporal data processing. This integrated approach enables the network to refine sensor data and dynamically predict covariance matrices, crucial for precise estimation of UAV trajectory, as depicted in Figure 5.2.

Let f in $g(z_t) = f(z_t; \Theta) = \Sigma_t$ represent our neural network architecture. It is structured as a VAE composed of fully connected layers and non-linear activation functions. The final layer incorporates LSTM units. Throughout the training process, optimization focuses on refining the parameters Θ . The VAE framework, with its encoder-decoder structure, transforms the high-dimensional input $z_t \in \mathbb{R}^n$ into a latent representation. This latent representation is processed through the fully connected layers and non-linear activation functions, capturing intricate features. The last layer, comprising LSTM units, introduces temporal dependencies to the encoded information. The resulting low-dimensional features are then vectorized, producing $f(z_t; \Theta) \in \mathbb{R}^r$. This approach draws inspiration from feature extraction methodologies seen in earlier studies [120, 121]. The final output Σ_t is approximated by a linear combination of the transformed features, expressed as Σ_t .

Starting with raw UWB range data vectors $\mathbf{R}_t$, the network employs fully connected layers to develop feature-rich vectors, which the VAE encoder compresses into a latent space. From here, two LSTM pathways diverge: one refines the range vector $\bar{\mathbf{R}}_t$ for positional accuracy, while the other computes the corresponding covariance matrix $\Sigma_{\bar{\mathbf{R}}_t}$. The first pathway readies data for Multilateration (MLAT), outputting positional coordinates $\mathbf{x}_t$, and the second

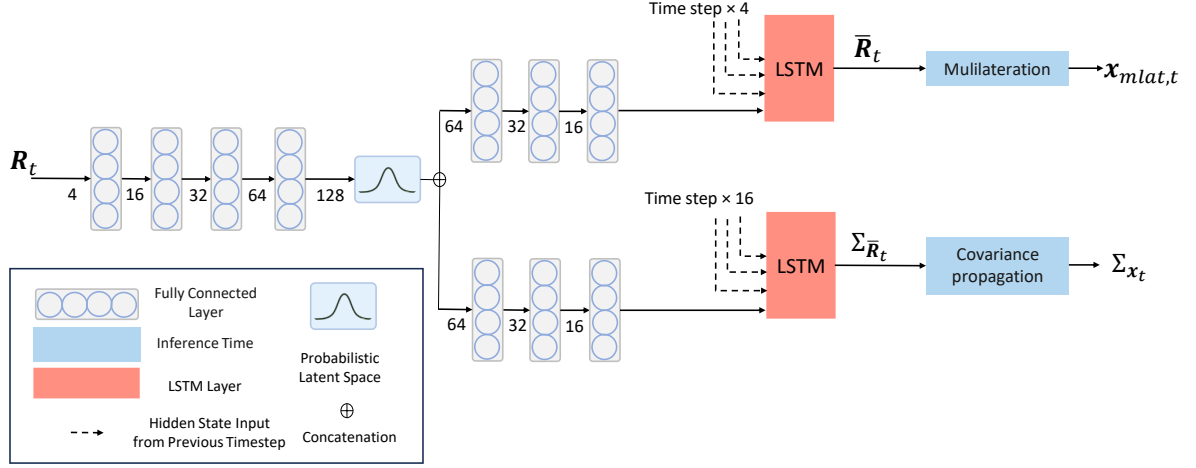


Figure 5.2 Architecture of the VAE-LSTM multitask network for UAV trajectory estimation. The network combines VAE units with an LSTM structure to refine sensor data and dynamic covariance matrices, facilitating a novel approach to trajectory refinement by accurately modelling sensor uncertainties.

delivers a 3x3 covariance matrix, capturing the dynamic positioning uncertainties. This dual-pathway approach ensures robust UAV trajectory estimation amid dynamic environmental changes.

5.4.1 Formulation of the Likelihood Loss Function

In Equation 5.2, the likelihood loss function is comprised of several key components and parameters that are critical for the VAE-LSTM network's optimization. Each term plays a specific role in the loss function's overall objective:

$$\begin{aligned} \mathcal{L}(\mathbf{R}_t, \Sigma_{\mathbf{R}_t} \mid h_t, p, q) = \\ \|\tilde{\mathbf{R}}_t - \bar{\mathbf{R}}_t\| + \|\Sigma_{\tilde{\mathbf{R}}_t} - \Sigma_{\bar{\mathbf{R}}_t}\| + \text{KL}[q(\psi_t \mid h_t) \parallel p(\psi_t)] \end{aligned} \quad (5.2)$$

The first reconstruction loss term for $\mathbf{R}_t$ seeks to minimize the disparity between the actual UWB range measurements and their reconstructions by the network, $\bar{\mathbf{R}}_t$, ensuring a faithful replication of observed range data essential for accurate UAV localization. The second reconstruction loss for $\Sigma_{\mathbf{R}_t}$ aims to diminish the differences between the real-world

measurement uncertainties and those estimated by the network, $\Sigma_{\bar{\mathbf{R}}_t}$, capturing the dynamic nature of these uncertainties accurately.

Furthermore, the third term of the Kullback-Leibler divergence serves as a regularization term, quantifying the deviation of the latent variables' probability distribution, conditioned on hidden variables $q(\psi_t | h_t)$, from a predefined prior distribution $p(\psi_t)$. This ensures the network's latent space is both structured and consistent with prior distributions, fostering the generation of meaningful, condensed representations of the input data.

These components, including $\bar{\mathbf{R}}_t$ for the genuine range measurements, $\Sigma_{\bar{\mathbf{R}}_t}$ indicating the covariance matrix of measurement uncertainties, h_t for hidden variables that might affect the data reconstruction, and ψ_t representing the latent variables crucial in the VAE's generative mechanism, collectively empower the network to not only recreate the range data and its uncertainties precisely but also ensure that the latent representations are coherent and significant.

Optimizing this loss function equips my network to support the DeepCovPG framework robustly, channelling refined range measurements and dynamically estimated covariance matrices into the MLAT and DCM modules. This facilitates an all-encompassing and precise UAV trajectory estimation process, which is adept at managing dynamically modelled uncertainties and significantly improving UAV positioning accuracy.

5.5 Dynamic Covariance Modeling

5.5.1 Multilateration Positioning Algorithm

My UAV positioning framework employs a TOA-based MLAT algorithm, crucial for deducing the UAV location by measuring its distance to a quadruple range vector $\mathbf{R}_t$. The MLAT algorithm uses these range measurements alongside the predetermined anchor positions $\mathbf{X}_{a_i} = [X_{a_i}, Y_{a_i}, Z_{a_i}]$ to calculate the UAV's estimated position at time t , represented by $\mathbf{x}_t = [x_t, y_t, z_t]$. This technique ensures accurate localization despite potential data loss from an anchor. For comprehensive insights into the MLAT algorithm, please see my

foundational work in chapter 3. Initial position estimates from MLAT are refined using the Levenberg-Marquardt optimization technique [122], effectively reducing the discrepancies between the computed and measured distances to anchors, thereby optimizing the UAV positional accuracy.

5.5.2 Dynamic Covariance Modeling

Accurate positioning of UAVs in the face of range measurement inaccuracies hinges on dynamic covariance modelling. Following insights from Section 5.7.3, the covariance between each UWB anchor and the UAV were learned from the VAE-LSTM network’s outputs.

Employing the Levenberg-Marquardt optimization in pair with the MLAT approach from Section 5.5.1, we compute the UAV’s position. The ranges to the i -th anchor at time t , denoted R_{t,a_i} , are important for this process. We have previously introduced a quadruple range vector $\mathbf{R}_t$, which aggregates the distances from the UAV to each of the four UWB anchors at time t , facilitating the calculation of the UAV’s location. The distance R_{t,a_i} from the UAV to any UWB anchor i at a specific time t is determined using the Euclidean distance formula. For the i -th anchor at coordinates $(X_{a_i}, Y_{a_i}, Z_{a_i})$ and the UAV positioned at $\mathbf{x}_t$ at the same time, this distance is calculated as:

$$R_{t,a_i}(x_t, y_t, z_t) = \sqrt{(x_t - X_{a_i})^2 + (y_t - Y_{a_i})^2 + (z_t - Z_{a_i})^2} \quad (5.3)$$

Through the first-order Taylor expansion, we evaluate the influence of range measurement errors on position estimation, leading to a sensitivity matrix represented by Equation 5.4. The variable R_{t,a_i} represents the range data measured by the i -th anchor at time t , while $\tilde{R}_{t,a_i}$ denotes the ground truth range measured by the i -th anchor, calculated using the ground truth data explained in Section 5.2.3. This forms the foundation for calculating the position covariance using the uncertainty propagation principle, as depicted in Equation 5.7.

$$\tilde{R}_{t,a_i} \approx R_{t,a_i}(\mathbf{x}_t) + \left. \frac{\partial R_{t,a_i}(\mathbf{x}_t)}{\partial x_t} \right|_{\mathbf{x}_t} \Delta x + \left. \frac{\partial R_{t,a_i}(\mathbf{x}_t)}{\partial y_t} \right|_{\mathbf{x}_t} \Delta y + \left. \frac{\partial R_{t,a_i}(\mathbf{x}_t)}{\partial z_t} \right|_{\mathbf{x}_t} \Delta z \quad (5.4)$$

The Jacobian matrix J_t , key in mapping range uncertainties to positional uncertainties, is shown in Equation 5.5.

$$\Delta R_t = \tilde{R}_t - \mathbf{R}_t = J_t \times \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} \quad (5.5)$$

The i -th row of J_t will be:

$$J_i^t = \left[\frac{x_t - X_{a_i}}{R_{t,a_i}(\mathbf{x}_t)}, \frac{y_t - Y_{a_i}}{R_{t,a_i}(\mathbf{x}_t)}, \frac{z_t - Z_{a_i}}{R_{t,a_i}(\mathbf{x}_t)} \right] \quad (5.6)$$

The residual error in the target position along the three axes can be expressed as a function of the differences ΔR between the ground truth and the measured ranges. By applying the law of uncertainty propagation as described in [123], the covariance of the position Σ_{x_t} can be derived from the covariance of the range Σ_{R_t} .

$$\Sigma_{x_t} = (J_t^T J_t)^{-1} J_t^T \Sigma_{R_t} J_t (J_t^T J_t)^{-1} \quad (5.7)$$

The final results, combining the estimated positions and their dynamic covariances, are then incorporated into the pose graph optimization, enhancing both accuracy and resilience of the localized UAV positioning. Equation 5.7 demonstrates the propagation of uncertainty from range measurements to the estimated position, providing a refined prediction of the UAV's localized position with quantified confidence levels.

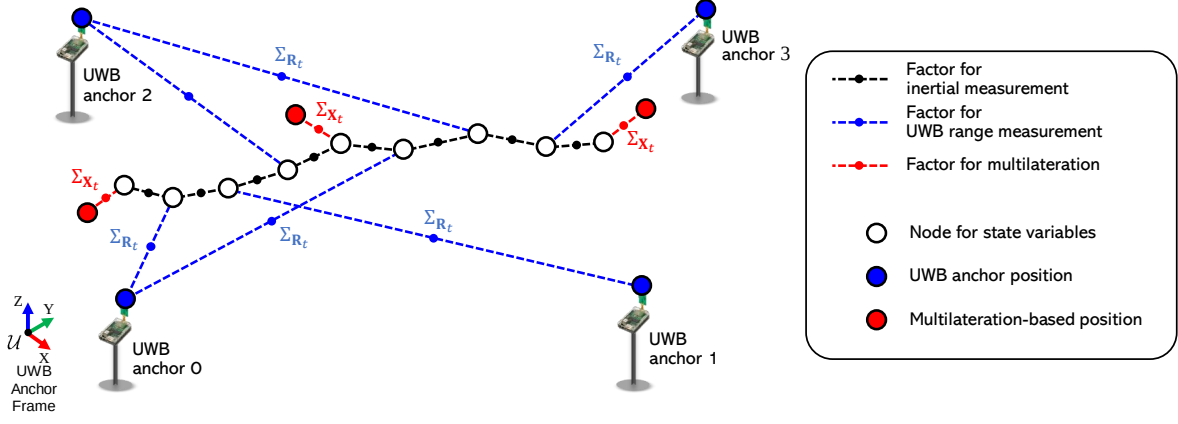


Figure 5.3 Illustration of incremental smoothing pose graph representing the proposed DCPG with dynamic covariance.

5.6 Dynamic Covariance Pose Graph

The DCPG is proposed to enhance the pose graph-based smoothing framework for precise 3D UAV positioning. My methodology can process diverse sensor measurements, integrating various measurement types. A pivotal addition is the integration of dynamic covariance, predicted by a deep neural network.

In my framework, sensor measurements are encapsulated as factors $f_i(x_i)$ within a Gaussian noise model:

$$f_i(x_i) \propto \exp \left(-\frac{1}{2} \|h_i(x_i) - z_i\|_{\Sigma_i}^2 \right) \quad (5.8)$$

Where z_i represents the observed measurement, $h_i(\cdot)$ the measurement function, and Σ_i the dynamic covariance matrix associated with each measurement i . This model underpins the objective function for UAV state estimation within the pose graph. By employing the iSAM2 algorithm [124], we achieve efficient inference, yielding MAP estimates for a comprehensive set of UAV states—position (x_{pos_t}), orientation (x_{ori_t}), velocity (x_{v_t}), and IMU biases (x_{b_t}).

The pose graph integrates three principal factors, as shown in Figure 5.3 to model the UAV's trajectory accurately, each corresponding to a different sensor measurement and

associated with its dynamic covariances Σ_t :

- **UWB Range Measurement Factor** ($f_{\text{range},t}$): This factor evaluates the discrepancy between measured UWB ranges and the UAV's estimated range from the anchors, taking into account the dynamic uncertainty of these measurements.

$$f_{\text{range},t}(x_{\text{pos},t}) \doteq \exp \left(-\frac{1}{2} \| \|x_{\text{pos},t} - \mathbf{X}_t\| - \mathbf{R}_t \|_{\Sigma_{\mathbf{R}_t}}^2 \right) \quad (5.9)$$

where $x_{\text{pos},t}$ represents the UAV's estimated position at time t , $\mathbf{X}_t$ is the matrix of positions of the UWB anchors, $\mathbf{R}_t$ is the quadruple range measurement vector from the UAV to each of the four UWB anchors at time t , and $\Sigma_{\mathbf{R}_t}$ is the dynamic covariance matrix for the range measurements at time t .

- **MLAT Measurement Factor** ($f_{\text{mlat},t}$): Applies MLAT techniques to enhance the UAV's position estimation from UWB range measurements, factoring in the cumulative uncertainty.

$$f_{\text{mlat},t}(x_{\text{pos},t}) \doteq \exp \left(-\frac{1}{2} \|x_{\text{pos},t} - x_{\text{mlat},t}\|_{\Sigma_{\mathbf{X}_t}}^2 \right) \quad (5.10)$$

where $x_{\text{mlat},t}$ denotes the position of the UAV as estimated through MLAT at time t , and $\Sigma_{\mathbf{X}_t}$ is the dynamic covariance matrix for the MLAT-based position estimates at time t .

- **Inertial Measurement Factor** ($f_{\text{IMU},t}$): Let $\mathbf{x}_{\text{pvb}}^{t-} = [\mathbf{x}_{\text{pos}}^{t-}, \mathbf{x}_{\text{ori}}^{t-}, \mathbf{x}_{\text{v}}^{t-}, \mathbf{x}_{\text{b}}^{t-}]$ represent the state at the previous time step t^- when preintegration began ($t^- < t$). Similarly, $\mathbf{x}_{\text{pv}}^t = [\mathbf{x}_{\text{pos}}^t, \mathbf{x}_{\text{ori}}^t, \mathbf{x}_{\text{v}}^t]$ denotes the current state at time t .

We define two factors:

$$f_{\text{ine}}^t(\mathbf{x}_{\text{pvb}}^{t-}, \mathbf{x}_{\text{pv}}^t) \doteq d(\mathbf{x}_{\text{pv}}^t - h(\mathbf{x}_{\text{pvb}}^{t-}, \Delta \mathbf{x}_{t^-,t})) \quad (5.11)$$

$$f_{\text{bias}}^t(\mathbf{x}_{\text{b}}^{t-}, \mathbf{x}_{\text{b}}^t) \doteq d(\mathbf{x}_{\text{b}}^t - g(\mathbf{x}_{\text{b}}^{t-})) \quad (5.12)$$

Here, $\Delta \mathbf{x}_{t^-,t}$ represents the relative motion from t^- to t obtained from pre-integrated inertial measurements, and $g(\mathbf{x}_b^{t^-})$ is a random walk model.

The DCPG estimates UAV's position and navigation by considering various factors. Each factor has a dynamic covariance matrix (Σ_t), which adapts over time based on sensor measurements. This dynamic approach provides accurate state estimation, improving the UAV's positioning and navigation. The DCPG has adopted the 'add-after-eliminating' strategy, which was suggested in chapter 4. This strategy involves adding two types of range-related factors into a graph without violating the factor graph condition. After obtaining multilateration results, range and corresponding IMU factors are removed from the graph. Then, a new multilateration factor and a singular IMU factor are introduced, ensuring all factors remain independent. Each range is used for one single factor.

Add-After-Eliminating

In a related work [49], we introduced an innovative UAV positioning algorithm that dynamically adjusts to the incoming UWB / IMU data. The algorithm evaluates the feasibility of multilateration using at least four data points. Without sufficient data for multilateration, a range factor is introduced into the pose graph. The "add-after-eliminating" strategy ensures that the graph remains independent, with each range measurement contributing to only one factor in the graph.

Learning-Based Probabilistic Smoothing

Real-world scenarios expose variables and measurements to many uncertainty sources, such as sensor noise or imprecise initial conditions. To account for these variances, we introduce a dynamic covariance pose graph model that adjusts in real-time.

In my previous approach to pose graph smoothing, we simplified by assuming that the measurement uncertainty remains the same throughout the entire drone flight. However, the drone encounters varying measurement uncertainties as it moves through different situations. To tackle this challenge, we introduce a dynamic covariance pose graph.

5.7 Experimental Results

5.7.1 UAV System and UWB Sensor

The UAV system is based on the DJI M100 quadcopter. At the UAV's base, we have installed an Intel NUC, a UWB tag, and a battery as part of the payload. In addition, a 360 degree mini prism is affixed to the central bottom of the UAV to provide ground-truth position data. We have integrated a total of five Time Domain PulsON 440 UWB modules for UWB sensing, capable of delivering accurate range measurements up to several hundred meters in clear line-of-sight scenarios. Of these modules, one serves as the UWB tag mounted on the UAV, while the other four function as UWB anchors.

A ROS Kinetic software on Ubuntu 16.04 was implemented to facilitate sensor data acquisition. The UAV's IMU provides data, including linear acceleration, angular velocity, and altitude, at a rate of 100 Hz. UWB range data is obtained by having the UWB tag cyclically request distances from the anchors. Each anchor provides range data at 10 Hz, resulting in a cumulative rate of 40 Hz from all four anchors. Additionally, the positioning algorithm is implemented using the GTSAM library [125] and my neural network was implemented using the TensorFlow KERAS framework. [126]

5.7.2 UAV Experimental Setup

As explained in chapter 3, the experiments were carried out at five distinct locations, including an indoor, open sports field, near a building with glasses, a semi-open tunnel and underneath a bridge. These experiments involved a cumulative flight time of 1 hour, 50 minutes and 28 seconds, covering approximately 4.3 kilometres while performing various flight maneuvers. Spatial relationships between the UAV and the UWB anchors were configured according to various patterns. More detailed information can be found in chapter 3.

5.7.3 Network Implementation and Training

The implementation leverages TensorFlow, a leading deep learning framework, for model development. Preprocessing involves formatting the data to suit the requirements of time-distributed data processing and reshaping the input data into time steps multiplied by 4. Ground-truth labelling data, collected using prism tracking, section 5.2.3. The training process involves partitioning the data sets from various environments into training and test sets, as detailed in chapter 3.

Each training set comprises data collected from at least one representative scenario, encompassing nearly 60% of the training dataset’s input data. This encompasses 3221 seconds of travelling time and a flight distance of 2390.8 meters. It should be noted that training is conducted separately for different environments. In training, we used a learning rate schedule with an initial rate of 0.001, decreasing by 0.9 every 10,000 steps. The batch size was set to 32 for efficient processing, balancing speed and convergence. We employed early stopping based on validation loss to determine the optimal number of epochs for each dataset, ensuring effective training without overfitting.

To optimize this VAE-LSTM network, we employ the Adam optimizer [127], selected for its adaptive learning rate capabilities, facilitating efficient model convergence even with noisy sensor data. This choice is particularly pertinent given the varying gradients and data sparsity encountered in UWB range measurements.

5.7.4 Results and Discussion: Neural Network Performance

Refinement of Range Data and MLAT Positioning Analysis

In this section, the performance of the VAE-LSTM neural network is analyzed. To accomplish this, the results of the dual outputs of the network—range and its covariance—are compared to their raw input. The MLAT position and its covariance, utilizing these refined values, are also examined.

The initial output of the network yields the refined range. Utilizing this refined range

via multilateration, we computed the refined MLAT positioning. A comparison of the RMSE between these two values and their respective refined ranges is presented in Table 5.1. Regarding range values, the field dataset exhibited the best result with an improvement of 0.19 meters, while the tunnel dataset showed the most significant improvement with a 0.41-meter enhancement. The tunnel, bridge, building, indoor, and field datasets improved progressively. This outcome was anticipated due to the varying levels of multipath interference present in these environments, with the tunnel dataset having a higher incidence of multipath and the field dataset showing fewer multipath outliers. Figure 5.4 shows the network’s capability in rectifying raw range data corrupted by multipath interference in one of the case studies in indoor environments where multipath occurs in a few instances.

Utilizing these refined ranges, the MLAT positioning follows a similar trend. For instance, the field dataset shows only marginal improvement, while the tunnel dataset exhibits the most significant enhancement in refined positioning accuracy.

Table 5.1 Comparison of RMSE between the Refined Range calculated by DeepCovPG and its Raw Value (left), and the MLAT Pose calculated using the Refined Range, along with its value using the raw ranges (right).

	RMSE [m]			RMSE [m]		
	Raw range ($\mathbf{R}$)	Refined range ($\hat{\mathbf{R}}$)		MLAT position ($\mathbf{x}_{MLAT}$)	Refined MLAT position ($\hat{\mathbf{x}}_{MLAT}$)	
		Training	Testing		Training	Testing
Indoor	0.56	0.14	0.26	1.43	0.75	0.97
Field	0.42	0.17	0.19	0.43	0.35	0.41
Building	0.57	0.21	0.26	2.76	1.21	1.92
Bridge	0.47	0.19	0.28	0.45	0.37	0.43
Tunnel	0.72	0.26	0.31	1.99	1.27	1.54
Average	0.54	0.19	0.26	1.416	0.78	1.04

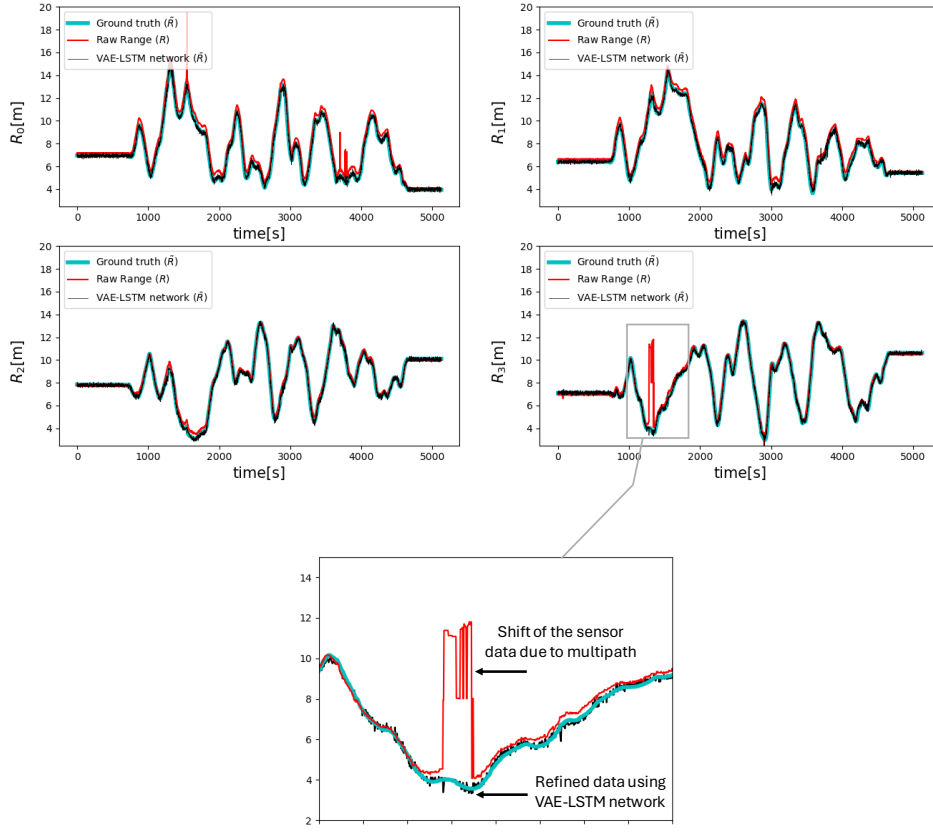


Figure 5.4 Illustrating the network’s capability in rectifying raw range data corrupted by multipath interference for indoor environment set 1.

Analysis of Range and MLAT Covariance

In this section, I explore the second output of the network: the range covariance. Additionally, the covariance of the MLAT positioning through uncertainty propagation is calculated. My aim is to compare the predicted covariances with the static covariances to assess their performance.

The constant covariance can either be a default value (as utilized in the incremental smoothing in the previous chapter) or calculated using ground truth values: $\Sigma_R = \frac{1}{N} \sum_{i=1}^N (R_i - \tilde{R})(R_i - \tilde{R})^T$. In the previous chapter, we assumed independence among the four range measurements, resulting in a diagonal covariance matrix. Each range measurement had a standard deviation of 0.3. However, this study considers interdependencies among

measurements due to factors such as signal noise, environmental dynamics, and clock drift, learning the covariance matrix using a VAE-LSTM network.

Table 5.2 presents a comprehensive overview of the RMSE values for both static and dynamic covariances of range and the calculated MLAT covariance across various environments. In the testing sets, the RMSE values for dynamic covariance ranged from as low as 0.32 in indoor scenarios to as high as 1.02 in tunnels. This demonstrates a limited three-fold variation in RMSE across vastly different settings. In contrast, static covariance shows a much greater variation, ranging from 0.54 in Field scenarios to 2.31 in Tunnels. The average RMSE for dynamic covariance in the testing sets was 0.60, compared to 1.23 for static covariance. This represents a substantial improvement of approximately 51% in RMSE when utilizing dynamic covariance.

In Figure 5.5, I present a 4x4 matrix illustrating covariances for one case study, indoor dataset set 1. The matrix represents the covariance structure of the positioning system, encompassing 8 parameters indicating 10 unique elements of the covariance matrix. These parameters are derived from combining 4 range values from 4 anchors. Specifically, the first two rows of the matrix correspond to the diagonal values, representing the variances of the position estimates. The remaining elements in the upper triangular portion of the matrix mirror the values in the lower triangular portion, capturing the covariances between the position estimates.

Importantly, the network output depicted in the figure aligns closely with the ground truth covariance, indicating the effectiveness of the VAE-LSTM network in capturing the underlying covariance relationships within the positioning data. The significant reduction in RMSE values for dynamic covariance, especially in complex structures such as tunnels and buildings, demonstrates the ability of DeepCovPG to adapt effectively to challenging navigation situations.

Table 5.2 Comparison of RMSE between the Predicted Covariance Matrix of range calculated by DeepCovPG and its static value (left) and the Covariance Matrix of MLAT Pose calculated by Covariance Propagation, along with its static value (right).

