Visual inertial odometry and lidar inertial odometry for mobile robot

Henawy, John Farid Nasry

2021

Henawy, J. F. N. (2021). Visual inertial odometry and lidar inertial odometry for mobile robot. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/146364>

<https://doi.org/10.32657/10356/146364>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 19:23:04 SGT

VISUAL INERTIAL ODOMETRY AND LIDAR INERTIAL ODOMETRY FOR MOBILE ROBOT

NANYANG
TECHNOLOGICAL
UNIVERSITY

School of Mechanical and Aerospace Engineering

JOHN FARID NASRY HENAWY

A thesis

submitted to Nanyang Technological University

in the partial fulfilment of the requirements for the degree of

Doctor of Philosophy

Singapore

2021

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

8-2-2021

.....

Date

.....

John Farid Nasry Henawy

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

8-2-2021

.....

Date

.....

Assoc Prof Seet Gim Lee, Gerald

Authorship Attribution Statement

Several parts of Chapter 2, 3, 4, 5 and 6 are published in J. Henawy, Z. Li, W. Y. Yau, and G. Seet, "Accurate imu factor using switched linear systems for vio," IEEE Transactions on Industrial Electronics, 2020.

The contributions of the co-authors are as follows:

- The manuscript was revised and corrected by Prof Gerald Seet.
- Dr. Yau Wei Yun initiated the idea and provided the initial direction. Also, he revised the final manuscript.
- Dr. Li Zhengguo provided feedback during the problem formulation, experimental evaluation and revised the manuscript.
- I prepared the manuscript drafts. All the experimental studies were conducted by me.

Several parts of Chapter 1, 2 , 3 and 6 are published in J. Henawy, Z. Li, W. Y. Yau, G. Seet, and K. W.Wan, "Accurate imu preintegration using switched linear systems for autonomous systems" in 2019 IEEE Intelligent Transportation Systems Conference (ITSC). IEEE, 2019, pp. 3839-3844.

The contributions of the co-authors are as follows:

- The manuscript was revised and corrected by Prof Gerald Seet.
- Dr. Yau Wei Yun initiated the idea and provided the initial direction. Also, he revised the final manuscript.
- Dr. Li Zhengguo provided feedback during the problem formulation, experimental evaluation and revised the manuscript.
- I prepared the manuscript drafts. All the experimental studies were conducted by me.
- Dr. Kong wah revised the manuscript.

Several parts of Chapter 2 are published in L. J. Chen, J. Henawy, B. B. Kocer, and G. G. L. Seet, "Aerial robots on the way to underground: An experimental evaluation of vins-mono on visual inertial odometry camera," in 2019 International Conference on Data Mining Workshops (ICDMW). IEEE, 2019, pp. 91-96.

The contributions of the co-authors are as follows:

- Prof Gerald Seet provided the initial direction.
- Eng Chen prepared the manuscript drafts. All the experimental studies were conducted by him.
- I revised and corrected the manuscript. Also, I provides some directions during the experimental studies.
- Dr Kocer revised and corrected the manuscript. Also, he helped during the experimental studies

8-2-2021

.....

Date

.....

John Farid Nasry Henawy

Abstract

Municipal infrastructure inspection and maintenance are vital to keep the integrity and safe operation of large-scale infrastructure such as pipelines, bridges and tunnels. For example, regular inspection and assessment of tunnels are essential to prevent any possible leakage or worse, collapse. Inspection of deep sewerage tunnels is even more challenging because of their complex environment where such tunnels are not only difficult to reach, they may also have dangerous bacteria and microorganisms and the sewer gas would also have mixture of flammable and toxic gases such as hydrogen sulphide, methane, ammonia and sulphur dioxide. In Singapore, deep tunnel sewerage system (DTSS) is 6 *m* in diameter and located about 20 to 50 *m* underground.

Recently, quadcopters are widely used for inspection because of their ability to do vertical take-off and landing with high stability, reach to great height as well as ease of use and low cost. However, quadcopter localization in deep sewerage tunnel is very challenging as the environment there is dark, wet, featureless and structureless. The darkness and featureless environment causes significant visual degradation, while structureless and wet environment causes lidar degradation. Thus, multiple sensors are required for proper robot localization to mitigate such sensor degradation. Moreover, employing an inertial measurement unit (IMU) as an additional sensor can dramatically improve both reliability and accuracy of visual and lidar odometry. Therefore, data fusion paradigm has become the major focus for the visual inertial odometry (VIO) and lidar inertial odometry (LIO) algorithms. Among these, the filtering data fusion paradigm is a common fusion technique because it is fast and efficient for online estimation. However, it produces cumulative linearization error which will cause gradual deterioration in the accuracy.

In this thesis, a data fusion paradigm based on nonlinear optimization called a full smoothing paradigm has been proposed as a better alternative to the filtering paradigm. The full smoothing paradigm takes into consideration the full history of the previous states without marginalization. Therefore, it is more accurate than the filtering paradigms. However, high IMU sampling rate represents a great challenge for online estimation. This is because the full state history grows rapidly making online estimation not feasible. Motion preintegration overcomes this problem whereby the IMU measurements can be integrated into a single equivalent value between two visual or lidar keyframes. This makes the full smoothing paradigm simpler and the nonlinear optimization process can be done at a lower frequency.

In this work, a novel IMU motion integration model was proposed for the continuous-time of IMU kinematics which are modelled using a switched linear system. The novel IMU model has a closed-form solution. Based on that, a novel IMU factor was formulated as a closed-form discrete factor to compute the mean measurements, covariance matrix and Jacobians. As such, the proposed model is more accurate and more efficient for online state estimation than the state-of-the-art. To evaluate the proposed IMU factor, it is integrated with the VIO and LIO frameworks as tightly coupled joint-optimization. The proposed VIO has been evaluated using simulated and real-world datasets. In addition, indoor experiments have been done to test its effectiveness. Results obtained show that the proposed framework outperforms the state-of-the-art approach by up to 22% and 38% on real datasets and indoor experiments, respectively.

A novel nonlinear de-skewing is then proposed to correct the motion distortion between all scanned point in LIO. The novel IMU motion integration model is further used to correct the distorted points. In addition, the novel IMU factor is used with lidar factor in joint-optimization to obtain high state estimation. Results obtained show that the proposed model outperforms the state-of-the-art by up to 21%.

Acknowledgements

I would like to thank my supervisors, *Assoc. Prof. Gerald Seet Gim Lee*, *Dr Yau Wei Yun* and *Dr Li Zhengguo*, from whom I have learnt a lot and who were generous with their continuous and kind support throughout my PhD journey till now. They have also regularly assisted me with helpful suggestions and guidance. Their effort is deeply appreciated.

I would also like to thank my family and friends for constantly cheering me on and pushing me to go forward in this long and arduous PhD journey.

I gratefully acknowledge funding by the A*STAR SINGA for providing me with 4 years of Ph.D. scholarship.

Lastly, I would like to thank my God, Jesus Christ, who I could not live without Him and I could not have done it without Him.

Thanks to all of you.

Contents

Abstract	vi
Acknowledgements	vii
List of Tables	xiv
List of Figures	xvi
List of Abbreviations	xx
1 Introduction	1
1.1 Inspection Methods	4
1.1.1 Visual Inspection Method	4
1.1.2 Ultrasonic Inspection Method	5
1.2 Tunnel Inspection Systems	5
1.3 Motivation	8
1.4 Problem Statement	11
1.4.1 Discrete IMU factor problem	12
1.4.2 Motion Distortion Problem	15
1.5 Summary	17
1.6 Organization of Contents	18

2	Literature Review	20
2.1	General Background	20
2.1.1	Global Navigation Satellite System (GNSS)	21
2.1.2	Wheel Odometry	22
2.1.3	Inertial Navigation System (INS)	23
2.1.4	Acoustic Systems	24
2.1.5	Ranging System	24
2.1.6	Visual Systems	25
2.1.7	Concluding Remarks On General Background	26
2.2	Visual Odometry (VO)	27
2.2.1	Feature-Based Methods	27
2.2.2	Direct Methods	28
2.3	Visual Inertial Odometry (VIO)	31
2.3.1	Fusion Paradigm	32
2.3.2	Optimization Paradigm	33
2.4	Lidar Inertial Odometry	43
2.5	Concluding Remarks	45
3	Methodology	49
3.1	Preliminaries	51
3.1.1	Basic Knowledge on Skew Matrix	51
3.1.2	Taylor Expansion of Cosine and Sine	52
3.2	Problem Formulation on Visual Inertial Odometry	53
3.3	A Novel IMU Integration Model Using Switched Linear Systems	55
3.3.1	A Linear Kinematic Model	55
3.3.2	The Proposed IMU Integration Model	57
3.4	A Novel IMU Preintegration Measurements	64

3.4.1	Definition	66
3.4.2	Preintegrated Rotational Measurements	68
3.4.3	Preintegrated Velocity Measurements	69
3.4.4	Preintegrated Position Measurements	70
3.5	A Novel Iterative Noise Propagation	72
3.5.1	Preintegrated Rotational Noise	72
3.5.2	Preintegrated Velocity Noise	73
3.5.3	Preintegrated Position Noise	73
3.5.4	Iterative Noise Form And Covariance Matrix	74
3.6	A Novel Analytic Biases Update	76
3.6.1	Preintegrated Rotational Measurements via Bias Correction	76
3.6.2	Preintegrated Velocity Measurements via Bias Correction	77
3.6.3	Preintegrated Position Measurements via Bias Correction	77
3.7	A Novel Preintegrated IMU Factor	79
3.7.1	Jacobian of Residual Errors	79
3.7.2	Jacobians of Position Residual Error ($r_{\Delta P(i,j)}$)	80
3.7.3	Jacobians of Velocity Residual Error ($r_{\Delta V(i,j)}$)	81
3.7.4	Jacobians of Rotational Residual Error ($r_{\Delta R(i,j)}$)	81
3.8	A Novel Nonlinear De-skewing for Lidar Inertial Odometry	82
3.8.1	Proposed Motion Distortion Solution	83
3.8.2	A Novel Nonlinear De-skewing	84
4	Results and Discussion	87
4.1	Simulated Dataset	87
4.2	Real World Dataset (EuRoC Dataset)	93
4.2.1	Window Size 10 (W10)	96
4.2.2	Window Size 5 (W5)	101

4.2.3	Overall Evaluation On Odometry Metric Error	103
4.3	Lidar Inertial Odometry Evaluation	104
4.3.1	Predicted Odometry	105
4.3.2	De-skewing And LIO Evaluation	109
4.3.3	Mapping Evaluation	114
5	Experimental Setup	118
5.1	Platform Selection	118
5.2	Experimental Setup	122
5.3	Communication Scheme	123
5.4	Experimental Results	125
5.4.1	Arbitrary Trajectory	125
5.4.2	Eight Trajectory	128
5.4.3	Square Trajectory	132
5.4.4	Analysis on Computational Cost	133
6	Conclusion	135
6.1	Contributions	138
6.1.1	A Novel IMU Motion Integration Model	138
6.1.2	A Novel IMU Preintegration Model	138
6.1.3	A Novel Iterative Noise Propagation	139
6.1.4	A Novel Analytic Bias Update	140
6.1.5	A Novel Preintegrated IMU Factor	140
6.1.6	A Novel Nonlinear De-skewing for LIO	141
6.2	Findings and Limitations	141
6.2.1	A Novel IMU Factor For VIO System	141
6.2.2	A Novel IMU Factor For LIO System	144
6.3	Significance of The Study	146

6.3.1	Academic Significance	146
6.3.2	Engineering Significance	147
6.4	Future Directions	147
6.4.1	On The Way To Underground	147
6.4.2	Multi-Sensors Fusion	148
6.4.3	High Dynamic Environment	149
A	Least-Square Error Minimization	150
B	Cubic Skew Matrix	155
C	Maximum A Posteriori (MAP)	156
D	State-of-The-Art	158
E	The Detailed Derivation of Chapter 3	162
E.1	The Proposed IMU Integration Model	162
E.2	A Novel Iterative Noise Propagation	165
E.2.1	Preintegrated Rotational Noise	165
E.2.2	Preintegrated Velocity Noise	166
E.2.3	Preintegrated Position Noise	167
E.3	A Novel Analytic Biases Update	168
E.3.1	Preintegrated Rotational Measurements via Bias Correction	168
E.3.2	Preintegrated Velocity Measurements via Bias Correction	169
E.3.3	Preintegrated Position Measurements via Bias Correction	170
E.4	A Novel Preintegrated IMU Factor	171
E.4.1	Jacobians of Position Residual Error ($r_{\Delta P(i,j)}$)	171
E.4.2	Jacobians of Velocity Residual Error ($r_{\Delta V(i,j)}$)	173
E.4.3	Jacobians of Rotational Residual Error ($r_{\Delta R(i,j)}$)	174

F	The Relationship Between Two Consecutive Points	177
G	VIO Error Analysis On Different Sequences	179
G.1	V1_01_easy	179
G.2	V1_02_medium	181
G.3	V2_01_easy	183
G.4	V2_02_medium	185
H	LIO Error Analysis On Different Sequences	187
H.1	Fast 2	187
H.2	Med 1	189
H.3	Med 2	191
H.4	Slow 1	193
I	ROS Node Graph	195
I.1	VINS_Mono	195
I.2	CPI-Forster-Proposed	196
I.3	Loam	197
I.4	LIO_Mapping	198
I.5	LIO Proposed	199
J	Publication	200
J.1	Journal Publications	200
J.2	Conference Publications	200
J.3	Workshop Publications	200
	Bibliography	202

List of Tables

2.1	Summary of Visual Odometry (VO)	29
2.2	Summary of Direct Methods	31
2.3	The Most Common Filtering Approaches	35
2.4	Summary of VI-SLAM (Filtering Approach) Over Past 10 Years	36
2.5	Differences Between VIO Approaches	38
2.6	Summary of VI-SLAM (Smoothing Approach) Over Past 10 Years	39
3.1	Coefficients of matrices $E(\theta)$, $\Gamma(\theta)$, $\Lambda(\theta)$ and $\Phi(\theta)$	66
4.1	RMSE On Position and Orientation For Different IMU Sampling Rates Compared to CPI-1 and CPI-2 on Simulated Dataset	92
4.2	RMSE On Position and Orientation For Different IMU Sampling Rates Compared to Discrete Model on Simulated Dataset	92
4.3	Comparison On Estimation Timing Of Different IMU Factors	93
4.4	RMSE On Position For Different Real Sequences (Window Size 10)	100
4.5	RMSE On Position For Different Real Sequences (Window Size 5)	101
4.6	Overall Odometry Metric Error	103
4.7	RMSE On Position For Different Real Sequences With Different Motion Speeds (Predicted Odometry)	106

4.8	RMSE On Position For Different Real Sequences With Different Motion Speeds Compared to LIO model (Lidar Inertial Odometry Only Without Mapping)	110
4.9	RMSE On Position For Different Real Sequences With Different Motion Speeds Compared to LOAM and LIO_mapping (Mapping)	115
5.1	RMSE On Position and Orientation For Different Trajectories (Indoor Experiments)	132
5.2	Analysis on Computational Cost	133
6.1	Conclusion Of The Novel IMU Factor For VIO (Finding)	144
6.2	Conclusion Of The Novel IMU Factor For LIO (Finding)	145

List of Figures

1.1	Deep Tunnel Sewerage System Layout(DTSS)	2
1.2	Example of Some Tunnel Defects	3
1.3	Visual Inspection Robot	5
1.4	ROBINSPECT EC Layout	6
1.5	Sewer Robots	7
1.6	Example of DTSS Features	9
1.7	Motion Distortion in a Sweep of Lidar. (a) is a given Environment. Blue points in (b) stand for a distorted scan, and Red points in (b) are a motion corrected scan [1]	17
2.1	GPS satellite Constellation	22
2.2	Optical Encoder	23
2.3	Direct Methods	29
2.4	Sparse Method (Semi-Direct Method)	30
2.5	Robustness to Frame-to-Frame Motion	30
2.6	Two Major Fusion Paradigms	32
2.7	Difference Between Front-End and Back-End	33
2.8	Illustration of Keyframe Algorithm x_i and y_i represent the variable state and observed features, respectively	40
2.9	Example of General Concept of Factor Graph	40

2.10	Key Idea of The Incremental Smoothing	41
2.11	IMU High Sampling Rate Affect on Factor Graph Representation	42
2.12	Combine All IMU Measurements into Equivalent Value	42
3.1	The difference between the state-of-the-art and proposed approach assumptions	50
3.2	Motion distortions in a sweep of LiDAR. (a) is a given environment. Blue points in (b) stand for a distorted scan, and red points in (b) are a motion corrected scan.	83
4.1	Estimation trajectories against ground-truth (Simulated Dataset)	89
4.2	RMSE on position and orientation of the proposed model against closed-form preintegration theory models and discerte approach	90
4.3	Analysis on Position Error of the proposed model against continuous preintegration theory models and discerte approach	91
4.4	Raw Estimated Trajectory Against Ground-Truth (V1_03_difficult)	95
4.5	Estimation Trajectories Against Ground-Truth	97
4.6	RMSE and position analysis on V1_03_difficult Sequence for the proposed model and VINS_mono	98
4.7	RMSE and position analysis on V2_03_difficult Sequence for the proposed model and VINS_mono	99
4.8	Failed Estimation On (V1_02_medium)	102
4.9	Over All Odometry Metric Error	104
4.10	Estimation trajectory against ground-truth On Fast 1 Dataset (Predicted Odometry)	107
4.11	RMSE and position analysis on Fast 1 Sequence for the proposed model and LIO_Mapping (Predicted Odometry)	108
4.12	Estimation Trajectories Against Ground-Truth	111

4.13 RMSE and position analysis on Fast 1 Sequence for the proposed model and LIO model	112
4.14 RMSE and position analysis on Slow 2 Sequence for the proposed model and LIO	113
4.15 Loam Estimation trajectory against ground-truth On Fast 1 Dataset . . .	116
5.1 Some Different Drone Platforms	119
5.2 DJI F450 Platform	121
5.3 Experimental Setup	123
5.4 Communication scheme	124
5.5 Estimation Trajectories Against Ground-Truth (Arbitrary Trajectory) . . .	126
5.6 RMSE and Position Analysis on Arbitrary Trajectory for the Proposed Model and VINS_Mono	127
5.7 Estimation Trajectories Against Ground-Truth	129
5.8 RMSE and Position Analysis on Eight Trajectory for the Proposed Model and VINS_Mono	130
5.9 RMSE and Position Analysis on Square Trajecotry for the Proposed Model and VINS_Mono	131
A.1 Graphical Problem	151
D.1 Different sampling rates	160
G.1 Estimation Trajectories Against Ground-Truth (V1_01_easy)	179
G.2 RMSE and position analysis on V1_01_easy Sequence for the proposed model and VINS_mono	180
G.3 Estimation Trajectories Against Ground-Truth (V1_02_medium)	181
G.4 RMSE and position analysis on V1_02_medium Sequence for the proposed model and VINS_mono	182

G.5	Estimation Trajectories Against Ground-Truth (V2_01_easy)	183
G.6	RMSE and position analysis on V2_01_easy Sequence for the proposed model and VINS_mono	184
G.7	Estimation Trajectories Against Ground-Truth (V2_02_medium)	185
G.8	RMSE and position analysis on V2_02_medium Sequence for the proposed model and VINS_mono	186
H.1	Estimation Trajectories Against Ground-Truth (Fast 2)	187
H.2	RMSE and position analysis on Fast 2 Sequence for the proposed model and Lio	188
H.3	Estimation Trajectories Against Ground-Truth (Med 1)	189
H.4	RMSE and position analysis on Med 1 Sequence for the proposed model and Lio	190
H.5	Estimation Trajectories Against Ground-Truth (Med 2)	191
H.6	RMSE and position analysis on Med 2 Sequence for the proposed model and Lio	192
H.7	Estimation Trajectories Against Ground-Truth (Slow 1)	193
H.8	RMSE and position analysis on Slow 1 Sequence for the proposed model and Lio	194
I.1	Two VIO Systems Running Simultaneously Without Affecting Each Other	195
I.2	CPI ROS Nodes	196
I.3	Loam ROS Nodes	197
I.4	LIO_Mapping ROS Nodes	198
I.5	LIO Proposed ROS Nodes	199

List of Abbreviations

BA	Bundle Adjustment
CPI	Closed Form Preintegration Theory
DST	Deep Sewerage Tunnel
DTSS	Deep Tunnel Sewerage System
DGNSS	Differential Global Navigation Satellite System
EKF	Extended Kalman Filter
FHWA	Federal Highway Administration
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GV	Ground Vehicle
HDR	High Dynamic Range
ICP	Iterative Closest Point
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
KF	Kalman Filter
LIO	Lidar Inertial Odometry
LOAM	Lidar Odometry And Mapping
MAP	Maximum A Posteriori
MAVs	Micro-Aerial Vehicles
MEMS	Microelectromechanical Systems
MSCKF	Multi-State Constrained Kalman Filter
MSE	Mean Square Error

NDI	Non-Destructive Inspection
OKVIS	Open Keyframe-based Visual Inertial SLAM
RTK	Real Time Kinematics
RC	Remote Control
RF	Radio Frequency
RK	Runge-Kutta Method
RMSE	Root Mean Square Error
ROS	Robot Operating System
RS	Ranging Systems
SLAM	Simultaneous Localization And Mapping
SVO	Semi-Direct Visual Odometry
UAV	Unmanned Aerial Vehicles
VIO	Visual Inertial Odometry
VI-SLAM	Visual Inertial SLAM
VO	Visual Odometry
VS	Visual System
V-SLAM	Visual SLAM

Chapter 1

Introduction

The basic physical and organizational structures and facilities (e.g., buildings, roads, and power supplies) needed for the operation of a society or enterprise are called infrastructure. Civil infrastructure is a system supporting human activities. Civil infrastructure represents one of the essential foundations in developed countries. These systems could include the design, analysis and management of electric power, oil and gas, water and waste-water, transportation, communication systems and building that make up urban and rural communities. Accordingly, to ensure the safe operation of civil infrastructure, their regular inspection and maintenance are vital. This is especially needed for the aging large-scale infrastructure such as pipelines, bridges, and tunnels to ensure the integrity of such systems is still intact. In the case of tunnels, they exist in different shapes and types (railway, transport, sewage tunnels, etc.). The number and length of these tunnels are larger in the more developed countries. In general, most of these tunnels are located underground which is an unpleasant hostile environment for human operation. There are also hazards to human health including lack of proper ventilation, visibility and acceptable noise level.