	RMSE [m^2]			RMSE [m^2]		
	Static Cov. ($\Sigma_{\mathbf{R}}$)	Dynamic Cov. ($\Sigma_{\bar{\mathbf{R}}}$)		Static Cov. ($\Sigma_{\mathbf{x}}$)	Dynamic Cov. ($\Sigma_{\bar{\mathbf{x}}}$)	
		Training	Testing		Training	Testing
Indoor	0.82	0.13	0.32	3.03	2.62	2.77
Field	0.54	0.23	0.42	2.89	1.74	2.01
Building	1.32	0.42	0.51	3.84	1.71	2.14
Bridge	1.25	0.54	0.63	3.53	1.83	2.07
Tunnel	2.31	0.76	1.02	4.88	2.59	2.97
Average	1.23	0.42	0.60	3.51	2.09	2.39

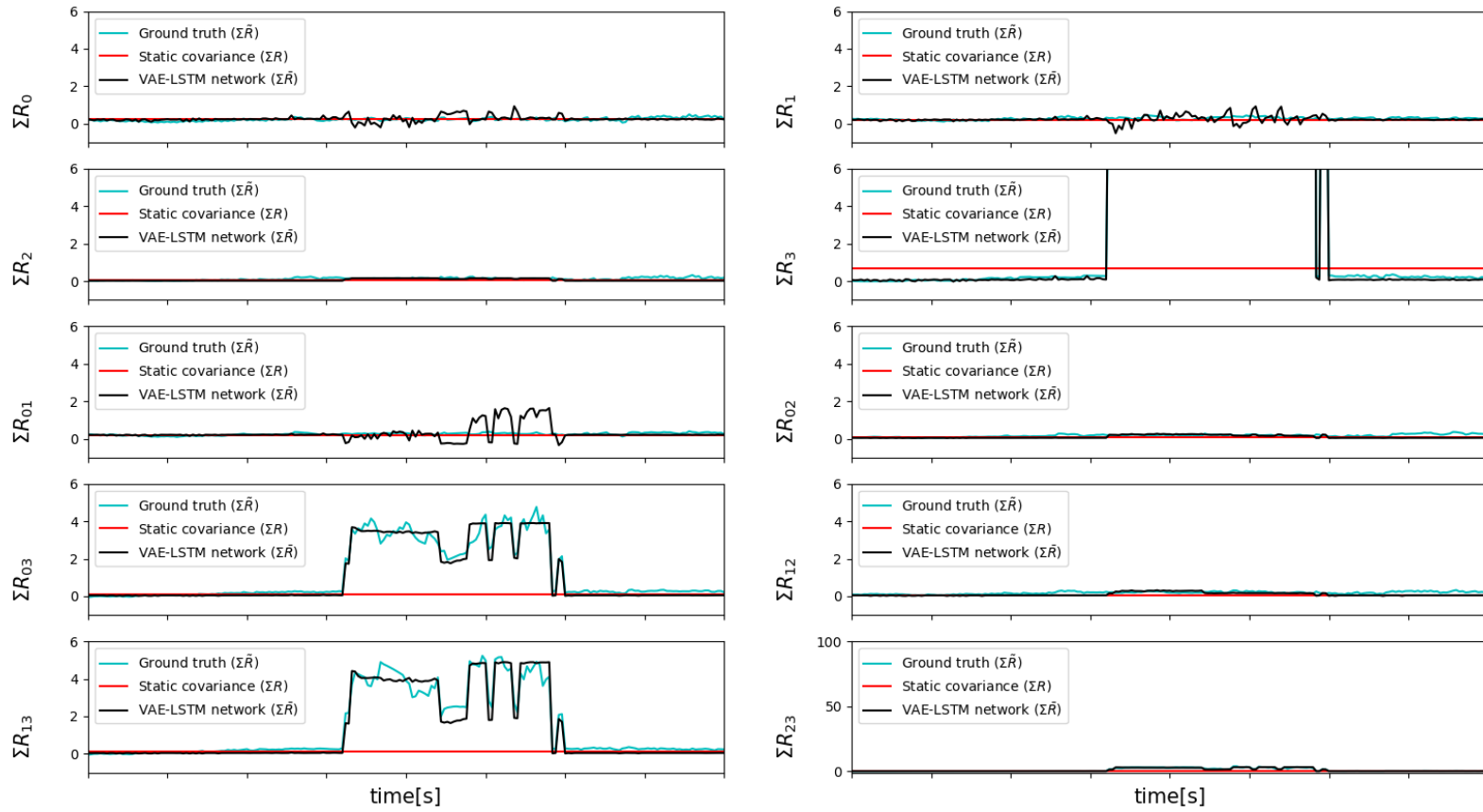


Figure 5.5 Covariances depicted in a 4x4 matrix for indoor dataset set 1, showcasing ground truth (cyan), static covariance (red), and predicted covariance by VAE-LSTM network (black), illustrating the alignment between network outcome and ground truth compared with static covariance.

5.7.5 Results and Discussion: Positioning Accuracy

I employed two baselines for comparison in my study. Baseline 1, referenced as [128], originates from my benchmark dataset paper, where UAV positions were determined using multilateration techniques. Baseline 2, referenced as [49], is from another of my papers. It involves a smoothing pose graph framework, which we utilized as a comparison baseline for my dynamic pose graph approach.

Figure 5.6 displays the trajectories of selected sample sequences generated using my method, and table 5.3 provides a comprehensive evaluation of the MAE for MLAT without Pose Graph, the baseline Pose Graph (proposed pose graph with static covariance), and the proposed DeepCovPG approach. For the baseline method, the overall MAE across all test environments was 0.264 m. In contrast, DeepCovPG achieved a lower overall MAE of approximately 0.169 m in the test scenarios. Even in open field conditions, DeepCovPG outperformed the baseline Pose Graph, showcasing an MAE of 0.192 m against 0.231 m (36% improvement). This advantage is not limited to open-field conditions; even in more complex settings such as buildings, tunnels, and bridges, DeepCovPG consistently produces lower MAE values.

The superior performance of DeepCovPG is largely attributed to its dynamic covariance modelling, allowing it to adapt effectively to different environmental challenges. This adaptability significantly enhances its resilience to issues like multipath effects, which are prevalent in complex settings. Upon breaking down the error metrics into individual dimensions (x, y, z), DeepCovPG also excels, particularly on the z-axis, demonstrating its utility in applications requiring high-precision spatial orientation data.

5.8 Summary

In this study, we presented DeepCovPG, a novel framework that integrates deep learning algorithms with UWB technologies to improve the positioning accuracy of UAVs substantially. Using VAEs and LSTM networks, DeepCovPG excels in refining and denoising UWB range

Table 5.3 The MAE of DeepCovPG outcomes (UNIT: METER).

-	-	Indoor		Field		Building		Bridge		Tunnel		Total	
Number of Sets	-	5		5		3		4		6		23	
Baseline 1: MLAT (chapter 3)	x	0.655		0.245		1.570		0.283		1.094		0.769	
	y	1.141		0.208		2.253		0.185		1.623		1.082	
	z	0.566		0.292		0.305		0.302		0.393		0.372	
	xyz	0.787		0.248		1.376		0.257		1.037		0.741	
Baseline 2: pose graph with two factors (chapter 3)	x	0.055		0.068		0.077		0.084		0.037		0.064	
	y	0.070		0.210		0.462		0.180		0.573		0.299	
	z	0.473		0.217		0.247		0.255		0.265		0.291	
	xyz	0.502		0.352		0.548		0.319		0.622		0.469	
Baseline 3: pose graph (chapter 4)	x	0.052		0.015		0.014		0.021		0.012		0.023	
	y	0.058		0.208		0.231		0.211		0.209		0.183	
	z	0.186		0.094		0.097		0.189		0.188		0.151	
	xyz	0.225		0.231		0.255		0.291		0.319		0.264	
Proposed Method: DeepCovPG	-	train	test	train	test	train	test	train	test	train	test	train	test
	x	0.043	0.048	0.010	0.012	0.013	0.015	0.014	0.017	0.011	0.012	0.014	0.021
	y	0.047	0.051	0.192	0.200	0.074	0.092	0.104	0.135	0.103	0.142	0.104	0.124
	z	0.097	0.126	0.075	0.089	0.082	0.091	0.093	0.102	0.092	0.131	0.088	0.108
	xyz	0.193	0.201	0.158	0.192	0.197	0.203	0.162	0.193	0.204	0.235	0.183	0.204

data while also dynamically predicting measurement covariances.

Through rigorous empirical assessments in GNSS-denied and UWB multipath-prone environments, we established the significant advantages of DeepCovPG over traditional approaches. Specifically, the framework improved MAE by approximately 36% and RMSE by approximately 51% in challenging contexts such as tunnels, bridges, and buildings. These improvements indicate the system’s adaptability and robustness and highlight its exceptional performance in complex, real-world situations.

Furthermore, my work lays the foundation for future adaptations that can integrate DeepCovPG with IMUs and other types of pose sensors. This extension would potentially lead to even more robust and accurate positioning solutions across various platforms, not just UAVs.

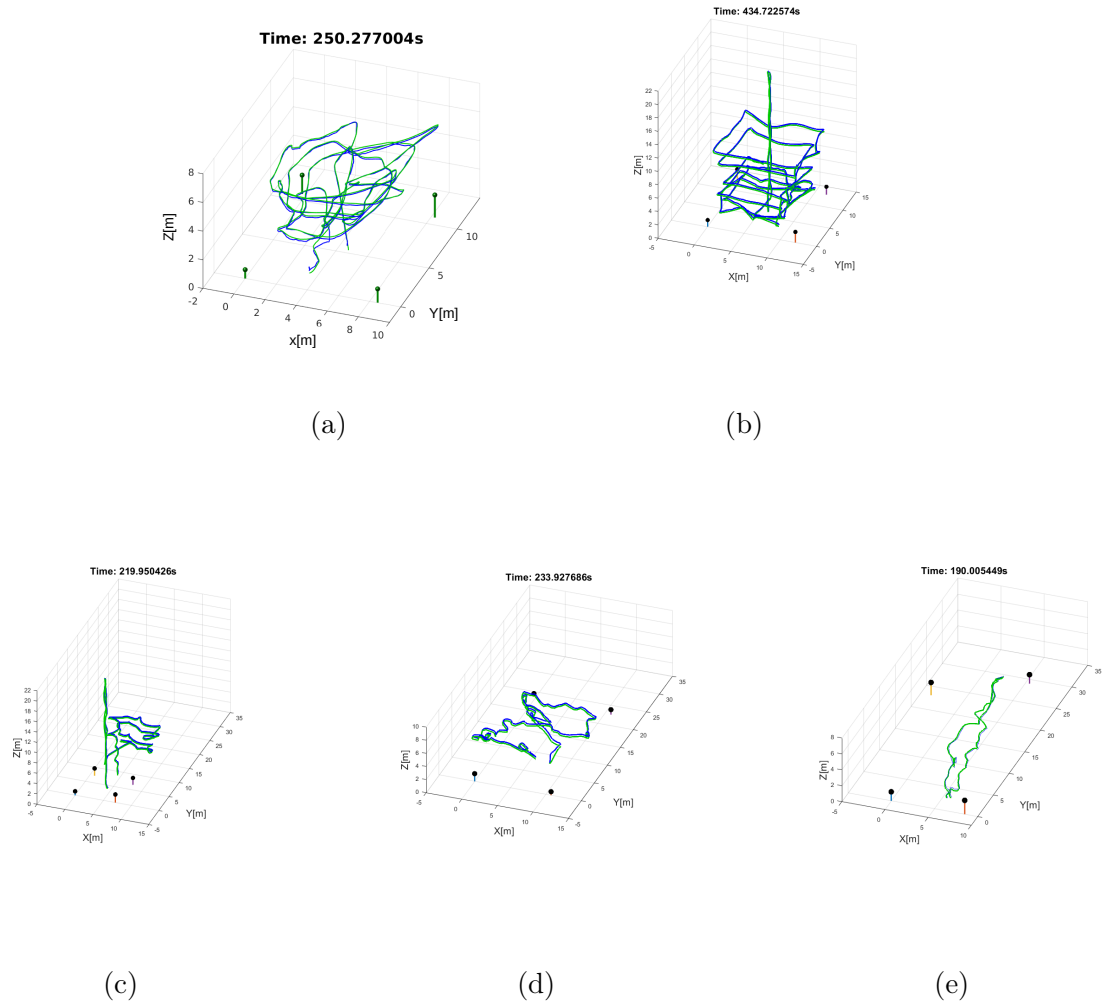


Figure 5.6 Comparison between estimated positions by DeepCovPG (blue) and ground truth (green) across various environments: (a) indoor, (b) field, (c) building, (d) bridge, and (e) tunnel. The complete trajectories across all datasets and environments are available in the appendix.

Chapter 6

Integrating AI and Geometric Methods for LiDAR SLAM

The primary objective of this chapter is to develop an autonomous, unmanned aerial photogrammetry system designed for inspecting vertical infrastructures that pose challenges for conventional access methods. A key focus in achieving this goal is the creation of a precise navigation and positioning system capable of functioning seamlessly in GNSS-denied environments. Initially, the plan was to employ a fusion of UWB and LiDAR sensor technologies to construct a robust positioning system.

Previous chapters established the foundation for precise UAV localization using UWB and IMU data. Chapter 3 introduced a multilateration-based method, enhanced with the Levenberg–Marquardt non-linear minimization algorithm, for offline tests in diverse environments. Chapter 4 and 5 then presented a novel approach combining high-rate 1D UWB range measurements with low-rate multilateration outcomes using an incremental smoothing-based method, significantly improving UAV positioning accuracy.

However, due to the pandemic, I could not collect new LiDAR data with the drone. As a result, I adapted my approach to use the KITTI dataset, which provides LiDAR data collected using a car instead of a drone. In this chapter, I advance the localization capabilities by proposing a fusion method that leverages the strengths of both geometric and deep-learning odometry techniques. My methodology overcomes the limitations inherent in each approach when employed independently. At its core, the localization method relies on geometry-based

LiDAR odometry, while deep learning techniques are utilized to handle inconsistencies and noise within the environment.

This work is my original contribution and is currently drafted for publication in:

Z. Arjmandi, G. Sohn, and C. Armenakis, “*Dynamic Recalibration in LiDAR SLAM: Integrating AI and Geometric Methods with Real-Time Feedback Using INAF Fusion*” in RA-L Journal, 2024.

6.1 Overview

In the subsequent sections, the discussion begins with an overview of the three different pipelines used in my approach. These include two single-branch setups (ICP-QE and DL-PE) and one fused setup combining both ICP-QE and DL-PE branches, followed by the Odometry Prediction Block and the Loop Closure and Optimization Block. Understanding these configurations is crucial for grasping the subsequent methodological choices.

Next, the focus shifts to the choice between ensemble learning and fusion networks for combining predictions from multiple models. This comparison provides the rationale behind selecting fusion techniques over ensemble learning, emphasizing the integration of complementary strengths of different branches within a unified framework.

Following this, the discussion moves to adapting networks to new data and unobserved motion. The innovative concept of Inferred Attention is introduced, dynamically adjusting attention weights post-training. This approach enables adaptation without retraining the network, refining attention scores based on kinematic changes. This section outlines the rationale for choosing Inferred Attention over incremental learning and traditional attention-based methods.

The chapter then explores the details of the fusion approach, examining three distinct fusion techniques: Direct Fusion, Attention-Based Fusion, and my unique contribution, the INAF. The INAF module stands out by employing advanced inference techniques to dynamically select the most relevant information from both geometric and deep learning

odometry, resulting in superior localization results, especially in challenging environments.

The chapter concludes with a clear thesis statement outlining the objectives and research questions, organized into two distinct phases. Step 1 focuses on analyzing various odometry and mapping techniques. In contrast, Step 2 aims to design a fusion module that optimizes performance under diverse conditions, introducing the unique INAF as a novel contribution.

6.2 Methodology

6.2.1 Odometry Pipelines

This study introduced three distinct odometry pipelines (Figure 6.1). The first pipeline, utilizing the geometric ICP methodology, named ICP-QE, begins with geometric computations between consecutive scans. This is followed by an odometry prediction block that computes the transformations. The loop closure and optimization block ultimately integrate these results to estimate the drone’s trajectory.

The second pipeline, DL-PE, leverages deep learning techniques to extract and predict motion features from sensor data. Similar to the first pipeline, this is followed by an odometry prediction block and subsequently by the loop closure and optimization block, which refines and integrates the estimated trajectory.

The third pipeline combines the strengths of both ICP-QE and DL-PE methodologies. Both pipelines contribute to the odometry prediction block, synthesizing their outputs there. This combined approach enhances the accuracy and robustness of the odometry estimation. Finally, as with the other pipelines, the loop closure and optimization block processes these estimates to refine the trajectory estimation for comprehensive mapping and localization tasks.

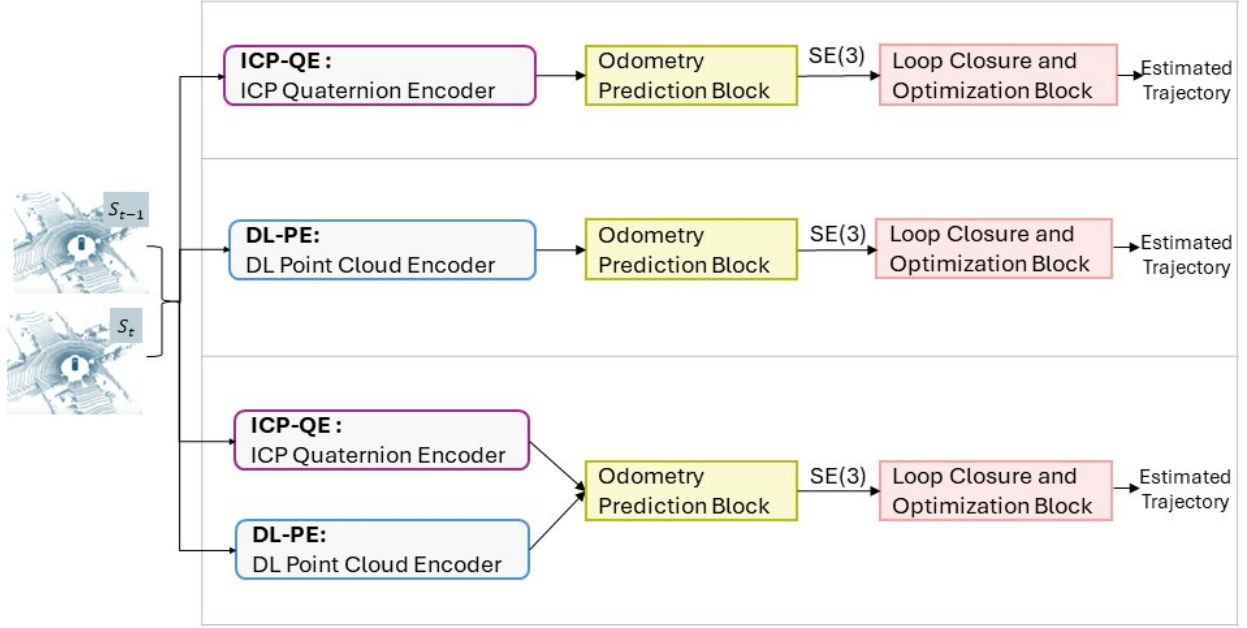


Figure 6.1 Pipelines illustrating three configurations: two single-branch setups (ICP-QE and DL-PE) and one fused setup combining both ICP-QE and DL-PE branches, followed by the Odometry Prediction Block and the Loop Closure and Optimization Block.

6.2.2 Ensemble Learning vs. Fusion Networks

Ensemble Learning and Fusion networks are integral methods that combine predictions from several models to improve performance and reliability.

Combining multiple models through ensemble learning results in better prediction performance than any single model could achieve. Bootstrap Aggregating, also known as Bagging, requires training multiple instances of a model on different subsets of this training data and then averaging their predictions. Bagging usually involves combining the results of multiple decision trees, such as Random Forests. On the other hand, Boosting sequentially trains models, with each subsequent model concentrating on adjusting errors made by previous ones. AdaBoost and Gradient Boosting Machines can be given as examples. Strong predictive models often result from boosting but can be overfitted if not managed carefully. Stacking is

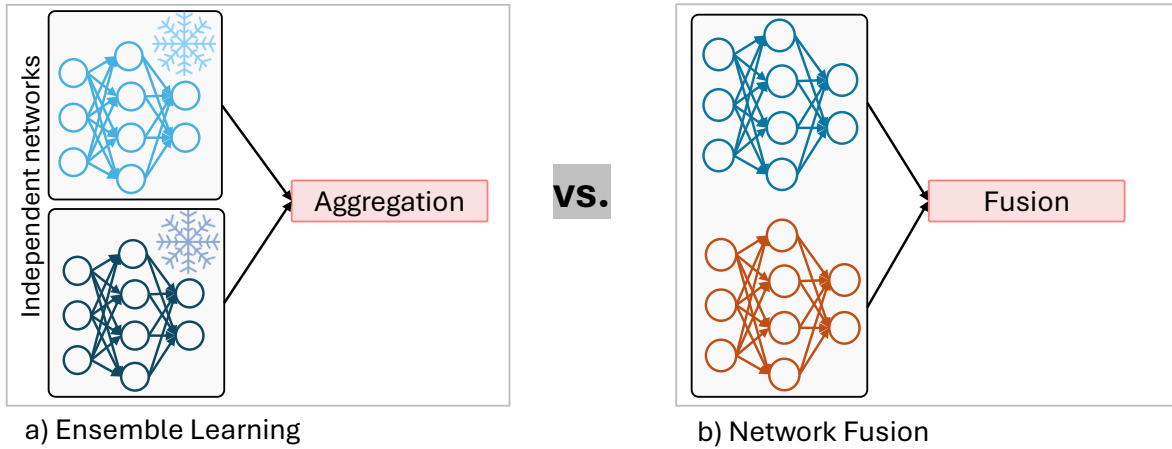


Figure 6.2 a) Ensemble Learning vs. b) Network Fusion. In ensemble learning, multiple networks are trained independently and remain frozen during combination. In contrast, network fusion involves training multiple networks together, allowing for integrated learning and mutual adaptation.

about using another model (level 1) that tries to learn how best to combine the predictions from the models trained at level 0 after simultaneously training some level 0 models. Level 1 attempts to learn what weightings to put on the predictions from level 0 models that will make them optimal for making accurate predictions. In the voting approach, which is simple, several various models are trained while their predictions are merged either by majority vote for classification or by averaging for regression tasks.

Fusion networks, also known as sensor fusion or multimodal fusion, integrate multiple data sources or models into a unified framework. Unlike ensemble learning, fusion networks typically involve joint optimization of parameters, allowing the model to learn how to combine different inputs during training best. Early fusion combines raw data from multiple sources at the input level before feeding it into the model. It is straightforward but may not always capture complex interactions between data sources. Late fusion combines the outputs of multiple models (each trained on different data sources) at the decision level. It is similar to ensemble learning but allows for more interaction between the models during the decision-making process. Hybrid fusion combines early and late fusion, where data is

partially combined at both the input and decision levels. This method aims to utilize the strengths of both approaches, allowing for more fine interactions between data sources.

Fusion networks were chosen over ensemble learning for this study due to their ability to dynamically adapt and optimize predictions by leveraging the complementary strengths of different models or data sources. While ensemble methods like bagging and boosting improve accuracy and robustness through independent training, they do not allow for the joint optimization of parameters during training. On the other hand, as is shown in Figure 6.2, fusion networks facilitate integrated learning and mutual adaptation, providing better performance in handling complex and dynamic tasks. This makes fusion networks more suitable for the localization system, which requires integrating two approaches to achieve high precision in odometry [129].

6.2.3 Advantages of Inferred Attention Over Incremental and Attention Based Approaches

Introducing Inferred Attention is motivated by its advantages over incremental learning and traditional attention-based methods.

Incremental Learning (Figure 6.3.a): Incremental learning involves continuously updating the model parameters with new data. While effective in some cases, it often requires careful management to prevent catastrophic forgetting and maintain stability in the learned representations. The iterative nature of incremental learning can also be computationally expensive and time-consuming, especially in scenarios where real-time adaptation is crucial.

Traditional Attention-Based Methods (Figure 6.3.b): Traditional attention mechanisms in deep learning typically involve fixed attention weights learned during training. These mechanisms selectively focus on specific parts of input data, enhancing model performance by prioritizing relevant information. However, they may struggle to adapt to unpredictable variations in input patterns or environmental conditions encountered during deployment. This lack of adaptability can limit the system’s robustness in dynamic settings.

Inferred Attention (Proposed Method)(Figure 6.3.c): Inferred Attention represents my

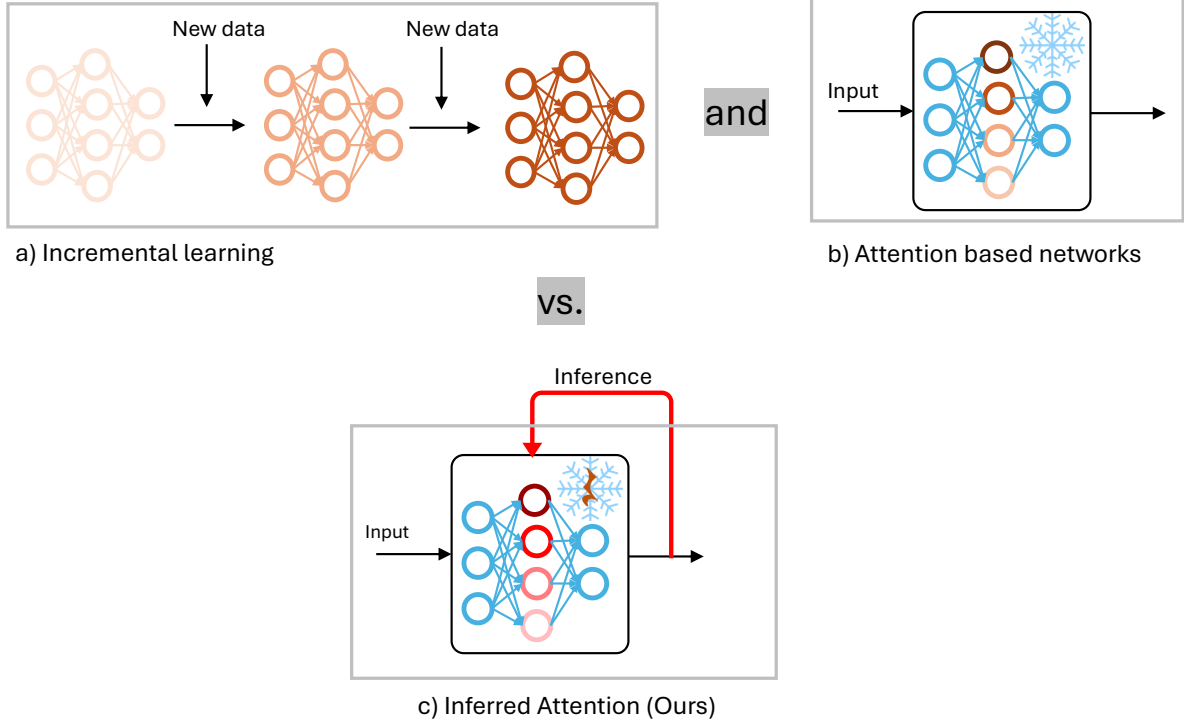


Figure 6.3 Adaptation of networks to new data across different methods: (a) Incremental learning, where the network is updated with new data over time; (b) Attention-based networks, which utilize attention mechanisms to focus on relevant parts of the input; and (c) Inferred Attention (Ours), where network parameters are continuously adapted based on feedback from the environment.

proposed method, introducing a novel approach that dynamically adjusts attention weights post-training. This dynamic adjustment is based on real-time feedback from the environment, enabling the network to respond effectively to unobserved motion and environmental changes. Inferred Attention optimizes model performance by continuously refining attention scores without requiring extensive retraining or manual intervention. This adaptive capability improves localization accuracy and enhances the system’s adaptability and robustness in challenging operational conditions.