One of the most important tunnels used in big cities such as Abu Dhabi [2], and developing countries such as United States of America [3], China [4] and Singapore [5] is the Deep

Figure 1.1: Deep Tunnel Sewerage System Layout(DTSS)

Sewerage Tunnel (DST). They are mainly used to collect sewage waste in one place and channel it for treatment elsewhere. This operation is essential for countries which have limited water resources such as Singapore. Therefore, the Deep Tunnel Sewerage System (DTSS)¹ has been established in Singapore which consists of two phases. The first phase is 48 *km* long while the second phase is 31 *km* long located about 20 to 50 *m* underground. The tunnel is 6 *m* in diameter, and it uses the gravity force to direct the water to three different reclamation plants as shown in Figure 1.1 [5].

The deep tunnel sewerage system environment has the same general characteristics as the other tunnels (e.g., railway, transport, etc.). In general, it is just an unpleasant environment for human operation where there is no proper ventilation, poor visibility, and possibly noisy. Additionally for the DST, there are presence of dangerous bacteria and microorganisms due to the nature of the environment. Besides the biological hazards, there are also toxic sewer gases from sewage such as hydrogen sulphide [6]. Eye irritation, breathless and queasiness are the most common symptoms of toxic gases exposure. Moreover, a human could die of asphyxiate if exposed for a prolong period to high

¹<https://www.pub.gov.sg/dtss/about>

concentration of these toxic gases.

On the other hand, the deep tunnel sewerage system can deteriorate over time because of a lot of factors such as ageing, poor maintenance, and change of environmental conditions. Therefore, proper maintenance process with regular inspection and assessment of the deep tunnel is required to prevent any possible leakage and collapse. However, the nature of the DTS environment represents a significant challenge for tunnel inspection because of the safety risk and hazard to human health. Consequently, automated tunnel assessment and inspection has generated great interest among the researchers in the last decade to develop an automated system for tunnel inspection [7, 8].

Proper inspection depends on reliable identification of defects. To aid this, the US Federal Highway Administration (FHWA) has classified the tunnel defects into different groups as mentioned in the TOMIE manual [9]. According to the manual, there are four major groups of tunnel defects. These are classified into concrete structures, steel structures, masonry structures and timber structures. However, the most common defects found in a tunnel in Singapore are related to the group of concrete structures such as spalling, cracks and leakage as shown in Figure 1.2. As such, the inspection methods must be able to detect and identify these common defects [10].

(a) Spalling

(b) Cracks

Figure 1.2: Example of Some Tunnel Defects

1.1 Inspection Methods

There are several methods available to detect the presence of defects in a tunnel. These methods can be classified into two major groups, namely the destructive methods and non-destructive methods. Destructive methods could have a negative effect on the concrete structure. Moreover, they are done manually and is therefore labour intensive. However, they are very accurate and able to also perform measurement of the ground condition surrounding the tunnel. Due to their destructive nature, these methods are not commonly used. The Non-Destructive Inspection methods (NDI) use different sensors such as visual, magnetic, thermography, ultrasonic, ground penetrating radar etc. to determine the properties of the tunnel and the ground surrounding the tunnel. Due to their non-destructive feature, they are more commonly used for automated inspection of tunnels. In this project, we will focus on the use of visual and ultrasonic sensors to identify the most common tunnel defects as mentioned before [11].

1.1.1 Visual Inspection Method

Visual method is considered the most important inspection method because it can obtain rich textural information with high resolution and the cost of the sensor is low. The information gathered will be analyzed and used to give an initial assessment of the tunnel's condition. Where required, other follow-up tests can then be performed to verify further the condition of the tunnel. The gathered information can be automatically processed using computer vision algorithms to detect tunnel's defects such as presence of crack, and the crack size such as length and thickness [12, 13].

For example, [14] used a ground robot with a CCD camera to scan for the presence of cracks on the wall automatically. The camera was mounted on a shock absorbing base to provide good stabilizing images. Furthermore, the ground robot has its own light source to provide good illumination for effective visual inspection. Based on the images taken,

the visual inspection system provide further details about the defects including cracks thickness, orientation and length. The inspection robot is shown in Figure 1.3.

1.1.2 Ultrasonic Inspection Method

Visual method alone is not sufficient for the inspection process because it cannot check the internal concrete structure. To do this, an ultrasonic sensor can be used. It generates ultrasonic pulses and by measuring the velocity of the pulses, it can provide information about the concrete's structural material. This is possible because the velocity of the pulse depends on the concrete's elasticity [15]. This would give indication of the depth of the crack and the density of the concrete to provide an overall picture of the overall structural strength when crack is detected. Additionally, it should be pointed out that the ultrasonic sensor requires that the sensor is in contact with the tunnel wall for effective inspection [16].

1.2 Tunnel Inspection Systems

Tunnel inspection systems are typically robotic systems with integrated inspection sensors for detecting tunnel defects. A fully autonomous system is preferred for deep sewerage

Figure 1.3: Visual Inspection Robot

1.2. TUNNEL INSPECTION SYSTEMS

tunnel inspection due to the safety and health risk and the confined nature of the tunnel [17]. Unfortunately, based on our study, there is no fully autonomous system available yet. Almost all of these inspection systems employ robotic system mounted with sensors and then record or send the data gathered for processing externally. These inspection process will still require human intervention [18, 19]. Accordingly, the ROBINSPECT EC project, which was supported by the European Union, was initiated to develop an autonomous tunnel inspection problem. It has an ultrasonic sensor and a camera [20] and the proposed design of the robotic system is shown in Figure 1.4.

Even though this project will overcome the autonomy problem, it cannot be used inside the sewage tunnels for inspection purposes because of the circular shape of the sewage tunnels. Their proposed mobile vehicle require a flat ground to move and to properly position the outriggers. The outriggers are usually placed at the four sides of the vehicle to increase the overall footprint to ensure the mobile vehicle is stable when the crane of the mobile vehicle holding the robotic manipulator is extended to a high position. The mobile vehicle will also require a counterweight to avoid toppling. Therefore, the overall weight of the proposed system is very high and may cause stress to the sewage tunnel which is usually not designed to carry heavy vehicles. As such, the proposed system is more suitable for inspecting motorway and railway tunnels instead of sewage tunnels.

Figure 1.4: ROBINSPECT EC Layout [20]

Figure 1.5: Sewer Robots [21]

On the other hand, there are also sewerage robots to inspect sewerage system [21]. However, these robotic systems are not suitable for deep tunnel sewerage system inspection because their size is small. Furthermore, sewerage robots are not fully autonomous, and are usually equipped with the visual inspection methods only as shown in Figure 1.5.

Thus, another system is needed which can work in the deep tunnel sewerage system environment. One possible choice for inspection system is to look into the use of Unmanned Aerial Vehicles (UAV). UAVs have attracted the attention of the research community and have been widely used recently. UAVs can be used in many different military and civilian applications, such as photography, border security, surveillance, search and rescue, forest fire monitoring, pipeline inspection, solar plant inspection and maintenance and volcano observation [22, 23]. Recently, researchers focused on the development and use of Micro-Aerial Vehicles (MAVs), made possible by the advancement in Microelectromechanical Systems (MEMS), control theory, computing and communication technologies. One of the most important MAVs is the quadcopter or drone because of its relatively low cost, high stability, easy to control and widespread availability. Moreover, they can be easily deployed without the need of a large space as they are able to perform vertical take-off and landing [24]. Even though the flight duration is considered as one of the quadcopter's major shortcomings, this can be overcome by using drone tethered

to the Ground or Surface Vehicle with a power cable or having a charging station on the Ground or Surface Vehicle (GV).

A recent project called the "Autonomous Inspection of Critical Infrastructure using Micro Aerial Vehicles" was started by Professor Vijay Kumar at the University of Pennsylvania about four years ago. This project uses the quadcopter for the inspection process with all onboard sensors as well as a light source such as small LED torch to overcome the darkness problem. Moreover, the drone can localize itself inside the tunnel using lidar and camera sensors.

In order to develop an autonomous system, the system have to fulfill two necessary requirements. Firstly, the drone must localize itself during the inspection process with respect to the available features of the environment. Secondly, the drone must have an autonomous control system [25]. From this brief introduction of deep tunnel inspection and the current state of inspection systems, *it can be emphasized that solving pose estimation (drone localization) problem as well as autonomous control system problem is the main requirement in developing an autonomous system which can work in this environment.*

1.3 Motivation

The primary motivation for this research is to provide an autonomous system for inspection purposes inside deep tunnel sewerage system (DTSS) in Singapore. The inspection system must have a good control system and can localize itself inside the tunnel. The autonomous control problem is being researched on by my colleague who has proposed the use of a drone with a manipulated arm [26, 27]. The main reason for using a drone with a manipulated arm is that the drone can carry the ultrasonic sensor which must be attached to the wall as mentioned before [28]. Thus, a high level control system is required to achieve the required stability for such a drone. Therefore,

Figure 1.6: Example of DTSS Features

this research excludes the autonomous control problem and instead focuses on solving the localization problem. The tunnel environment represents a significant challenge because it is usually dark, wet and texture-less with a homogeneous structure. Accordingly, the main motivation of this research is to solve the pose estimation problem (Agent Localization) under these environmental challenges so that the drone can navigate autonomously in the tunnel to perform the inspection process.

The easiest and incipient way for pose estimation is using the Global Positioning System (GPS). However, the GPS signal is not available underground. Alternatively, pose estimation can be obtained by integrating both the linear acceleration and angular velocity of an Inertial Measurement Unit (IMU) sensor assuming that the discrete outputs are constant between two successive sampling measurements. Initially, IMU was used in the aerospace community [29]. However, IMU measurements are associated with noise and bias. Consequently, the error will be accumulated because of the integration process. Thus, there will be drift in the pose estimation [30] which increase with distance moved. So, aided Inertial Navigation System (INS) is used to compensate the IMU measurements error using the GPS information. Unfortunately, GPS is not available inside the deep sewerage tunnel [31].

On the other hand, visual navigation or Visual Odometry (VO) is a perfect choice to

1.3. MOTIVATION

---

solve the pose estimation problem because camera provides abundant information at low cost. However, visual odometry depends on environment features. In other words, if rich and unique features are not observed, visual odometry cannot be used. Fortunately, the deep sewerage tunnel has some features on the wall which can still be used as a landmark for visual odometry. Features on the tunnel's wall are shown in Figure 1.6. Although the features exist, however when the features are not observable for a short period in tunnel space, because of the nature of the tunnel environment, the visual odometry will fail as well. This could happen because of low light, change in illumination conditions, or presence of fog due to gas evaporation. One more reason that makes the features not observable is when the drone is flying very close to the tunnel walls. As aforementioned reasons, pose estimation using visual odometry only will not be robust choice [32].

Fortunately, IMU measurements could assist in this issue when the features are not observable. By employing IMU measurements with camera information, the dynamic behaviour of the system could be estimated for a short time until the features are observable again. Using IMU and camera sensor together is called Visual Inertial Odometry (VIO). So, the camera and IMU are complimentary with one another. In other words, the IMU measurement errors will be compensated using the abundant information from the camera. Even though employing IMU measurements increases the robustness of the pose estimation, the homogeneous structure of the tunnel walls could be an issue for pose estimation using visual inertial odometry. This is because vision-based methods depend on the presence of rich and unique features. Thus, a vision-based method only is insufficient for pose estimation inside the tunnel environment.

Another way to obtain the pose estimation is to use ranging methods such as lidar. Lidar is capable of providing high accuracy pose estimation, and is reliable and does not require lighting. However, lidar can suffer from the wet environment especially if there is fog or high vapour content and the presence of mud that covers the wall. Another issue with the use of lidar is that it can suffer from the structureless environment such as along axis of

the tunnel. Thus, using the range methods (lidar) only without other additional sensors in this environment are not suitable for this application because lidar *cannot localize the drone along the tunnel axis (X-axis)* [33]. However, to take advantage of the high accuracy and reliability of the lidar, lidar could be used to localize the drone within the other two axes (Y-axis and Z-axis) of the tunnel.

Accordingly, the aforementioned discussion can be summarized into the following three key points:

- Vision-based methods will suffer from darkness, texture-less environment and lighting variation conditions.
- Range-based methods will suffer from wet environment, mud along the wall or structure-less environment such as a long tunnel axis.
- Combination of vision-based and range-based methods is the best way to estimate the drone localization.

In other words, the pose estimation which can be obtained using one localization technique is not reliable. However, to increase its estimation reliability, it is preferable to estimate the pose using fusion of multiple sensing systems (Inertial - Visual - Ranging). *Thus, the main motivation of this research is to develop a robotic system which can localize itself with respect to the features in the environment using multiple sensing systems.*

1.4 Problem Statement

The problem statement can be classified into two main problems which are driven and raised by the primary research objective and motivation. These two main problems can be identified as follows:

- Discrete IMU factor problem.
- Motion distortion problem.

1.4.1 Discrete IMU factor problem

As aforementioned in the motivation section 1.3, the VO is not robust and reliable enough to be used inside the tunnel environment. Moreover, initialization and estimation accuracy could be another problem. Thus, the VO problem could be defined in terms of robustness, initialization and accuracy of the VO approaches. Regarding robustness, the robustness and the reliability could be dramatically improved by employing an inertial measurement unit as an additional sensor. The high dynamic change in the system is taken into consideration using the inertial measurement unit (IMU) with the condition that proper initialization can be obtained. With IMU, the robustness and reliability of VO are improved [34]. On the other hand, the IMU errors and bias will be compensated using the visual measurements so that the accuracy will be improved as well. Regarding initialization, proper initialization could be obtained by using traditional techniques such as those mentioned in [35] and [36]. With the aforementioned reasons, *this research excludes the robustness and initialization problem while it focuses on how to obtain a high accuracy state estimation in real-time.*

Regarding accuracy, the accuracy of the VO is improved by employing an IMU into the visual system (called VIO), because the IMU could help to observe and take into consideration the kinematic of the system when the visual measurements are not observed for a short time [37]. For example, when the robot is too near to the wall or inside a narrow corridor, the environment is dark or in the presence of smoke or fog. Thus, the accuracy could be improved in the aforementioned cases. The accuracy of the VIO system depends on the fusion data problem from different sensors. Fusion data problem generally can be categorized into two main classifications: fusion approach and optimization approach [38].

By using the fusion approach, the fusion data methods could be classified into two major categories. First, loosely coupled fusion data where each state estimation from a different

sensor is done separately. Then, one sensor data is used to update the other in order to obtain the state estimation. For example, the loosely coupled considers the visual odometry as a black box and does not update the odometry with IMU measurements while it estimates the pose [39, 40]. The second category is called tightly coupled where the state estimation is done using all data sensors at the same time. Tightly coupled methods are more accurate than the loosely coupled, but they are more computationally intensive than the loosely coupled one [41, 42]. However, advanced computer technologies make the tightly coupled methods feasible are the focus of the research presently.

According to the optimization approach, data fusion techniques could categorized into filtering methods, fixed-lag smoothing methods or full smoothing methods. Currently, real-time implementation uses the filtering methods over full smoothing methods for data fusion because it is fast and efficient for online-estimation. The reason is that the filtering methods only consider the previous state of the robot and marginalized all the other historical robot states. However, the error is cumulative over some time period as a result of the marginalization. On the contrary, full smoothing methods smooth all the robot trajectory over time, thus they are more accurate than the filtering methods. However, they are computational costly and is not feasible for real-time estimation. Thus, previous researchers try to decrease the computational cost of the full smoothing methods so that it could be used in real-time [43]. Until now, the greatest challenge is the trade-off between computational complexity and accuracy.

Accordingly, full smoothing methods and tightly coupled approaches give a higher state estimation in terms of accuracy and robustness. Typically, for high computational cost of the full smoothing methods, the sliding window-based is used in the optimization approach. Thus, tightly coupled with sliding window-based approach tries to minimize the residual errors of all the measurements. In the VIO system, there are two types of residuals errors. First, the IMU residual measurements which are called *IMU factor* and second, the visual residual measurements which are called *Visual factor* [44]. In other words, a

maximum posterior estimation is obtained by minimizing the residual errors of both IMU and visual factors. One easy way to represent and formulate this optimization problem is the factor graph which is a graphical model [45, 46]. An alternative terminology for using a factor graph to solve the sliding window optimization problem is called graph-based optimization. So, from now on, this terminology will be utilized and expressed in this research.

The graph-based optimization approaches are typically performed using the least-square error minimization approach [47]. Once a new measurement is observed or obtained, the graph-based optimization will perform the optimization by minimizing the residual error of both the IMU and visual factors. However, the high sampling rate of the IMU represents a great challenge for online state estimation because the trajectory grows very fast and becomes intractable. To solve this problem, Lupton *et al.* [37] established the pre-integration theory based on the fact that the IMU has a higher sampling rate than all the other sensors used in the motion estimation problem. The pre-integration theory means that the IMU measurements are integrated into a single value between two keyframes. This means that graph-based optimization can be performed at a lower frequency. In other words, the optimization process will be conducted only when the graph-based optimization receives a new measurement from the camera. Thus, the computational cost will be reduced, and the graph-based optimization approaches could be used for online state estimation because the graph-based optimization has been simplified.