By leveraging these capabilities, Inferred Attention represents a significant advancement over traditional incremental learning and fixed attention mechanisms, offering a more

responsive and adaptive solution.

6.2.4 Proposed Inferred Attention Fusion

During training, the networks undergo parameter updates through forward and backward passes, optimizing their internal representations based on the training data. However, once training is complete, the network typically operates statically, processing inputs based on fixed parameters and producing outputs accordingly. This static behaviour may limit the network's ability to handle variations in input patterns and environmental dynamics encountered during deployment.

To address the challenge of adapting to new data and unobserved motion, I propose the "Inferred Attention" approach. Unlike conventional attention-based methods, where attention weights are predetermined during training, Inferred Attention introduces a dynamic inference mechanism. This mechanism continuously adjusts attention weights in real-time, utilizing feedback from the environment to prioritize relevant information and ignore noise or irrelevant signals. By dynamically updating attention weights post-training, the network can adapt its focus to changing conditions, enhancing robustness and accuracy in challenging operational scenarios.

Inferred Attention advances static attention mechanisms and traditional incremental learning approaches. It empowers the network to maintain performance without extensive retraining, making it particularly suited for applications requiring real-time responsiveness and adaptive behaviour in dynamic environments.

6.2.5 Backend Optimization

After extracting odometry data, the system's backend integrates two crucial components: scan loop closure and graph optimization. Scan loop closure employs scan context techniques [130], which enable the system to identify and close loops in the trajectory based on similarities between scans obtained at different times. This process enhances the accuracy and consistency of the trajectory estimation by correcting accumulated errors over time.

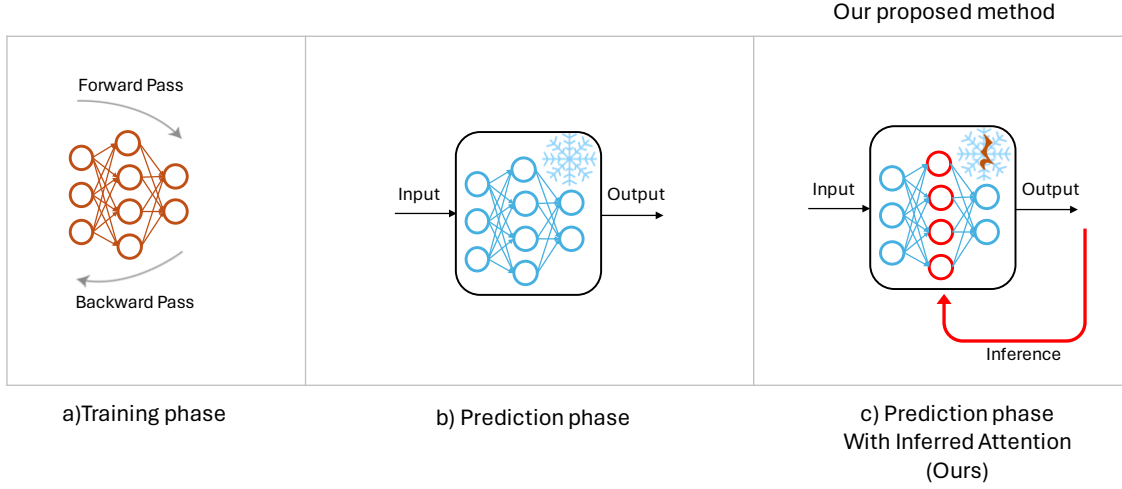


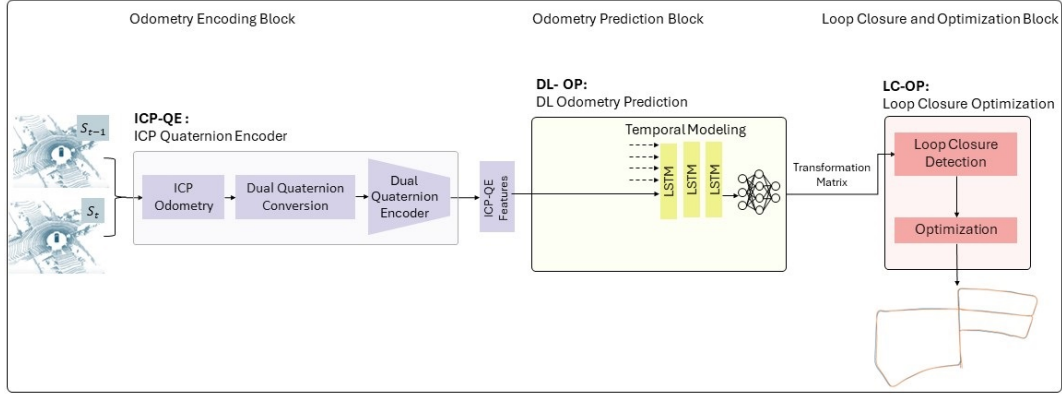
Figure 6.4 Overview of the three phases in my network: (a) Training phase, where the network learns from data; (b) Prediction phase, where the trained network is frozen (indicated by a snowflake symbol) and used for inference, and (c) Prediction phase with Inferred Attention Fusion (INAF), where some network parameters continue to adapt based on feedback from the environment.

Additionally, graph optimization is employed using minisam [131], a universal framework for solving large-scale nonlinear optimization problems commonly found in robotics and computer vision. Minisam optimizes the entire trajectory, considering constraints from odometry measurements and loop closures.

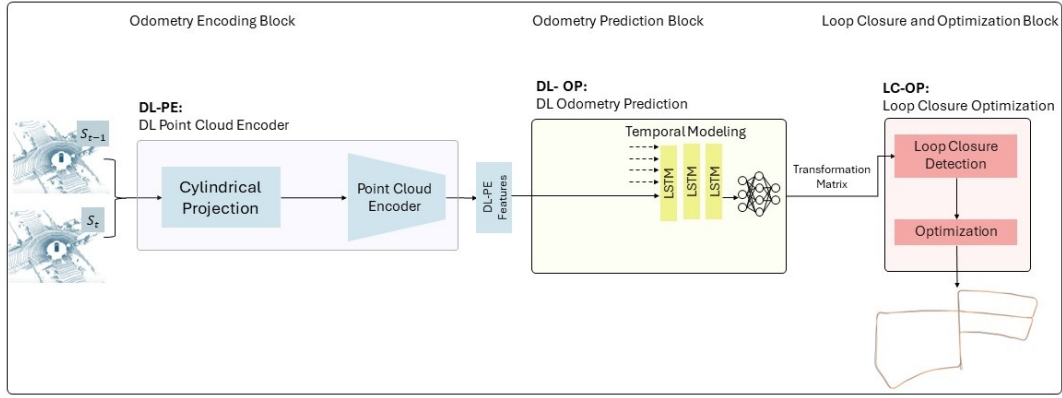
6.3 Explaining System Modules

In my LiDAR SLAM approach, I have developed three distinct odometry pipelines. The detailed architecture is shown in Figure 6.5. These pipelines share a common structure composed of three key blocks: the odometry encoding block, the odometry prediction block, and the loop closure and optimization block. Each pipeline incorporates these blocks but employs different methodologies and combinations to exploit their strengths.

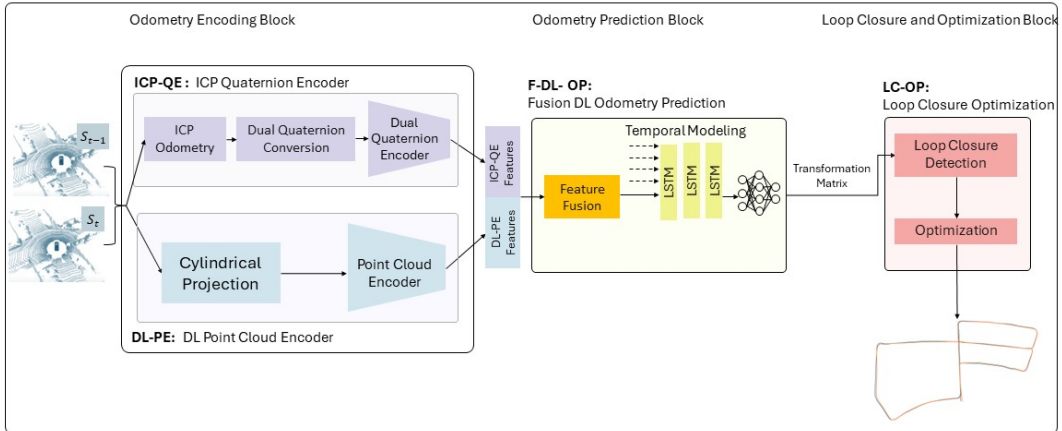
The first pipeline employs the geometric-based ICP-QE branch within its odometry encoding block. This branch focuses on computing transformations between consecutive



(a) ICP-QE block followed by the DL-OP module and the loop closure optimization block.



(b) DL-PE block followed by the DL-OP module and the loop closure optimization block.



(c) Fusion of ICP-QE and DL-PE branches, followed by the F-DL-OP module and the loop closure optimization block.

Figure 6.5 The proposed SLAM system, illustrating three configurations: (a) Fusion of ICP-QE and DL-PE, (b) ICP-QE and (c) DL-PE followed by the Odometry Prediction and Loop Optimization Blocks

scans, relying on geometric principles to provide accurate odometry data. The odometry prediction block in this pipeline consists of a temporal modelling module, followed by fully connected layers for task-solving.

The second pipeline utilizes the DL-PE branch in its odometry encoding block. This branch employs deep learning techniques to extract and predict motion features from sensor data, providing an alternative approach to odometry computation. Like the first pipeline, the odometry prediction block includes a temporal modelling module and fully connected layers for task-solving.

The third pipeline combines the strengths of both the ICP-QE and DL-PE branches within its odometry encoding block. By integrating geometric and deep learning approaches, this pipeline aims to enhance the accuracy and robustness of the odometry estimation. The odometry prediction block in this pipeline includes a feature fusion module to adjust the outputs from both branches, followed by temporal modelling and task-solving layers.

Finally, the loop closure and optimization block are consistent across all three pipelines. This block is crucial in refining the estimated trajectory by integrating loop closure factors and performing graph optimization.

6.3.1 ICP-QE: ICP Quaternion Encoder

The ICP-QE module, designed explicitly for geometric-based odometry, consists of three submodules: ICP Odometry, Dual Quaternion Conversion, and the Dual Quaternion Encoder.

The ICP Odometry submodule employs the Iterative Closest Point algorithm [90] to process LiDAR point cloud data. This algorithm iteratively refines and aligns the point cloud data, using geometric information to estimate the odometry parameters. This alignment process minimizes the difference between successive point clouds, resulting in a transformation matrix representing the motion between scans.

The second submodule, Dual Quaternion Conversion, transforms the obtained transformation matrix into an eight-parameter dual quaternion representation. Dual quaternions provide a compact and efficient way to represent rotations and translations, facilitating the

subsequent encoding.

The Dual Quaternion Encoder’s third submodule consists of dense layers and activation functions designed to encode the dual quaternion parameters into a feature vector. This encoding process captures the essential geometric transformations, providing a robust representation for further processing within the odometry prediction block.

The overall transformation within the Dual Quaternion Encoder is:

$$\theta_{ICP-QE} = f_{\text{Dual Quaternion Encoder}}((q_0, q_1, q_2, q_3) + \varepsilon(q_{e0}, q_{e1}, q_{e2}, q_{e3})) \quad (6.1)$$

With these encoded features, the ICP-QE module fully represents the geometric transformations. These features are prepared for the next stage in the odometry pipeline.

6.3.2 DL-PE: DL Point Cloud Encoder

The DL-PE (Point Cloud Encoder) module, designed for extracting features from LiDAR point cloud data, consists of two submodules: Cylindrical Projection and Point Cloud Encoder.

Point clouds are a set of data points (p^x, p^y, p^z) defined in Cartesian coordinates, offering a comprehensive representation of the 3D structure within a given space. Typically generated by LiDAR devices, these point clouds serve as valuable datasets for various applications, such as environmental mapping and object detection. However, the inherent sparse structure and irregular format of point cloud data make direct processing challenging for neural networks [132].

To address this challenge and enable effective utilization by convolutional neural networks, the Cylindrical Projection submodule (Section 6.6.2) converts the point clouds into a regular matrix format through 2D spherical projection. This transformation reduces the irregularities in the data and aligns with CNNs’ structured input requirements, facilitating more efficient learning and feature extraction by the neural network.

In the Point Cloud Encoder submodule, point cloud data undergo further processing to extract meaningful features. This submodule employs multiple layers of ResNet [133]

to learn the essential characteristics from the point cloud data. Through a combination of convolutional layers, adaptive average pooling, and fully connected layers, the input matrix x_P is transformed into a d -dimensional point cloud feature vector $a_P \in \mathbb{R}^d$. The transformation function is represented as:

$$\theta_{DL-PE} = f_{\text{Pointcloud Encoder}}(u, v, p^z, \rho, \iota, N_\xi, N_v, N_\zeta) \quad (6.2)$$

These features are then passed on to the odometry prediction block for further processing.

6.3.3 Odometry Prediction Block

The Odometry Prediction Module is a critical component within the framework. It is tasked with predicting the transformation matrix based on features received from the Odometry Encoding Block. This module is consistent across the single-branch and dual-branch pipelines, with minor variations to accommodate the fusion of multiple feature sets. It consists of the following submodules: Feature Fusion (for dual branch pipelines), Temporal Modeling, and Task Solver.

In the single branch cases, the module comprises two submodules:

Temporal Modeling: This submodule incorporates temporal dependencies to enhance odometry predictions. A recurrent neural network (RNN), specifically an LSTM, processes the input feature representation z_t at time step t along with its preceding hidden states h_{t-1} , capturing dynamics and temporal correlations. The relationship is formulated as follows:

$$y_t = \text{RNN}(z_t, h_{t-1}) \quad (6.3)$$

Central to effective state estimation is incorporating temporal dependencies, particularly in modelling accurate system states such as relative poses. The LSTM network processes the combined feature representation z_t at time step t , along with its preceding hidden states h_{t-1} , effectively capturing the dynamics and connections across a sequence of features. The hidden states h_t retain the historical information of features relevant to the task.

Task Solver: This submodule consists of dense layers and activation functions that transform the temporal features obtained from the LSTM into the system state. Specifically, a fully connected layer functions as the regressor, converting the learned features into a transformation matrix representing the relative pose.

Feature Fusion: For the dual branch pipeline, the module includes an additional submodule to handle integrating features from both branches. This submodule integrates high-level representations from the ICP-QE and DL-PE branches. The fusion function g combines the features $\theta_{\text{ICP-QE}}$ and $\theta_{\text{DL-PE}}$ into a unified feature vector z :

$$z = g(\theta_{\text{ICP-QE}}, \theta_{\text{DL-PE}}) \quad (6.4)$$

Fusion strategies include direct concatenation, attention-based fusion, and the INAF method. These approaches are designed to capture the most relevant information from both modalities, enhancing the system’s overall performance.

6.3.4 Loss Function

The network outputs relative poses could be $y = [p, r]$, consisting of a 3-dimensional translation vector $p \in \mathbb{R}^3$ and a 3-dimensional Euler rotation vector $r \in \mathbb{R}^3$.

To optimize the neural network parameters θ , the objective is to minimize the Mean Squared Error (MSE) of the relative poses:

$$L(\theta)^2 = \|\hat{p} - p\|^2 + \lambda^2 \|\hat{r} - r\|^2 \quad (6.5)$$

Here, $[\hat{p}, \hat{r}]$ represents the ground truth values, and λ^2 is a scale factor used to balance translational and rotational errors.

Dual quaternions are adopted to eliminate the need for manual adjustment of λ^2 and address the inherent imbalance between translational and rotational losses. Dual quaternions provide an 8-dimensional representation for rigid transformations, effectively managing both rotations and translations in a unified and balanced manner. Compared to other methods,

Their computational efficiency and robustness make dual quaternions highly suitable for this task. Further details on dual quaternions can be found in Section 6.4.

$$L(\theta)^2 = \|\hat{q} - q\|^2 \quad (6.6)$$

This equation represents the squared Euclidean distance between the ground truth dual quaternion $\hat{q}$ and the predicted dual quaternion q , illustrating the network’s ability to learn accurate representations of relative poses.

6.4 Dual Quaternions

Accurately representing rigid transformations is crucial in computer graphics, robotics, and animation. These transformations involve rotations and translations, allowing manipulation of objects in 3D space. Dual quaternions offer a powerful solution in this domain.

In 3D space, the Special Euclidean group (SE), represented as $SE(3)$, encompasses all rigid transformations, merging rotations and translations. This forms the basis for coherent object transformations.

Various methods exist for handling rigid transformations. Euler angles provide an intuitive approach for expressing rotations. Still, they can encounter issues like Gimbal lock, which results in losing one degree of freedom in a three-dimensional space, making certain orientations unachievable [134]. While mathematically precise, rotation matrices can be computationally expensive and prone to numerical errors due to the accumulation of floating-point inaccuracies over successive operations [134]. Quaternions offer a concise and clear way to represent rotations, avoiding Gimbal lock and efficiently interpolating between orientations, thus providing a more robust solution for handling rotations in three-dimensional space [134].

However, combining quaternions with translation vectors presents a challenge, potentially leading to unbalanced loss functions during neural network optimization. Dual quaternions extend the quaternion framework by introducing dual numbers. This extension seamlessly

combines rotation and translation information while maintaining a consistent scale and preventing loss function imbalance. Dual quaternions excel in complex rigid body systems, ensuring smooth transitions between poses and reducing sudden jumps caused by singularities. Compared to rotation matrices, they require fewer operations, speeding up computations essential for real-time applications. While not entirely immune to singularities, dual quaternions manage them more effectively than Euler angles, reducing ambiguities arising from multiple rotation representations. This is supported by their ability to handle complex transformations and maintain stability in computations [135, 136].

Dual quaternions are powerful tools in representing rigid body transformations, combining rotation and translation into a single mathematical entity. They are defined as follows:

$$q = q_0 + \varepsilon q_e \quad (6.7)$$

where q_0 and q_e are standard quaternions, and ε is the dual unit with the properties $\varepsilon^2 = 0$ and $\varepsilon \neq 0$. A standard quaternion q_0 is composed of four parameters representing rotation, and the dual part q_e also consists of four parameters:

$$q = (q_0, q_1, q_2, q_3) + \varepsilon(q_{e0}, q_{e1}, q_{e2}, q_{e3}) \quad (6.8)$$

This encapsulation allows the dual quaternion to represent rotation and translation in 3D space. For a rotation q_r and translation $t = [t_x, t_y, t_z]$, the corresponding dual quaternion DQ is given by:

$$q = q_r + \frac{\varepsilon}{2} q_r \otimes t \quad (6.9)$$

where $\otimes$ denotes quaternion multiplication. Here, q_r represents the quaternion for rotation, and the term $\frac{\varepsilon}{2} q_r \otimes t$ integrates translation, simplifying the computation of combined transformations.

This formulation is particularly useful in computer graphics and robotics for smoothly interpolating motions and handling complex kinematics scenarios without the Gimbal lock

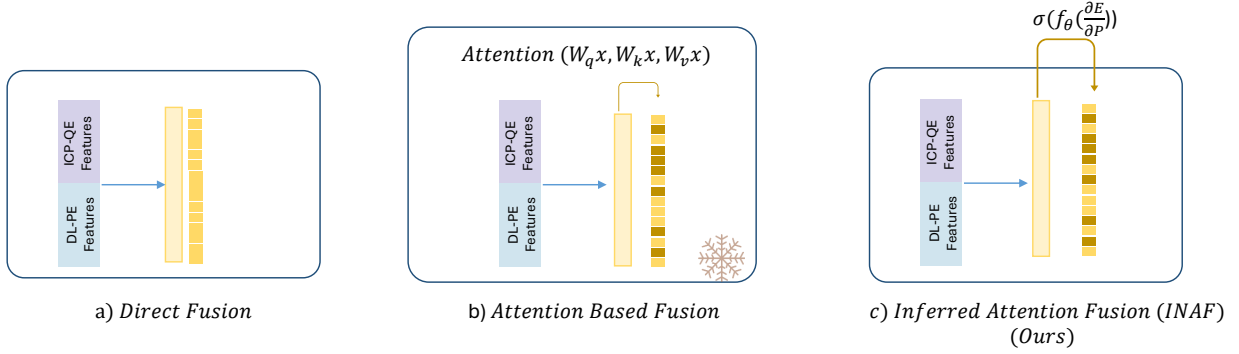


Figure 6.6 Illustration of three fusion methods: (a) Direct Fusion, where inputs are concatenated directly; (b) Attention Fusion, which leverages multihead self-attention mechanisms; and (c) Inferred Attention Fusion (INAF) (Ours), which adapts network parameters based on environmental feedback for improved performance.

that affects Euler angles. It provides a more stable and efficient way of handling 3-D transformations than the separate handling of rotation matrices and translation vectors.

6.5 Fusion Modules

In this section, feature fusion modules are explored. Recognizing the limitations of considering all features equally important without accounting for degradation and dynamic environmental factors. My perspective highlights the potential drifts and errors from such an approach. To address this, I introduce the INAF Fusion methodology, featuring a dynamically operating module that recalibrates feature importance in real-time based on motion feedback, aiming to enhance system accuracy adaptively.

Two alternative fusion methods, direct fusion and Attention-Based fusion, are presented. These serve as baseline models, providing benchmarks for evaluating the efficacy of the proposed INAF Fusion model in reducing drift and improving overall accuracy. The visual representation of these fusion methods is illustrated in Figure 6.6, offering a graphical comparison of the presented fusion methods.

6.5.1 Direct Fusion

An effective strategy for implementing sensor fusion involves the application of MLPs to merge features from distinct sensor modality channels. In this ideal scenario, the system learns to discern pertinent features for predictions in an end-to-end manner. Therefore, direct fusion can be expressed as follows:

$$g_{\text{direct}}(\theta_{\text{ICP-QE}}, \theta_{\text{DL-PE}}) = [\theta_{\text{ICP-QE}}; \theta_{\text{DL-PE}}] \quad (6.10)$$

In this formulation, the notation $[\theta_{\text{ICP-QE}}; \theta_{\text{DL-PE}}]$ represents an operation function that concatenates features $\theta_{\text{ICP-QE}}$ and $\theta_{\text{DL-PE}}$, where these features are extracted from Modality One and Modality Two, respectively. This straightforward approach aims to leverage the collective information from both modalities to enhance the overall prediction capabilities of the system [132].

6.5.2 Attention Based Fusion

The self multi-head attention mechanism applied to fusing the outputs of the branches $\theta_{\text{ICP-QE}}$ and $\theta_{\text{DL-PE}}$ uses distinct weight matrices to transform these branch outputs into queries, keys, and values:

$$\text{Queries} = W_q \times \theta, \quad \text{Keys} = W_k \times \theta, \quad \text{Values} = W_v \times \theta \quad (6.11)$$

where θ represents the concatenated outputs of the branches, and W_q , W_k , and W_v are the weight matrices for queries, keys, and values, respectively.

The attention scores are then computed by:

$$\text{Attention Scores} = \text{softmax} \left(\frac{(\text{Queries} \cdot \text{Keys}^T)}{\sqrt{d_k}} \right) \quad (6.12)$$

where d_k is the dimension of the key vectors. These scores prioritize parts of the input based on their computed relevance to the task.

The output from the attention mechanism is a weighted sum of the values:

$$\text{Output} = \text{Attention Scores} \cdot \text{Values} \quad (6.13)$$

Multiple heads are used to independently process the inputs, allowing the model to explore different aspects and relationships within the data. The final output of the multi-head attention mechanism is:

$$\text{Multi-Head Output} = \text{Concatenate}(\text{Head}_1, \text{Head}_2, \dots, \text{Head}_h) \times W^O \quad (6.14)$$

where W^O is a linear transformation applied to the concatenated outputs of all heads, effectively integrating the distinct features from $\theta_{\text{ICP-QE}}$ and $\theta_{\text{DL-PE}}$ into a unified representation.

6.5.3 Inferred Attention Fusion(Propose Method)

Dynamically operating module recalibrating feature importance based on real-time motion feedback. Following network training and parameter stabilization, the network output is used to align rigid body transformations at each time step precisely. The transformations are applied to the source scan to generate subsequent time steps. The comparative analysis contrasts the transformed source scan with the actual target scan, facilitated by three loss functions: point-to-point, point-to-plane, and plane-to-plane loss.

The derivation of partial derivatives of error concerning the fusion feature space unfolds in two stages. First, the calculation involves the interaction between the error and the transformation parameters during the inference phase. Second, this partial derivative is obtained numerically in the network’s backpropagation phase. The multiplication of these two partial derivatives ascertains the sensitivity of the error to the fusion feature space parameters, empowering the network to make informed decisions regarding feature prioritization from both branches. This adaptability ensures robustness within the network architecture.

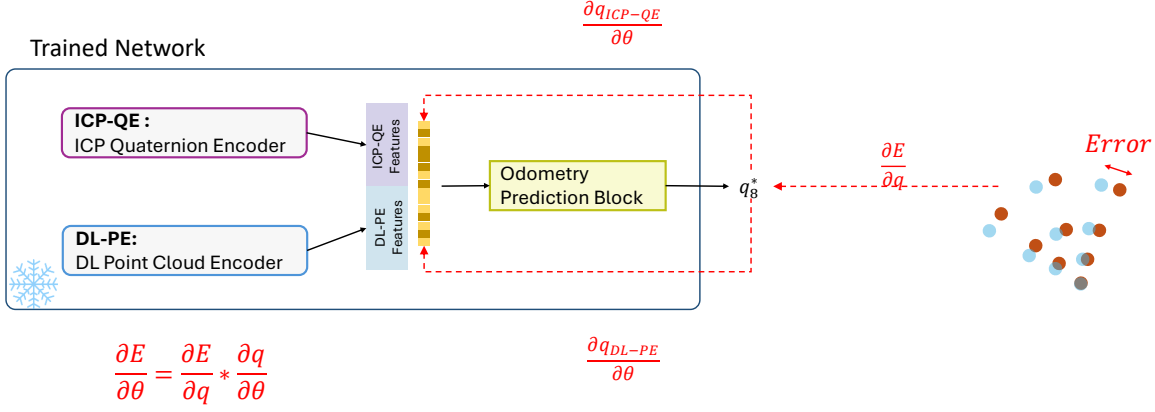


Figure 6.7 Illustration of the chain rule in calculating error gradients for the INAF fusion method, tracing back from the output error to the features derived from the branches $\theta_{\text{ICP-QE}}$ and $\theta_{\text{DL-PE}}$.

$$\mathcal{L}_{\text{point to point}} = \frac{1}{n_k} \sum_{p=1}^N \left| \left(\widehat{\text{transformed source}_p} - \text{target}_p \right) \right|_2^2 \quad (6.15)$$

$$\mathcal{L}_{\text{point to plane}} = \frac{1}{n_k} \sum_{p=1}^N \left| \left(\widehat{\text{transformed source}_p} - \text{target}_p \right) \cdot \mathbf{n}_p \right|_2^2 \quad (6.16)$$

$$\mathcal{L}_{\text{plane to plane}} = \frac{1}{n_k} \sum_{p=1}^N \left| (\mathbf{n} \cdot \hat{\mathbf{p}} - \mathbf{n}_p) \right|_2^2 \quad (6.17)$$

The total error is a summation of the individual errors calculated from the above equations:

$$E = \mathcal{L}_{\text{point to point}} + \mathcal{L}_{\text{point to plane}} + \mathcal{L}_{\text{plane to plane}} \quad (6.18)$$

I use the chain rule to calculate the gradient of the total error with respect to the branch features θ . Let q represent the dual quaternion parameters, and the gradients are calculated as follows:

$$\frac{\partial E}{\partial \theta} = \frac{\partial E}{\partial q} \cdot \frac{\partial q}{\partial \theta} \quad (6.19)$$

$$\theta = [\theta_{\text{ICP-QE}}; \theta_{\text{DL-PE}}] \quad (6.20)$$

Here, $\frac{\partial E}{\partial q}$ represents the gradient of the error with respect to the dual quaternion parameters, and $\frac{\partial q}{\partial P}$ captures how changes in the dual quaternion parameters affect the branch features.