Accordingly, this research focuses on how to use graph-based optimization methods for data fusion to obtain high accuracy state estimation in real-time. Chapter 2 describes the state-of-the-art in this area. As mentioned before, the VIO problem consists of the IMU factor and visual factor, and as the tunnel environment is usually dark, featureless and there is a possibility of the presence of fog, there is a high chance of the visual sensor degradation happening. Thus, the system must provide a high accuracy state estimation using IMU measurements until the visual measurements are observed again.

From that point of view, this research focuses and relies on the *IMU factor*. Moreover, the IMU factor has a great effect on state estimation because of its hidden noise and bias. Essentially, graph-based optimization is a nonlinear optimization process because of nonlinearity concerning IMU noise and bias. However, the nonlinear optimization problem is difficult to solve because the least square error function is complex and there is no closed-form solution [48]. However, numerical approaches could be used to solve the nonlinear optimization problem where the squared error function is linearized using the Taylor expansion [49] around a good initial guess point. This requires an accurate Jacobians estimation. Second, the estimation of the information matrix (i.e. the inverse of the covariance matrix) for the squared error. Therefore, the optimization problem depends on three crucial factors. First, the prediction value (i.e. mean value) of the optimization process where the linearization can be obtained; second, the estimation of the Jacobian matrix; and third, the estimation of the covariance matrix which represents the weighted values in the optimization process. For further details on the least-square error minimization, the reader is referred to Appendix A.

The state-of-the-art shows that the prediction value, covariance matrix and Jacobians have been obtained using discrete methods. The discrete methods hinder the accuracy of the state estimation. From this short note, *the academic research problem can be stated as follows; to obtain an accurate IMU factor in such way that it has good predictive value or good initial guess (mean value) for the optimization process as well as to find a closed-form solution of both the Jacobian and covariance matrices for IMU factor*. Thus, a high accuracy motion estimation could be obtained in real-time [50, 51].

1.4.2 Motion Distortion Problem

One key advantage of using lidar for pose estimation problem is that lidar can provide high-frequency range measurements with good accuracy. Furthermore, it is insensitive to ambient lighting and visual texture in the scenes. Lidar rangefinder is one of the

most widely used sensors for indoor position tracking among all available sensors. It is also commonly used in autonomous vehicles and outdoor robots. Pose estimation of rangefinder can be estimated by matching two different scans [52]. Iterative Closest Point (ICP) algorithm is one of the dominant solutions for the scan matching problem by iteratively finding the closest points and geometrically aligning the 3D points [53].

The motion of a scanning device is usually assumed to be very small compared to the scanning frequency of the ICP algorithm. However, as the rangefinder itself is mounted on a robot and thus is moving when performing scanning, this causes motion distortion among all points obtained in a sweep. The motion distortion caused by the movement of the robot was addressed by Minguez *et al.* [54]. They assumed that the sensor measures the distance to the same object between two frames at the same angle. Nevertheless, this algorithm is not applicable when the robot rotates and moves fast. An example is illustrated in Figure 1.7. Black line at Figure 1.7(a) represents an environment and the movement of the rangefinder is indicated by the arrow. Blue points at Figure 1.7(b) are the raw scan data. Obviously, the set of the point is distorted.

Moreover, the aforementioned assumption could be an issue when the motion is complex such as when the robot with the lidar needs to move forward in straight line and then perform a sharp turn to orientate itself and to be in close proximity to the wall. Such movement will cause motion distortion. Thus, *an accurate Lidar Odometry requires knowledge of the lidar pose during continuous laser ranging. Fortunately, the knowledge of the lidar pose can be provided by an IMU.*

On the other hand, the IMU-based odometry integrates small incremental motions between two successive sweeps. Thus, it suffers from drift and necessary attention is required to reduce the drift using an ICP algorithm. The drift was not addressed in [55] even though they discussed about having assistance from an IMU in the experiment.

Another issue related to the real-world scenarios is that lidar may only receive a few usable points due to sensor degradation especially in the narrow corridor which represents the

Figure 1.7: Motion Distortion in a Sweep of Lidar. (a) is a given Environment. Blue points in (b) stand for a distorted scan, and Red points in (b) are a motion corrected scan [1]

same environment of the deep tunnel. This is because only a few points can be observed from the ceiling and ground, while most of the received points are from the side walls. That leads to ill-constrained pose estimation. Another challenge is that the lidar provides low update rate or measurements which causes great challenge for applications that require a fast feedback pose estimation for the controller [56].

For these reasons, this research focuses on developing an ICP algorithm by employing the IMU measurements to correct the distortions due to the movement of the robot. In addition, the proposed *IMU Factor* will be employed with the lidar as a tightly coupled fusion to obtain high accuracy motion estimation. The state of the art and the conclusion to this problem will be mentioned and discussed in chapter 2.

1.5 Summary

It can be summarized that, in this thesis, the UAV was proposed to perform autonomous inspection task in DTSS which has harsh environment. To enable autonomous operation, this system should localize itself with respect to the features of the environment. Moreover, it must have a reliable and robust autonomous control system. Based on

the aforementioned objective, it can be concluded that the main motivation of this thesis work is to develop a system that could localize itself with respect to the features in the environment using multiple sensing systems, i.e., inertial, visual and lidar systems.

The research problem was formulated according to VIO and LIO. The VIO system consists of the IMU factor and the visual factor. The IMU factor has been obtained under specific assumptions. These assumptions hindered its accuracy due to discretization. On the other hand, LIO suffers from motion distortion in the scanned points during the robot movement.

In this thesis, an accurate IMU factor was proposed based on practical assumptions and derived the IMU factor as a closed-form solution, including the mean value, the covariance matrix and the Jacobians. Therefore, the proposed IMU factor is able to achieve better accuracy at lower computational complexity than the state-of-the-art. Moreover, the proposed IMU factor has been adopted to correct the distortion caused by the robot or UAV movement. The performance of the proposed VIO and LIO were evaluated by using public dataset and real world experiments. Experimental results obtained show that the proposed VIO and LIO algorithm outperforms the state-of-the-art algorithms.

1.6 Organization of Contents

The report is organized as follows. Chapter 2 provides a general overview and background work by discussing the general concept of VO methods followed by literature review on VIO. In addition, literature review was also done on the LIO and the available methods in the literature to compensate for motion distortion.

Chapter 3, titled Methodology, introduces in full detail the proposed approach. Basic motion integration model under specific assumption is introduced which is used to determine motion integration between two consecutive frames followed by finding the Jacobians and covariance matrices in closed form representation for accurate motion

estimation. In addition, this chapter also described the proposed approach to compensate for the motion distortion caused by the robot's movement.

Chapter 4 shows the experimental results of the proposed approach and compared it with the state-of-the-art using simulated and real-world datasets. This is followed by discussion and concluding remarks of the experimental results obtained. Subsequently, the experimental results of employing the proposed approach into LIO and evaluated on real-world datasets is being described and compared with the other state-of-the-art results for lidar odometry.

Chapter 5, titled by Experimental Setup, details the prototype system setup developed, the considerations made in choosing the MAV platform and the real-world experimental setup in the lab to comprehensively evaluate the proposed approach in solving the localization challenge faced in DTSS. Lastly, conclusion and recommended future works are stated in Chapter 6.

Chapter 2

Literature Review

In this chapter, the literature review of mobile robot localization will be stated. The literature review is divided into four main sections. The first section provides general background on mobile robot localization in section 2.1. The second section regards the visual odometry approaches is described in section 2.2. This will be further sub-sectioned into two; i.e. the feature-based method and the direct method. The third section describes the past works on visual-inertial odometry in section 2.3. In this section, VIO literature will be further classified into three major approaches; (i) filtering approach; (ii) fixed-lag approach; and (iii) full smoothing approach. Section 2.4 describes the literature on lidar inertial odometry. Finally, the concluding remarks will be given at the end of this chapter.

2.1 General Background

One of the fundamental challenges in autonomous navigation, obstacle avoidance and motion tracking is mobile robot localization. *Mobile robot localization is knowing the robot's position over the travelled distance and keep updating its position constantly.* The technology that describes the mobile robot localization is called Simultaneous Localization And Mapping (SLAM). SLAM initially was proposed by Smith [57, 58]. The main purpose

was to build a real-time mapping for an unknown environment using sensor information and to localize itself in such environment. This was revised by Durrant-Whyte and Bailey [59, 60] in 2006. SLAM consists of two parts; (i) mobile robot localization, (i.e. the location of the robot); and (ii) environment mapping (i.e. the feature's location in the environment). Both mobile robot localization and environment mapping are inter-dependent (i.e. the robot uses the salient features of the environment to define its location. On the other hand, to construct the map, the robot's location must be known). However, this research focuses on mobile robot localization only and in the future, this work will be extended to include the mapping process. Many techniques and systems were provided by researchers over the last years to detect and determine the robot position. There is no unique approach that can be used for all applications. However, every approach has its strength and weakness. These approaches can be categorized based on the sensors used such as global position system (Satellite), wheel odometry (Encoder), Lidar or Ultrasonic odometry (Lidar), an inertial navigation system (inertial measurement unit(IMU)), Visual odometry (Camera sensor). Also, these techniques could be used for both indoor or outdoor applications. Only the Global Positioning System (GPS) cannot be used in indoor application because the signal could not penetrate walls.

2.1.1 Global Navigation Satellite System (GNSS)

Global Navigation Satellite System is a navigation system which depends on satellite-based signals to provide the location information which includes longitude, latitude, and altitude/elevation. Anyone who has a GNSS receiver on the surface of the earth can determine his/her position. Thus, GNSS is considered the most popular techniques for mobile robot localization. A popular satellite system is the Global Positioning System (GPS) operated by the US. GPS consists of twenty-four operational satellites which orbit the Earth. The GPS satellite constellation is shown in Figure 2.1 [61].

The advantages of the GNSS are that it can find the absolute position without error accumulation over the travelled distance, and it is also stable. GNSS accuracy can be in the order of meters for commercial GNSS. This may not be sufficient for applications requiring high positioning accuracy. Even though the accuracy of GNSS can be improved to 3 cm using Differential Global Navigation Satellite System (DGNSS) such as Real Time Kinematics (RTK) techniques, GNSS signals are blocked by walls and are affected by the environmental factors such as weather, metal structures etc. Thus, it is not suitable for indoor applications. For the aforementioned reasons, GNSS cannot be used inside the underground tunnels [62].

2.1.2 Wheel Odometry

Odometry is an ancient word which derives from Greek words. First word is *hodos* and second word is *metron*. Hodos regards a journey or travel while metron means measure [63]. Therefore, the odometry term has first been used in wheel odometry where the agent motion can be obtained by accounting the number of wheel revolutions of the agent over the travelled distance. Thus, linear displacement can be obtained by translating the wheel revolutions. Usually, the wheel revolutions are measured by Encoders. An Encoder is shown in Figure 2.2.

Even though the wheel odometry is a simple technique, inexpensive and provides a relative

Figure 2.1: GPS satellite Constellation [61]

position, it suffers from slippage. Thus, the error will be accumulated over the travelled distance which increases proportionally with the distance covered. Therefore, wheel odometry is suitable only for short-term navigation. Also, its estimation accuracy could be improved by employing an additional sensor such as IMU [64]. On the other hand, the wheel odometry is not applicable for DTSS because the wheel odometry is used for the ground mobile robot. However, the drone will be used in this study as an inspection platform [65, 66].

2.1.3 Inertial Navigation System (INS)

The inertial navigation system provides the position and orientation of the mobile robot relative to a known point. The inertial measurement unit (IMU) is usually used in such an inertial navigation system. It consists of three accelerometers, which provides three linear accelerations, and three gyroscopes, which provides three rotational rates around three axes. Both linear velocity and position can be obtained by integrations of the linear acceleration with respect to time. On the other hand, the agent orientation can be obtained by integration of the rotational rates with respect to the time. The commercial inertial navigation systems are cheap, but they are associated with noise and bias. Thus, the mobile robot position and orientation have a huge drift from the actual state because of the accumulated error due to the integration process. However, the INS can be corrected

Figure 2.2: Optical Encoder

by employing it with another navigation system [67, 68].

In conclusion, the inertial sensor is not reliable to determine the position and orientation of the mobile robot alone. However, it can be utilized to supplement another navigation system to increase its reliability and accuracy such as GPS, camera or lidar. Although high accuracy inertial sensor provides a good motion estimation, it is expensive which makes using it difficult for commercial applications [69].

2.1.4 Acoustic Systems

The acoustic system utilizes the ultrasonic sensor to provide acoustic energy in measuring the distance between the obstacle and the system location. The acoustic system transmits ultrasonic pulses, and it receives them when they are reflected by surfaces and objects. Thus, the distance is calculated by measuring the travelling time of ultrasonic pulses. The acoustic system can provide high accuracy motion estimation. However, its accuracy depends on the reflection of the ultrasonic waves. The reflected signal is affected by the orientation of surfaces as well as the object materials. Regardless of the reflecting surface's materials and orientations, the environment noise has the same frequency of the ultrasonic pulse. Another disadvantage of the acoustic system is that it could receive multiple echo reflections which is a high likely to happen in DTSS. Moreover, the tunnel is a structure-less environment with a long tunnel axis. Thus, the acoustic system is not suitable for this application [70, 71].

2.1.5 Ranging System

The ranging system utilizes laser radar, light detection or lidar. It shares the same idea of the acoustic system where it transmits a laser pulse toward the object and then compares the reflected pulse with the original one. Ranging system technique depends on either travelling time of the laser pulse or the phase-shifting. For the travelling time, the distance

is calculated based on the travelling time of the laser pulse, beginning from the system to the object, and till the reflected pulse is received. On the contrary, phase-shifting is a comparison between two signals where two scans are compared to each other and finding the phase-shift. Lidar is considered as one of the most accurate motion estimation sensors because the drift is not accumulated over the travelled distance [72]. Also, it is insensitive to environmental conditions where it is not affected by lighting conditions, noise, or ambient temperature. Moreover, lidar can be combined with other motion estimation sensors such as GPS or inertial navigation system to improve the accuracy of the motion estimation [73, 74].

However, there are two essential drawbacks of lidar. First, lidar is a costly sensor compared to the other motion estimation sensors especially for 3D lidar such as Velodyne [75]. Not only it is expensive, but lidar is computationally expensive. Secondly, lidar fails in a structure-less environment such as along the tunnel axis (i.e. X-axis) [76, 77]. Thus, to take advantage of the lidar, lidar will be utilized to localize the system in two dimensions, the horizontal axis (i.e. Y-axis), and the vertical axis (i.e. Z-axis).

2.1.6 Visual Systems

Nowadays, visual systems attract the research communities because of their ability to perform various tasks. In one of these tasks is mobile robot localization, the robot utilizes a camera sensor to provide abundant information about the environment. With the help of the computer vision system, it can analyze the captured information from images and uses it to find the position and orientation of the mobile robot [78, 79]. The camera is considered a low-cost sensor as compared to the other motion estimation sensors. Even though the camera is inexpensive and provides abundant information, there are many drawbacks in using it. The camera sensor is sensitive to ambient conditions such as illumination conditions, texture in the environment, motion blurs and shadows. Moreover, almost all of the visual navigation systems are based on the static scene. If

there is a dynamic object or dynamic environment which changes frequently, it is a great challenge for the visual navigation system [80, 81]. In addition to previous drawbacks, computer vision algorithms are computationally expensive due to features of extractions and matching images. Hence the trade-off between the computational complexity and the accuracy is significant for the autonomous navigation system using the camera sensor. In the last few decades, a lot of visual-based localization algorithms have been developed and implemented to improve the robustness and reliability of the visual system navigation [82, 83].

2.1.7 Concluding Remarks On General Background

Based on the aforementioned background on mobile robot localization system, It can be concluded that achieving mobile robot localization in deep sewerage tunnel is challenging. It cannot be solved by using only one localization technique. Hence, Multiple sensing localization systems are recommended in this situation because there is a high chance that one of the sensors is degraded.

GPS, Acoustic system and wheel odometry are not suitable for this application. Firstly the GPS signal is not available inside the deep tunnel. Secondly echo reflection of the acoustic system could be an issue inside the tunnel. Moreover, the tunnel is considered as a structure-less environment. Thirdly wheel odometry is not applicable for drone localization which is the main inspection system for this application. Thus, the other three localization systems (i.e. Inertial Navigation System (INS) - Ranging Systems (RS) - Visual System (VS)) are applicable to be utilized inside the deep tunnel. However, these systems cannot be used alone. Therefore, the inertial navigation system will be utilized to improve the accuracy and robustness of both ranging system and the visual system. The ranging system will be used for mobile robot localization within two axes (Y-axis and Z-axis), while visual systems will be used for mobile robot localization along the tunnel axis (X-axis). Literature reviews on visual-based and ranging-based localization will be

discussed in the next sections.

2.2 Visual Odometry (VO)

The general technology of using the visual system to localize the robot in an unknown environment is called Visual SLAM methods (V-SLAM). V-SLAM is improved and developed by researchers over the last 30 years [84]. There are several famous V-SLAM algorithms and could be listed in chronological order as RatSLAM [85], MonoSLAM [86], PTAM [87], DTAM [88], Kinect Fusion [89] and ORB-SLAM [90]. The V-SLAM purpose is to define the robot location and to construct the environment mapping. Visual odometry is a part of V-SLAM that which it is responsible for estimating the mobile robot pose. VO was inspired by the wheel odometry concept. The motion estimation between two consecutive camera frames using the camera measurements over time to obtain the mobile robot pose (translation and orientation) related to its initial position is called VO [91, 92]. Furthermore, VO can be classified into two primary methods, namely, feature-based methods and direct methods. Feature-based methods was built on the extraction and matching of the features between two consecutive frames, while the direct-based methods was built on the pixel-level intensity [84, 93].

2.2.1 Feature-Based Methods

D. Nister [94] and F. Moreno-Noguer *et al.* [95] proposed two different solutions of the motion estimation problem using feature-based methods. A sparse set of salient image features is extracted, and they are matched in successive video frames using robust feature descriptors. The camera motion is first estimated using epipolar geometry and then refined through reprojection error minimization [96].

Existing robust feature detectors and descriptors such as SIFT [97] or SURF [98, 99] allow matching of feature points between frames even at the large inter-frame movement. thus,

the feature-based methods are good for the large frame to frame motion. Possible issues of feature-based methods are that most feature detectors are optimized for speed rather than precision and the necessity of robust estimation to deal with wrong correspondence. Many feature measurements are thus required to compensate the drift in the motion estimation. Moreover, the feature-based methods are relatively slow compared to the directed-based methods due to the features extraction, feature matching and the outliers which they are generally computationally expensive [100, 101].

2.2.2 Direct Methods

The direct methods use the pixels intensity values to estimation the motion. The motion is estimated by minimization the photometric error [102]. The direct methods exploit all the information in video frames (i.e. they utilize all the pixels intensity values in the image). Thus, they outperform the feature-based methods regarding robustness in scenes with little texture [103] or defocus-camera cases and motion blur [88].

The computation of photometric error is more intensive than the reprojection error as it involves integrating large image regions. The photometric consistency could be an issue, especially for outdoor scenarios. Besides, the convergence of the direct methods could be another issue when the agent moves around environment with waving grass, forest, and so on. Although, the precision and robustness of the direct methods are very good, they are limited to the frame-to-frame motion. The summary of feature-based methods and direct methods is stated in Table 2.1.