Let $\theta_{\text{ICP-QE}}$ and $\theta_{\text{DL-PE}}$ be the feature vectors from different branches, and let e represent the rate of change of error with respect to a parameter, defined as $e = \frac{\partial \text{Error}}{\partial \theta}$. First, normalize e to prepare it for input into the neural network:

$$e' = \frac{e - \mu_e}{\sigma_e} \quad (6.21)$$

where μ_e and σ_e are the mean and standard deviation of the vector e , respectively.

A neural network f_θ is used to map the normalized error rate vector e' to a set of weights, applying a sigmoid activation function to ensure the weights are within the range $[0, 1]$:

$$w = \sigma(f_\theta(e')) \quad (6.22)$$

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \quad (6.23)$$

The learned weights are then applied to the features from each branch:

$$\theta'_{\text{ICP-QE}} = w \odot \theta_{\text{ICP-QE}} \quad (6.24)$$

$$\theta'_{\text{DL-PE}} = w \odot \theta_{\text{DL-PE}} \quad (6.25)$$

Here, $\odot$ denotes element-wise multiplication, producing the weighted feature vectors $\theta'_{\text{ICP-QE}}$ and $\theta'_{\text{DL-PE}}$.

This approach integrates an error-aware weighting mechanism into the model, allowing dynamic adjustment of feature contributions based on associated estimation error rates. This enhances reliability and accuracy when changing data conditions.

6.6 Model Training

6.6.1 Data Preparation

For this project, I utilized the KITTI Lidar Odometry Dataset [137], which is a benchmark dataset commonly used for evaluating the performance of visual/LiDAR odometry, SLAM, and 3D object detection algorithms. The dataset includes high-resolution LIDAR point clouds, colour and grayscale stereo images, and precise 6-DoF ground truth poses, making it ideal for developing and testing odometry models. Specifically, sequences 00 to 10 were employed for training and validation purposes. These sequence sets capture a wide variety of urban, rural, and highway scenarios to provide diverse testing for good model performance. That is challenging and comprehensive. The KITTI dataset serves as a very important resource for improving the state-of-the-art field of autonomous navigation and perception systems. Sequences 00 to 10 of the KITTI Lidar Odometry Dataset provide rich and diverse data points for training and validation [137]. Here are some key statistics about these sequences:

Table 6.1 Statistics of KITTI Lidar Odometry Dataset Sequences 00 to 10.

Sequence	Number of Frames	Distance (km)	Environment
00	4,541	3.7	Urban
01	1,110	2.2	Highway
02	4,466	5.1	Urban
03	801	0.8	Urban
04	271	0.4	Residential
05	2,760	2.2	Urban
06	1,100	1.2	Urban
07	1,101	0.7	Residential
08	4,068	3.2	Urban
09	1,599	1.7	Urban
10	1,201	1.2	Urban

These sequences collectively cover a distance of about 22.3 km and provide a comprehensive set of environments, including urban streets, highways, and residential areas. The diversity and size of this data make it ideal for training and validating robust odometry and perception

models for autonomous driving applications.

6.6.2 Data Transformation

Spherical projection and normal vector estimation were performed to prepare the lidar data for data transformation. Spherical projection involves mapping the 3D lidar points onto a spherical coordinate system, which enables the representation of the point cloud in a more structured and efficient manner. This transformation allows for easier visualization and analysis of the lidar data and simplifies subsequent processing steps, such as feature extraction and object detection. Normal vector estimation was also employed to calculate the surface orientation at each lidar point. Estimating the normal vectors provides valuable information about the local geometry of the environment, enhancing the richness of the lidar data representation.

Spherical Projection

To acquire the image representation, I employ a geometric mapping function [138] denoted as $\psi : \mathbb{R}^{n \times 3} \rightarrow \mathbb{R}^{4 \times H \times W}$, where H and W signify the height and width of the resulting image, respectively. The computation of image coordinates (u, v) involves discretizing azimuth and polar angles in spherical coordinates. Notably, this process ensures that only the nearest point is retained at each pixel location. The selection of H aligns with the number of vertical scan lines of the sensor, while W is typically set to be smaller than the number of points per ring. This strategic choice aims to achieve a dense image.

Equation 6.26 transforms spherical coordinates into 2D image coordinates. In this equation, u and v represent the resulting horizontal and vertical image coordinates, respectively. The transformation is conducted systematically from spherical to Cartesian coordinates, incorporating essential considerations for the field of view and scaling.

For u , the azimuthal angle θ is determined by the arctangent of the ratio of p^y to p^x . This angle is then adjusted by subtracting it from half of the horizontal field of view ($f_h/2$). The resulting value is further scaled by the parameter δ_h .

Similarly, for v , the polar angle ϕ is calculated using the arctangent of the ratio of p^z to d .

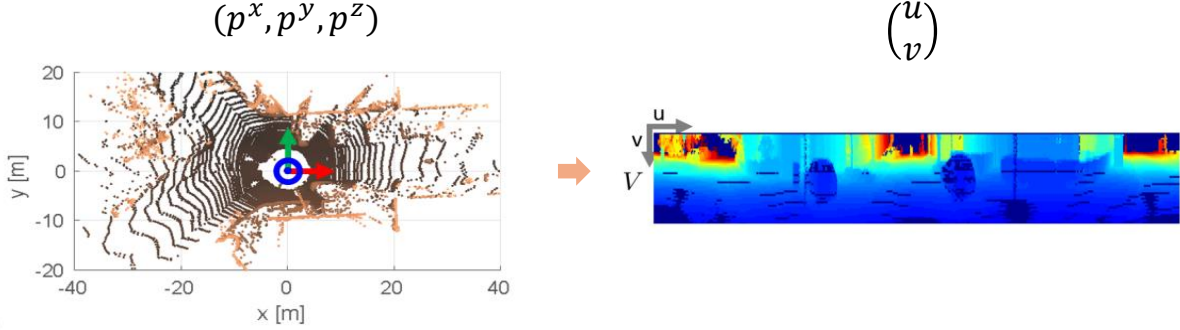


Figure 6.8 The left side displays unprocessed point clouds with the central local axis (RGB representing the XYZ axis). On the right side, the input vertices are presented and colour-coded based on their distance from the origin to enhance visualization[138].

The angle is then adjusted by subtracting it from a term related to the vertical field of view (f_{vu}). The outcome is scaled by the parameter δ_v .

These transformations are integral in computer graphics and computer vision, particularly in scenarios where the conversion from 3D spherical coordinates to 2D image coordinates is required. The formulation accommodates the nuances of field-of-view adjustments and scaling, contributing to the accurate mapping of spherical images onto a 2D surface.

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} (\frac{f_h}{2} - \arctan(p^y, p^x))/\delta_h \\ (f_{vu} - \arctan(p^z, d))/\delta_v \end{pmatrix} \quad (6.26)$$

In addition to the 3D point coordinates, the image incorporates intensity, range, and normal vectors, resulting in a comprehensive representation for each valid pixel of the image expressed as $(u, v, p^z, \rho, \iota, N_\xi, N_v, N_\zeta)$. This augmented information is collectively denoted as $I = \psi(S)$, where S represents the original dataset. This multi-dimensional approach enhances the richness of the image representation, containing not only spatial coordinates but also additional attributes like Normal vectors for a better understanding of the scene.

Normal Vector Estimation

To determine the normal vector of the point cloud (N_ξ, N_v, N_ζ) , I employed the Open3D library for estimating normal vectors within the point cloud. The methodology involves a

covariance analysis, where the principal axis of adjacent points is computed. This process is facilitated by utilizing a KD-tree structure with hybrid search parameters, specifying a search radius of 0.1 and considering a maximum of 30 nearest neighbours. Notably, the covariance analysis inherently produces two opposing directions as potential normal vectors, a phenomenon known as the normal orientation problem. To address this, Open3D attempts to align the normals with the original orientation when available; otherwise, it reverts to a random guess. This robust methodology enhances the understanding of local surface characteristics within the point cloud, contributing to the efficacy of subsequent analyses and applications in 3D point cloud processing.

6.6.3 Model Architecture

The network architecture employed in this project is a modular design customized to process geometric and point cloud features, perform feature fusion, engage in temporal modelling, and solve specific tasks. All the parameters in the function headers, such as layer sizes, activation functions, dropout rates, and the number of units in LSTM layers, are carefully defined through hyperparameter tuning explained in Section 6.6.5 to optimize the model's performance.

Dual Quaternion Feature Encoder

The dual quaternion feature encoder consists of several layers dedicated to extracting and processing geometric features from the ICP dual quaternion input data. The architecture incorporates several layers to process geometric features effectively, shown in the Algorithm 2.

Point Cloud Feature Encoder

The point cloud feature encoder is designed to extract features from the transformed lidar data $(u, v, p^z, \rho, \iota, N_\xi, N_v, N_\zeta)$ using convolutional and fully connected layers as detailed in Algorithm 3.

Feature Fusion

The fusion module integrates features from the ICP-QE and DL-PE branches. It is explained in Algorithm 4.

Algorithm 2 Dual Quaternion Feature Encoder Architecture

- 1: **Input:** Dual quaternion parameters $q = (q_0, q_1, q_2, q_3) + \varepsilon(q_{e0}, q_{e1}, q_{e2}, q_{e3})$
 - 2: **Output:** Encoded feature vector
 - 3: **Begin**
 - 4: Apply time-distributed batch normalization:
 - 5: **Normalization Layer:** Batch Normalization
 - 6: Apply three sequential fully connected layers with activation:
 - 7: **First Encoding Layer:** Dense Layer with activation, size parameterized
 - 8: **Second Encoding Layer:** Dense Layer with activation, size parameterized
 - 9: **Third Encoding Layer:** Dense Layer with activation, size parameterized
 - 10: Apply dropout to enhance regularization:
 - 11: **First Dropout Layer:** Dropout, rate parameterized
 - 12: Additional dense layer for feature decoding with dropout:
 - 13: **Decoding Layer:** Dense Layer with activation, size parameterized
 - 14: **Second Dropout Layer:** Dropout, rate parameterized
 - 15: Apply two LSTM layers to capture temporal dynamics:
 - 16: **First LSTM Layer:** LSTM with specified activation functions, size parameterized
 - 17: **Second LSTM Layer:** LSTM with specified activation functions, size parameterized
 - 18: **End**
-

Algorithm 3 Point Cloud Feature Encoder Architecture Using ResNet

- 1: **Input:** Transformed lidar data $(u, v, p^z, \rho, \iota, N_\xi, N_v, N_\zeta)$
 - 2: **Output:** Processed features capturing spatial and temporal dependencies
 - 3: **Begin**
 - 4: **Initialize ResNet:** Configure network with optimizable parameters
 - 5: Parameters include filter counts and activation details
 - 6: **Apply Batch Normalization**
 - 7: **Apply Convolutional Layer:**
 - 8: Time-distributed Conv2D with specified kernel size, strides, and padding
 - 9: **Apply Max Pooling Layer:**
 - 10: Specified pool size and strides with appropriate padding
 - 11: **Apply Activation and ResNet Blocks:**
 - 12: **for** each required ResNet block set **do**
 - 13: Apply consecutive ResNet blocks configured for feature transformation
 - 14: **end for**
 - 15: **Apply Average Pooling**
 - 16: **Flatten Output:** Prepare output for dense layer processing
 - 17: **Apply Fully Connected Layer:**
 - 18: Dense layer with activation function to finalize feature representation
 - 19: **End**
-

Algorithm 4 Feature Fusion Architecture

- 1: **Input:** Outputs from geometric ($\theta_{\text{ICP-QE}}$) and lidar ($\theta_{\text{DL-PE}}$) branches
 - 2: **Output:** Integrated feature representation
 - 3: **Begin**
 - 4: **Batch Normalization:**
 - 5: Apply separate batch normalization to each branch output
 - 6: **BN_geo_branch:** Normalize $\theta_{\text{ICP-QE}}$
 - 7: **BN_Li_branch:** Normalize $\theta_{\text{DL-PE}}$
 - 8: **Feature Fusion Options:**
 - 9: **Option 1: Direct Fusion**
 - 10: Concatenate normalized outputs along the specified axis
 - 11: Apply a dense layer and dropout for final fusion
 - 12: **Option 2: Attention-Based Fusion**
 - 13: Apply self-multi-head attention to concatenated outputs:
 - 14: **Queries, Keys, Values** from concatenated features
 - 15: Compute attention scores and obtain weighted features
 - 16: **Option 3: INAF Fusion**
 - 17: Apply INAF mechanism:
 - 18: Use error feedback to adjust attention weights dynamically
 - 19: Fuse features based on adjusted attention scores
 - 20: **End**
-

Task-Solving Layers

The final module uses a sequence of neural network layers, including batch normalization, densely connected layers, and LSTM units, to effectively capture spatial features and temporal dynamics. The structure ensures that the predictive model learns from the data's static properties and adapts to the sequential patterns as explained in Algorithm 5.

Algorithm 5 Temporal Modeling and Prediction Architecture

- 1: **Input:** Integrated feature vector from the fusion process
 - 2: **Output:** Predicted quaternion parameters $\bar{q} = (\bar{q}_0, \bar{q}_1, \bar{q}_2, \bar{q}_3, \bar{q}_{e0}, \bar{q}_{e1}, \bar{q}_{e2}, \bar{q}_{e3})$
 - 3: **Begin**
 - 4: **Enhanced Feature Integration:**
 - 5: Apply a densely connected layer to enhance feature integration
 - 6: **Enhancement Layer:** Dense layer with activation
 - 7: Apply dropout to enhance model generalization
 - 8: **Generalization Layer:** Dropout layer
 - 9: **Sequential Temporal Processing:**
 - 10: Incorporate LSTM layers to model temporal dynamics effectively
 - 11: **Temporal Layer 1:** LSTM configured with specific activation
 - 12: **Temporal Layer 2:** LSTM configured with specific activation
 - 13: **Predictive Dense Layers:**
 - 14: Sequentially apply densely connected layers to refine the prediction
 - 15: **Predictive Layer 1:** Dense layer with activation
 - 16: **Predictive Layer 2:** Dense layer with activation
 - 17: **Output Prediction Layer:** Dense layer without activation for final prediction
 - 18: **Quaternion Normalization:**
 - 19: Normalize the final output to ensure valid quaternion format
 - 20: **Normalization Step:** Custom normalization for quaternions
 - 21: **End**
-

6.6.4 Training Procedure

The training procedure for my network involves several critical steps, including data preparation, optimization, and model validation. Each component is carefully designed and fine-tuned.

Data Preparation and Splitting

The dataset used for training the model is split into training and validation subsets. This split

is crucial for evaluating the model's performance on unseen data and preventing overfitting. The data is combined into a single TensorFlow dataset and then shuffled to ensure a random distribution of samples. The dataset is split based on a specified validation ratio, with 20% of the data allocated for validation and the remaining 80% for training.

Optimization Strategy

The Adam optimizer optimizes the network parameters, which is well-suited for handling sparse gradients and large datasets. An exponential decay schedule is employed to dynamically adjust the learning rate during training. This schedule starts with an initial learning rate of 0.001, which decays by a factor of 0.9 every 10,000 steps. A decaying learning rate helps fine-tune the learning process, allowing the model to converge more effectively and avoid overshooting minima in the loss function landscape.

Custom Loss Function

The model employs a combined loss function designed to optimize performance by integrating MAE and MSE, which is crucial for balancing the trade-offs between precision and sensitivity to outliers. The function is defined as follows:

$$L_{\text{combined_loss}} = (1 - \alpha) \times \text{MAE}(y_{\text{true}}, y_{\text{pred}}) + \alpha \times \text{MSE}(y_{\text{true}}, y_{\text{pred}}) \quad (6.27)$$

Where α is a weighting factor set to 0.8, which adjusts the emphasis between MAE and MSE.

Early Stopping and Model Checkpointing

Early stopping is implemented to enhance the training process by halting it when no further improvement in validation performance is detected over a span of 15 epochs. This "patience" threshold is strategically set to ensure efficient use of computational resources and to prevent the model from overfitting by stopping training when it no longer makes significant progress. Additionally, model checkpointing is utilized to save the best-performing model weights during the training phase. This approach ensures that the most effective version of the model is retained for further evaluation and deployment, thereby increasing the model's practical effectiveness.

6.6.5 Hyperparameter Tuning

Before the training process starts, a machine-learning model is configured using parameters called hyperparameters. These parameters can include the batch size, learning rate, the number of layers and units in each layer and their activation functions. Due to their substantial influence on the model’s performance and capacity to generalize to fresh data, these hyperparameters must be tuned. While improperly selected hyperparameters can result in poor performance, delayed training, or inefficient learning, properly adjusted hyperparameters can increase accuracy, expedite the training process, and reduce overfitting. Thus, creating effective machine learning models requires rigorous hyperparameter optimization.

There are various kinds of hyperparameter search algorithms, and they all have different approaches to effectively spanning the search space. Random Search provides ease and variety in exploration by selecting hyperparameter values randomly from the search space. Hyperband optimizes resource utilization by dynamically allocating resources to promising configurations and eliminating less promising ones early. Using probabilistic models to forecast performance, Bayesian optimization iteratively chooses hyperparameter values to optimize a given acquisition function while balancing between exploration and exploitation.

Because Hyperband [139] efficiently manages computing resources and expedites the search for optimal configurations, I employ it for hyperparameter optimization in my complex model with a huge dataset. Because of the intricacy of my model and the large dataset, standard methods are not feasible because they would take too much time and money to investigate every potential combination of hyperparameters. This problem is solved by hyperband, which dynamically allocates resources to the most promising hyperparameter combinations while quickly eliminating the less successful ones. This method greatly cuts down on the total time and expense of the optimization process by ensuring that my computational efforts are focused on the configurations that are most likely to produce the best results.

Building on the Successive Halving (SHA) technique, Hyperband is an effective hyperparameter optimization strategy that dynamically distributes resources (e.g., epochs) across various configurations. It works by executing many brackets, each with a distinct starting

resource allocation, progressively eliminating less promising configurations while increasing the resources allocated to the more effective ones. For each bracket s , the initial number of configurations n and resources per configuration r are given by:

$$n = \left\lceil \frac{s_{\max} + 1}{s + 1} \cdot \eta^s \right\rceil \quad (6.28)$$

$$r = \frac{R}{\eta^s} \quad (6.29)$$

where R is the maximum resource and η is the reduction factor. This equation ensures that the number of configurations n increases exponentially as the initial resource allocation per configuration decreases. This method is especially well-suited for complex models and large datasets because it balances exploration and exploitation, making it possible to navigate large search spaces efficiently. Hyperband dramatically lowers the computational cost using a resource allocation theory compared to conventional approaches [139].

In my hyperparameter tuning setup, I utilize the Keras Tuner library to systematically explore the hyperparameter space and optimize the performance of my neural network model. The Hyperband algorithm, chosen for its efficiency in resource utilization, conducts a series of trials with varying hyperparameter configurations. The objective is to minimize the mean absolute error (`val mae`) on the validation dataset, ensuring that the model generalizes well to unseen data. With a maximum of 20 epochs per trial and a reduction factor of 3, the tuner intelligently allocates resources to promising configurations while discarding less promising ones early. Early stopping is implemented based on the validation loss, with a patience of 15 epochs to prevent overfitting. This comprehensive approach to hyperparameter tuning allows me to optimize the model effectively, leading to improved performance and generalization capabilities.

6.7 Backend Loop Closure and Optimization

The system’s backend integrates two key components: scan loop closure and graph optimization, which are important in improving performance and accuracy. Loop closure identifies previously visited locations by matching features between different scans, enabling the system to correct accumulated errors and maintain global consistency. Graph optimization optimizes the trajectory based on the loop closure constraints, refining the estimated poses to minimize errors and improve accuracy. This section explains both components’ contributions to the system’s functionality and performance.

6.7.1 Loop Closure

Scan context is a novel global descriptor derived from 3D LiDAR scans, offering distinct advantages over traditional histogram-based methods. Unlike previous approaches, scan context directly captures the 3D structure of visible space from a sensor without relying on histograms or prior training. Furthermore, it introduces a similarity score to measure the distance between two scan contexts and employs a two-phase search algorithm for efficient loop detection. This method ensures loop detection remains invariant to changes in LiDAR viewpoint, enabling the identification of loops in scenarios such as reverse revisit and corners. Extensive evaluations conducted on benchmark datasets demonstrate the superior performance of scan context in loop detection tasks, underscoring its efficacy in enhancing mapping optimization processes [130].

The process flow of the scan context loopback detection algorithm is illustrated in Figure 3. Initially, a rectangular image representing the scan context is constructed using the point cloud data acquired during a single scan. Subsequently, a KD tree is created based on the ring key vector (a rotation-invariant point cloud descriptor), facilitating the efficient search for nearest neighbours. Multiple frames exhibiting similarities to the current frame, along with their corresponding ring key translations, are identified through this search. A similarity score is computed for each potential loopback frame, and those with high scores are filtered out.

Next, the algorithm calculates each sector’s minimum offset and similarity score, ultimately selecting the frame with the highest similarity score as the loopback frame. The algorithm determines the pose relationship between the current frame and the loopback frame, thus achieving loop detection [130, 140].

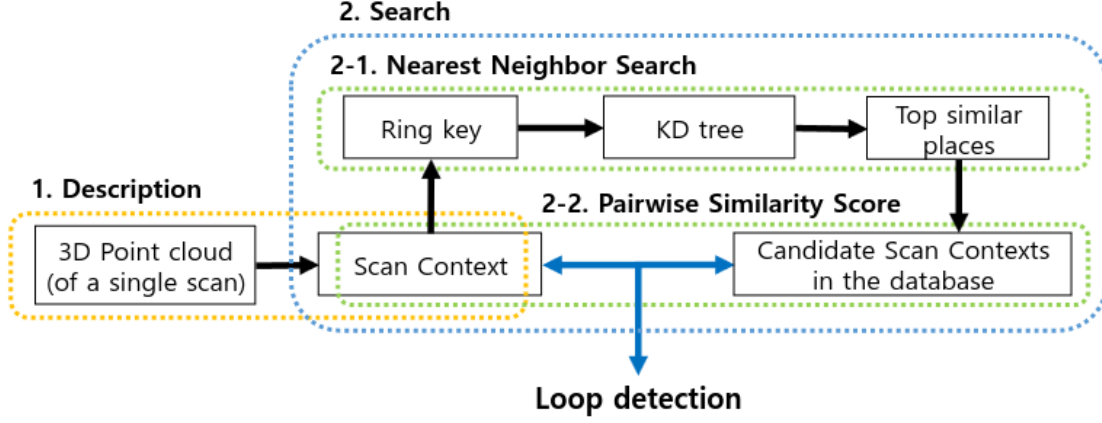


Figure 6.9 Scan Context overview: A point cloud from a single 3D scan is first encoded into scan context. Next, an N_r -dimensional vector is derived from the scan context to retrieve the nearest candidates and construct the KD tree. Finally, the retrieved candidates are compared to the query scan context, with the loop being identified as the candidate meeting the acceptance threshold and closest to the query [130].

6.7.2 Graph Optimization

For graph optimization, I utilized miniSAM, a lightweight framework inspired by GTSAM [117]; both employ factor graphs to model non-linear least square problems. The optimization process is defined by Equation 6.30.

$$x^* = \underset{x}{\operatorname{argmin}} \sum_i \rho_i \|f_i(x)\|_{\Sigma_{R_i}}^2 \quad (6.30)$$

Where $x \in M$ represents a point on an n -dimensional manifold, x^* denotes the solution, $f_i \in \mathbb{R}^m$ is a vector-valued error function, ρ_i signifies a robust kernel function, and $\Sigma_i \in \mathbb{R}^{m \times m}$ is a covariance matrix. The Mahalanobis distance is calculated as $\|v\|_2^\Sigma = v^\top \Sigma^{-1} v$, where

$v \in \mathbb{R}^m$ and Σ^{-1} denotes the information matrix. By performing Cholesky factorization on the information matrix, $\Sigma^{-1} = R^\top R$ where R is upper triangular, the distance simplifies to $\|v\|_2^\Sigma = \|Rv\|_2$ [131].

Solving the pose graph problem using miniSAM involves several key steps. First, a factor graph is constructed to represent the problem. In miniSAM, this is achieved using the **FactorGraph** data structure, where each variable is uniquely indexed by a combination of a character and an unsigned integer (e.g., **x1**). Their key list defines Factors in miniSAM, indicating connected variables and their loss function, including covariance Σ_i and an optional robust kernel ρ_i . The pose graph example typically employs two types of factors: unary **PriorFactor** and binary **BetweenFactor**. Once the factor graph is established, initial variable values are provided as the linearization point. These variable values are stored in the **Variables** structure, with each variable indexed by its key. Finally, the problem is solved using a non-linear least square solver, such as Levenberg-Marquardt. This iterative process aims to optimize the variables to minimize the error between observed and predicted values, ultimately producing the optimal solution for the pose graph problem [131].

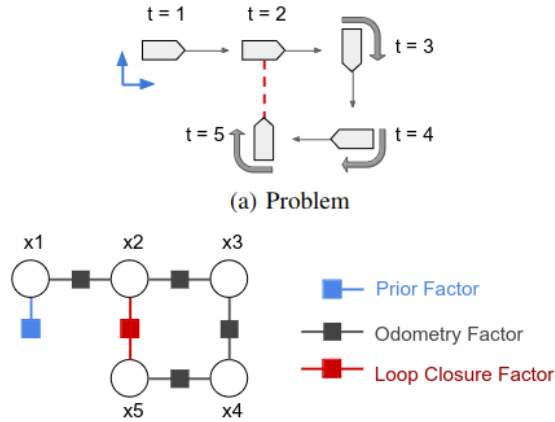


Figure 6.10 In a 2D pose graph problem example, pose variables are depicted as white circles, a prior is represented by a blue factor, black factors denote four odometry measurements, and a red factor indicates a single loop closure measurement [131].

6.8 Results

6.8.1 Performance Evaluation of the Proposed Method

The relative pose error measures trajectory precision within a time interval Δ , crucial for visual odometry evaluation. It's calculated at time step i as:

$$E_i = (Q_i^{-1}Q_{i+\Delta})^{-1}(P_i^{-1}P_{i+\Delta}) \quad (6.31)$$

Yielding $m = n - \Delta$ errors from n poses. For overall accuracy, compute:

$$\text{RMSE}(E_{1:n}, \Delta) = \left(\sum_{i=1}^m \frac{1}{m} \|\text{trans}(E_i)\|^2 \right)^{1/2} \quad (6.32)$$

With $\text{trans}(E_i)$ as translational components.

Average $\text{RMSE}(E_{1:n}, \Delta)$ over all intervals Δ :

$$\text{RMSE}(E_{1:n}) = \frac{1}{n} \sum_{\Delta=1}^n \text{RMSE}(E_{1:n}, \Delta) \quad (6.33)$$

6.8.2 Comparison with Classical SLAM

This section compares the performance of various odometry estimation methods, including a traditional approach without neural networks (ICP) and one neural network-based method: the end-to-end network using the ICP-QE branch (Figure 6.5.a).

While learning-based methods achieve similar translation error rates, traditional methods consistently exhibit lower orientation errors. This difference occurs because deep neural networks excel at extracting features and estimating translations directly from raw point clouds.

The ICP method, while not the most advanced traditional approach, serves as a fundamental technique that effectively illustrates the integration of AI with traditional methods.