Direct methods can be classified into two main methods, namely Dense and Semi-Dense methods. Every pixel in the frame is involved and tracked using the dense method. Thus, the dense method can track over 300,000 pixels. [88]. On the other hand, strong gradient pixels-level intensities are tracked in the semi-dense method. Over 10,000 Pixels could be tracked approximately using the semi-dense method [104]. Comparison of dense and semi-dense methods are shown in Figure 2.3.

Table 2.1: Summary of Visual Odometry (VO)

Visual Odometry (VO)			
Feature-Based Methods		Direct Methods	
Reprojection error minimization		Photometric error minimization	
Advantages	Disadvantages	Advantages	Disadvantages
• Large frame to frame motion	• Computationally expensive due to features extraction	• All information in the image can be exploited (High Precision - Robust)	• Limited frame to frame motion

Based on previous observations between feature-based methods and direct methods, C. Forster [106] has introduced the Semi-Direct Visual Odometry (SVO) which is a combination of feature-based and direct methods. SVO is considered a direct method because it is working on pixel-level intensity. However, it works on the pixel-level intensity of the existing features, so it is called a sparse direct method as well. The SVO can be tracked over 2,000 pixels. The sparse method is shown in Figure 2.4. Furthermore, SVO gives the same effect compared to the dense and semi-dense methods in ordinary cases. We should point out that the dense method is beneficial with defocus and motion blur [105].

(a) Dense Method

(b) Semi-Dense Method

Figure 2.3: Direct Methods [105]

Figure 2.4: Sparse Method (Semi-Direct Method) [105]

Moreover, the comparison between dense, semi-dense and sparse methods has been made to show their robustness of frame-to-frame motion. The results show that both dense and semi-dense give roughly the same results because the semi-dense works on the strong gradient in the image. On the other hand, the sparse method gives almost the same results compared to dense and semi-dense when the overlap between two frames are around 30 %. However, if the overlap between two consecutive frames is less than 30 %, the robustness of the sparse method decreases as compared with dense and semi-dense methods as shown in Figure 2.5. A summary of direct methods is stated in Table 2.2.

The deep tunnel environment is usually dark, feature-less with a homogeneous structure as shown in Figure 1.6. As such, the best method to be used in such environment is the dense method because it gives a better result in such cases of low light, defocuses or motion

Figure 2.5: Robustness to Frame-to-Frame Motion [105]

Table 2.2: Summary of Direct Methods

Direct Methods		
Dense Method	Semi-Dense Method	Sparse Method
every pixel in the image is used	pixels with high intensity gradient are used	selective corner intensity pixels or along strong gradient edges pixels are used
Track around 300,000+ pixels	Track around 10,000+ pixels	Track around 2,000+ pixels

blur. However, it is computationally expensive. So, the SVO works on pixel intensity but with particular corner intensity pixel or with strong gradient edges which make it very useful in the deep tunnels. As a result of previous literature, SVO is recommended to be used in this research which represents the visual factor in our graph-optimization problem. Moreover, SVO¹ is available as open-source code.

2.3 Visual Inertial Odometry (VIO)

The robust and accurate state estimation can be achieved by using a camera with IMU. The camera is considered as an exteroceptive sensor that can provide abundant information about the environment. However, in the case of the monocular camera, it suffers from the scale of the ambiguity. Moreover, it is very sensitive to low texture environment, High Dynamic Range (HDR) applications and high-speed applications due to motion blur. On the other hand, IMU is a proprioceptive sensor that can provide a metric scale of monocular setup [107]. Furthermore, it is an independent sensor. Thus, it is a suitable choice for the aforementioned cases. However, it suffers from noise and bias. Thus, the state estimation can not be obtained using IMU alone because the drift is accumulating rapidly. For the aforementioned reasons, camera and IMU are considered

---

¹https://github.com/uzh-rpg/rpg_svo

complementary to each other. Fusion information or data from the camera and IMU is called Visual Inertial Odometry (VIO) or Visual-Inertial Simultaneous Localization and Mapping (VI-SLAM) etc [108].

The main problem of the Visual Inertial Odometry (VIO) is the data fusion from different sensors because each sensor works at different rate. The data fusion can be categorized into two major paradigms, namely, the fusion paradigm and the optimization paradigm. The first one is the fusion paradigm can be further categorized into two sub-paradigms, namely loosely coupled and tightly coupled, which express the fusion type [109]. The second one is the optimization paradigm, which also can be called back-end approaches [51], can be further categorized into three sub-paradigms, namely, filtering approach, fixed-lag approach and smoothing approach.

2.3.1 Fusion Paradigm

The accuracy of the visual-inertial odometry system depends on the fusion paradigm. In the literature, there are two major fusion paradigms. The first one is loosely coupled paradigm whereby the visual odometry and position estimation from IMU are separately computed. Then the output of both algorithms is fused to obtain the state estimation. For example, The output orientation from IMU is usually fused with the output of visual odometry to loosely-coupled paradigm [110, 111]. In contrast, tightly coupled paradigm

Figure 2.6: Two Major Fusion Paradigms [108]

occurs when the IMU information is fused directly into visual odometry. Thus, the tightly coupled paradigm is more accurate than the loosely coupled paradigm, but it is also more computationally intensive [112, 113]. The comparison between loosely and tightly coupled fusion paradigms is shown in Figure 2.6. Moreover, Tables 2.4 and 2.6 classify the major work which has been done over the last 10 years in VI-SLAM. They show that, due to the advanced technologies, the tightly coupled paradigm is being used recently over the loosely coupled paradigm because it is more accurate.

2.3.2 Optimization Paradigm

As mentioned in section 1.4.1, the VI-SLAM problem can be represented graphically by a factor graph. The VI-SLAM factor graph can be divided into two major approaches: the front-end approaches and the back-end approaches [51]. The front-end approaches deal with the graph construction and sensor data association which they are crucial for the back-end approaches [114, 115], while the back-end approaches deal with the optimization problem. Figure 2.7 shows the difference between front-end and back-end paradigms. In this subsection, the back-end paradigms will be stated in details. In the literature, researchers consider the optimization paradigm as the major factor of the VIO classification problem [108, 116, 117]. Thus, the VIO or VI-SLAM problem can be sectioned along with three major paradigms according to the back-end approaches: filtering approach, fixed-lag approach and full smoothing approach.

Figure 2.7: Difference Between Front-End and Back-End

The state estimation can be found by solving the Maximum A Posteriori (MAP) of fusion data problem. However, motion estimation is very challenging to be achieved in real-time because of computational complexity. Thus, the literature on VI-SLAM emphasizes *the trade-off between computational complexity and accuracy*.

Filtering Approach

Filtering approach is the fastest and most efficient way for fusion information because it enables a fast interference between data. The main reason that filtering approach is very fast is that all previous camera poses in the navigation history is marginalized. Only the latest state is considered in the estimation using the filtering approaches.

Kalman Filter (KF) is widely used for many years. It is considered a powerful algorithm for state estimation in many scientific areas such as flight control, navigation and indoor localization system. KF is also known as an optimal filter because it minimizes the mean square error (MSE). Moreover, linear system behaviour is estimated using the KF under the Gaussian noise. The main reasons of using KF are its simplicity, very fast and efficient for real-time application [118]. KF has been designed initially for a linear system. However, most sensors have nonlinear behaviour. Thus, it has poor robustness against nonlinearities in the models [119].

The Extended Kalman Filter (EKF) has been developed to deal with the nonlinearity of the system, whereby the nonlinear system will be linearized about the mean using the concept of Taylor series expansion [49] of the nonlinear system up to the first-order. EKF has two stages, namely the predicted state and the updated state. The Jacobian matrices are used to estimate the state-transition matrix [120]. For more detail about mathematics derivation of both KF and EKF, the reader is referred to Thrun *et al.* [119] which considers the main reference of basic concepts of the SLAM problem.

Although the EKF is a very efficient and powerful tool for real-time estimation, the complexity growth and inaccuracy are two main problems faced by EKF. Last studies

Table 2.3: The Most Common Filtering Approaches

Filter	Pros	Cons
Kalman Filter (KF)	• Simple • Fast • Used for linear system	• Poor robustness against the nonlinearities
Extended Kalman Filter (EKF)	• Deal with nonlinear systems • Very efficient for real-time estimation	• the complexity grows with the number of the extracted features quadratically • Inaccuracy problem where it marginalizes out the previous states
Multi-State Constrained Kalman Filter (MSCKF)	• Deal with complexity growth problem by using the structure-less approach	• Inaccuracy Problem

show that the complexity grows with the number of the extracted features that are tracked in real-time quadratically. So, only a small number of features can be tracked in real-time to reduce the complexity [86], where the tracked features must have a 3-D position in the vector state [121]. To overcome this issue, the structure-less approach has been introduced. The structure-less approach does not require the 3-D positions of the tracked feature in vector state as well as some of the previous poses have been marginalized out of the vector state. This concept has been introduced for the first time in Multi-State Constraint Kalman Filter (MSC-KF) [122]. Although MSC-KF is very efficient and solves the complexity of EKF problem, all measurements of the features must be obtained first. This causes a significant delay. The pros and cons of the most widely filtering fusion approaches are mentioned in Table 2.3.

Regarding the accuracy issue, filtering approaches take into consideration the latest state only to obtain motion estimation, and they marginalize out all other previous states.

2.3. VISUAL INERTIAL ODOMETRY (VIO)

Table 2.4: Summary of VI-SLAM (Filtering Approach) Over Past 10 Years

Fusion Paradigm	
Tightly Coupled	Loosely Coupled
2007, Mourikis [122]	2010, Tardif [123]
2007, Veth [124]	2011, Weiss [111]
2011, Kelly [125]	2011, Achtelik [126]
2011, Jones [121]	2012, Li [113]
2012, Lupton [37]	2012, Weiss [127]
2013, Li [128]	2012, Weiss [129]
2013, Guo [130]	2013, Sa [131]
2014, Guo [132]	2013, Lynen [40]
2014, Asadi [133]	2016, Munguia [134]
2015, Wu [135]	2016, Plaezieux [136]
2015, Tanskanen [137]	
2015, Bloesch [34]	
2015, Clement [138]	
2017, Bloesch [139]	
2017, Sa [140]	
2018, Brossard [141]	
2018, Sun [112]	
2018, Zhang [142]	

Thus, accuracy suffers because of linearization errors. In consequence, the actual state drifted, causing the filter inconsistency. As a result of such inconsistency, the filter will be over-confident. Therefore, the fusion information problem will not be optimally estimated, and this leads to an erroneous and fast-growing error in the state estimation [143]. Table 2.4 shows the most common work has been done on filtering approaches over the past 10 years. It shows that, although the computational efficiency increased due to the advanced technology nowadays, many researchers are keen to use the filtering approaches because they are efficient in real-time state estimation. For the aforementioned reasons, over the past three years, the filtering approaches are formulated as a tightly coupled paradigm

over the loosely coupled paradigm because it is more accurate.

The filtering approach was used in the tunnel inspection research that was done by Pennsylvania University. The study shows that there is a significant drift has occurred when the drone navigates along the tunnel axis. For 40 *m* flight, the recorded drift was 1 *m* from the actual state. This error is enormous [144]. Accordingly, this filtering approach should be replaced by some alternative approach that gives better accuracy and has less computational complexity.

Fixed-Lag Smoothing Approach

As mentioned before, the effect of marginalization leads to huge drift accumulated over the travelled distance. So, the fixed-lag smoothing approach takes into consideration some of the previous states within a sliding window. For example, if the size of the sliding window is ten, the fixed-lag smoothing approach takes into consideration the last ten previous states into the optimization process. This process gives better results over the filtering approach because the fixed-lag smoother approach allows for smoothing some of the history states within the given window size [145]. The fixed-lag smoothing is efficient and more accurate than filtering approach with a low computational cost that can be applied in real-time. However, as the fixed-lag smoothing approach has marginalized part of the history state in the motion estimation, it shares the same issue with the filtering approach where it is drifting over the travelled distance because of the consistent linearization error [42]. Moreover, up to now, there is no basic concept or solid definition of how to select the sliding window size.

Full Smoothing Approach

The highest motion estimation accuracy can be obtained by the full smoothing approaches, whereby, the full history of the previous states will be taken into consideration to solve the nonlinear optimization problem. Also, the full optimization problem is known as Bundle

2.3. VISUAL INERTIAL ODOMETRY (VIO)

Table 2.5: Differences Between VIO Approaches

Filtering Approach	Fixed-Lag Smoothing Approach	Full Smoothing Approach
It marginalizes out all history of the previous states and keeps the latest state	It takes into consideration some of the previous states within a sliding window and marginalizes out the other states	The full history of the previous states takes into consideration to solve the nonlinear optimization problem.

Adjustment (BA) where it optimizes the full state trajectory when every new measurement is observed. However, the full nonlinear optimization problem is computational costly in real-time because the entire history trajectory grows over time. Therefore, solving a comprehensive optimization problem in real-time becomes infeasible. Thus, solving the full optimization problem has to be done offline [146]. The main differences between VIO optimization paradigms are summarized in Table 2.5. Table 2.6 shows the work that has been done over the past 10 years according to the fusion paradigm and smoothing approach. It should be pointed out that both fixed-lag smoothing and full smoothing are considered as smoothing approach in Table 2.6. The reason is that, both of these two approaches share the same concept and differ from filtering approach. Also, Tables 2.4 and 2.6 show that the most recent researches are based on smoothing approach and tightly coupled fusion paradigm.

The full optimization problem can be represented by Markov Random Field [119] as shown in Figure 2.8. The most recent researches focus on using the full smoothing approach in real-time. Firstly, a famous technique has been proposed to solve this problem, called *Keyframe BA*. The fundamental idea of this algorithm is to discard some of the camera's frames out of the estimation under the assumption that all measurements will be connected and there is no marginalization. In other words, all measurements must be observed when some of the camera frames are discarded [169]. As shown in Figure 2.8, the states X_1 and X_2 are discarded from the optimization problem. Thus, the optimization problem is simplified.

Table 2.6: Summary of VI-SLAM (Smoothing Approach) Over Past 10 Years

Fusion Paradigm	
Tightly Coupled	Loosely Coupled
2013, Leutenegger [147]	2015, Brunetto [148]
2015, Forster [149]	2016, Falquez [150]
2015, Burri [151]	
2015, Huai [152]	
2016, Keivan [153]	
2016, Concha [154]	
2016, Usenko [155]	
2017, Mur-Artal [156]	
2017, Piao [157]	
2017, Liu [158]	
2017, Yang [159]	
2017, Kasyanov [160]	
2017, Laidlow [161]	
2017, Li [162]	
2018, Qin [163]	
2018, Lin [164]	
2018, Chen [165]	
2018, Liu [166]	
2019, Echenhoff [167]	
2019, Henawy [168]	

Another technique to solve the full optimization problem in real-time is using a parallel thread. There are some examples of using the dual architecture in tracking and mapping [87, 170]. As mentioned in section 1.4.1, a factor graph is a graphical model to solve the optimization problem. The data can be fused from different sensor in full optimization approach using factor graph [46]. Factor graph consists of nodes, and each unknown *variable state* can be represented by a node which it is connected with a number of probabilistic information called *factors* which depends on the measurements or prior

Figure 2.8: Illustration of Keyframe Algorithm [169]

x_i and y_i represent the variable state and observed features, respectively

knowledge. Consequently, when a new measurement is observed, it simply adds to the factor graph representation. Moreover, if any sensor becomes unavailable, it is merely compensated in the factor graph. The general concept of the factor graph approach is shown in Figure 2.9. Solving the factor graph representation is a linearized least-square problem (see Appendix A) where the nonlinear optimization will be linearized around the predicted point or initial value (the mean value) by calculating the measurement Jacobian matrix. As a result of the linearization, the new Jacobian matrix must be calculated every time a new measurement is observed. Therefore, the computational cost is very high over time [171].

However, the incremental smoothing approaches are considered a breakthrough in the full

Figure 2.9: Example of General Concept of Factor Graph [171]

smoothing techniques whereby high computational cost is reduced. The key idea of the incremental smoothing is found in the nature of calculating the Jacobian matrix [172, 173]. On the contrary to full smoothing approach, the incremental smoothing does not require calculating the new Jacobian matrix every time a new measurement is observed. In other words, incremental smoothing updates the Jacobian matrix with the new variables when it is observed in the factor graph as shown in Figure 2.10.

Even though that the incremental smoothing reduced the computational cost and is considered as a breakthrough, the fact that the IMU has a higher sampling rate than the other sensors used in the current motion estimation represents a great challenge in VIO problem. The reason is that, every new measurement observed from IMU is represented with a new node in the factor graph representation and optimization should be done once a new node is observed. Therefore, the trajectory will grow very fast and become computationally infeasible in real-time. The effect of IMU higher sampling rate on the factor graph can be shown in Figure 2.11.

Accordingly, Lupton *et al.* [37] established the pre-integration theory whereby the linear acceleration and angular velocity are integrated into the body frame before they are transferred to the world frame between two consecutive keyframes. However, previously both of the linear acceleration and the angular velocity were transferred to the world frame at each time stamp to estimate the state before being integrated.

The main idea of the pre-integration theory is that, all IMU measurements can be

$$J' = \begin{bmatrix} \mathbf{x}_1 & \mathbf{c}_1 & \mathbf{x}_2 & \mathbf{c}_2 & \mathbf{x}_3 & \mathbf{c}_3 \\ \times & & & & & \\ & \times & & & & \\ \times & \times & \times & & & \\ & \times & & \times & & \\ & & \times & & & \\ & & \times & \times & \times & \\ & & \times & \times & \times & \times \end{bmatrix},$$

Figure 2.10: Key Idea of The Incremental Smoothing [171]

Figure 2.11: IMU High Sampling Rate Affect on Factor Graph Representation [171]

combined into one single and equivalent observation between two consecutive keyframes. This concept is shown in Figure 2.12. As a result of the pre-integration theory, the graph-based SLAM can work in slower frequency such as camera frequency thus the graph-based optimization problem is simplified. Consequently, the agent motion can be estimated in the real-time to obtain high accuracy. Moreover, Forster *et al.* [149] has introduced a novel pre-integration theory on the manifold $SO(3)$ on top of the work of Lupton *et al.* [37]. This is considered as the first step towards achieving a high accuracy state estimation with less computational cost in real-time. However, the pre-integration theory has been built on discrete assumptions which hinder its accuracy. These discretization assumptions have been relaxed and mitigated their effect by high sampling IMU. Moreover, Jacobians and covariance matrices also have been hindered

Figure 2.12: Combine All IMU Measurements into Equivalent Value [171]

based on these discretization assumptions.

Therefore, in chapter 3, *a novel pre-integration IMU model is proposed using more reasonable assumptions to obtain higher motion estimation accuracy over the travelled distance*. This allows us to obtain high accuracy *IMU factor* with a closed-form solution of mean value, covariance matrix and analytic Jacobians.

2.4 Lidar Inertial Odometry

One of the dominant solutions for the scanning matching problem is ICP algorithms [174, 175]. However, the existing ICP algorithms [176, 54] have not dealt with the motion distortion problem. The motion distortion represents the main problem of lidar that occurs among all points in the sweep caused by the movement of the robot as mentioned in section 1.4.2. Hong *et al.* [1] estimated velocity of the robot numerically over ICP iterations under an assumption that the velocity is constant during the scanning time. As a result, the distortion of a scan due to motion can be compensated using the estimated velocity. Also, Pomerleau *et al.* [177] provides a general comparison between existing ICP algorithms.

A more accurate and fast-motion tolerant correction method is required for indoor navigation applications. By assuming that lidar motion is modelled with constant angular and linear velocities during the sweep, Zhang *et al.* [55, 178] provided an interesting lidar odometry method by recursively estimating the 6-DOF motion during the sweep, and gradually including more points. The assumption could be an issue when the motion is complicated. For example, a sweep can be scanned when the lidar goes straight and followed by a turn. Thus, *accurate Lidar Odometry requires knowledge of the lidar pose during continuous laser ranging. Fortunately, knowledge can be provided by an IMU*. The linear acceleration from IMU was utilized by Zhang *et al.* [55] to address the nonlinear motion of the agent. Experimental results indicated that the performance of the ICP can

indeed be improved using the IMU measurements.