Using ICP, the differences and highlight the potential enhancements provided by neural network-based methods can be exaggerated. Despite its simplicity, ICP’s reliance on basic geometric principles makes it an ideal candidate for demonstrating how AI can improve accuracy and efficiency in odometry estimation. Furthermore, its straightforward implementation and widespread recognition in the field provide a clear benchmark for comparing and contrasting the benefits of incorporating AI-driven techniques.

The ICP algorithm is the basic way of aligning 3D shapes initially proposed in [141] but later revised by [142] in 1994. This algorithm aims to transform one shape into a form that matches another. The first step involves selecting the nearest points from each set to obtain matching pairs. Then, it determines the optimal transformation for minimizing such distance between these pairs by solving a least squares optimization problem for computing the rotation matrix and translation vector that aligns the source point cloud with the target point cloud. This process continues until convergence using a threshold on change in total alignment error as a stopping criterion. ICP has found many applications in computer vision, robotics, or any other field that uses 3D shape registration because of its simplicity and efficiency, as explained in both influential work of [141] and [142] on algorithms for iterative repositioning of three-dimensional models.

The classic approach to odometry estimation, known as ICP, aligns successive LiDAR scans to estimate the vehicle’s motion. While this method has been a staple in the field due to its straightforward implementation and effectiveness in certain scenarios, it has notable limitations, particularly in environments with insufficient geometric features, sparse point clouds, or noisy data. The traditional ICP approach serves as my baseline for comparison, providing a reference point for evaluating the enhancements introduced by neural network-based methods.

The first Deep Learning (DL)-based method, using the ICP-QE branch (Figure 6.5.a), incorporates ICP outputs as input features. This approach leverages the strengths of ICP for initial alignment while using the neural network to refine the odometry estimates. Integrating the geometric branch significantly improves the performance of the traditional ICP method.

The geometric branch demonstrates enhanced accuracy and robustness by learning to correct the residual errors in ICP outputs. Outperforming the classic ICP approach in both rotational and translational accuracy.

Table 6.2 Positioning results comparing ICP, a geometric method, and ICP-QE, a deep learning network.

Odometry Encoding Block	Feature Weighting	Back-End	Training Set		Test Set	
			Trans. err. (%)	Rot. err. (deg/100m)	Trans. err. (%)	Rot. err. (deg/100m)
ICP [141]	–	–	–	–	101.23	31.32
	–	✓	–	–	11.83	15.65
ICP-QE (this study)	Direct	–	7.82	5.75	8.22	8.16
	Direct	✓	5.37	4.22	6.31	6.83

6.8.3 Comparison with AI SLAM

The second DL-based method, the DL-PE module, handles point cloud data by first applying it to Cylindrical Projection before feeding it into the network. This approach bypasses the need for explicit ICP alignment, relying entirely on the neural network’s capability to extract and utilize point cloud data features. The LiDAR branch performs better in estimating rotational motion than the traditional ICP method and the geometric branch. This improvement is attributed to the network’s ability to capture intricate spatial relationships within the point cloud data, which are crucial for precise rotational estimation. However, regarding translational error, the LiDAR branch does not outperform the geometric branch with ICP input. This suggests that while the neural network excels at interpreting point cloud data for rotational movements, the initial alignment provided by ICP still plays a critical role in accurately estimating translational motion. Comparative Analysis

6.8.4 Comparison with State-of-the-Art Methods

In this section, the performance of the proposed INAF Fusion method is analyzed against several state-of-the-art approaches, encompassing both deep learning-based and geometry-

Table 6.3 Positioning results comparing ICP, a geometric method, and ICP-QE, and DL-PE two deep learning networks.

Odometry Encoding Block	Feature Weighting	Back-End	Training Set		Test Set	
			Trans. err. (%)	Rot. err. (deg/100m)	Trans. err. (%)	Rot. err. (deg/100m)
ICP [141]	–	–	–	–	101.23	31.32
	–	✓	–	–	11.83	15.65
ICP-QE (this study)	Direct	–	7.82	5.75	8.22	8.16
	Direct	✓	5.37	4.22	6.31	6.83
DL-PE (this study)	Direct	–	8.36	4.18	10.42	4.25
	Direct	✓	5.21	4.07	5.85	3.86

based methods. The detailed results are presented in Table 6.4.

Methods	Training Sets 00-08		Test Sets 09 and 10	
	Trans. err. (%)	Rot. err. (deg/100m)	Trans. err. (%)	Rot. err. (deg/100m)
DLL-based Approaches				
INAF Fusion (this study)	2.93	1.18	5.72	2.01
INAF Fusion (this study) + Loop Closure	2.15	0.82	2.26	0.92
Velas et al. [143]	2.94	NA	4.11	NA
DeLORA [41]	3.00	1.38	6.25	2.58
Gemotery Based Approaches				
ICP [141]	NA	NA	101.23	31.32
CT-ICP (ours) [144]	NA	NA	0.48	NA
LOAM [91]	NA	NA	1.36	0.51

Table 6.4 Performance Comparison of Various Methods on KITTI Sets. The table presents translational and rotational errors for deep learning-based and geometry -based methods.

The INAF Fusion method’s strength lies in dynamically integrating multiple methodologies using a novel attention-based mechanism. This mechanism allows selective focus on the most relevant features, enhancing robustness and accuracy. Including loop closure further improves performance by revisiting and correcting previously traversed paths, thus ensuring spatial consistency and accuracy.

Velas et al. [143] employs a Convolutional Neural Network (CNN) for IMU-assisted odometry estimation using Velodyne LiDAR. This approach performs competitively due to CNN’s ability to learn complex features from IMU and LiDAR data. However, its reliance on these inputs can lead to variability in performance across different environments, particularly

when sensor conditions change or are less than ideal.

DeLORA [41], which combines depth information with visual odometry, demonstrates the effectiveness of integrating multiple data types. However, integrating depth information poses challenges, especially when depth cues vary significantly across different scenes. This can result in increased errors when the model encounters environments different from those seen during training.

Geometry-based methods like ICP and its variants offer a different set of strengths and weaknesses. The traditional ICP algorithm, although widely used for its simplicity and effectiveness in aligning point clouds, is highly susceptible to local minima and requires good initial alignment, which is often not guaranteed in real-world scenarios. CT-ICP [144], which incorporates temporal continuity, achieves impressive translational accuracy and could be a good candidate to replace ICP. However, ICP was chosen to exaggerate the difference between the two methods.

LOAM [91], another geometry-based method, excels in processing LiDAR data for odometry and mapping. Its robustness and efficiency in handling LiDAR data contribute to its strong performance. However, LOAM includes mapping capabilities, which cannot be directly integrated into the network, making it less suitable for comparison in this context.

6.8.5 Qualitative Analysis of INAF Fusion Technique

The next analysis I conducted focused on combining the two branches. As a baseline, I implemented two fusion methods: the first is a straightforward approach that directly concatenates the outputs, and the second uses an attention-based method incorporating multihead self-attention. These methods are then compared with the proposed novel method called INAF fusion.

Handling Large Angle Rotations in KITTI Dataset for Robust Odometry Estimation

It is important to consider the challenges posed by the dataset’s nature, particularly when dealing with rotations. The KITTI dataset, known for its high-frequency scans, presents

instances where the vehicle rotates with large angles. This characteristic complicates the task for traditional geometric methods, such as those relying on ICP, which often struggle with rapid changes in orientation. Moreover, the KITTI dataset’s high-frequency scans can lead to motion distortion, where the movement of the sensor during the acquisition of a scan causes points to appear distorted [145]. This issue is particularly pronounced when the vehicle’s motion involves rapid rotation, as illustrated in Figure 6.12. The scan examples show instances where the angle exceeds 1 degree.

The histogram analysis, as depicted in Figure 6.11, highlights the frequency of such rotations, emphasizing the difficulty in maintaining accuracy. This variability affects the robustness of geometric methods and introduces complexities in training neural networks due to a low number of such cases.

The network must handle these rapid transitions to ensure reliable odometry estimation. The proposed INAF fusion method addresses these challenges by leveraging advanced attention mechanisms to dynamically adjust to environmental conditions, thereby enhancing the overall performance and stability of the fusion process in the face of such rotational dynamics.

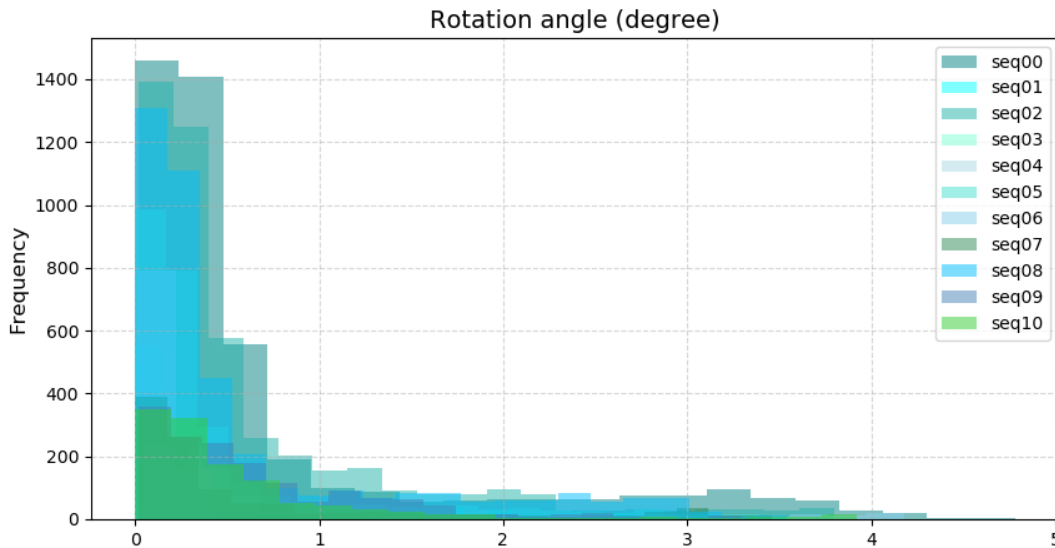
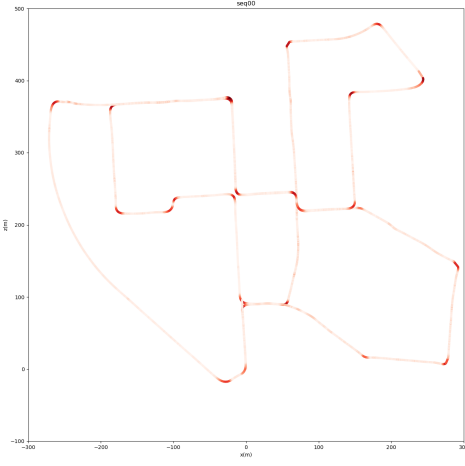
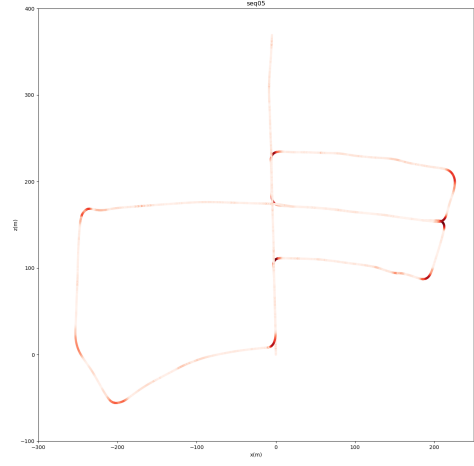


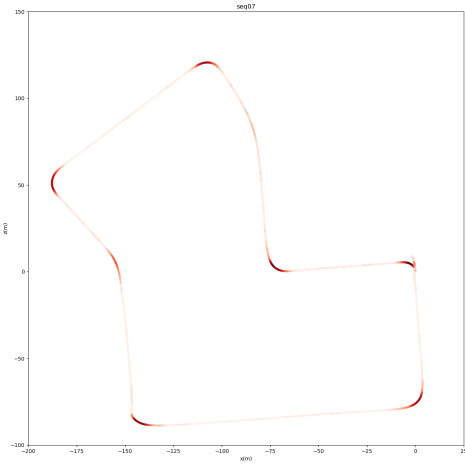
Figure 6.11 Histogram of all KITTI sequences showing the rotation angle in degrees.



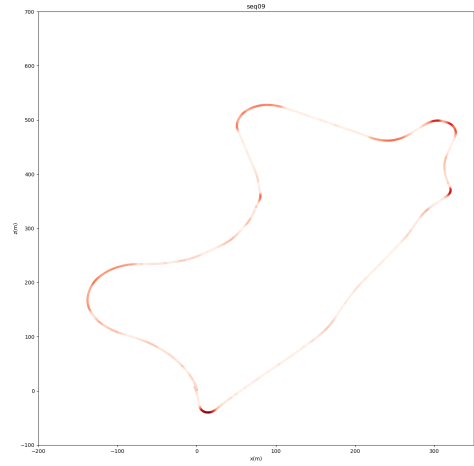
(a) Sequence 00



(b) Sequence 05



(c) Sequence 07



(d) Sequence 09

Figure 6.12 Examples of four sequences from the KITTI dataset (00, 05, 07, and 09). The trajectories illustrate situations where the rotation angle exceeds 1 degree, highlighted in red.

Comparing INAF Fusion with Direct and Attention-Based Approaches

The results of the INAF fusion method, compared with the direct and attention-based fusion methods, demonstrate its best performance in translation and rotation error metrics. This comparison was conducted with and without including loop closure optimization in the

training and test sets. The choice to compare these methods with and without loop closure optimization allows for a comprehensive understanding of their performance under different conditions, as loop closure is a common technique to correct drift in odometry and SLAM systems, particularly relevant in long-duration and complex environments.

Without loop closure optimization, the direct fusion method resulted in a translation error of 3.92% and a rotation error of 2.61 degrees per 100 meters in the training set, with slightly higher errors in the test set. When loop closure optimization was applied, the errors significantly reduced to 2.81% and 1.82 degrees in the training set and to 2.73% and 1.52 degrees in the test set. This reduction in errors when using loop closure optimization shows its importance in improving the accuracy of fusion methods by addressing accumulated errors over time.

The attention-based fusion method performed slightly better than the direct fusion, achieving a translation error of 3.42% and a rotation error of 2.13 degrees in the training set without loop closure. These errors decreased to 2.73% and 1.41 degrees in the training set and 2.51% and 1.16 degrees in the test set with loop closure. The improvement over the direct fusion method suggests that the attention mechanism effectively prioritizes relevant features, enhancing the model’s ability to handle diverse and dynamic environmental conditions.

The INAF fusion method outperformed both the direct and attention-based methods. Without loop closure, it achieved a translation error of 2.93% and a rotation error of 1.18 degrees in the training set, which was reduced to 2.15% and 0.82 degrees with loop closure. In the test set, the errors were 5.72% and 2.01 degrees without loop closure and further reduced to 2.26% and 0.92 degrees with loop closure. The superior performance of the INAF method can be attributed to its novel fusion approach, which effectively combines the strengths of both geometric and learning-based methods while incorporating advanced techniques to adjust to the environment dynamically.

These results highlight the efficacy of the INAF fusion method, particularly in reducing rotation errors, which is crucial for accurate odometry estimation in environments with significant rotational dynamics. The advanced attention mechanisms in the INAF method

enable it to dynamically adapt to varying environmental conditions, thereby enhancing the overall stability and accuracy of the fusion process. This adaptability is essential in real-world applications, where the environment can vary widely and unpredictably, requiring robust and flexible fusion methods to maintain high accuracy.

Table 6.5 Positioning results comparing ICP-QE + DL-PE, with different Fusion methods.

Odometry Encoding Block	Fusion	Back-End	Training Set		Test Set	
			Trans. err. (%)	Rot. err. (deg/100m)	Trans. err. (%)	Rot. err. (deg/100m)
ICP-QE + DL-PE	Direct	–	3.92	2.61	6.16	3.62
	Direct	✓	2.81	1.82	2.73	1.52
	Attention	–	3.42	2.13	6.04	3.54
	Attention	✓	2.73	1.41	2.51	1.16
	INAF	–	2.93	1.18	5.72	2.01
	INAF	✓	2.15	0.82	2.26	0.92

Adaptability of Sensor Fusion in Diverse Driving Scenarios

In estimating straight-driving translation, my investigation reveals a fluctuating allocation of importance among different modalities. Notably, the ICP branch sometimes assumes a higher weighting, highlighting its key role in capturing specific aspects of the driving dynamics. On the other hand, there are instances where the LiDAR branch exhibits greater influence, indicating its significant contribution to accurate estimations of straight-line motion. This dynamic distribution highlights the sensor fusion system’s adaptive nature, which adjusts each modality’s importance based on the overall driving conditions.

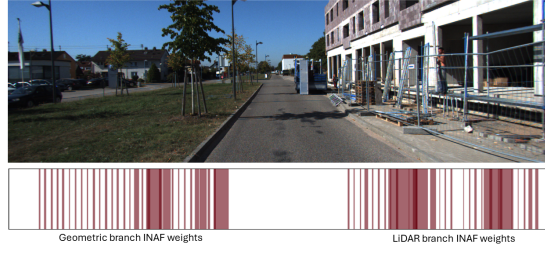
Contrarily, the LiDAR-based branch features exhibit greater influence in scenarios characterized by rapid rotation and the presence of moving objects. This implies that the features extracted from the AI-based branch are particularly adept at capturing the details of dynamic environments, where quick changes in orientation and moving objects pose significant challenges to accurate estimation. Thus, my analysis highlights the adaptive nature of the INAF fusion system, which allocates weights to different modalities based on the specific dynamics of the driving scenario.

Furthermore, I conducted rigorous experiments over the KITTI dataset to validate these findings across various environmental conditions and driving scenarios. My analysis of the sensor fusion masks consistently indicates the frequency of these trends in multiple trials.

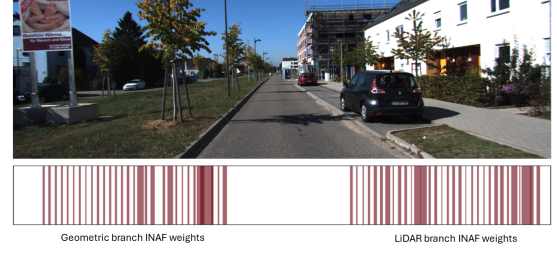
Thus, the INAF fusion approach enhances my understanding of the individual strengths inherent in each modality and provides practical evidence to support these observations. These insights are instrumental in guiding the development of future multimodal systems, ensuring their efficacy and adaptability to diverse real-world scenarios.

By employing visualization techniques to analyze the sensor fusion masks, as illustrated in Figure 6.13, a detailed examination of the relationship between the selected features of each branch and the dynamics of the environment or measurements was provided. The investigation reveals a robust correlation between these features and the overall environmental conditions.

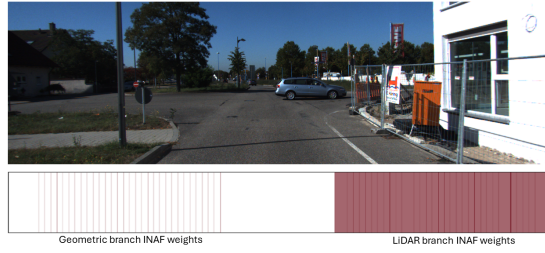
The sensor fusion masks are critical for understanding how different sensors contribute to



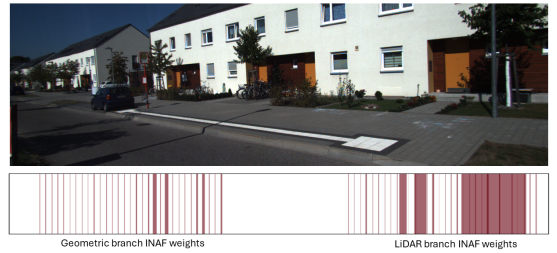
(a) Driving straight



(b) Driving straight



(c) Moving object



(d) Turning

Figure 6.13 Visualization of the learned weights under different conditions for INAF fusion. In scenario a, the vehicle drives straight, while in scenario b, a moving object (a car) is present. Scenario d depicts a turning maneuver. The weights of selected features in the geometric branch and LiDAR branch illustrate the increasing importance of different branches in various scenarios, highlighting the adaptive nature of the fusion process.

the final odometry estimation. These masks highlight which features are prioritized under various driving scenarios, such as straight driving, turning, or the presence of moving objects. For instance, in straight driving scenarios, the geometric features from the LiDAR branch may dominate due to their ability to capture static structural information. On the other hand, during turning maneuvers or when encountering moving objects, the importance may shift towards features that can better handle dynamic changes, thus demonstrating the adaptive nature of the fusion process.

Further, the visualization of these masks provides insight into the adaptability of the proposed INAF fusion method. By dynamically adjusting the importance of features from different branches based on the environmental context, the INAF method enhances the robustness and accuracy of odometry estimation. This adaptability is crucial for handling the variability and complexity of real-world driving conditions, as evidenced by improved

performance in scenarios with rapid rotations or significant motion distortions.

Such an approach underscores the value of integrating advanced attention mechanisms within the fusion process. These mechanisms enable the model to focus on the most relevant features at any given time, thereby improving its ability to maintain accurate and stable odometry estimates. This detailed analysis and understanding of the sensor fusion masks contribute significantly to developing more reliable and efficient odometry systems capable of performing well in diverse and challenging environments.

6.9 Summary

This chapter has highlighted significant advancements in integrating AI and geometric methods for LiDAR SLAM, specifically targeting UGV positioning in GNSS-denied environments. The following contributions have been made:

Utilization of the KITTI Dataset: Due to the pandemic-related constraints, I adapted the approach by utilizing the KITTI dataset, which provides LiDAR data collected from a car. This allowed me to continue developing and testing my methods despite the lack of new drone-collected LiDAR data.

Introduction of the INAF Fusion Method: A key innovation of this work is the INAF method. This novel approach dynamically selects the most relevant information from geometric and deep learning-based odometry methods, enhancing localization results, particularly in challenging environments with large rotations and motion distortions.

Comparison with Baseline Fusion Methods: The chapter compared the INAF method with two baseline fusion methods: direct and attention-based fusion. The results demonstrated that INAF significantly outperforms these baselines in both translation and rotation error metrics.

Handling Large Angle Rotations: The INAF fusion method effectively addresses the challenges posed by the KITTI dataset’s high-frequency scans and large angle rotations, complicating traditional geometric methods and neural network training.

Specific Contributions of INAF Fusion Method

Dynamic Feature Selection: INAF employs advanced inference techniques to dynamically adjust attention weights, selecting the most relevant features from geometric and DL branches based on the current environmental context.

Improved Odometry Accuracy: The method substantially improved odometry accuracy, particularly in scenarios with rapid rotations and motion distortions. INAF achieved lower translation and rotation errors than direct and attention-based fusion methods.

Adaptability to Environmental Conditions: By visualizing sensor fusion masks, INAF demonstrated the ability to adaptively prioritize features from different methods, ensuring robust performance across various driving conditions.

Future research will further refine and test the INAF fusion method in more diverse and complex environments. Expanding the dataset to include additional scenarios and implementing real-time testing on UAVs and UGVs will be crucial. Integrating additional sensor modalities, like cameras, could provide even more comprehensive situational awareness and enhance system resilience. Additionally, classical and deep learning approaches could be updated with state-of-the-art methods like CT-ICP [144] for the geometric-based approach.

Chapter 7

Conclusions and Future Work

This thesis presents significant advancements in UAV and UGV positioning, particularly in GNSS-denied environments, through innovative methods and comprehensive datasets. Each chapter contributes to the overall improvement of autonomous navigation systems. This chapter summarizes each chapter’s key contributions and findings and outlines future research directions.

7.1 Conclusions

7.1.1 Q-Drone UWB Benchmark Dataset

Chapter 3 introduced the Q-Drone UWB benchmark dataset, designed to support research in UAV positioning across challenging GNSS-denied environments. This dataset includes synchronized IMU and UWB data, with ground truth positions established using a robotic total station. Calibration procedures refined the sensor relationships, and rigorous validation measures ensured data accuracy. The Q-Drone benchmark dataset is publicly accessible, providing a valuable resource for developing and validating positioning methods. This dataset is a critical tool for researchers to test and improve their algorithms in diverse and challenging real-world conditions.

7.1.2 Incremental Smoothing-Based Method

Chapter 4 presented an incremental smoothing-based method that integrates high-rate UWB range measurements and low-rate multilateration outcomes into a unified factor graph framework. This method demonstrated improved positioning accuracy through experiments under various scenarios. The proposed technique effectively combined different types of measurements to provide more robust and accurate UAV positioning, especially in GNSS-denied environments. However, further development is needed to address the proper setting of uncertainty values for factors, which could enhance the method’s robustness and overall effectiveness.

7.1.3 DeepCovPG Framework

In Chapter 5, the DeepCovPG framework was introduced, combining deep learning algorithms with UWB technologies to enhance UAV positioning accuracy. By using VAEs and LSTM networks, DeepCovPG effectively refines and denoises UWB range data and dynamically predicts measurement covariances. Rigorous empirical assessments showed significant improvements in MAE and RMSE in GNSS-denied and UWB multipath-prone environments. This framework lays the foundation for future adaptations integrating IMUs and other pose sensors, potentially leading to even more robust positioning solutions. The success of DeepCovPG highlights the potential of combining deep learning with traditional methods to tackle complex positioning challenges.

7.1.4 Integrating AI and Geometric Methods for LiDAR SLAM

Chapter 6 highlighted significant advancements in integrating AI and geometric methods for LiDAR SLAM, specifically targeting UGV positioning in GNSS-denied environments. The KITTI dataset was used to adapt the approach, enabling continued development and testing despite pandemic constraints. A key innovation of this work is the INAF method, which dynamically selects the most relevant information from geometric and deep learning-

based odometry methods. This novel approach enhances localization results, particularly in challenging environments with large rotations and motion distortions. The chapter compared the INAF method with two baseline fusion methods: direct concatenation and attention-based fusion with multi-head self-attention. The results demonstrated that INAF significantly outperforms these baselines in translation and rotation error metrics. Additionally, INAF effectively addresses challenges posed by the KITTI dataset’s high-frequency scans and large-angle rotations, complicating traditional geometric methods and neural network training.

Specific contributions of the INAF fusion method include dynamic feature selection, where advanced inference techniques are used to dynamically adjust attention weights and select the most relevant features from geometric and deep learning branches based on the current environmental context. This adaptability makes INAF highly effective in diverse and complex scenarios.

7.2 Future Work

Building on the advancements presented in this thesis, several areas for future research are identified to enhance UAV and UGV positioning technologies further.

7.2.1 Enhancing the Q-Drone UWB Benchmark Dataset

Future work will focus on refining and expanding the Q-Drone UWB benchmark dataset introduced in Chapter 3. Expanding the dataset to include more varied scenarios and environmental conditions will be essential. This includes capturing data in different weather conditions, terrains, and urban settings. Additionally, more sophisticated calibration and validation procedures will improve data accuracy and reliability. Ensuring the dataset remains publicly accessible and updated will continue to support the research community in developing and testing innovative positioning methods.

7.2.2 Improving the Incremental Smoothing-Based Method

For the incremental smoothing-based method presented in Chapter 4, future research will aim to develop more precise methods for setting uncertainty values for factors. This involves considering factors like the dilution of precision and other environmental variables that can affect the accuracy of multilateration outcomes. Advanced statistical and machine learning techniques can be employed to dynamically adjust these uncertainty values based on real-time data, enhancing the robustness and accuracy of the method. Furthermore, extending the application of this method to other types of sensors and platforms will broaden its utility and impact.

7.2.3 Advancing the DeepCovPG Framework

Future work on the DeepCovPG framework, as discussed in Chapter 5, will involve integrating this framework with IMUs and other pose sensors to create more robust and accurate positioning solutions. This includes training the network across a wider range of environments to improve generalization and adaptability. Real-world testing and extending the applicability of DeepCovPG to other platforms beyond UAVs, such as UGVs and autonomous marine vehicles, will be crucial. Additionally, exploring the integration of other advanced deep learning techniques and sensor fusion strategies will further enhance the performance and reliability of the framework.