There are two major paradigms to fuse the lidar measurements and IMU measurements, either loosely coupled or tightly coupled. Both of loosely and tightly coupled shared the same definition as mentioned in section 2.3.1. For loosely coupled paradigm, the orientation of the robot pose has been calculated by IMU measurements in lidar Odometry And Mapping (LOAM) algorithm which was proposed by Zhang *et al.* [55]. LOAM used the IMU measurements as the only prior factor for the algorithm. As a result of that, the IMU measurements have not been used in the optimization process. Another effective algorithm, Tang *et al.* [179] proposed a filtering fusion method between lidar and IMU measurements using the EKF. However, this algorithm was based on 2D scenarios and was not applicable for a 3D environment such as an unmanned aerial vehicle. On the other hand, Lynen *et al.* [40] proposed a robust multi-sensors fusion approach applied to an unmanned aerial vehicle using EKF. This approach can use both lidar and IMU measurements. Similarly to the general conclusion of loosely coupled paradigm in the vision system, they are less accurate and less computational than the tightly coupled paradigm.

For tightly coupled paradigm, Soloviev *et al.* [41] proposed an algorithm for 2D lidar scanning and matching, where IMU orientation measurements have been used to compensate the tilted lidar. Moreover, the drift in IMU measurements was corrected and updated using the KF. Another recent tightly coupled algorithm was proposed by Hemann *et al.* [180], where the IMU error state was propagated using the KF. It showed good and robust results over long navigation period. However, it failed without prior information about the environment. Accordingly, Ye *et al.* [56] proposed a first 3D tightly coupled fusion algorithm between lidar and IMU measurements. This algorithm was inspired by the work has been done by Qin *et al.* [163]. However, there were two major issues related to this algorithm. First, the transformation matrix, which used to correct the motion distortion among all points in the sweep, was based on discrete IMU

model. Thus, its accuracy was hindered by discretization. Second, the *IMU factor* was based also in discrete propagation. Therefore, The mean value, covariance matrix and the Jacobian matrix are all represented in discrete form.

2.5 Concluding Remarks

From the literature review, it can be concluded that obtaining a high accuracy motion estimation is crucial for mobile robot localization. In GPS denied area, camera and lidar can be used to localize the robot with respect to the environment. However, the camera sensor is suffering from darkness or illumination changing conditions and featureless environment (e.g. homogeneous environment), while lidar is suffering from the structure-less environment as well as a wet wall. Therefore, multiple sensing (e.g. camera and lidar) system must be used inside the deep sewerage tunnel to localize the robot with respect to the environment.

This literature review focused on robot localization using the vision system and lidar. For localization using the vision system, there are two main approaches to localize a mobile robot indoor. First is the visual odometry (VO) and second is the visual-inertial odometry (VIO).

Visual odometry is classified into two majors methods: feature-based method and direct based method. The feature-based method is based on feature extraction and matching using reprojection error minimization. Moreover, it is suitable for a large frame to frame motion. However, it is computationally expensive. The direct based method is based on pixel intensity using photometric error minimization. It utilizes all the information in the image. It is suitable for a poor feature environment as well as motion blur, but it has a limited frame to frame motion. To overcome this problem, a semi-direct method is introduced where it uses a direct method based on pixel intensity, and it works on strong gradient features such as a long edge. VO is not reliable when no feature is observed or

2.5. CONCLUDING REMARKS

---

when the agent is too close to the wall which it cannot observe the feature.

Accordingly, employing an inertial measurement unit (IMU) as an additional sensor can dramatically improve both reliability and accuracy of Visual Odometry (VO) where the high dynamic change in the system is taken into consideration using the inertial measurement units. Visual-inertial odometry can be expressed as a fusion problem from different sensors. Thus, the VIO can be classified into two primary paradigms: fusion paradigm and optimization paradigm. Fusion paradigm can also sub-divided into two major paradigms: loosely coupled and tightly coupled. In loosely coupled paradigm, state estimation is done using an inertial system and the visual system separately first. One system output will be fused with other system output to correct the drift. On the other hand, tightly coupled paradigm fuses both systems data directly. Thus, it is more accurate but more computational intensive than loosely coupled paradigm.

Optimization paradigm can be sectioned into three main approaches to solve the sensor fusion problem, namely, filtering approaches, fixed-lag smoothing approaches and full smoothing approaches. Filtering approach is very efficient, fast and suitable for real-time application. However, the accuracy deteriorates for long travelled distance. The main reason is that the filtering approach marginalizes out all history of the previous states and takes into consideration the latest state only. So, the error is accumulated due to the linearization process. To overcome this problem, fixed-lag smoothing is introduced where it takes into consideration some of the previous states within a sliding window and removes the other states. It gives a better result over the filtering approaches. However, it still shares the same problem of having low accuracy over long travelled distance.

Finally, high accuracy motion estimation is obtained by full smoothing approach where the full history of the previous states is taken into consideration to solve the nonlinear optimization problem. However, it is expensive computationally where the trajectory grows over time. Thus, two approaches are introduced to solve this problem, namely the key-frame and the dual architecture in tracking and mapping. Moreover, a breakthrough

has been in the full smoothing techniques is called incremental smoothing approaches where it does not require to calculate the new Jacobian matrix every time a new measurement is observed, but it updates the Jacobian matrix.

Although the incremental smoothing approaches have a significant advantage to solve the full optimization problem, high IMU sampling rate could be an issue. The full optimization problem can be represented graphically by a factor graph. A new node is added to the factor graph once it receives a new measurement. Thus, the factor graph trajectory grows very fast due to high IMU sampling rate. The preintegration motion theory was introduced to overcome this problem by combining all measurement between two consecutive frames into one equivalent value. However, the current motion estimation depends on specific assumptions which affect the motion estimation accuracy. The high IMU sampling rate mitigates its discretization effect.

Accordingly, this research aims to propose *a novel motion integration model that provides an analytical solution*. Based on this novel motion integration model, a closed-form solution is proposed to estimate mean value, covariance matrix and Jacobians matrices which are required to represent the IMU factor. The proposed motion integration model and IMU factor will be stated in details in Chapter 3.

For localization using a ranging system, the main problem, in the ranging system, is the motion distortion among all the scanned points. There are some algorithms to deal with this issue. However, they are useful when the robot moves with a constant velocity or of low speed. In real-world scenarios, especially in cases of using UAVs, the movement is usually more complicated. Another issue with lidar is that it can receive only a few points in the tunnel environment. Fortunately, the IMU can help in this case where the kinematic changes can be observed by IMU for a short time with high updating rate. Thus, in DTSS case, lidar inertial odometry is recommended where it fuses measurements from lidar and IMU.

There are two main paradigms for fusing measurements of lidar and IMU: loosely coupled

2.5. CONCLUDING REMARKS

---

and tightly coupled. Loosely coupled is less accurate than the tightly coupled and it considers lidar odometry is a black box. On the other hand, the most accurate motion estimation and reliable one are obtained by tightly coupled paradigm.

From this Literature review, *a novel way to correct the motion distortion* caused by robot movement is proposed, where IMU measurements is used to correct all distorted points. Second, the proposed IMU factor from the visual part will be used as tightly coupled joint-optimization with the lidar factor to achieve high accuracy motion estimation. The novel algorithm for correcting the motion distortion will be stated in details in Chapter [3](#).

Chapter 3

Methodology

In this chapter, a brief introduction related to the proposed model is stated first. This is followed by preliminaries definition in section 3.1. Visual inertial odometry problem is formulated in section 3.2. Novel IMU motion integration model and preintegration IMU measurements are stated in details in sections 3.3 and 3.4. Next, the novel covariance matrix and analytic biases updated are shown in sections 3.5 and 3.6. Based on the previous sections, the novel IMU factor is stated as a closed-form solution in section 3.7. Finally, the novel nonlinear de-skewing is proposed in detail in section 3.8.

As mentioned in section 1.4.1, the preintegration theory has been proposed by Lupton *et al.* [37] to simplify the graph-based optimization in real-time, whereby the IMU measurements are integrated into a single equivalent value between two successive keyframes. Thus, this is a first step towards using a full smoothing approach in online state estimation. With advanced computational technology, the motion pre-integration theory has become widely used nowadays. However, the pre-integration theory is based on discrete motion IMU integration model under some assumptions. These assumptions are that *the linear acceleration in the world frame and the angular velocity in the body frame are assumed to be constant between two consecutive IMU measurements* as demonstrated by Forster *et al.* and Qin *et al.* [116, 163]. Consequently, the agent's orientation has also

Figure 3.1: The difference between the state-of-the-art and proposed approach assumptions

been assumed to be constant for each integration interval between two successive IMU measurements. These assumptions may be applicable when the IMU sampling rate is high. However, these assumptions could be a problem when the agent is travelling very fast, which is equivalent to a low sampling IMU (i.e., the high IMU sampling rate mitigates the discretization effect) or when there are high dynamic changes in the agent's motion. Thus, the accuracy of the discrete methods experienced by Forster *et al.* and Qin *et al.* [116, 163] is hindered because of discretization. In contrast, numerical methods are recommended when the IMU sampling rate is low or when high accuracy state estimation is obtained. These numerical methods such as the Runge-Kutta method with a fourth-order method (RK4) [181, 182] can be a good and efficient solution in these cases. However, the high sampling rate of IMU represents a great challenge because of high computational cost of the numerical methods.

In this chapter, a novel IMU integration model is introduced by using switched linear systems [183]. *The linear acceleration is assumed to be constant in the local frame instead of the world frame by the new model.* The dynamic of UAV is represented very well by the new model and thus the assumption that the linear acceleration is constant in the local frame is reasonable [184]. The difference between the state-of-the-art and the proposed model assumptions is shown in Figure 3.1. A simple state-space representation for a discrete linear system is first developed to represent the kinematic between two successive IMU measurements. Since both the linear acceleration and the angular velocity between two successive visual frames are usually time-varying, the overall system between two successive visual frames is a switched linear system [183]. The IMU preintegration is then obtained by integrating all the IMU measurements between two video frames. The new IMU integration model provides a closed-form solution for the IMU kinematics. It also provides a closed-form solution for the state transition matrix which is used to compute the covariance matrix. Moreover, all Jacobian matrices with respect to biases and residual error Jacobian matrices are represented by the closed-form solution in a discrete manner. Clearly, 1) the novel model takes the high dynamic changes of the agent motion into consideration, 2) it is very useful for high-speed application. In order to illustrate the efficiency of the proposed approach, it is integrated with a structureless model of visual measurements to form a VIO pipeline. The pipeline is formulated as a factor graph optimization problem which enables the application of incremental smoothing algorithms.

3.1 Preliminaries

3.1.1 Basic Knowledge on Skew Matrix

The angular velocity of the agent measured by a gyroscope in the body frame can be expressed by $\omega_B = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 \end{bmatrix}^T$. The matrix $\omega_B^\wedge$ is the skew matrix which can be

expressed by

$$\omega_B^\wedge = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \quad (3.1)$$

From (3.1) It can be easily derived that

$$\begin{cases} (\omega_B^\wedge)^2 = \begin{bmatrix} -\omega_2^2 - \omega_3^2 & \omega_1\omega_2 & \omega_1\omega_3 \\ \omega_1\omega_2 & -\omega_1^2 - \omega_3^2 & \omega_2\omega_3 \\ \omega_1\omega_3 & \omega_2\omega_3 & -\omega_1^2 - \omega_2^2 \end{bmatrix}, \\ (\omega_B^\wedge)^3 = -\|\omega_B\|^2 \omega_B^\wedge \end{cases}, \quad (3.2)$$

where $\|\omega_B\|^2$ is $(\omega_1^2 + \omega_2^2 + \omega_3^2)$. For more details of $(\omega_B^\wedge)^3$, the reader is referred to Appendix B.

Similarly, from (3.2) it can be easily derived that

$$\begin{cases} (\omega_B^\wedge)^{(2m+1)} = (-1)^m \|\omega_B\|^{2m} (\omega_B^\wedge) \\ (\omega_B^\wedge)^{(2m+2)} = (-1)^m \|\omega_B\|^{2m} (\omega_B^\wedge)^2 \end{cases}. \quad (3.3)$$

3.1.2 Taylor Expansion of Cosine and Sine

Sine function can be expressed by Maclaurin series as the following:

$$\begin{aligned} \sin \|\omega_B\| &= \sum_{m=0}^{\infty} \frac{(-1)^m}{(2m+1)!} \|\omega_B\|^{(2m+1)} \\ &= \|\omega_B\| - \frac{\|\omega_B\|^3}{3!} + \frac{\|\omega_B\|^5}{5!} - \frac{\|\omega_B\|^7}{7!} + \dots \end{aligned} \quad (3.4)$$

Cosine function can be also expressed by Maclaurin series as the following:

$$\begin{aligned} \cos \|\omega_B\| &= \sum_{m=0}^{\infty} \frac{(-1)^m}{(2m)!} \|\omega_B\|^{(2m)} \\ &= 1 - \frac{\|\omega_B\|^2}{2!} + \frac{\|\omega_B\|^4}{4!} - \frac{\|\omega_B\|^6}{6!} + \dots \end{aligned} \quad (3.5)$$

3.2 Problem Formulation on Visual Inertial Odometry

We consider an agent, which is equipped with an IMU and a monocular camera. The IMU frame is assumed to coincide with the body frame to be tracked. The transformation between the camera and the IMU is fixed and known from prior calibration. In addition, a *front-end* is supposed to provide image measurements of the 3-D landmarks at unknown positions. A subset of images called *keyframes* is selected to estimate the pose of the agent. The state of the agent X at time k is described by the IMU's orientation, position, velocity and biases

$$x(k) \doteq \begin{bmatrix} R_{BW}(k) & P_W(k) & V_W(k) & b(k) \end{bmatrix}, \quad (3.6)$$

where $P_W = \begin{bmatrix} P_1 & P_2 & P_3 \end{bmatrix}^T$ and $V_W = \begin{bmatrix} V_1 & V_2 & V_3 \end{bmatrix}^T$ are the position of the camera center and the linear velocity of the agent having the camera in the world reference frame. R_{BW} is the rotation matrix from the body frame to the world reference one. $b(k) = \begin{bmatrix} b^g(k) & b^a(k) \end{bmatrix} \in \mathbb{R}^6$. $b^g(k)$ and $b^a(k)$ are the gyroscope and accelerometer bias, respectively.

Let $C(i)$ be the image measurements at keyframe i . Multiple landmarks $z_l(i)$ are observed by the i th keyframe, i.e., $z_l(i) \in C(i)$. Let $L(i, j)$ denote the set of IMU measurement acquired between two consecutive keyframes i and j . The set of measurements collected up to time k is

$$Z(k) \doteq \{C(i), L(i, j)\}_{(i,j) \in \mathfrak{K}(k)}, \quad (3.7)$$

where $\mathfrak{K}(k)$ is the set of all keyframes up to time k . The states of all keyframes in $\mathfrak{K}_k$ is defined as

$$X(k) \doteq \{x(i)\}_{i \in \mathfrak{R}(k)}. \quad (3.8)$$

The posterior probability of the variables $X(k)$, given the available visual and inertial measurements $Z(k)$ and prior $p(X(0))$, $p(X(k)|Z(k))$ is proportional to

$$\prod_{(i,j) \in \mathfrak{R}(k)} p(L(i,j)|x(i), x(j)) \prod_{i \in \mathfrak{R}(k)} \prod_{l \in C(i)} p(z_l(i)|x(i)). \quad (3.9)$$

The Maximum A Posteriori (MAP) (i.e, find the set of parameters X that maximize the likelihood of a set of measurements or observations) estimation problem is then formulated as

$$\arg \min_{X(k)} \left\{ \sum_{i,j \in \mathfrak{R}(k)} \|r_{L(i,j)}\|_{\Sigma_{ij}}^2 + \sum_{i \in \mathfrak{R}(k), l \in C(i)} \|r_{z_l(i)}\|_{\Sigma_z}^2 \right\}, \quad (3.10)$$

where $r_{L(i,j)}$ and $r_{z_l(i)}$ are the residual errors associated to the corresponding measurements, and Σ_{ij} , and Σ_z are the corresponding covariance matrices. For more details of the MAP, the reader is referred to Appendix C.

This is a nonlinear optimization problem. Generally, first-order Taylor expansion is adopted to convert the problem into a quadratic optimization problem which can be easily solved. One standard approach for optimization on manifold is to use a retraction $\mathcal{R}_x$, i.e.

$$\min_{x \in M} f(x) \Rightarrow \min_{\delta x} f(\mathcal{R}_x(\delta x)). \quad (3.11)$$

The reparametrization is usually called lifting. In this research, the following retraction is used for $SO(3)$:

$$\mathcal{R}_R(\delta\phi) = RE(\delta\phi); \delta\phi \in \mathbb{R}^3, \quad (3.12)$$

and the following for $SE(3)$ at $T \doteq (R, p)$

$$\mathcal{R}_T(\delta\phi, \delta p) = (RE(\delta\phi), p + R\delta p), \begin{bmatrix} \delta\phi & \delta p \end{bmatrix} \in \mathbb{R}^6, \quad (3.13)$$

In this research, the *IMU factor* will be the focus. First, a novel IMU motion integration model will be proposed. Then, based on the novel IMU model, the novel closed form solution of residual errors $r_{L(i,j)}$ with the corresponding covariance matrix Σ_{ij} and Jacobians matrices will be derived.

3.3 A Novel IMU Integration Model Using Switched Linear Systems

An IMU provides both linear acceleration and angular velocity in the body frame in the form of a 3-axis accelerometer and 3-axis gyroscope data denoted by a_B and ω_B , respectively. The body frame will be expressed by the suffix B notation, whereas the world frame will be expressed by the suffix W notation.

3.3.1 A Linear Kinematic Model

The motion integration can be obtained by introducing the IMU kinematic model as given by Forster *et al.* and Mur-Artal *et al.* [116, 156]:

$$\begin{cases} \dot{P}_W = V_W \\ \dot{V}_W = a_W \\ \dot{R}_{BW} = R_{BW}\omega_B^\wedge \end{cases} \quad (3.14)$$

where $\omega_B = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 \end{bmatrix}^T$ is the angular velocity of the agent measured by a gyroscope in the body frame. a_W is the linear acceleration in the world frame, and it can be expressed

3.3. A NOVEL IMU INTEGRATION MODEL USING SWITCHED LINEAR SYSTEMS

---

by:

$$a_W = R_{BW}a_B + g_W \quad (3.15)$$

$a_B = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}^T$ is the linear acceleration in the body frame that is measured with an accelerometer. $g_W = \begin{bmatrix} 0 & 0 & -9.81 \end{bmatrix}^T$ with g_W being the gravity acceleration in the world frame.

Let $R_{WB} = \begin{bmatrix} R_1 & R_2 & R_3 \end{bmatrix}$ be rotation matrix from the world reference to the body reference. It can be derived from equation (3.14) and $R_{WB}R_{BW} = I_{3 \times 3}$ that the kinematic of R_{WB} is given as the following:

$$\dot{R}_{WB} = -\omega_B^\wedge R_{WB}, \quad (3.16)$$

and it is equivalent to

$$\dot{R}_i = -\omega_B^\wedge R_i, i \in \{1, 2, 3\}. \quad (3.17)$$

A new kinematics model will be built up using the new state vector whereby a new state sub-vector is introduced as $\begin{bmatrix} P_W^T & V_W^T & R_1^T & R_2^T & R_3^T \end{bmatrix}^T$. The kinematic of the system can be expressed by:

$$\begin{bmatrix} \dot{P}_W \\ \dot{V}_W \\ \dot{R}_1 \\ \dot{R}_2 \\ \dot{R}_3 \end{bmatrix} = \begin{bmatrix} V_W \\ R_{BW}a_B \\ -\omega_B^\wedge R_1 \\ -\omega_B^\wedge R_2 \\ -\omega_B^\wedge R_3 \end{bmatrix} + \begin{bmatrix} 0_{3 \times 1} \\ g_W \\ 0_{3 \times 1} \\ 0_{3 \times 1} \\ 0_{3 \times 1} \end{bmatrix}. \quad (3.18)$$

Using the equality that $R_{BW} = R_{WB}^T$ and the matrix theory [185], it can be derived that

$$R_{BW}a_B = \begin{bmatrix} R_1^T a_B \\ R_2^T a_B \\ R_3^T a_B \end{bmatrix} = \begin{bmatrix} a_B^T R_1 \\ a_B^T R_2 \\ a_B^T R_3 \end{bmatrix}. \quad (3.19)$$

It follows that

$$\begin{bmatrix} \dot{P}_W \\ \dot{V}_W \\ \dot{R}_1 \\ \dot{R}_2 \\ \dot{R}_3 \end{bmatrix} = \Psi(1, a_B, \omega_B) \begin{bmatrix} P_W \\ V_W \\ R_1 \\ R_2 \\ R_3 \end{bmatrix} + \begin{bmatrix} 0_{3 \times 1} \\ g_W \\ 0_{3 \times 1} \\ 0_{3 \times 1} \\ 0_{3 \times 1} \end{bmatrix}, \quad (3.20)$$

where the matrix $\Psi(s, a, \omega)$ is given by

$$\Psi(s, a, \omega) = \begin{bmatrix} 0_{3 \times 3} & sI_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & \hat{A}_1(a) & \hat{A}_2(a) & \hat{A}_3(a) \\ 0_{3 \times 3} & 0_{3 \times 3} & -\omega^\wedge & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & -\omega^\wedge & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & -\omega^\wedge \end{bmatrix}, \quad (3.21)$$

and the matrices $\hat{A}_i(a) (1 \leq i \leq 3)$ are

$$\begin{cases} \hat{A}_1(a) = \begin{bmatrix} a_{3 \times 1} & 0_{3 \times 1} & 0_{3 \times 1} \end{bmatrix}^T \\ \hat{A}_2(a) = \begin{bmatrix} 0_{3 \times 1} & a_{3 \times 1} & 0_{3 \times 1} \end{bmatrix}^T \\ \hat{A}_3(a) = \begin{bmatrix} 0_{3 \times 1} & 0_{3 \times 1} & a_{3 \times 1} \end{bmatrix}^T \end{cases}. \quad (3.22)$$

3.3.2 The Proposed IMU Integration Model

The new kinematics model (3.20) is a time varying linear system. Consider the small interval $[k\Delta t, (k+1)\Delta t]$. Assume that the values of a_B and ω_B are fixed within the small interval. The matrix $\Psi(1, a_B, \omega_B)$ is a constant matrix in the interval. Subsequently, the system in the equation (3.20) is a linear time invariant system during the interval. Using the linear control system theory [186], a discrete model can be derived from the equation (3.20) by Horn *et al.* and Kailath [185, 186]

3.3. A NOVEL IMU INTEGRATION MODEL USING SWITCHED LINEAR SYSTEMS

$$\begin{bmatrix} P_W(k+1) \\ V_W(k+1) \\ R_1(k+1) \\ R_2(k+1) \\ R_3(k+1) \end{bmatrix} = e^{\Psi(\Delta t, \Delta t a_B, \theta_B)} \begin{bmatrix} P_W(k) \\ V_W(k) \\ R_1(k) \\ R_2(k) \\ R_3(k) \end{bmatrix} + \begin{bmatrix} g_W \frac{\Delta t^2}{2} \\ g_W \Delta t \\ 0_{3 \times 1} \\ 0_{3 \times 1} \\ 0_{3 \times 1} \end{bmatrix}, \quad (3.23)$$

where the matrix $e^{\Psi(s, a, \omega_B)}$ can be calculated as

$$e^{\Psi} = I + \Psi + \frac{\Psi^2}{2!} + \frac{\Psi^3}{3!} + \frac{\Psi^4}{4!} + \frac{\Psi^5}{5!} + \dots \quad (3.24)$$

and

$$\Psi^2 = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} & s\hat{A}_1(a) & s\hat{A}_2(a) & s\hat{A}_3(a) \\ 0_{3 \times 3} & 0_{3 \times 3} & \hat{A}_1(a)(-\omega_B^\wedge) & \hat{A}_2(a)(-\omega_B^\wedge) & \hat{A}_3(a)(-\omega_B^\wedge) \\ 0_{3 \times 3} & 0_{3 \times 3} & (-\omega_B^\wedge)^2 & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & (-\omega_B^\wedge)^2 & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & (-\omega_B^\wedge)^2 \end{bmatrix}, \quad (3.25)$$

$$\Psi^3 = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} & s\hat{A}_1(a)(-\omega_B^\wedge) & s\hat{A}_2(a)(-\omega_B^\wedge) & s\hat{A}_3(a)(-\omega_B^\wedge) \\ 0_{3 \times 3} & 0_{3 \times 3} & \hat{A}_1(a)(-\omega_B^\wedge)^2 & \hat{A}_2(a)(-\omega_B^\wedge)^2 & \hat{A}_3(a)(-\omega_B^\wedge)^2 \\ 0_{3 \times 3} & 0_{3 \times 3} & (-\omega_B^\wedge)^3 & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & (-\omega_B^\wedge)^3 & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & (-\omega_B^\wedge)^3 \end{bmatrix}, \quad (3.26)$$

$$\Psi^4 = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} & s\hat{A}_1(a)(-\omega_B^\wedge)^2 & s\hat{A}_2(a)(-\omega_B^\wedge)^2 & s\hat{A}_3(a)(-\omega_B^\wedge)^2 \\ 0_{3 \times 3} & 0_{3 \times 3} & \hat{A}_1(a)(-\omega_B^\wedge)^3 & \hat{A}_2(a)(-\omega_B^\wedge)^3 & \hat{A}_3(a)(-\omega_B^\wedge)^3 \\ 0_{3 \times 3} & 0_{3 \times 3} & (-\omega_B^\wedge)^4 & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & (-\omega_B^\wedge)^4 & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & (-\omega_B^\wedge)^4 \end{bmatrix}, \quad (3.27)$$

$$\Psi^5 = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} & s\hat{A}_1(a)(-\omega_B^\wedge)^3 & s\hat{A}_2(a)(-\omega_B^\wedge)^3 & s\hat{A}_3(a)(-\omega_B^\wedge)^3 \\ 0_{3 \times 3} & 0_{3 \times 3} & \hat{A}_1(a)(-\omega_B^\wedge)^4 & \hat{A}_2(a)(-\omega_B^\wedge)^4 & \hat{A}_3(a)(-\omega_B^\wedge)^4 \\ 0_{3 \times 3} & 0_{3 \times 3} & (-\omega_B^\wedge)^5 & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & (-\omega_B^\wedge)^5 & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & (-\omega_B^\wedge)^5 \end{bmatrix}. \quad (3.28)$$

By substituting equations (3.25-3.28) in equation (3.24). The matrix $e^{\Psi(s,a,\theta)}$ is given as

$$e^{\Psi(s,a,\theta)} = \begin{bmatrix} I_{3 \times 3} & sI_{3 \times 3} & \Psi_{13} & \Psi_{14} & \Psi_{15} \\ 0_{3 \times 3} & I_{3 \times 3} & \Psi_{23} & \Psi_{24} & \Psi_{25} \\ 0_{3 \times 3} & 0_{3 \times 3} & \Psi_{33} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & \Psi_{44} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & \Psi_{55} \end{bmatrix}. \quad (3.29)$$

Here Ψ_{13} can be calculated as:

$$\begin{aligned} \Psi_{13} &= \frac{1}{2!} s\hat{A}_1(a) + \frac{1}{3!} s\hat{A}_1(a)(-\omega_B^\wedge) + \frac{1}{4!} s\hat{A}_1(a)(-\omega_B^\wedge)^2 \\ &+ \frac{1}{5!} s\hat{A}_1(a)(-\omega_B^\wedge)^3 + \frac{1}{6!} s\hat{A}_1(a)(-\omega_B^\wedge)^4 \\ &+ \frac{1}{7!} s\hat{A}_1(a)(-\omega_B^\wedge)^5 + \frac{1}{8!} s\hat{A}_1(a)(-\omega_B^\wedge)^6 + \dots \end{aligned} \quad (3.30)$$

From (3.3) and (3.30)

3.3. A NOVEL IMU INTEGRATION MODEL USING SWITCHED LINEAR SYSTEMS

$$\begin{aligned}\Psi_{13} &= s\hat{A}_1(a) \left[\frac{I_{3 \times 3}}{2!} + \frac{1}{\|\omega_B\|^3} \sum_{m=1}^{\infty} \frac{(-1)^m}{(2m+1)!} \|\omega_B\|^{(2m+1)} \omega_B^\wedge \right. \\ &\quad \left. + \frac{1}{\|\omega_B\|^4} \sum_{m=2}^{\infty} \frac{(-1)^m}{(2m)!} \|\omega_B\|^{(2m)} (\omega_B^\wedge)^2 \right].\end{aligned}\quad (3.31)$$

For more details of Equation (3.31) derivation, the reader is referred to Appendix E.1.

From (3.4), (3.5) and (3.31)

$$\begin{aligned}\Psi_{13} &= s\hat{A}_1(a) \left[\frac{I_{3 \times 3}}{2!} + \frac{\sin \|\omega_B\| - \|\omega_B\|}{\|\omega_B\|^3} \omega_B^\wedge + \frac{\cos \|\omega_B\| - 1 + \frac{\|\omega_B\|^2}{2}}{\|\omega_B\|^4} (\omega_B^\wedge)^2 \right] \\ &= s\hat{A}_1(a) \Lambda(-\omega_B),\end{aligned}\quad (3.32)$$

where $\Lambda(\omega_B)$ is given as:

$$\Lambda(\omega_B) = \frac{I_{3 \times 3}}{2!} + \frac{\|\omega_B\| - \sin \|\omega_B\|}{\|\omega_B\|^3} \omega_B^\wedge + \frac{2 \cos \|\omega_B\| - 2 + \|\omega_B\|^2}{2\|\omega_B\|^4} (\omega_B^\wedge)^2 \quad (3.33)$$

Similarly, both of Ψ_{14} and Ψ_{15} are gives as:

$$\Psi_{14} = s\hat{A}_2(a) \Lambda(-\omega_B) ; \Psi_{15} = s\hat{A}_3(a) \Lambda(-\omega_B). \quad (3.34)$$

Ψ_{23} can be calculated as:

$$\begin{aligned}\Psi_{23} &= \hat{A}_1(a) + \frac{1}{2!} \hat{A}_1(a) (-\omega_B^\wedge) + \frac{1}{3!} \hat{A}_1(a) (-\omega_B^\wedge)^2 + \frac{1}{4!} \hat{A}_1(a) (-\omega_B^\wedge)^3 \\ &\quad + \frac{1}{5!} \hat{A}_1(a) (-\omega_B^\wedge)^4 + \frac{1}{6!} \hat{A}_1(a) (-\omega_B^\wedge)^5 \\ &\quad + \frac{1}{7!} \hat{A}_1(a) (-\omega_B^\wedge)^6 + \frac{1}{8!} \hat{A}_1(a) (-\omega_B^\wedge)^7 + \dots\end{aligned}\quad (3.35)$$

From (3.3) and (3.35)

$$\begin{aligned}\Psi_{23} &= \hat{A}_1(a) \left[I_{3 \times 3} + \frac{1}{\|\omega_B\|^2} \sum_{m=1}^{\infty} \frac{(-1)^m}{(2m)!} \|\omega_B\|^{(2m)} \omega_B^\wedge \right. \\ &\quad \left. - \frac{1}{\|\omega_B\|^3} \sum_{m=1}^{\infty} \frac{(-1)^m}{(2m+1)!} \|\omega_B\|^{(2m+1)} (\omega_B^\wedge)^2 \right].\end{aligned}\quad (3.36)$$

For more details of Equation (3.36) derivation, the reader is referred to Appendix E.1.

From (3.4), (3.5) and (3.36)

$$\begin{aligned}\Psi_{23} &= \hat{A}_1(a) \left[I_{3 \times 3} + \frac{\cos \|\omega_B\| - 1}{\|\omega_B\|^2} \omega_B^\wedge + \frac{\|\omega_B\| - \sin \|\omega_B\|}{\|\omega_B\|^3} (\omega_B^\wedge)^2 \right] \\ &= \hat{A}_1(a) \Gamma(-\omega_B),\end{aligned}\quad (3.37)$$

where $\Gamma(\omega_B)$ is given as:

$$\Gamma(\omega_B) = I_{3 \times 3} + \frac{1 - \cos \|\omega_B\|}{\|\omega_B\|^2} \omega_B^\wedge + \frac{\|\omega_B\| - \sin \|\omega_B\|}{\|\omega_B\|^3} (\omega_B^\wedge)^2 \quad (3.38)$$

Similarly, both of Ψ_{24} and Ψ_{25} are gives as:

$$\Psi_{24} = \hat{A}_2(a) \Gamma(-\omega_B) ; \quad \Psi_{25} = \hat{A}_3(a) \Gamma(-\omega_B). \quad (3.39)$$

$\Psi_{33} = \Psi_{44} = \Psi_{55}$ can be calculated as:

$$\begin{aligned}\Psi_{33} &= I_{3 \times 3} + (-\omega_B^\wedge) + \frac{1}{2!} (-\omega_B^\wedge)^2 + \frac{1}{3!} (-\omega_B^\wedge)^3 + \frac{1}{4!} (-\omega_B^\wedge)^4 \\ &\quad + \frac{1}{5!} (-\omega_B^\wedge)^5 + \frac{1}{6!} (-\omega_B^\wedge)^6 + \frac{1}{7!} (-\omega_B^\wedge)^7 + \frac{1}{8!} (-\omega_B^\wedge)^8 + \dots\end{aligned}\quad (3.40)$$

From (3.3) and (3.40)

$$\begin{aligned}\Psi_{33} &= I_{3 \times 3} - \frac{1}{\|\omega_B\|} \sum_{m=0}^{\infty} \frac{(-1)^m}{(2m+1)!} \|\omega_B\|^{(2m+1)} \omega_B^\wedge \\ &\quad - \frac{1}{\|\omega\|^2} \sum_{m=1}^{\infty} \frac{(-1)^m}{(2m)!} \|\omega_B\|^{(2m)} (\omega_B^\wedge)^2.\end{aligned}\quad (3.41)$$

3.3. A NOVEL IMU INTEGRATION MODEL USING SWITCHED LINEAR SYSTEMS

For more details of Equation (3.41) derivation, the reader is referred to Appendix E.1.

From (3.4), (3.5) and (3.41)

$$\begin{aligned}\Psi_{33} &= I_{3 \times 3} - \frac{\sin \|\omega_B\|}{\|\omega_B\|} \omega_B^\wedge + \frac{1 - \cos \|\omega_B\|}{\|\omega_B\|^2} (\omega_B^\wedge)^2 \\ &= E(-\omega_B),\end{aligned}\tag{3.42}$$

where $E(\omega_B)$ is given as:

$$\boxed{E(\omega_B) = I_{3 \times 3} + \frac{\sin \|\omega_B\|}{\|\omega_B\|} \omega_B^\wedge + \frac{1 - \cos \|\omega_B\|}{\|\omega_B\|^2} (\omega_B^\wedge)^2}\tag{3.43}$$

Overall, the matrix $e^{\Psi(s,a,\theta)}$ is given as [187, 168]

$$e^{\Psi(s,a,\theta)} = \begin{bmatrix} I_{3 \times 3} & sI_{3 \times 3} & s\hat{A}_1(a)\Lambda(-\theta) & s\hat{A}_2(a)\Lambda(-\theta) & s\hat{A}_3(a)\Lambda(-\theta) \\ 0_{3 \times 3} & I_{3 \times 3} & \hat{A}_1(a)\Gamma(-\theta) & \hat{A}_2(a)\Gamma(-\theta) & \hat{A}_3(a)\Gamma(-\theta) \\ 0_{3 \times 3} & 0_{3 \times 3} & E(-\theta) & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & E(-\theta) & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & E(-\theta) \end{bmatrix},\tag{3.44}$$

where the matrices $E(\theta)$, $\Gamma(\theta)$ and $\Lambda(\theta)$ are defined as (3.43), (3.38) and (3.33). Using the following equation

$$\begin{cases} a_B^T(k)\Lambda(-\theta_B(k))R_i(k) = R_i^T(k)\Lambda(\theta_B(k))a_B(k) \\ a_B^T(k)\Gamma(-\theta_B(k))R_i(k) = R_i^T(k)\Gamma(\theta_B(k))a_B(k) \\ (E(-\theta_B(k))R_i)^T = R_i^T E(\theta_B(k)) \end{cases},\tag{3.45}$$

it can be shown that the equation (3.23) is equivalent to

$$\begin{cases} P_W(k+1) = P_W(k) + V_W(k)\Delta t \\ \quad + (\frac{1}{2}g_W + R_{BW}(k)\Lambda(\theta_B(k))a_B(k))\Delta t^2 \\ V_W(k+1) = V_W(k) + (g_W + R_{BW}(k)\Gamma(\theta_B(k))a_B(k))\Delta t \\ R_{BW}(k+1) = R_{BW}(k)E(\theta_B(k)) \end{cases}.\tag{3.46}$$

Similar to Forster *et al.* [116], we also assume that two video frames are captured at the time instances $k = i$ and $k = j$. Defining a matrix $\hat{A}(k)$ as

$$\hat{A}(k) = \exp^{\Psi(\Delta t, \Delta t a_B(k), \theta_B(k))}, \quad (3.47)$$

and a vector $\hat{b}$ as

$$\hat{b} = \begin{bmatrix} g_W \frac{\Delta t^2}{2} & g_W \Delta t & 0 & 0 & 0 \end{bmatrix}^T, \quad (3.48)$$

the kinematics of the agent between the two video frames can be represented by the following time variant system:

$$\begin{bmatrix} P_W(k+1) \\ V_W(k+1) \\ R_1(k+1) \\ R_2(k+1) \\ R_3(k+1) \end{bmatrix} = \hat{A}(k) \begin{bmatrix} P_W(k) \\ V_W(k) \\ R_1(k) \\ R_2(k) \\ R_3(k) \end{bmatrix} + \hat{b}, \quad (3.49)$$

where $i \leq k \leq j$.

The values of a_B and ω_B may be changed in the next small time interval $[(k+1)\Delta t, (k+2)\Delta t]$. This implies that the matrix $\hat{A}(k)$ could be different from the matrix $\hat{A}(k+1)$. Thus, the time invariant system in the equation (3.49) is a switched linear system [183]. The discrete kinematic model in the equation (3.49) will be used to integrate all the IMU data between the two video frames as follows:

$$\begin{aligned} P_W(j) &= P_W(i) + \Theta(i, j) + R_{BW}(i)\zeta(i, j) \\ V_W(j) &= V_W(i) + g_W \sum_{k=i}^{j-1} \Delta t + R_{BW}(i)\mu(i, j) \\ R_{BW}(j) &= R_{BW}(i)F(i, j) \end{aligned}$$

(3.50)

where the matrix $F(i, j)$ and the vectors $\Theta(i, j)$, $\zeta(i, j)$ and $\mu(i, j)$ are computed as

$$\left\{ \begin{array}{l} F(i, j) = \prod_{k=i}^{j-1} E(\theta_B(k)) \\ \Theta(i, j) = V_W(i) \sum_{k=i}^{j-1} \Delta t + \frac{g_W}{2} \left(\sum_{k=i}^{j-1} \Delta t \right)^2 \\ \zeta(i, j) = \sum_{k=i}^{j-1} (F(i, k) \Lambda(\theta_B(k)) a_B(k) \Delta t^2 + \mu(i, k) \Delta t) \\ \mu(i, j) = \sum_{k=i}^{j-1} F(i, k) \Gamma(\theta_B(k)) a_B(k) \Delta t \end{array} \right. . \quad (3.51)$$

The translation matrix between the two video frames i and j , $T(i, j)$ is finally computed as

$$T(i, j) = \begin{bmatrix} F^T(i, j) & t(i, j) \\ 0 & 1 \end{bmatrix}, \quad (3.52)$$

where the translation vector $t(i, j)$ is $(-F^T(i, j)(\zeta(i, j) + R_{WB}(i)\Theta(i, j)))$.