7.2.4 Refining the INAF Fusion Method

The INAF fusion method from Chapter 6 will undergo further refinement and testing in diverse and complex environments. Implementing real-time testing on UAVs and UGVs will validate its effectiveness in operational settings. Integrating additional sensor modalities like cameras and radar will provide more comprehensive situational awareness and enhance system resilience. Updating both classical and deep learning approaches with state-of-the-art methods like CT-ICP for geometric-based approaches will further improve performance.

These enhancements aim to push the boundaries of autonomous navigation, contributing to developing more advanced and reliable UAV and UGV systems.

In conclusion, the research presented in this thesis provides a comprehensive framework for improving UAV and UGV positioning in challenging environments. The contributions made through the Q-Drone UWB benchmark dataset, incremental smoothing-based method, DeepCovPG framework, and INAF fusion method collectively advance the field of autonomous navigation, offering robust and accurate solutions for real-world applications.

Bibliography

References

- [1] William W Greenwood, Jerome P Lynch, and Dimitrios Zekkos. “Applications of UAVs in civil infrastructure”. In: *Journal of infrastructure systems* 25.2 (2019), p. 04019002.
- [2] Federal Railroad Administration. *Automated track centerline following for drone flight automation*. URL: <https://railroads.dot.gov/elibrary/automated-track-centerline-following-drone-flight-automation>.
- [3] Hazim Shakhatreh et al. “Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges”. In: *Ieee Access* 7 (2019), pp. 48572–48634.
- [4] *Automating Reality Capture on Job Sites with Drones, Ground Robots, and More: Users Perspective*. Mar. 2023. URL: <https://dronelife.com/2023/10/12/automating-reality-capture-on-job-sites-with-drones-ground-robots-and-more/>.
- [5] He Ren et al. “A review of UAV monitoring in mining areas: Current status and future perspectives”. In: *International Journal of Coal Science & Technology* 6 (2019), pp. 320–333.
- [6] Sophie Jordan et al. “State-of-the-art technologies for UAV inspections”. In: *IET Radar, Sonar & Navigation* 12.2 (2018), pp. 151–164.
- [7] Jennifer Chu. *Researchers develop virtual-reality testing ground for drones*. URL: <https://news.mit.edu/2018/virtual-reality-testing-ground-drones-0517>.

- [8] João Antunes. <https://www.commercialuavnews.com/infrastructure/hard-to-reach-places-smart-drones-making-infrastructure-inspection-safer>. Mar. 2022.
- [9] Faine Greenwood. *Assessing the impact of drones in the global covid response*. July 2021. URL: <https://www.brookings.edu/articles/assessing-the-impact-of-drones-in-the-global-covid-response/>.
- [10] Mohinder S Grewal. “Global navigation satellite systems”. In: *Wiley Interdisciplinary Reviews: Computational Statistics* 3.4 (2011), pp. 383–384.
- [11] Elliott D Kaplan and Christopher Hegarty. *Understanding GPS/GNSS: principles and applications*. Artech house, 2017.
- [12] Abdul Mueed Hafiz and Ghulam Mohiuddin Bhat. “Deep network ensemble learning applied to image classification using CNN trees”. In: *arXiv preprint arXiv:2008.00829* (2020).
- [13] NA Barkhatov et al. “Low-frequency oscillations while magnetic storms as a tool to determine the types of solar plasma flows”. In: *E3S Web of Conferences*. Vol. 220. EDP Sciences. 2020, p. 01090.
- [14] Purba Dutta et al. “Analysis of jamming and anti jamming techniques for Galileo GNSS”. In: *Materials Today: Proceedings* 58 (2022), pp. 489–495.
- [15] Desmond Schmidt et al. “A survey and analysis of the GNSS spoofing threat and countermeasures”. In: *ACM Computing Surveys (CSUR)* 48.4 (2016), pp. 1–31.
- [16] Fabricio S Prol et al. “Position, navigation, and timing (PNT) through low earth orbit (LEO) satellites: A survey on current status, challenges, and opportunities”. In: *IEEE Access* 10 (2022), pp. 83971–84002.
- [17] Paloma Carrasco et al. “Multi-sensor fusion for aerial robots in industrial GNSS-denied environments”. In: *Applied Sciences* 11.9 (2021), p. 3921.

- [18] Sukchang Yun, Young Jae Lee, and Sangkyung Sung. “IMU/Vision/Lidar integrated navigation system in GNSS denied environments”. In: *2013 IEEE Aerospace Conference*. IEEE. 2013, pp. 1–10.
- [19] Kailash Krishnaswamy et al. “Sensor fusion for GNSS denied navigation”. In: *2008 IEEE/ION Position, Location and Navigation Symposium*. IEEE. 2008, pp. 541–551.
- [20] Hui Liu et al. “Survey of wireless indoor positioning techniques and systems”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 37.6 (2007), pp. 1067–1080.
- [21] Fazeelat Mazhar, Muhammad Gufran Khan, and Benny Sällberg. “Precise indoor positioning using UWB: A review of methods, algorithms and implementations”. In: *Wireless Personal Communications* 97.3 (2017), pp. 4467–4491.
- [22] Guowei Shi and Ying Ming. “Survey of indoor positioning systems based on ultra-wideband (UWB) technology”. In: *Wireless Communications, Networking and Applications: Proceedings of WCNA 2014*. Springer. 2016, pp. 1269–1278.
- [23] Günther Retscher, Daniel Kiss, and Jelena Gabela. “Fusion of GNSS pseudoranges with UWB ranges based on clustering and weighted least squares”. In: *Sensors* 23.6 (2023), p. 3303.
- [24] Abdulrahman Alarifi et al. “Ultra wideband indoor positioning technologies: Analysis and recent advances”. In: *Sensors* 16.5 (2016), p. 707.
- [25] Aitor Guisasola et al. “Reliable and Low-Power Communications System Based on IR-UWB for Offshore Wind Turbines”. In: *Electronics* 11.4 (2022), p. 570.
- [26] Marina Md Din et al. “Indoor positioning: technology comparison analysis”. In: *International Journal of Engineering & Technology* 7.2 (2018), p. 14.
- [27] Shilpa Shyam, Sujitha Juliet, and Kirubakaran Ezra. “A UWB Based Indoor Asset Tracking Architecture for Industry 4.0”. In: *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*. IEEE. 2022, pp. 501–506.

- [28] Javier San Martín et al. “Precise positioning of autonomous vehicles combining UWB ranging estimations with on-board sensors”. In: *Electronics* 9.8 (2020), p. 1238.
- [29] Daniel Minoli and Benedict Occhiogrosso. “Ultrawideband (UWB) technology for smart cities IoT applications”. In: *2018 IEEE international smart cities conference (ISC2)*. IEEE. 2018, pp. 1–8.
- [30] Yu Xianjia et al. “Applications of uwb networks and positioning to autonomous robots and industrial systems”. In: *2021 10th Mediterranean Conference on Embedded Computing (MECO)*. IEEE. 2021, pp. 1–6.
- [31] Arnis Cirulis. “Ultra wideband tracking potential for augmented reality environments”. In: *Augmented Reality, Virtual Reality, and Computer Graphics: 6th International Conference, AVR 2019, Santa Maria al Bagno, Italy, June 24–27, 2019, Proceedings, Part II* 6. Springer. 2019, pp. 126–136.
- [32] Szymon Czyża et al. “Assessment of accuracy in unmanned aerial vehicle (uav) pose estimation with the real-time kinematic (rtk) method on the example of dji matrice 300 rtk”. In: *Sensors* 23.4 (2023), p. 2092.
- [33] Kai Guo et al. “An accurate and robust method for absolute pose estimation with UAV using RANSAC”. In: *Sensors* 22.15 (2022), p. 5925.
- [34] Ruben Mascaro et al. “Gomsf: Graph-optimization based multi-sensor fusion for robust uav pose estimation”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 1421–1428.
- [35] Yassine Salih Alj, Charles Despins, and Sofiene Affes. “Impact of multipath interference on the performance of an UWB fast acquisition system for ranging in an indoor wireless channel”. In: *6th Annual Communication Networks and Services Research Conference (cnsr 2008)*. IEEE. 2008, pp. 390–396.
- [36] Mohammed Dahiru Buhari et al. “Statistical LOS/NLOS Classification for UWB Channels”. In: *arXiv preprint arXiv:2308.07726* (2023).

- [37] Cung Lian Sang et al. “Identification of NLOS and multi-path conditions in UWB localization using machine learning methods”. In: *Applied Sciences* 10.11 (2020), p. 3980.
- [38] Airu Zou et al. “An Improved UWB/IMU Tightly Coupled Positioning Algorithm Study”. In: *Sensors* 23.13 (2023), p. 5918.
- [39] Pengcheng Shi et al. “Covariance estimation for pose graph optimization in visual-inertial navigation systems”. In: *IEEE Transactions on Intelligent Vehicles* (2023).
- [40] Ziqi Lu et al. “SLAM-supported self-training for 6D object pose estimation”. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2022, pp. 2833–2840.
- [41] Julian Nubert, Shehryar Khattak, and Marco Hutter. “Self-supervised learning of lidar odometry for robotic applications”. In: *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2021, pp. 9601–9607.
- [42] Changhao Chen et al. “Selective sensor fusion for neural visual-inertial odometry”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 10542–10551.
- [43] Federal Communications Commission. *Fcc Amends Rules to Permit New Wideband Unlicensed Devices and Affirms Rules to Authorize The Deployment of Ultra-Wideband Technology*. Report. 2004.
- [44] Kunwoo Park et al. “Multilateration under flip ambiguity for UAV positioning using ultrawide-band”. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 1 (2020), pp. 317–323.
- [45] Jiaxin Li et al. “Accurate 3D localization for MAV swarms by UWB and IMU fusion”. In: *2018 IEEE 14th International Conference on Control and Automation (ICCA)*. IEEE. 2018, pp. 100–105.

- [46] Janis Tiemann, Florian Schweikowski, and Christian Wietfeld. “Design of an UWB indoor-positioning system for UAV navigation in GNSS-denied environments”. In: *2015 international conference on indoor positioning and indoor navigation (IPIN)*. IEEE. 2015, pp. 1–7.
- [47] Mark W Mueller, Michael Hamer, and Raffaello D’Andrea. “Fusing ultra-wideband range measurements with accelerometers and rate gyroscopes for quadcopter state estimation”. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 1730–1736.
- [48] S Zahran et al. “Micro-radar and UWB aided UAV navigation in GNSS denied environment”. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 42 (2018), pp. 469–476.
- [49] Jungwon Kang et al. “Ultra-wideband aided UAV positioning using incremental smoothing with ranges and multilateration”. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 4529–4536.
- [50] Yang Song and Li-Ta Hsu. “Tightly coupled integrated navigation system via factor graph for UAV indoor localization”. In: *Aerospace Science and Technology* 108 (2021), p. 106370.
- [51] Qinghua Zeng et al. “An improved multi-sensor fusion navigation algorithm based on the factor graph”. In: *Sensors* 17.3 (2017), p. 641.
- [52] Jun Dai et al. “UAV localization algorithm based on factor graph optimization in complex scenes”. In: *Sensors* 22.15 (2022), p. 5862.
- [53] Anton Ledergerber and Raffaello D’Andrea. “Ultra-wideband angle of arrival estimation based on angle-dependent antenna transfer function”. In: *Sensors* 19.20 (2019), p. 4466.
- [54] Anton Ledergerber. “Dataset to" Ultra-wideband angle of arrival estimation based on angle-dependent antenna transfer function"”. In: (2019).

- [55] Adrien Van den Bossche et al. “Locura: A new localisation and uwb-based ranging testbed for the internet of things”. In: *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE. 2018, pp. 1–6.
- [56] Matteo Ridolfi et al. “Experimental evaluation of UWB indoor positioning for sport postures”. In: *Sensors* 18.1 (2018), p. 168.
- [57] Milad Heydariaan, Hessam Mohammadmoradi, and Omprakash Gnawali. “Toward standard non-line-of-sight benchmarking of ultra-wideband radio-based localization”. In: *2018 IEEE Workshop on Benchmarking Cyber-Physical Networks and Systems (CPSBench)*. IEEE. 2018, pp. 19–24.
- [58] Mark W Mueller, Michael Hamer, and Raffaello D’Andrea. “Fusing ultra-wideband range measurements with accelerometers and rate gyroscopes for quadcopter state estimation”. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 1730–1736.
- [59] Kexin Guo et al. “Ultra-wideband-based localization for quadcopter navigation”. In: *Unmanned Systems* 4.01 (2016), pp. 23–34.
- [60] Emil Fresk, Kristoffer Ödmark, and George Nikolakopoulos. “Ultra wideband enabled inertial odometry for generic localization”. In: *IFAC-PapersOnLine* 50.1 (2017), pp. 11465–11472.
- [61] Brandon Dewberry and Mike Einhorn. “Indoor aerial vehicle navigation using ultra wideband active two-way ranging”. In: *Proceedings of the AUVSI Xponential Conference*. 2016.
- [62] Chen Wang et al. “Ultra-wideband aided fast localization and mapping system”. In: *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE. 2017, pp. 1602–1609.
- [63] Francisco Javier Perez-Grau et al. “Multi-modal mapping and localization of unmanned aerial robots based on ultra-wideband and RGB-D sensing”. In: *2017 IEEE/RSJ inter-*

- national conference on intelligent robots and systems (IROS)*. IEEE. 2017, pp. 3495–3502.
- [64] Chen Wang et al. “Graph optimization approach to localization with imu and ultra-wideband measurements”. In: *CoRR* (2018).
 - [65] Erik Strömberg. *Smoothing and mapping of an unmanned aerial vehicle using ultra-wideband sensors*. 2017.
 - [66] Qian Meng et al. “Resilient tightly coupled INS/UWB integration method for indoor UAV navigation under challenging scenarios”. In: *Defence Technology* 22 (2023), pp. 185–196.
 - [67] Karthikeyan Gururaj et al. “Real-time identification of NLOS range measurements for enhanced UWB localization”. In: *2017 international conference on indoor positioning and indoor navigation (IPIN)*. IEEE. 2017, pp. 1–7.
 - [68] Xin Li and Yang Wang. “Research on a factor graph-based robust UWB positioning algorithm in NLOS environments”. In: *Telecommunication Systems* 76 (2021), pp. 207–217.
 - [69] Leonardo A Fagundes-Junior et al. “Machine Learning for Unmanned Aerial Vehicles Navigation: An Overview”. In: *SN Computer Science* 5.2 (2024), p. 256.
 - [70] Yirga Yayeh Munaye et al. “UAV positioning for throughput maximization using deep learning approaches”. In: *Sensors* 19.12 (2019), p. 2775.
 - [71] Assefinew Wondosen et al. “Bayesian optimization for fine-tuning EKF parameters in UAV attitude and heading reference system estimation”. In: *Aerospace* 10.12 (2023), p. 1023.
 - [72] Chamali Sandamini et al. “A Review of Indoor Positioning Systems for UAV Localization with Machine Learning Algorithms”. In: *Electronics* 12.7 (2023), p. 1533.

- [73] Uthman Olawoye and Jason N Gross. “Uav position estimation using a lidar-based 3d object detection method”. In: *2023 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. IEEE. 2023, pp. 46–51.
- [74] H-A Loeliger. “An introduction to factor graphs”. In: *IEEE Signal Processing Magazine* 21.1 (2004), pp. 28–41.
- [75] Szymon Czyża et al. “Assessment of Accuracy in Unmanned Aerial Vehicle (UAV) Pose Estimation with the REAL-Time Kinematic (RTK) Method on the Example of DJI Matrice 300 RTK”. In: *Sensors* 23.4 (2023), p. 2092.
- [76] Kai Guo et al. “An Accurate and Robust Method for Absolute Pose Estimation with UAV Using RANSAC”. In: *Sensors* 22.15 (2022), p. 5925.
- [77] Ruben Mascaro et al. “Gomsf: Graph-optimization based multi-sensor fusion for robust uav pose estimation”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 1421–1428.
- [78] David J Yoon et al. “Unsupervised learning of lidar features for use in a probabilistic trajectory estimator”. In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 2130–2138.
- [79] Valentin Peretroukhin and Jonathan Kelly. “Dpc-net: Deep pose correction for visual localization”. In: *IEEE Robotics and Automation Letters* 3.3 (2017), pp. 2424–2431.
- [80] Xuewei Zhou and Ruining Huang. “A state-of-the-art review on SLAM”. In: *International Conference on Intelligent Robotics and Applications*. Springer. 2022, pp. 240–251.
- [81] Xiaobin Xu et al. “A review of multi-sensor fusion slam systems based on 3D LIDAR”. In: *Remote Sensing* 14.12 (2022), p. 2835.
- [82] Team Project 2 tp2@mss.isunet.edu Students of the International Space University. “A GNSS Literature Review and a Navigation System for Future Space Vehicles”. In: *Satellite Navigation Systems: Policy, Commercial and Technical Interaction* (2003), pp. 45–52.

- [83] Weifeng Chen et al. “An overview on visual slam: From tradition to semantic”. In: *Remote Sensing* 14.13 (2022), p. 3010.
- [84] Xiangdi Yue et al. “LiDAR-based SLAM for robotic mapping: state of the art and new frontiers”. In: *Industrial Robot: the international journal of robotics research and application* 51.2 (2024), pp. 196–205.
- [85] Samer Karam, Ville Lehtola, and George Vosselman. “Strategies to integrate IMU and LiDAR SLAM for indoor mapping”. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 1 (2020), pp. 223–230.
- [86] Zhongli Wang et al. “IMU-assisted 2D SLAM method for low-texture and dynamic environments”. In: *Applied Sciences* 8.12 (2018), p. 2534.
- [87] Kunyi Li et al. “Dns slam: Dense neural semantic-informed slam”. In: *arXiv preprint arXiv:2312.00204* (2023).
- [88] Mingrui Li et al. “DDN-SLAM: Real-time Dense Dynamic Neural Implicit SLAM with Joint Semantic Encoding”. In: *arXiv preprint arXiv:2401.01545* (2024).
- [89] Patryk Mazurek and Tomasz Hachaj. “SLAM-OR: simultaneous localization, mapping and object recognition using video sensors data in open environments from the sparse points cloud”. In: *Sensors* 21.14 (2021), p. 4734.
- [90] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. “Generalized-icp.” In: *Robotics: science and systems*. Vol. 2. 4. Seattle, WA. 2009, p. 435.
- [91] Ji Zhang and Sanjiv Singh. “LOAM: Lidar odometry and mapping in real-time.” In: *Robotics: Science and systems*. Vol. 2. 9. Berkeley, CA. 2014, pp. 1–9.
- [92] Julian Nubert, Shehryar Khattak, and Marco Hutter. “Self-supervised learning of lidar odometry for robotic applications”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 9601–9607.
- [93] Qing Li et al. “Lo-net: Deep real-time lidar odometry”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 8473–8482.

- [94] Zhichao Li and Naiyan Wang. “Dmlo: Deep matching lidar odometry”. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 6010–6017.
- [95] Brandon Wagstaff, Valentin Peretroukhin, and Jonathan Kelly. “Self-supervised deep pose corrections for robust visual odometry”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 2331–2337.
- [96] Jiaxin Zhang et al. “Deep online correction for monocular visual odometry”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 14396–14402.
- [97] Neelesh Mungoli. “Adaptive Ensemble Learning: Boosting Model Performance through Intelligent Feature Fusion in Deep Neural Networks”. In: *arXiv preprint arXiv:2304.02653* (2023).
- [98] Neelesh Mungoli. “Adaptive feature fusion: enhancing generalization in deep learning models”. In: *arXiv preprint arXiv:2304.03290* (2023).
- [99] Firuz Juraev, Shaker El-Sappagh, and Tamer Abuhmed. “Explainable Dynamic Ensemble Framework for Classification Based on the Late Fusion of Heterogeneous Multimodal Data”. In: *Intelligent Systems Conference*. Springer. 2023, pp. 555–570.
- [100] Matteo Ridolfi et al. “Experimental evaluation of UWB indoor positioning for sport postures”. In: *Sensors* 18.1 (2018), p. 168.
- [101] Wenda Zhao et al. “Learning-based bias correction for ultra-wideband localization of resource-constrained mobile robots”. In: *arXiv preprint arXiv:2003.09371* (2020).
- [102] Wei Cui et al. “Dynamic multidimensional scaling algorithm for 3-D mobile localization”. In: *IEEE Transactions on Instrumentation and Measurement* 65.12 (2016), pp. 2853–2865.

- [103] João Sena Esteves, Adriano Carvalho, and Carlos Couto. “Generalized geometric triangulation algorithm for mobile robot absolute self-localization”. In: *2003 IEEE international symposium on industrial electronics (Cat. No. 03TH8692)*. Vol. 1. IEEE. 2003, pp. 346–351.
- [104] Yifeng Zhou, Jun Li, and Louise Lamont. “Multilateration localization in the presence of anchor location uncertainties”. In: *2012 IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2012, pp. 309–314.
- [105] Jorge J Moré. “The Levenberg-Marquardt algorithm: implementation and theory”. In: *Numerical analysis: proceedings of the biennial Conference held at Dundee, June 28–July 1, 1977*. Springer. 2006, pp. 105–116.
- [106] Wenda Zhao et al. “UTIL: An Ultra-wideband Time-difference-of-arrival Indoor Localization Dataset”. In: *International Journal of Robotics Research (IJRR)*. 2022.
- [107] Paola Torrico Morón et al. “Benchmarking UWB-based infrastructure-free positioning and multi-robot relative localization: dataset and characterization”. In: *2023 IEEE Sensors Applications Symposium (SAS)*. IEEE. 2023, pp. 1–6.
- [108] Usman Raza et al. “Dataset: Indoor localization with narrow-band, ultra-wideband, and motion capture systems”. In: *Proceedings of the 2nd Workshop on Data Acquisition to Analysis*. 2019, pp. 34–36.
- [109] Quentin Vey et al. “Indoor UWB localisation: LocURa4IoT testbed and dataset presentation”. In: *2022 IEEE 47th Conference on Local Computer Networks (LCN)*. IEEE. 2022, pp. 258–260.
- [110] Yifeng Zhou, Jun Li, and Louise Lamont. “Multilateration localization in the presence of anchor location uncertainties”. In: *2012 IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2012, pp. 309–314.
- [111] Abdelmoumen Norrdine. “An algebraic solution to the multilateration problem”. In: *Proceedings of the 15th international conference on indoor positioning and indoor navigation, Sydney, Australia*. Vol. 1315. 2012.

- [112] Jiahong Li et al. “A novel robust trilateration method applied to ultra-wide bandwidth location systems”. In: *Sensors* 17.4 (2017), p. 795.
- [113] Pablo Coteria et al. “Indoor robot positioning using an enhanced trilateration algorithm”. In: *International Journal of Advanced Robotic Systems* 13.3 (2016), p. 110.
- [114] Vadim Indelman et al. “Information fusion in navigation systems via factor graph based incremental smoothing”. In: *Robotics and Autonomous Systems* 61.8 (2013), pp. 721–738.
- [115] Michael Kaess et al. “iSAM2: Incremental smoothing and mapping using the Bayes tree”. In: *The International Journal of Robotics Research* 31.2 (2012), pp. 216–235.
- [116] Todd Lupton and Salah Sukkarieh. “Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions”. In: *IEEE Transactions on Robotics* 28.1 (2011), pp. 61–76.
- [117] Frank Dellaert. “Factor graphs and GTSAM: A hands-on introduction”. In: *Georgia Institute of Technology, Tech. Rep 2* (2012), p. 4.
- [118] Lionel G  nev  , Olivier Kermorgant, and Edouard Laroche. “Limits of trilateration-based sensor placement algorithms”. In: *2017 IEEE Sensors Applications Symposium (SAS)*. IEEE. 2017, pp. 1–6.
- [119] Lukasz Zwirello et al. “UWB localization system for indoor applications: Concept, realization and analysis”. In: *Journal of Electrical and Computer Engineering* 2012 (2012), pp. 1–11.
- [120] Humphrey Hu and George Kantor. “Parametric covariance prediction for heteroscedastic noise”. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2015, pp. 3052–3057.
- [121] Katherine Liu et al. “Deep inference for covariance estimation: Learning gaussian noise models for state estimation”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 1436–1443.

- [122] Jorge J Moré. “The Levenberg-Marquardt algorithm: implementation and theory”. In: *Numerical analysis: proceedings of the biennial Conference held at Dundee, June 28–July 1, 1977*. Springer. 2006, pp. 105–116.
- [123] Joffray Guillory, Daniel Truong, and Jean-Pierre Wallerand. “Multilateration with self-calibration: uncertainty assessment, experimental measurements and Monte-Carlo simulations”. In: *Metrology* 2.2 (2022), pp. 241–262.
- [124] Michael Kaess et al. “iSAM2: Incremental smoothing and mapping using the Bayes tree”. In: *The International Journal of Robotics Research* 31.2 (2012), pp. 216–235.
- [125] Frank Dellaert. “Factor graphs and GTSAM: A hands-on introduction”. In: *Georgia Institute of Technology, Tech. Rep* 2 (2012), p. 4.
- [126] Ferdin Joe John Joseph, Sarayut Nonsiri, and Annop Monsakul. “Keras and TensorFlow : A hands-on experience”. In: *Advanced deep learning for engineers and scientists: A practical approach* (2021), pp. 85–111.
- [127] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).
- [128] Zahra Arjmandi et al. “Benchmark dataset of ultra-wideband radio based UAV positioning”. In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2020, pp. 1–8.
- [129] Ke-Lin Du and Madisetti NS Swamy. *Neural networks and statistical learning*. Springer Science & Business Media, 2013.
- [130] Giseop Kim and Ayoung Kim. “Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 4802–4809.
- [131] Jing Dong and Zhaoyang Lv. “miniSAM: A flexible factor graph non-linear least squares optimization framework”. In: *arXiv preprint arXiv:1909.00903* (2019).

- [132] Changhao Chen et al. “Learning selective sensor fusion for state estimation”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2022).
- [133] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.
- [134] Evan G Hemingway and Oliver M O’Reilly. “Perspectives on Euler angle singularities, gimbal lock, and the orthogonality of applied forces and applied moments”. In: *Multibody system dynamics* 44 (2018), pp. 31–56.
- [135] Ben Kenwright. “A beginners guide to dual-quaternions: what they are, how they work, and how to use them for 3D character hierarchies”. In: (2012).
- [136] Tobias Löw and Sylvain Calinon. “Geometric algebra for optimal control with applications in manipulation tasks”. In: *IEEE Transactions on Robotics* 39.5 (2023), pp. 3586–3600.
- [137] Andreas Geiger et al. “Vision meets Robotics: The KITTI Dataset”. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.
- [138] Younggun Cho, Giseop Kim, and Ayoung Kim. “Unsupervised geometry-aware deep lidar odometry”. In: *2020 IEEE International Conference on robotics and Automation (ICRA)*. IEEE. 2020, pp. 2145–2152.
- [139] Lisha Li et al. “Hyperband: A novel bandit-based approach to hyperparameter optimization”. In: *Journal of Machine Learning Research* 18.185 (2018), pp. 1–52.
- [140] Guanghui Xue et al. “LeGO-LOAM-SC: An Improved Simultaneous Localization and Mapping Method Fusing LeGO-LOAM and Scan Context for Underground Coalmine”. In: *Sensors* 22.2 (2022), p. 520.
- [141] PJ Besl and ND McKay. “A method for registration of 3-D shapes, IEEE Trans.” In: *P pattern Anal. and Machine Intelligence* 14 (1992), p. 23.
- [142] Zhengyou Zhang. “Iterative point matching for registration of free-form curves and surfaces”. In: *International journal of computer vision* 13.2 (1994), pp. 119–152.

- [143] Martin Velas et al. “CNN for IMU assisted odometry estimation using velodyne LiDAR”. In: *2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. IEEE. 2018, pp. 71–77.
- [144] Pierre Dellenbach et al. “Ct-icp: Real-time elastic lidar odometry with loop closure”. In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 5580–5586.
- [145] Qiku Cao et al. “An Iterative Closest Point Method for Lidar Odometry with Fused Semantic Features”. In: *Applied Sciences* 13.23 (2023), p. 12741.

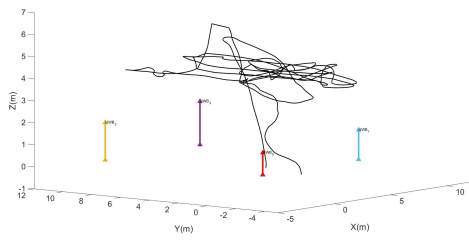
Appendices

Appendix A

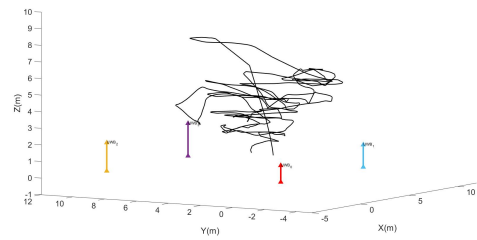
UWB Positioning Experimental Results Figures

This appendix presents the UWB positioning experimental results, showing the outcomes of ground truth and three other methods. For each method, five figures illustrate the results in various environments, including indoor, field, building, bridge, and tunnel. The sections are organized as follows: Benchmark Dataset (Chapter 3), MLAT Results (Chapter 3), Incremental Smoothing Results (Chapter 4), and the DeepCovPG Results (Chapter 5).

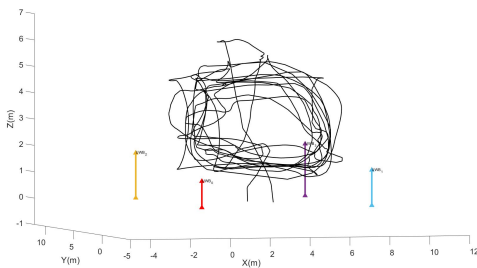
A.1 Benchmark Dataset



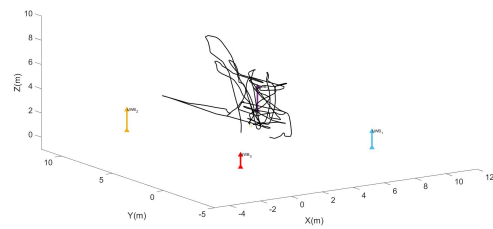
(a)



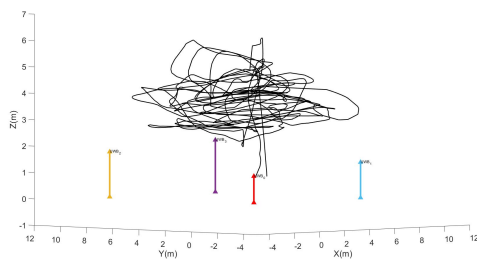
(b)



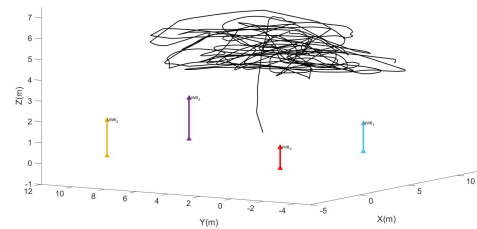
(c)



(d)

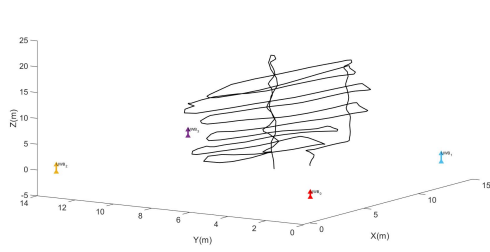


(e)

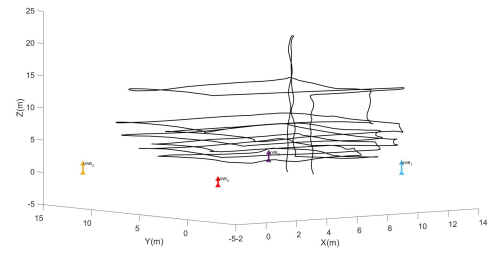


(f)

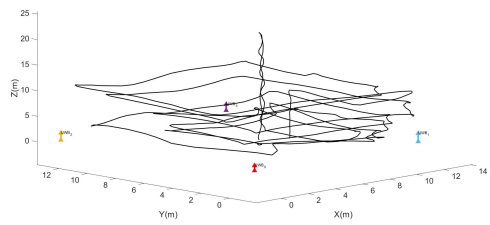
Figure A.1 Indoor datasets collected using the Q-Drone system, with trajectories generated based on ground truth.



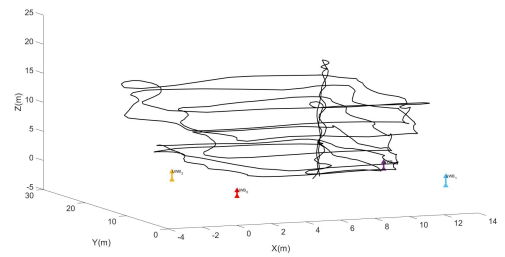
(a)



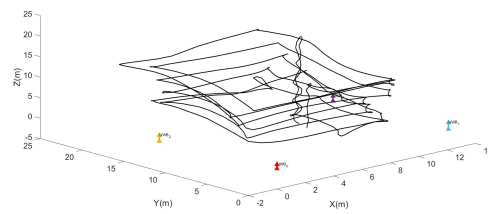
(b)



(c)

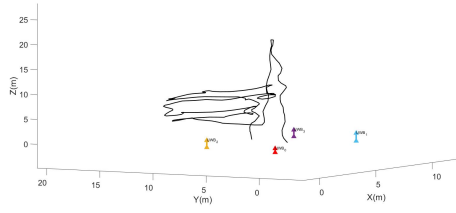


(d)

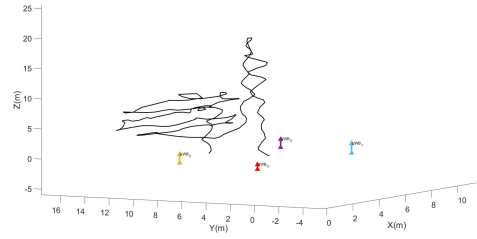


(e)

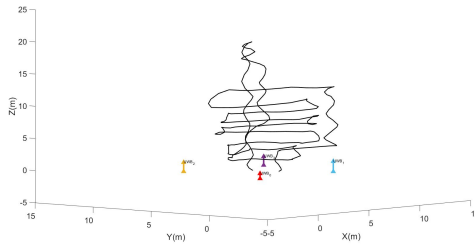
Figure A.2 Field datasets collected using the Q-Drone system, with trajectories generated based on ground truth.



(a)

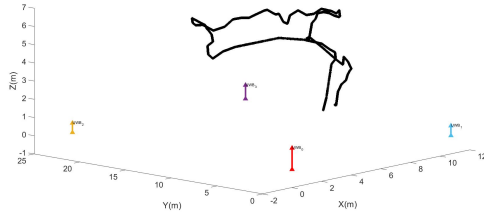


(b)

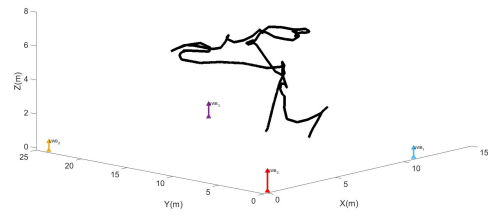


(c)

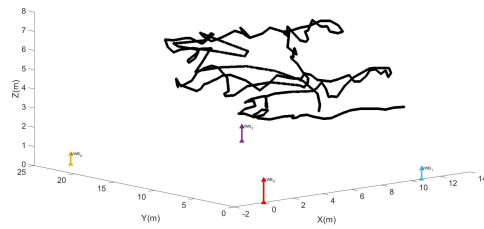
Figure A.3 Building datasets collected using the Q-Drone system, with trajectories generated based on ground truth.



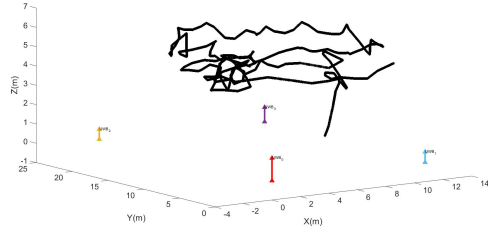
(a)



(b)

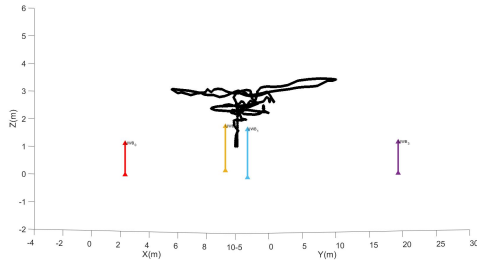


(c)

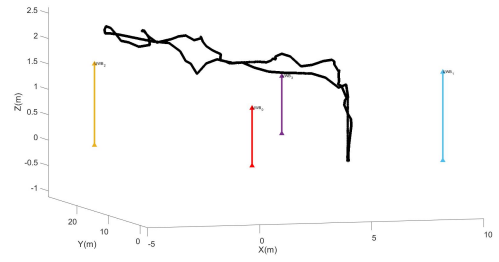


(d)

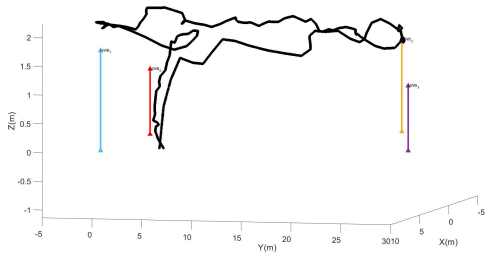
Figure A.4 Bridge datasets collected using the Q-Drone system, with trajectories generated based on ground truth.



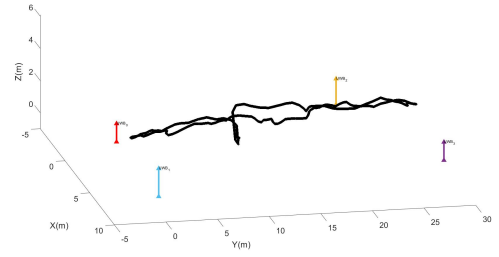
(a)



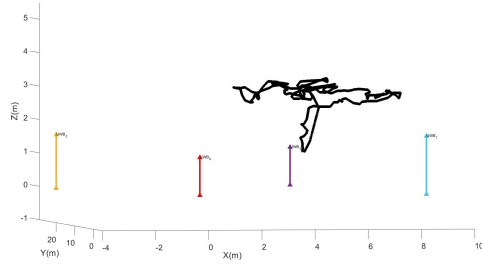
(b)



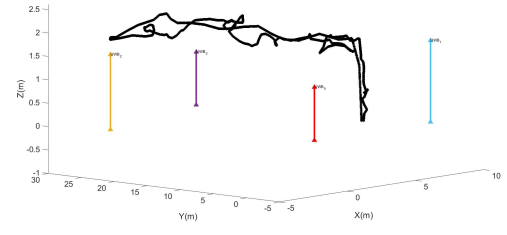
(c)



(d)



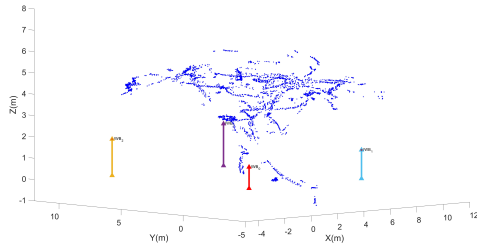
(e)



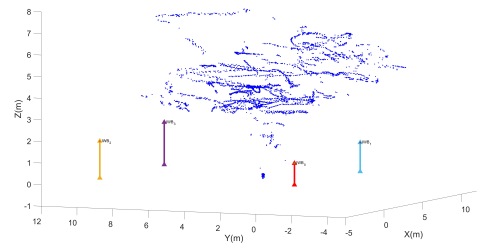
(f)

Figure A.5 Tunnel datasets collected using the Q-Drone system, with trajectories generated based on ground truth.

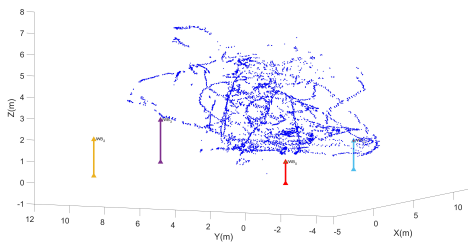
A.2 MLAT Results



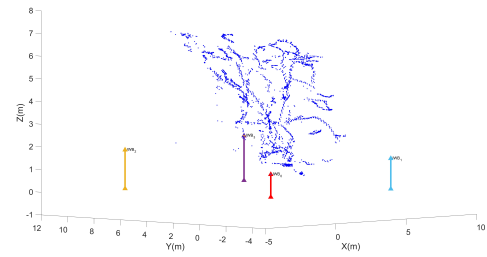
(a)



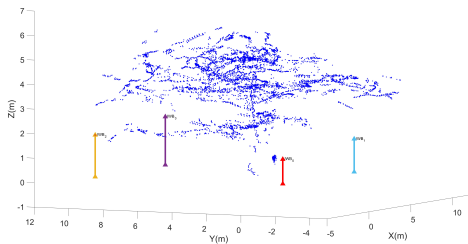
(b)



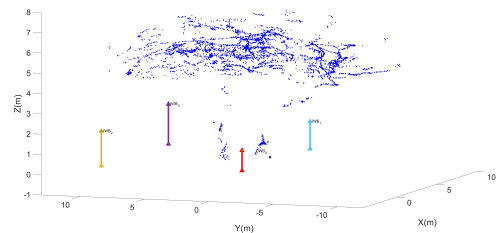
(c)



(d)

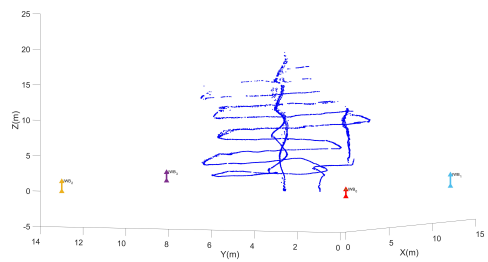


(e)

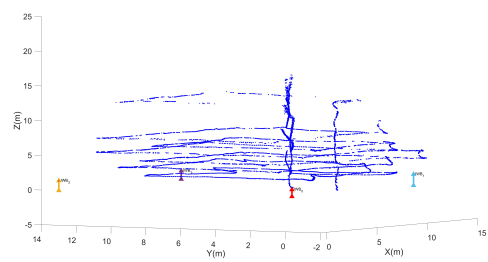


(f)

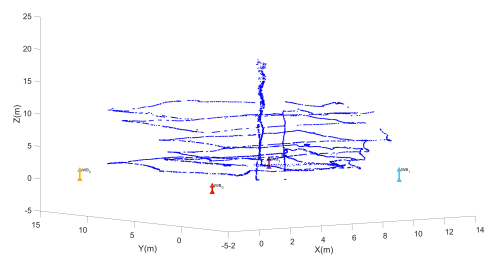
Figure A.6 Indoor datasets collected using the Q-Drone system, with trajectories generated based on MLAT.



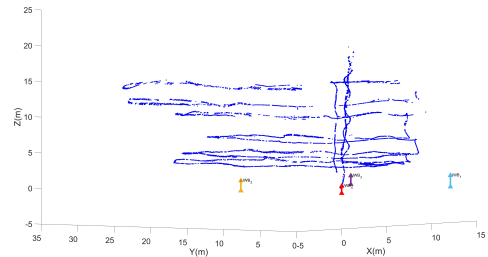
(a)



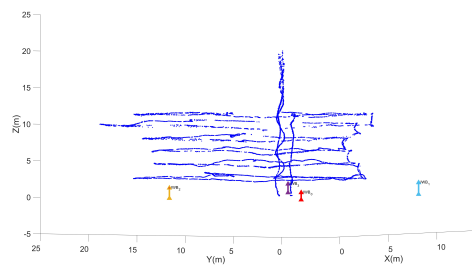
(b)



(c)

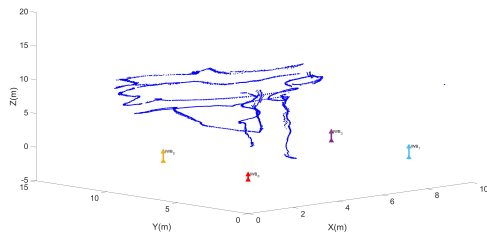


(d)

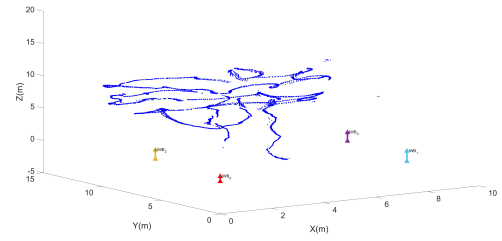


(e)

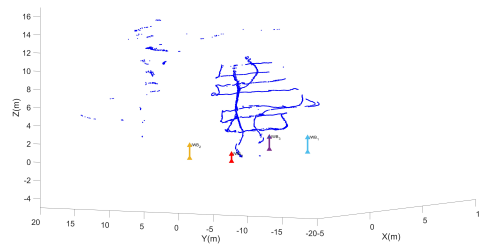
Figure A.7 Field datasets collected using the Q-Drone system, with trajectories generated based on MLAT.



(a)

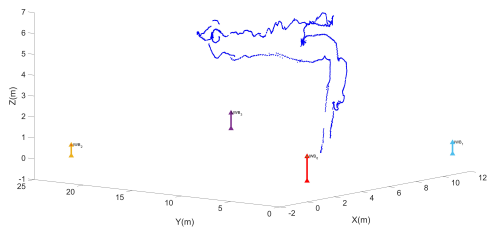


(b)

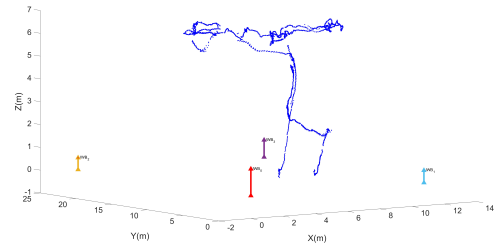


(c)

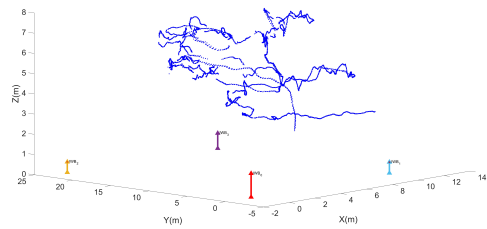
Figure A.8 Building datasets collected using the Q-Drone system, with trajectories generated based on MLAT.



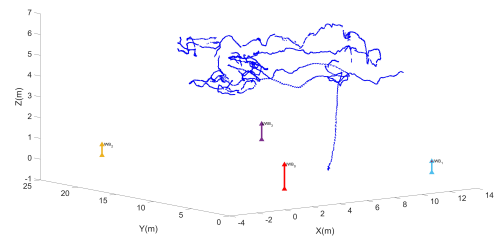
(a)



(b)

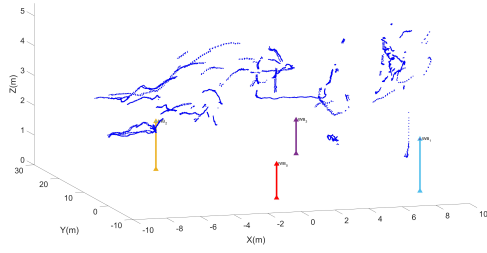


(c)

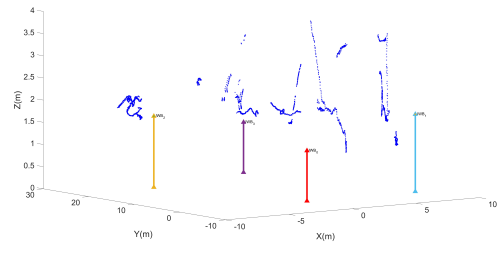


(d)

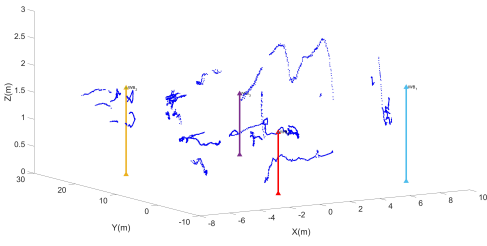
Figure A.9 Bridge datasets collected using the Q-Drone system, with trajectories generated based on MLAT.



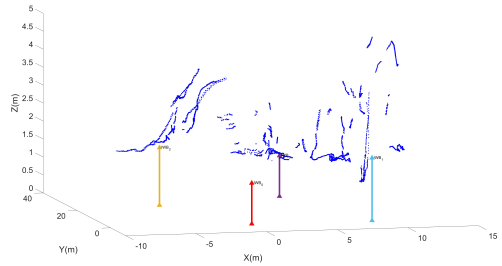
(a)



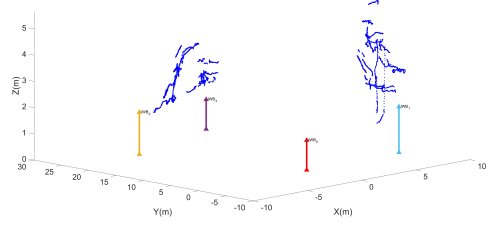
(b)



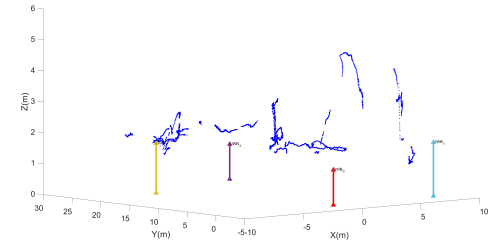
(c)



(d)



(e)



(f)

Figure A.10 Tunnel datasets collected using the Q-Drone system, with trajectories generated based on ground truth.

A.3 Incremental Smoothing Result

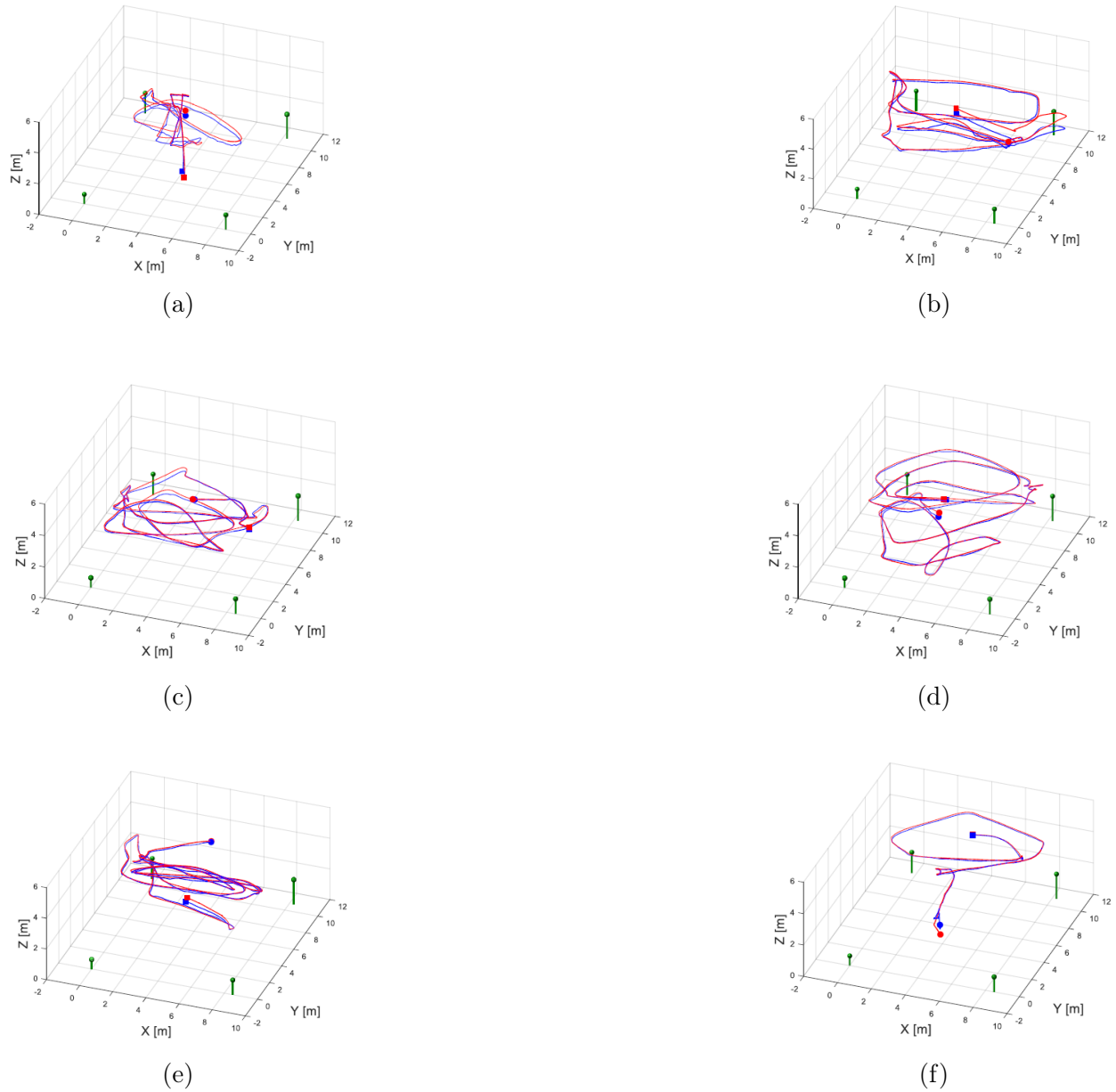
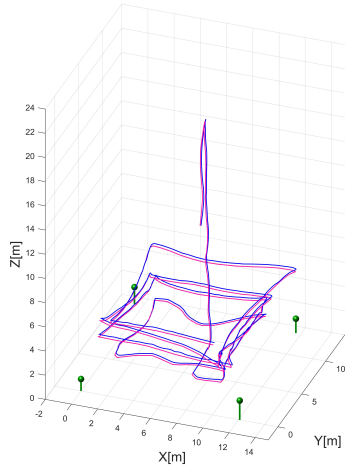
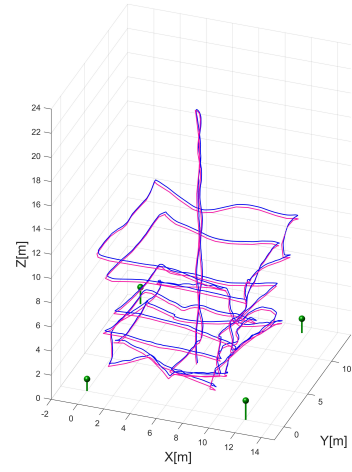


Figure A.11 Indoor datasets collected using the Q-Drone system, with trajectories generated using incremental smoothing method.

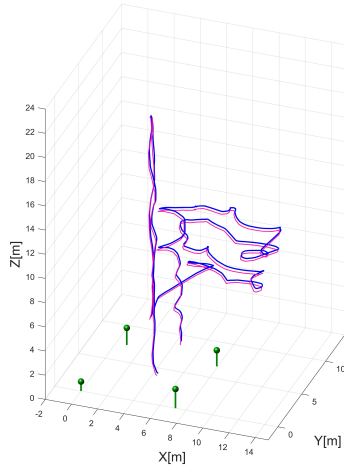


(a)

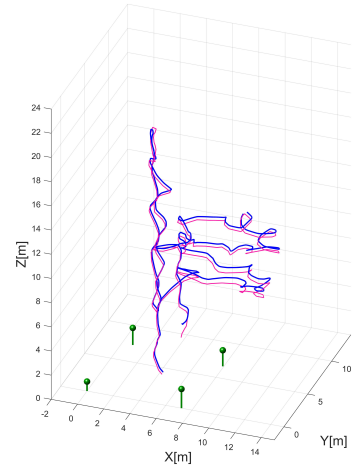


(b)

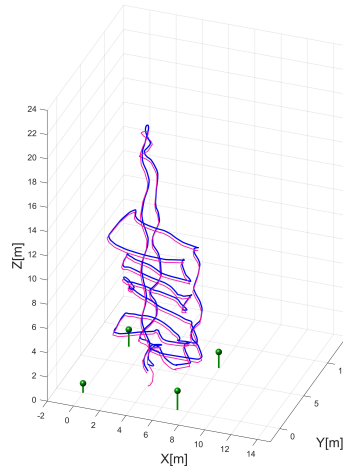
Figure A.12 Field datasets collected using the Q-Drone system, with trajectories generated using incremental smoothing method.