The novel IMU integration model between two keyframes is stated in (3.50), while the state-of-the-art IMU model is stated in (D.7). For more details of the state-of-the-art IMU model, the reader is referred to Appendix D.

3.4 A Novel IMU Preintegration Measurements

The novel IMU motion integration model in Equation (3.50) has a drawback, since it depends on the position, linear velocity and rotational matrix at t_i . This is because the model in Equation (3.50) needs to repeat its integration whenever the system is linearized at t_i [42]. To avoid repeating the integration again and again whenever the linearized point is changed at t_i , the preintegration principles by Lupton *et al.* [37] will be used to reformulate the model in Equation (3.50). This could be defined when the position, linear velocity and rotational matrix at t_i with the gravity vector are transferred into

the left-hand side. Thus, three relative motion increments that are independent of the position, linear velocity and rotational matrix at t_i are defined as

$$\begin{aligned}
 \Delta R_{BW}(i, j) &\doteq R_{BW}^T(i) R_{BW}(j) = F(i, j) \\
 \Delta V_W(i, j) &\doteq R_{BW}^T(i) (V_W(j) - V_W(i) - g \sum_{k=i}^{j-1} \Delta t) = \mu(i, j) \\
 \Delta P_W(i, j) &\doteq R_{BW}^T(i) (P_W(j) - P_W(i) - \Theta(i, j)) = \zeta(i, j)
 \end{aligned} \tag{3.53}$$

Obviously, the right-hand side in Equation (3.53) becomes independent from the state at t_i . Moreover, since the right-hand side depends only on the IMU measurements, it can be computed between two keyframes directly from the IMU measurements. It should be pointed out that, the *delta* in Equation (3.53) does not imply a physical change in the position ΔP_W and the linear velocity ΔV_W . However, it is a physical change in the case of the rotational matrix ΔR_{BW} .

The measurements, namely $\tilde{a}_B(k)$, and $\tilde{\omega}_B(k) (k \in [i, j])$, are affected by additive white noise η and a slowly varying sensor bias b as explained by Forster *et al.* [116].

$$\begin{cases} \tilde{a}_B(k) = a_B(k) + \bar{b}^a(i) + \eta^a(k) \\ \tilde{\omega}_B(k) = \omega_B(k) + \bar{b}^g(i) + \eta^g(k) \end{cases} . \tag{3.54}$$

By substituting the noisy IMU measurements from Equation (3.54) in the preintegration model in Equation (3.53), it is observed that the right-hand side depends on the noise values in the IMU measurements. Unfortunately, it is very complicated for MAP estimation because MAP estimation requires an accurately defined log-likelihood of the measurements. Please refer to Appendix C for more details of the relationship between MAP and log-likelihood. Thus, the noise will be isolated from the IMU measurements. The IMU bias (i.e., gyro bias b^g and accelerometer bias b^a) are assumed to be constant between two keyframes, while the changing in the bias will be discussed later in section 3.6.

Table 3.1: Coefficients of matrices $E(\theta)$, $\Gamma(\theta)$, $\Lambda(\theta)$ and $\Phi(\theta)$

Matrix	Coefficient of I	Coefficient of $\theta^\wedge$	Coefficient of $(\theta^\wedge)^2$
$E(\theta)$	1	$\frac{h_1(\ \theta\ )}{\ \theta\ }$	$\frac{h_2(\ \theta\ )}{\ \theta\ ^2}$
$\Gamma(\theta)$	1	$\frac{h_2(\ \theta\ )}{\ \theta\ ^2}$	$\frac{h_3(\ \theta\ )}{\ \theta\ ^3}$
$\Lambda(\theta)$	$\frac{1}{2}$	$\frac{h_3(\ \theta\ )}{\ \theta\ ^3}$	$\frac{h_4(\ \theta\ )}{2\ \theta\ ^4}$
$\Phi(\theta)$	$\frac{1}{6}$	$\frac{h_4(\ \theta\ )}{2\ \theta\ ^4}$	$\frac{h_5(\ \theta\ )}{6\ \theta\ ^5}$

3.4.1 Definition

The three matrices $E(\theta)$, $\Gamma(\theta)$ and $\Lambda(\theta)$ are summarized from Equations (3.43), (3.38) and (3.33) in Table 3.1 and the matrix $\Phi(\theta)$ has been derived with same concept of the matrices $E(\theta)$, $\Gamma(\theta)$ and $\Lambda(\theta)$. The four matrices can be expressed with the functions $h_i(z)$ ($i = 1, 2, 3, 4, 5$) being defined as

$$\begin{cases} h_1(z) = \sin z \\ h_2(z) = 1 - \cos z \\ h_3(z) = z - \sin z \\ h_4(z) = 2 \cos z - 2 + z^2 \\ h_5(z) = 6 \sin z - 6z + z^3 \end{cases} . \quad (3.55)$$

The first order approximation of $E(\theta)$, $\Gamma(\theta)$ and $\Lambda(\theta)$ are given as

$$\begin{cases} E(\theta) \approx I_{3 \times 3} + \theta^\wedge \\ \Gamma(\theta) \approx I_{3 \times 3} + \frac{1}{2}\theta^\wedge \\ \Lambda(\theta) \approx \frac{I_{3 \times 3}}{2} + \frac{1}{6}\theta^\wedge \end{cases} , \quad (3.56)$$

and they will be utilized to study the effect of noise and bias. $\text{SO}(3)$ is the group of 3-D rotation matrices, and it is formally defined as

$$SO(3) \doteq \{R \in \mathbb{R}^{3 \times 3}, R^T R = I, \det(R) = 1\}. \quad (3.57)$$

The group operation is the conventional matrix multiplication, and the inverse is the conventional matrix transpose. The group $SO(3)$ is a smooth manifold. The tangent space to the manifold (at the identity) is represented by $SO(3)$, which is usually called the *Lie algebra* and is the same as the space of 3×3 skew symmetric matrices. Let R be a matrix in the set $SO(3)$. It can be derived that

$$\begin{cases} E(R\theta) = RE(\theta)R^T \\ \Gamma(R\theta) = R^T \Gamma(\theta) R \\ \Lambda(R\theta) = R^T \Lambda(\theta) R \end{cases} \quad (3.58)$$

Another exponential map property that will be used later is

$$E(\theta)R = RE(R^T \theta) \quad (3.59)$$

A useful property of the skew matrices is given as

$$a^\wedge b = -b^\wedge a \quad (3.60)$$

Using the first-order approximation, it can be obtained that

$$\begin{cases} E(\theta + \delta\theta) \approx E(\theta)E(\Gamma(-\theta)\delta\theta) \\ \Gamma(\theta + \delta\theta) \approx \Gamma(\theta)\Gamma(\Lambda(-\theta)\delta\theta) \\ \Lambda(\theta + \delta\theta) \approx 2\Lambda(\theta)\Lambda(\Phi(-\theta)\delta\theta) \end{cases} \quad (3.61)$$

Defining an operation $Log(\cdot)$ as given by Forster *et al.* [116]

$$Log(E(\theta)) = \theta, \quad (3.62)$$

it can be derived that

$$\text{Log}(E(\theta)E(\delta\theta)) \approx \theta + \Gamma^{-1}(\theta)\delta\theta, \quad (3.63)$$

where $\Gamma^{-1}(\theta)$ is given in (3.64)

$$\Gamma^{-1} = I_{3 \times 3} - \frac{h_2(\|\theta\|)(h_1(\|\theta\|) + h_3(\|\theta\|))}{\|\theta\|(h_1^2(\|\theta\|) + h_2^2(\|\theta\|))}\theta^\wedge + \frac{h_2^2(\|\theta\|) - h_1(\|\theta\|)h_3(\|\theta\|)}{\|\theta\|^2(h_1^2(\|\theta\|) + h_2^2(\|\theta\|))}(\theta^\wedge)^2 \quad (3.64)$$

Let $(\tilde{a}_B(k) - \bar{b}^a(i))$ and $(\tilde{\omega}_B(k) - \bar{b}^g(i))$ in Equation (3.54) be denoted as $\bar{a}_B(k)$ and $\bar{\omega}_B(k)$, respectively. This is followed by replacing $a_B(k)$ and $\omega_B(k)$ by $\bar{a}_B(k)$ and $\bar{\omega}_B(k)$ in Equation (3.51). Then, in the next sub-sections the noise from the preintegration measurements will be isolated.

3.4.2 Preintegrated Rotational Measurements

The preintegrated rotational measurements from Equation (3.53) can be expressed as the following:

$$\begin{aligned} \Delta R_{BW}(i, j) &= \prod_{k=i}^{j-1} E(\theta_B(k)) \\ &\stackrel{\text{eq. (3.54)}}{=} \prod_{k=i}^{j-1} E((\tilde{\omega}_B(k) - \bar{b}^g(i) - \eta^g(k))\Delta t) \\ &\stackrel{\text{eq. (3.61)}}{\approx} \prod_{k=i}^{j-1} E(\bar{\theta}_B(k)) E(-\Gamma(-\bar{\theta}_B(k))\eta^g(k)\Delta t) \\ &\stackrel{\text{eq. (3.59)}}{=} \bar{F}(i, j) \prod_{k=i}^{j-1} E(-\bar{F}^T(k+1, j)\Gamma(-\bar{\theta}_B(k))\eta^g(k)\Delta t) \\ &\doteq \bar{F}(i, j) E(-\delta\phi_{BW}^\eta(i, j)), \end{aligned} \quad (3.65)$$

where $\bar{F}(i, j)$ is the *preintegrated rotational measurements* and *its noise* $\delta\phi_{BW}^\eta$. It can be derived that

$$\begin{aligned}
 \delta\phi_{BW}^\eta(i, j) &= -\text{Log} \left(\prod_{k=i}^{j-1} E(-\bar{F}^T(k+1, j)\Gamma(-\bar{\theta}_B(k))\eta^g(k)\Delta t) \right) \\
 &\stackrel{\text{eq. (3.63)}}{\approx} \sum_{k=i}^{j-1} \bar{F}^T(k+1, j)\Gamma(-\bar{\theta}_B(k))\eta^g(k)\Delta t.
 \end{aligned} \tag{3.66}$$

3.4.3 Preintegrated Velocity Measurements

The preintegrated rotational measurements from Equation (3.53) can be expressed as the following:

$$\begin{aligned}
 \Delta V_W(i, j) &= \sum_{k=i}^{j-1} F(i, k)\Gamma(\theta_B(k))a_B(k)\Delta t \\
 &\stackrel{\text{eq. (3.54), (3.65)}}{\approx} \sum_{k=i}^{j-1} \bar{F}(i, k)E(-\delta\phi_{BW}^\eta(i, k))\Gamma(\bar{\theta}_B(k) - \eta^g(k)\Delta t) \\
 &\quad (\bar{a}_B(k) - \eta^a(k))\Delta t \\
 &\stackrel{\text{eq. (3.61)}}{\approx} \sum_{k=i}^{j-1} \bar{F}(i, k)E(-\delta\phi_{BW}^\eta(i, k))\Gamma(\bar{\theta}_B(k))\Gamma(-\Lambda(-\bar{\theta}_B(k))\eta^g(k)\Delta t) \\
 &\quad (\bar{a}_B(k) - \eta^a(k))\Delta t \\
 &\stackrel{\text{eq. (3.56)}}{\approx} \sum_{k=i}^{j-1} \bar{F}(i, k)(I_{3 \times 3} - \delta\phi_{BW}^\eta(i, k)^\wedge)\Gamma(\bar{\theta}_B(k))(I_{3 \times 3} - \frac{1}{2}(\Lambda(-\bar{\theta}_B(k))\eta^g(k)\Delta t)^\wedge) \\
 &\quad (\bar{a}_B(k) - \eta^a(k))\Delta t \\
 &\approx \sum_{k=i}^{j-1} [\bar{F}(i, k)\Gamma(\bar{\theta}_B(k))\bar{a}_B(k)\Delta t - \bar{F}(i, k)\Gamma(\bar{\theta}_B(k))\eta^a(k)\Delta t \\
 &\quad + \bar{F}(i, k)(-\delta\phi_{BW}^\eta(i, k)^\wedge)\Gamma(\bar{\theta}_B(k))\bar{a}_B(k)\Delta t \\
 &\quad + \bar{F}(i, k)\Gamma(\bar{\theta}_B(k))(-\frac{1}{2}(\Lambda(-\bar{\theta}_B(k))\eta^g(k)\Delta t)^\wedge)\bar{a}_B(k)\Delta t] \\
 &\stackrel{\text{eq. (3.60)}}{=} \sum_{k=i}^{j-1} \bar{F}(i, k)\Gamma(\bar{\theta}_B(k))\bar{a}_B(k)\Delta t \\
 &\quad - \sum_{k=i}^{j-1} \bar{F}(i, k)\Gamma(\bar{\theta}_B(k))\eta^a(k)\Delta t \\
 &\quad + \sum_{k=i}^{j-1} \bar{F}(i, k)(\Gamma(\bar{\theta}_B(k))\bar{a}_B(k))^\wedge \delta\phi_{BW}^\eta(i, k)\Delta t
 \end{aligned}$$

$$\begin{aligned}
 & + \frac{1}{2} \sum_{k=i}^{j-1} \bar{F}(i, k) \Gamma(\bar{\theta}_B(k)) (\bar{a}_B(k))^\wedge \Lambda(-\bar{\theta}_B(k)) \eta^g(k) \Delta t^2 \\
 & \doteq \bar{\mu}(i, j) - \delta V_W^\eta(i, j),
 \end{aligned} \tag{3.67}$$

where $\bar{\mu}(i, j)$ is the preintegrated velocity measurements and its noise $\delta V_W^\eta(i, j)$.

3.4.4 Preintegrated Position Measurements

The preintegrated rotational measurements from Equation (3.53) can be expressed as the following:

$$\begin{aligned}
 \Delta P_W(i, j) &= \sum_{k=i}^{j-1} [F(i, k) \Lambda(\theta_B(k)) a_B(k) \Delta t^2 + \mu(i, k) \Delta t] \\
 &\stackrel{\text{eqs. (3.54), (3.65), (3.67)}}{\approx} \sum_{k=i}^{j-1} [\bar{F}(i, k) E(-\delta \phi_{BW}^\eta(i, k)) \Lambda(\bar{\theta}_B(k) - \eta^g(k) \Delta t) \\
 &\quad (\bar{a}_B(k) - \eta^a(k)) \Delta t^2 + (\bar{\mu}(i, k) + \delta V_W^\eta(i, k)) \Delta t] \\
 &\stackrel{\text{eq. (3.61)}}{\approx} \sum_{k=i}^{j-1} [\bar{F}(i, k) E(-\delta \phi_{BW}^\eta(i, k)) 2\Lambda(\bar{\theta}_B(k)) \Lambda(-\Phi(-\bar{\theta}_B(k)) \eta^g(k) \Delta t) \\
 &\quad (\bar{a}_B(k) - \eta^a(k)) \Delta t^2 + (\bar{\mu}(i, k) + \delta V_W^\eta(i, k)) \Delta t] \\
 &\stackrel{\text{eq. (3.56)}}{\approx} \sum_{k=i}^{j-1} \left[\bar{F}(i, k) (I - \delta \phi_{BW}^\eta(i, k)^\wedge) \Lambda(\bar{\theta}_B(k)) (I - \frac{1}{3} (\Phi(-\bar{\theta}_B(k)) \eta^g(k) \Delta t)^\wedge) \right. \\
 &\quad \left. (\bar{a}_B(k) - \eta^a(k)) \Delta t^2 + (\bar{\mu}(i, k) + \delta V_W^\eta(i, k)) \Delta t \right] \\
 &\approx \sum_{k=i}^{j-1} [\bar{F}(i, k) \Lambda(\bar{\theta}_B(k)) \bar{a}_B(k) \Delta t^2 - \bar{F}(i, k) \Lambda(\bar{\theta}_B(k)) \eta^a(k) \Delta t^2 \\
 &+ \bar{F}(i, k) (-\delta \phi_{BW}^\eta(i, k)^\wedge) \Lambda(\bar{\theta}_B(k)) \bar{a}_B(k) \Delta t^2 \\
 &+ \bar{F}(i, k) \Lambda(\bar{\theta}_B(k)) (-\frac{1}{3} (\Phi(-\bar{\theta}_B(k)) \eta^g(k) \Delta t)^\wedge) \bar{a}_B(k) \Delta t^2 \\
 &+ (\bar{\mu}(i, k) + \delta V_W^\eta(i, k)) \Delta t] \\
 &\stackrel{\text{eq. (3.60)}}{=} \sum_{k=i}^{j-1} \bar{F}(i, k) \Lambda(\bar{\theta}_B(k)) \bar{a}_B(k) \Delta t^2 + \sum_{k=i}^{j-1} \bar{\mu}(i, k) \Delta t \\
 &- \sum_{k=i}^{j-1} \bar{F}(i, k) \Lambda(\bar{\theta}_B(k)) \eta^a(k) \Delta t^2
 \end{aligned}$$

$$\begin{aligned}
 & + \sum_{k=i}^{j-1} \bar{F}(i, k) (\Lambda(\bar{\theta}_B(k)) \bar{a}_B(k))^\wedge \delta \phi_{BW}^\eta(i, k) \Delta t^2 \\
 & + \frac{1}{3} \sum_{k=i}^{j-1} \bar{F}(i, k) \Lambda(\bar{\theta}_B(k)) (\bar{a}_B(k))^\wedge \Phi(-\bar{\theta}_B(k)) \eta^g(k) \Delta t^3 \\
 & + \sum_{k=i}^{j-1} \delta V_W^\eta(i, k) \Delta t \\
 & \doteq \bar{\zeta}(i, j) - \delta P_W^\eta(i, j),
 \end{aligned} \tag{3.68}$$

where $\bar{\zeta}(i, j)$ is the preintegrated position measurements and its noise $\delta P_W^\eta(i, j)$.