(a)

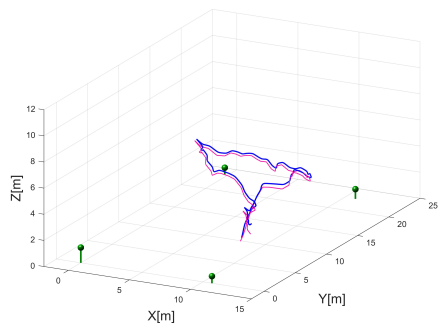


(b)

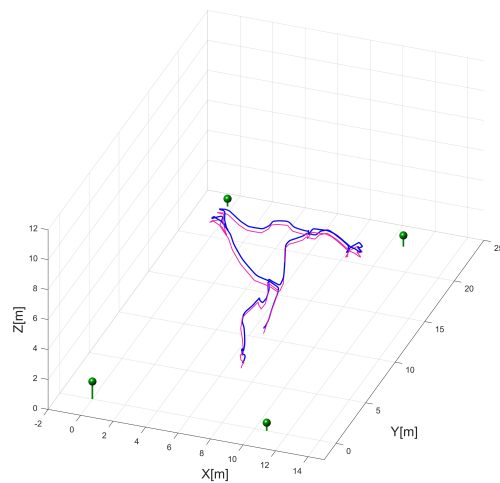


(c)

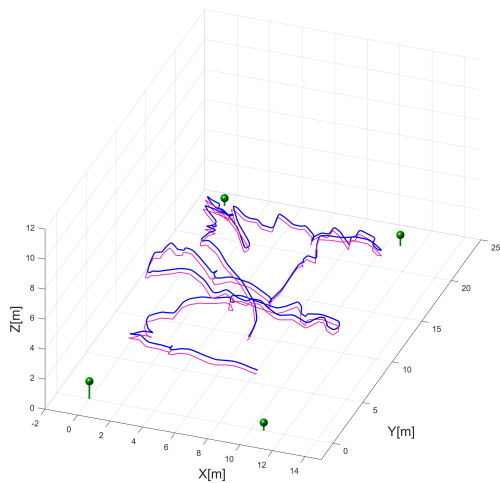
Figure A.13 Building datasets collected using the Q-Drone system, with trajectories generated using incremental smoothing method.



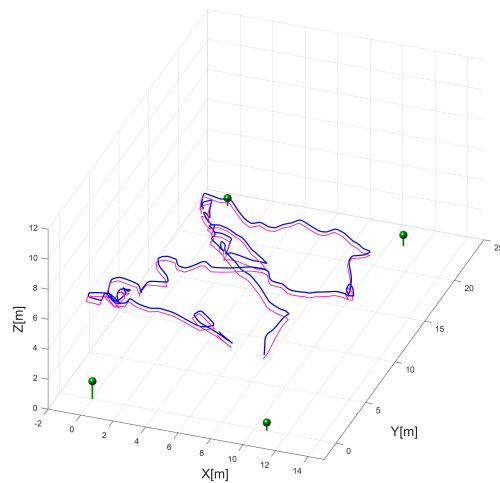
(a)



(b)

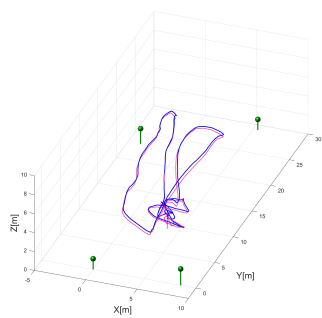


(c)

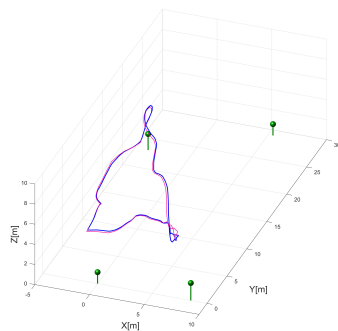


(d)

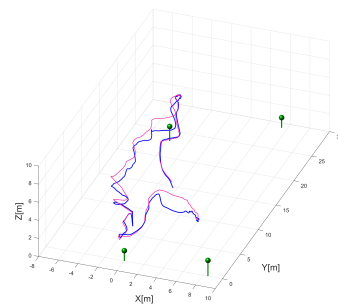
Figure A.14 Bridge datasets collected using the Q-Drone system, with trajectories generated using incremental smoothing method.



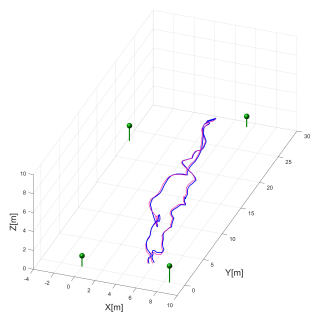
(a)



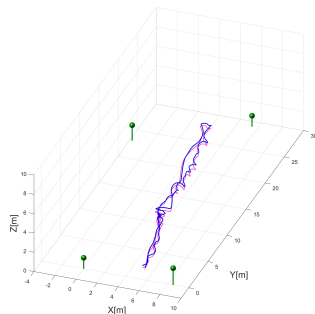
(b)



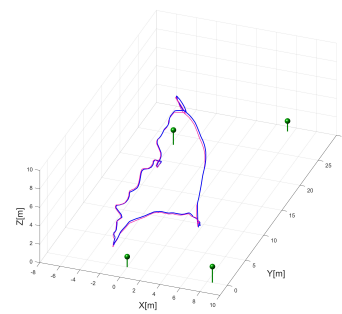
(c)



(d)



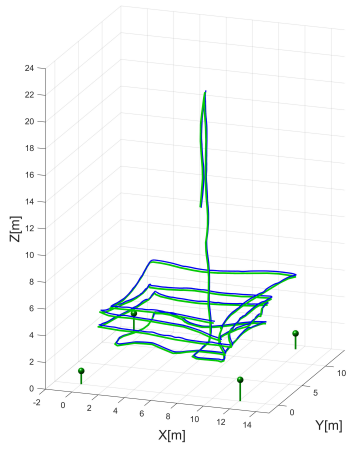
(e)



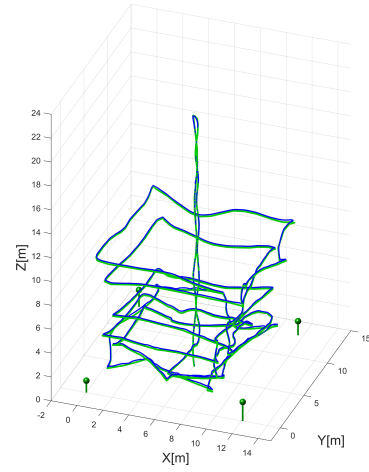
(f)

Figure A.15 Tunnel datasets collected using the Q-Drone system, with trajectories generated using incremental smoothing method.

A.4 DeepCovPG Result

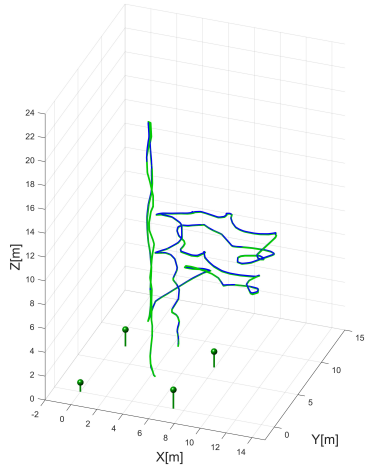


(a)

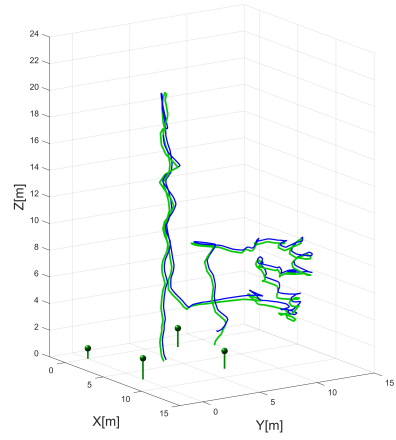


(b)

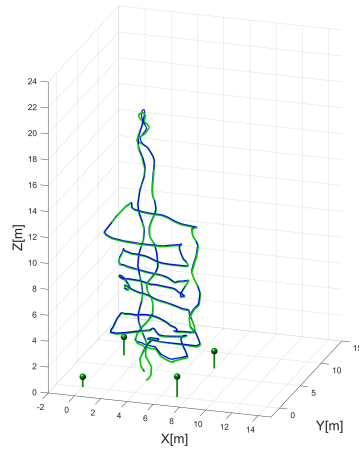
Figure A.16 Field datasets collected using the Q-Drone system, with trajectories generated using the DeepCovPG method.



(a)

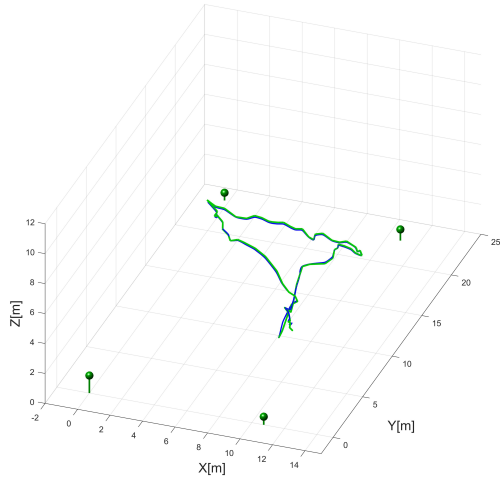


(b)

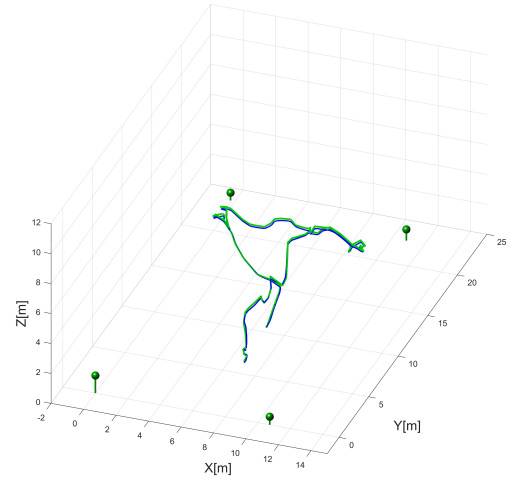


(c)

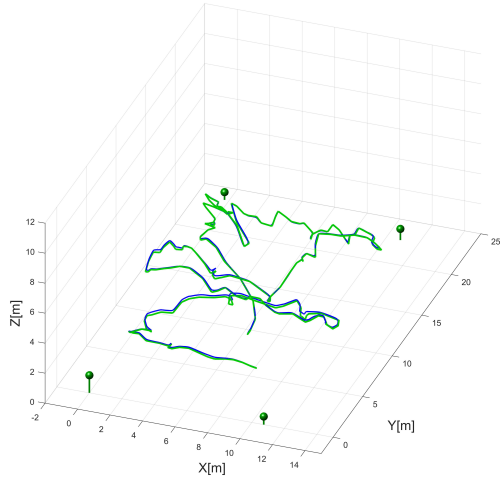
Figure A.17 Building datasets collected using the Q-Drone system, with trajectories generated using the DeepCovPG method.



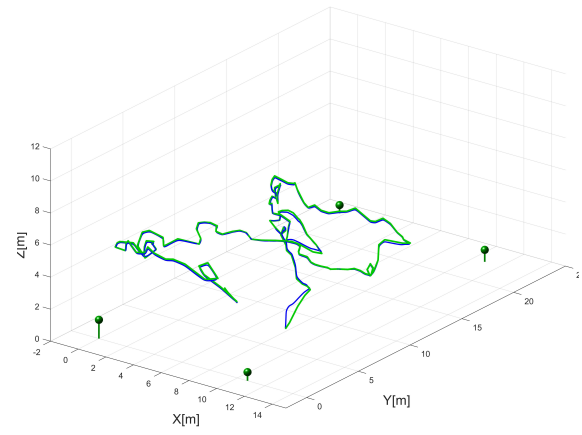
(a)



(b)

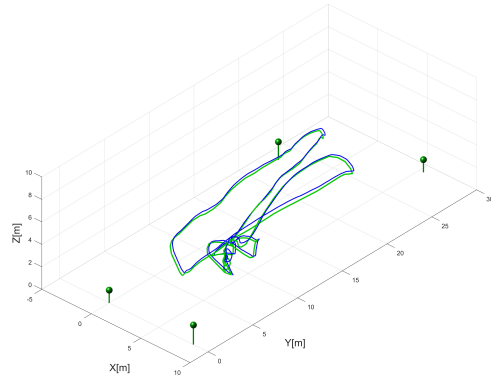


(c)

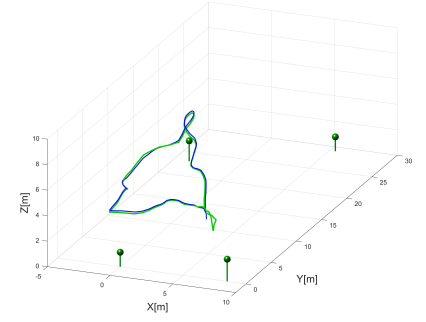


(d)

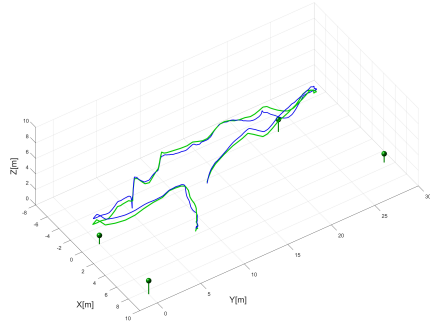
Figure A.18 Bridge datasets collected using the Q-Drone system, with trajectories generated using the DeepCovPG method.



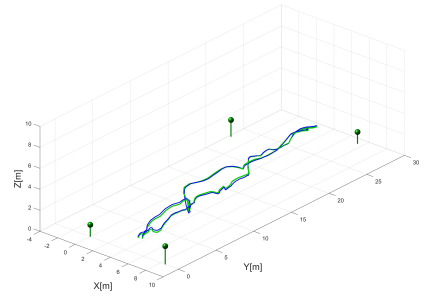
(a)



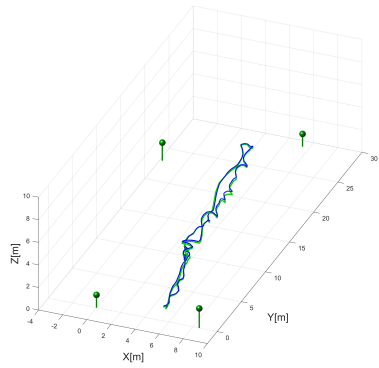
(b)



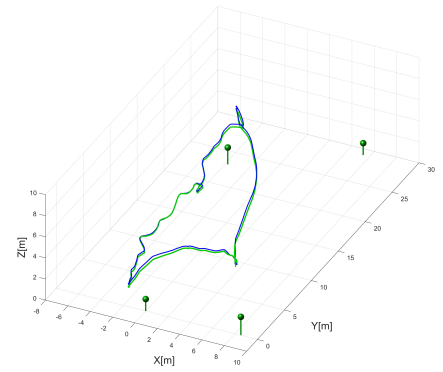
(c)



(d)



(e)



(f)

Figure A.19 Tunnel datasets collected using the Q-Drone system, with trajectories generated using the DeepCovPG method.

Appendix B

LiDAR Positioning Experimental Results Figures

This appendix includes figures showing the LiDAR positioning experimental results for the ICP-QE odometry method, DL-PE odometry and the ICP-QE plus DL-PE with the INAF method (6). These results do not include loop closure.

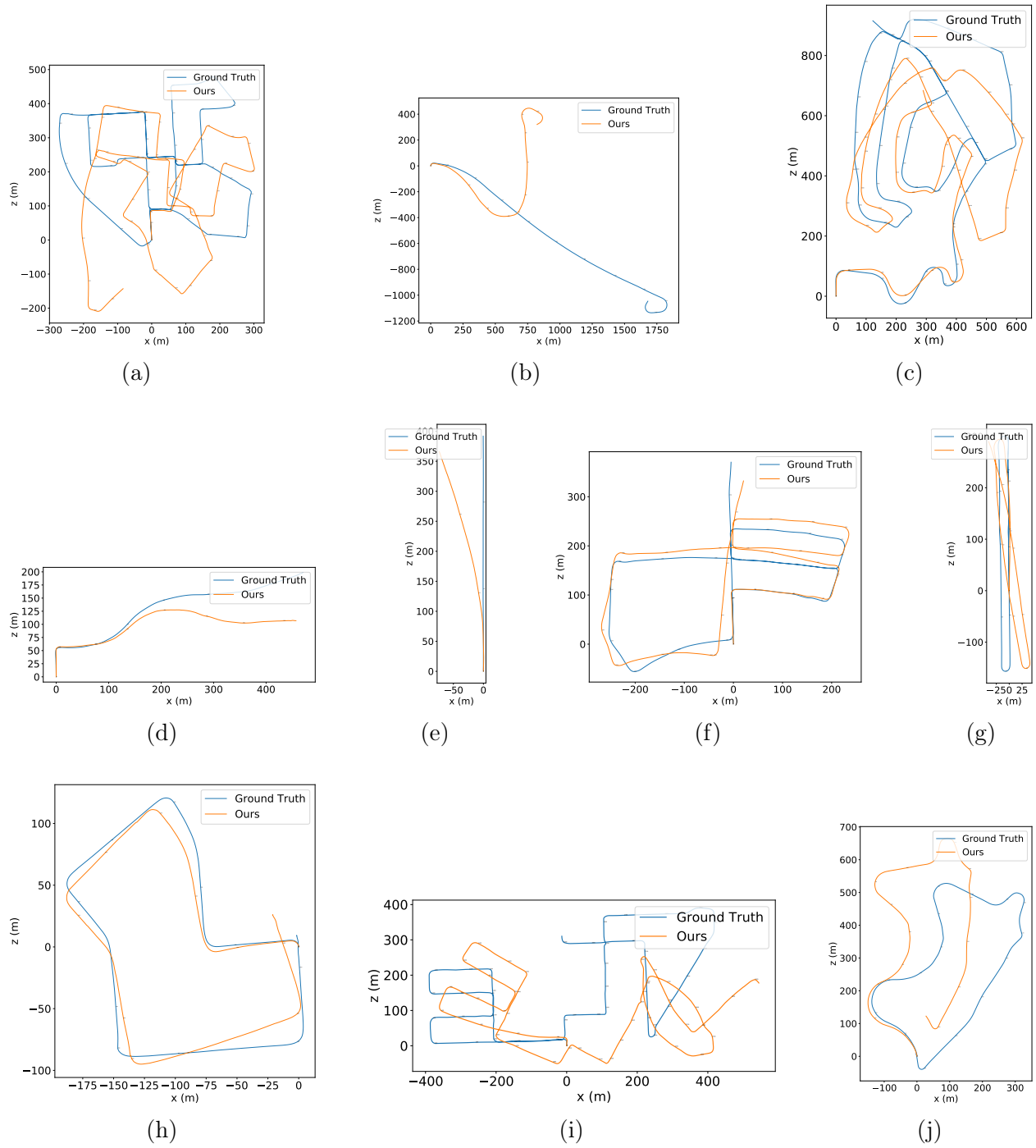


Figure B.1 The figure compares the results of ICP-QE odometry, and the ground truth poses from sequences 00 to 09 of the KITTI dataset.

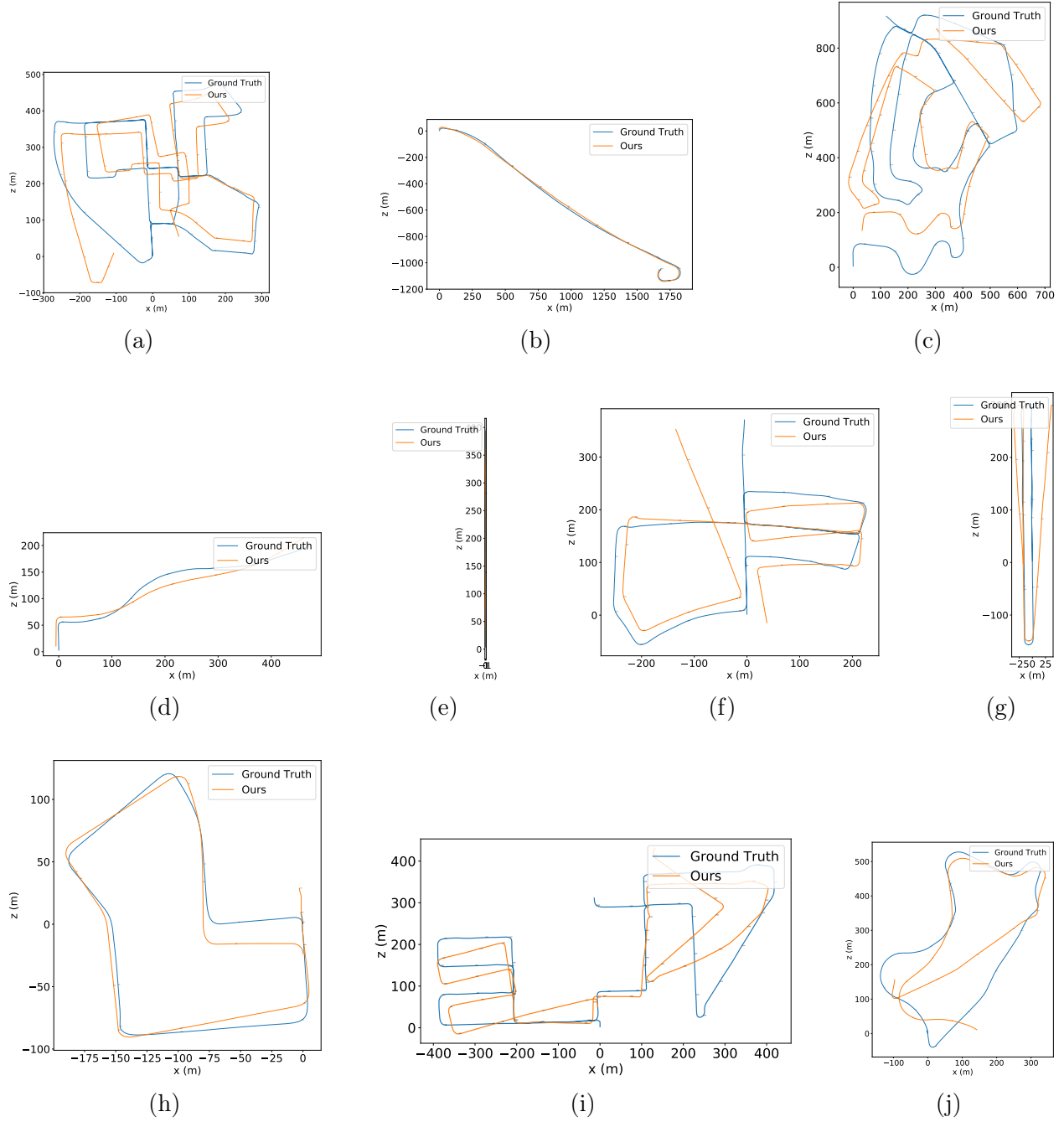


Figure B.2 The figure compares the results of DL-PE odometry and the ground truth poses from sequences 00 to 09 of the KITTI dataset.

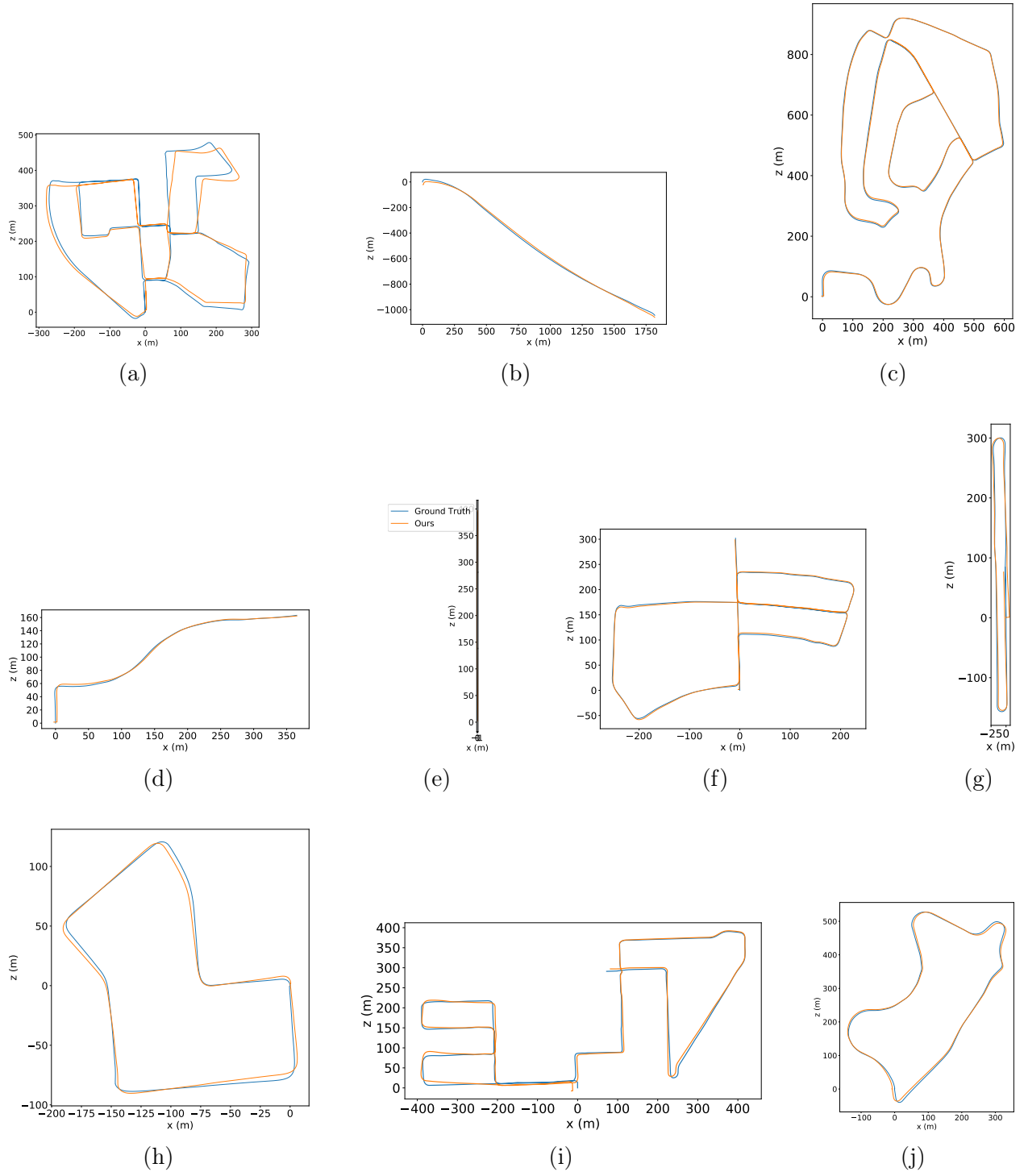


Figure B.3 The figure compares the results of INAF odometry, and the ground truth poses from sequences 00 to 09 of the KITTI dataset. Loop Closure included.