Obviously, from Equations (3.65)-(3.68), it shows that the mean value of the preintegrated measurements, e.g. $\bar{F}(i, j)$, $\bar{\mu}(i, j)$ and $\bar{\zeta}(i, j)$, in closed form solution with their separate noise. Thus, the preintegration measurements between two keyframes can be expressed with the mean preintegrated measurements and their noise as the following:

$$\begin{aligned}
 \Delta R_{BW}(i, j) &= \bar{F}(i, j) E(-\delta \phi_{BW}^\eta(i, j)) \\
 \Delta V_W(i, j) &= \bar{\mu}(i, j) - \delta V_W^\eta(i, j) \\
 \Delta P_W(i, j) &= \bar{\zeta}(i, j) - \delta P_W^\eta(i, j)
 \end{aligned} \tag{3.69}$$

where the mean preintegrated measurements can be summarized and expressed as the following:

$$\begin{aligned}
 \bar{F}(i, j) &= \prod_{k=i}^{j-1} E(\bar{\theta}_B(k)) \\
 \bar{\mu}(i, j) &= \sum_{k=i}^{j-1} \bar{F}(i, k) \Gamma(\bar{\theta}_B(k)) \bar{a}_B(k) \Delta t \\
 \bar{\zeta}(i, j) &= \sum_{k=i}^{j-1} \bar{F}(i, k) \Lambda(\bar{\theta}_B(k)) \bar{a}_B(k) \Delta t^2 + \sum_{k=i}^{j-1} \bar{\mu}(i, k) \Delta t
 \end{aligned} \tag{3.70}$$

while the preintegrated noise can be expressed as:

$$\begin{aligned}
 \delta\phi_{BW}^\eta(i, j) &= \sum_{k=i}^{j-1} \bar{F}^T(k+1, j) \Gamma(-\bar{\theta}_B(k)) \eta^g(k) \Delta t \\
 \delta V_W^\eta(i, j) &= - \sum_{k=i}^{j-1} \bar{F}(i, k) (\Gamma(\bar{\theta}_B(k)) \bar{a}_B(k))^\wedge \delta\phi_{BW}^\eta(i, k) \Delta t \\
 &\quad - \frac{1}{2} \sum_{k=i}^{j-1} \bar{F}(i, k) \Gamma(\bar{\theta}_B(k)) (\bar{a}_B(k))^\wedge \Lambda(-\bar{\theta}_B(k)) \eta^g(k) \Delta t^2 \\
 &\quad + \sum_{k=i}^{j-1} \bar{F}(i, k) \Gamma(\bar{\theta}_B(k)) \eta^a(k) \Delta t \\
 \delta P_W^\eta(i, j) &= - \sum_{k=i}^{j-1} \bar{F}(i, k) (\Lambda(\bar{\theta}_B(k)) \bar{a}_B(k))^\wedge \delta\phi_{BW}^\eta(i, k) \Delta t^2 \\
 &\quad - \frac{1}{3} \sum_{k=i}^{j-1} \bar{F}(i, k) \Lambda(\bar{\theta}_B(k)) (\bar{a}_B(k))^\wedge \Phi(-\bar{\theta}_B(k)) \eta^g(k) \Delta t^3 \\
 &\quad + \sum_{k=i}^{j-1} \bar{F}(i, k) \Lambda(\bar{\theta}_B(k)) \eta^a(k) \Delta t^2 + \sum_{k=i}^{j-1} \delta V_W^\eta(i, k) \Delta t
 \end{aligned} \tag{3.71}$$

It is clearly that, the preintegrated measurements expressed in Equation (3.69) has an advantage over the the preintegrated measurements expressed in Equation (3.53). The separated expression of the noise makes the definition of the log-likelihood is straightforward whereby the noise is considered to be a normal distribution with zero-mean.

3.5 A Novel Iterative Noise Propagation

This section aims to write the preintegrated noise propagation in iterative form. So, the covariance matrix (Σ) of the preintegrated measurements can be obtained which it is crucial and has strong influence on the MAP estimator.

3.5.1 Preintegrated Rotational Noise

Simply, the last term ($k = j$) has been taken out of summation of the preintegrated rotational noise $\delta\phi_{BW}^\eta$. Accordingly, the preintegrated in Equation (3.71) can be written

in iterative way as following:

$$\begin{aligned}
 \delta\phi_{BW}^\eta(i, j+1) &\doteq \sum_{k=i}^j \bar{F}^T(k+1, j+1) \Gamma(-\bar{\theta}_B(k)) \eta^g(k) \Delta t \\
 &= \bar{F}^T(j, j+1) \underline{\delta\phi_{BW}^\eta(i, j)} + \Gamma(-\bar{\theta}_B(j)) \Delta t \underline{\eta^g(j)}
 \end{aligned} \tag{3.72}$$

For more details of Equation (3.72) derivation, the reader is referred to Appendix E.2.1.

3.5.2 Preintegrated Velocity Noise

By repeating same process in Equation (3.72) for δV_B^η as follows:

$$\begin{aligned}
 \delta V_W^\eta(i, j+1) &\doteq - \sum_{k=i}^j \bar{F}(i, k) (\Gamma(\bar{\theta}_B(k)) \bar{a}_B(k))^\wedge \delta\phi_{BW}^\eta(i, k) \Delta t \\
 &\quad - \frac{1}{2} \sum_{k=i}^j \bar{F}(i, k) \Gamma(\bar{\theta}_B(k)) (\bar{a}_B(k))^\wedge \Lambda(-\bar{\theta}_B(k)) \eta^g(k) \Delta t^2 \\
 &\quad + \sum_{k=i}^j \bar{F}(i, k) \Gamma(\bar{\theta}_B(k)) \eta^a(k) \Delta t \\
 &= \underline{\delta V_W^\eta(i, j)} - \bar{F}(i, j) (\Gamma(\bar{\theta}_B(j)) \bar{a}_B(j))^\wedge \Delta t \underline{\delta\phi_{BW}^\eta(i, j)} \\
 &\quad - \frac{1}{2} \bar{F}(i, j) \Gamma(\bar{\theta}_B(j)) (\bar{a}_B(j))^\wedge \Lambda(-\bar{\theta}_B(j)) \Delta t^2 \underline{\eta^g(j)} \\
 &\quad + \bar{F}(i, j) \Gamma(\bar{\theta}_B(j)) \Delta t \underline{\eta^a(j)}
 \end{aligned} \tag{3.73}$$

For more details of Equation (3.73) derivation, the reader is referred to Appendix E.2.2.

3.5.3 Preintegrated Position Noise

By repeating same process in Equation (3.72) for δP_B^η as follows:

$$\begin{aligned}
 \delta P_W^\eta(i, j+1) &\doteq - \sum_{k=i}^j \bar{F}(i, k) (\Lambda(\bar{\theta}_B(k)) \bar{a}_B(k))^\wedge \delta \phi_{BW}^\eta(i, k) \Delta t^2 \\
 &\quad - \frac{1}{3} \sum_{k=i}^j \bar{F}(i, k) \Lambda(\bar{\theta}_B(k)) (\bar{a}_B(k))^\wedge \Phi(-\bar{\theta}_B(k)) \eta^g(k) \Delta t^3 \\
 &\quad + \sum_{k=i}^j \bar{F}(i, k) \Lambda(\bar{\theta}_B(k)) \eta^a(k) \Delta t^2 + \sum_{k=i}^j \delta V_W^\eta(i, k) \Delta t \\
 &= \underline{\delta P_W^\eta(i, j)} + \Delta t \underline{\delta V_W^\eta(i, j)} \\
 &\quad - \bar{F}(i, j) (\Lambda(\bar{\theta}_B(j)) \bar{a}_B(j))^\wedge \Delta t^2 \underline{\delta \phi_{BW}^\eta(i, j)} \\
 &\quad - \frac{1}{3} \bar{F}(i, j) \Lambda(\bar{\theta}_B(j)) (\bar{a}_B(j))^\wedge \Phi(-\bar{\theta}_B(j)) \Delta t^3 \underline{\eta^g(j)} \\
 &\quad + \bar{F}(i, j) \Lambda(\bar{\theta}_B(j)) \Delta t^2 \underline{\eta^a(j)}
 \end{aligned} \tag{3.74}$$

For more details of Equation (3.74) derivation, the reader is referred to Appendix E.2.3.

3.5.4 Iterative Noise Form And Covariance Matrix

The preintegrated noise vector can be represented as the following with IMU manufacture specification:

$$\begin{aligned}
 \eta_W(i, j) &= [(\delta \phi_{BW}^\eta(i, j))^T, (\delta V_W^\eta(i, j))^T, (\delta P_W^\eta(i, j))^T]^T \\
 \eta_{IMU}(k) &= [(\eta^g(k))^T, (\eta^a(k))^T]^T
 \end{aligned} \tag{3.75}$$

Equations (3.72), (3.73) and (3.74), the noise propagation, can be represented in compact matrix form as the following:

$$\begin{bmatrix} \delta \phi_{BW}^\eta(i, j+1) \\ \delta V_W^\eta(i, j+1) \\ \delta P_W^\eta(i, j+1) \end{bmatrix} = A(j) \begin{bmatrix} \delta \phi_{BW}^\eta(i, j) \\ \delta V_W^\eta(i, j) \\ \delta P_W^\eta(i, j) \end{bmatrix} + B(j) \begin{bmatrix} \eta^g(j) \\ \eta^a(j) \end{bmatrix} \tag{3.76}$$

Defining a (3×3) matrix $A(j)$ as

$$\begin{aligned}
 A_{11}(j) &= \bar{F}^T(j, j+1) \\
 A_{12}(j) &= A_{13}(j) = 0_{3 \times 3} \\
 A_{21}(j) &= -\bar{F}(i, j)(\Gamma(\bar{\theta}_B(j))\bar{a}_B(j))^{\wedge} \Delta t \\
 A_{22}(j) &= I_{3 \times 3} \\
 A_{23}(j) &= 0_{3 \times 3} \\
 A_{31}(j) &= -\bar{F}(i, j)(\Lambda(\bar{\theta}_B(j))\bar{a}_B(j))^{\wedge} \Delta t^2 \\
 A_{32}(j) &= I_{3 \times 3} \Delta t \\
 A_{33}(j) &= I_{3 \times 3}
 \end{aligned} \tag{3.77}$$

and a (3×2) matrix $B(j)$ as

$$\begin{aligned}
 B_{11}(j) &= \Gamma(-\bar{\theta}_B(j)) \Delta t \\
 B_{12}(j) &= 0_{3 \times 3} \\
 B_{21}(j) &= -\frac{1}{2} \bar{F}(i, j) \Gamma(\bar{\theta}_B(j)) (\bar{a}_B(j))^{\wedge} \Lambda(-\bar{\theta}_B(j)) \Delta t^2 \\
 B_{22}(j) &= \bar{F}(i, j) \Gamma(\bar{\theta}_B(j)) \Delta t \\
 B_{31}(j) &= -\frac{1}{3} \bar{F}(i, j) \Lambda(\bar{\theta}_B(j)) (\bar{a}_B(j))^{\wedge} \Phi(-\bar{\theta}_B(j)) \Delta t^3 \\
 B_{32}(j) &= \bar{F}(i, j) \Lambda(\bar{\theta}_B(j)) \Delta t^2
 \end{aligned} \tag{3.78}$$

Subsequently, the preintegrated measurements covariance can be computed iteratively in the closed-form as

$$\boxed{\Sigma(i, j+1) = A(j) \Sigma(i, j) A^T(j) + B(j) \Sigma_{\eta}(j) B^T(j)} \tag{3.79}$$

Where the $\Sigma_{\eta} \in \mathbb{R}^{6 \times 6}$ is the raw IMU measurements noise η_{IMU}

A great advantage from computing the covariance matrix in an iterative manner is that, when a new IMU measurement is received, *the covariance matrix is updated immediately rather than calculating from scratch*. Such an iterative form is less compute intensive and hence is suitable for online estimation.

3.6 A Novel Analytic Biases Update

Sensor bias is the small offset in the average measurement output compared to the actual measurement or the ground truth. In the previous section, the bias is assumed to be constant during the video keyframes i and j . However, in practice, the bias will be changed by a small amount. The measurements can then modeled by

$$\begin{cases} \tilde{a}_B(k) = a_B(k) + \bar{b}^a(i) + \delta b^a(i) \\ \tilde{\omega}_B(k) = \omega_B(k) + \bar{b}^g(i) + \delta b^g(i) \end{cases}. \quad (3.80)$$

Thus, we are trying to find a expression of preintegration measurements update when there are changes in the bias estimation. The main idea is to find a small correction which it can be added to the old the preintegrated measurements. So, the new bias $\hat{b}(i)$ is equal to the old bias $\bar{b}(i)$ plus small update bias correction $\delta b(i)$.

$$\begin{aligned} \hat{b}^g(i) &= \bar{b}^g(i) + \delta b^g(i) \\ \hat{b}^a(i) &= \bar{b}^a(i) + \delta b^a(i) \end{aligned} \quad (3.81)$$

By substituting the new value of bias estimation in Equation (3.81) in preintegration model in Equation (3.51), the preintegrated measurements can be updated.

3.6.1 Preintegrated Rotational Measurements via Bias Correction

$$\begin{aligned} \Delta R_{BW}(i, j) &\stackrel{\text{eq. (3.81)}}{=} \prod_{k=i}^{j-1} E((\tilde{\omega}_B(k) - \bar{b}^g(i) - \delta b^g(i))\Delta t) \\ &= \bar{F}(i, j) E\left(\frac{\partial \bar{F}(i, j)}{\partial b^g} \delta b^g(i)\right) \\ &\doteq \bar{F}(i, j) E(\delta \phi_{BW}^b(i, j)) \end{aligned} \quad (3.82)$$

For more details of Equation (3.82) derivation, the reader is referred to Appendix E.3.1.

3.6.2 Preintegrated Velocity Measurements via Bias Correction

$$\begin{aligned}
 \Delta V_W(i, j) &\stackrel{\text{eq. (3.81), (3.82)}}{=} \sum_{k=i}^{j-1} \bar{F}(i, k) E(\delta \phi_{BW}^b(i, k)) \Gamma(\bar{\theta}_B(k) - \delta b^g(i) \Delta t) \\
 &\quad (\bar{a}_B(k) - \delta b^a(i)) \Delta t \\
 &\doteq \bar{\mu}(i, j) + \frac{\partial \bar{V}(i, j)}{\partial b^a} \delta b^a(i) + \frac{\partial \bar{V}(i, j)}{\partial b^g} \delta b^g(i) \\
 &\doteq \bar{\mu}(i, j) + \delta V_W^b(i, j).
 \end{aligned} \tag{3.83}$$

For more details of Equation (3.83) derivation, the reader is referred to Appendix E.3.2.

3.6.3 Preintegrated Position Measurements via Bias Correction

$$\begin{aligned}
 \Delta P_W(i, j) &\stackrel{\text{eq. (3.81), (3.83)}}{=} \sum_{k=i}^{j-1} [\bar{F}(i, k) E(\delta \phi_{BW}^b(i, k)) \Lambda(\bar{\theta}_B(k) - \delta b^g(i) \Delta t) \\
 &\quad (\bar{a}_B(k) - \delta b^a(i)) \Delta t^2 + (\bar{\mu}(i, k) + \delta V_W^b(i, k)) \Delta t] \\
 &\doteq \bar{\zeta}(i, j) + \frac{\partial \bar{P}(i, j)}{\partial b^a} \delta b^a(i) + \frac{\partial \bar{P}(i, j)}{\partial b^g} \delta b^g(i) \\
 &\doteq \bar{\zeta}(i, j) + \delta P_W^b(i, j)
 \end{aligned} \tag{3.84}$$

For more details of Equation (3.84) derivation, the reader is referred to Appendix E.3.3.

Accordingly, the preintegrated measurements can be expressed by the mean preintegrated measurements plus the change in measurements due to bias changing, whereby the Jacobians $\left\{ \frac{\partial \bar{F}(i, j)}{\partial b^g}, \frac{\partial \bar{V}(i, j)}{\partial b^g}, \dots \right\}$ describe the measurements changing due to bias changing. Thus, from Equations (3.82)-(3.84), the preintegrated measurements with bias correction update can be summarized as the following:

$$\begin{aligned}
 \Delta R_{BW}(i, j) &= \bar{F}(i, j) E \left(\frac{\partial \bar{F}(i, j)}{\partial b^g} \delta b^g(i) \right) \\
 \Delta V_W(i, j) &= \bar{\mu}(i, j) + \frac{\partial \bar{V}(i, j)}{\partial b^a} \delta b^a(i) + \frac{\partial \bar{V}(i, j)}{\partial b^g} \delta b^g(i) \\
 \Delta P_W(i, j) &= \bar{\zeta}(i, j) + \frac{\partial \bar{P}(i, j)}{\partial b^a} \delta b^a(i) + \frac{\partial \bar{P}(i, j)}{\partial b^g} \delta b^g(i)
 \end{aligned} \tag{3.85}$$

where the Jacobian matrices with respect to the bias $\left\{ \frac{\partial \bar{F}(i, j)}{\partial b^g}, \frac{\partial \bar{V}(i, j)}{\partial b^g}, \dots \right\}$ can be expressed in the closed-form as

$$\begin{aligned}
 \frac{\partial \bar{F}(i, j)}{\partial b^g} &= - \sum_{k=i}^{j-1} \bar{F}^T(k+1, j) \Gamma(-\bar{\theta}_B(k)) \Delta t \\
 \frac{\partial \bar{V}(i, j)}{\partial b^g} &= - \sum_{k=i}^{j-1} \bar{F}(i, k) (\Gamma(\bar{\theta}_B(k)) \bar{a}_B(k))^\wedge \left(\frac{\partial \bar{F}(i, k)}{\partial b^g} \right) \Delta t \\
 &\quad + \frac{1}{2} \sum_{k=i}^{j-1} \bar{F}(i, k) \Gamma(\bar{\theta}_B(k)) (\bar{a}_B(k))^\wedge \Lambda(-\bar{\theta}_B(k)) \Delta t^2 \\
 \frac{\partial \bar{V}(i, j)}{\partial b^a} &= - \sum_{k=i}^{j-1} \bar{F}(i, k) \Gamma(\bar{\theta}_B(k)) \Delta t \\
 \frac{\partial \bar{P}(i, j)}{\partial b^g} &= - \sum_{k=i}^{j-1} \bar{F}(i, k) (\Lambda(\bar{\theta}_B(k)) \bar{a}_B(k))^\wedge \left(\frac{\partial \bar{F}(i, k)}{\partial b^g} \right) \Delta t^2 \\
 &\quad + \frac{1}{3} \sum_{k=i}^{j-1} \bar{F}(i, k) \Lambda(\bar{\theta}_B(k)) (\bar{a}_B(k))^\wedge \Phi(-\bar{\theta}_B(k)) \Delta t^3 \\
 &\quad + \sum_{k=i}^{j-1} \left(\frac{\partial \bar{V}(i, k)}{\partial b^g} \right) \Delta t \\
 \frac{\partial \bar{P}(i, j)}{\partial b^a} &= - \sum_{k=i}^{j-1} \bar{F}(i, k) \Lambda(\bar{\theta}_B(k)) \Delta t^2 + \sum_{k=i}^{j-1} \left(\frac{\partial \bar{V}(i, k)}{\partial b^a} \right) \Delta t
 \end{aligned} \tag{3.86}$$

It can be shown from Equations (3.77), (3.78) and (3.86) that the proposed IMU factor provides a closed-form solution for the state transition matrix in a discrete manner. Thus, the covariance matrix and Jacobian are more accurate than that obtained by Forster *et al.* and Qin *et al.* [116, 163].

3.7 A Novel Preintegrated IMU Factor

A novel preintegrated IMU factor $\left\{ \sum_{i,j \in \mathcal{R}(k)} \|r_{L(i,j)}\|_{\Sigma_{ij}}^2 \right\}$ can be expressed in terms of the preintegrated measurements in Equation (3.69), and their noise is represented by zero-mean Gaussian distribution with covariance matrix in Equation(3.79). Thus, the residual errors can be written as $r_{\Delta} = [r_{\Delta R(i,j)}^T, r_{\Delta V(i,j)}^T, r_{\Delta P(i,j)}^T]^T \in \mathbb{R}^9$ which also included the bias estimation update from Equation (3.85) as follows:

$$\begin{aligned}
 r_{\Delta R(i,j)} &= \text{Log}((\bar{F}(i,j)E(\frac{\partial F(i,j)}{\partial b^g(i)}\delta b^g(i)))^T R_{BW}^T(i)R_{BW}(j)) \\
 r_{\Delta V(i,j)} &= R_{BW}^T(i)(V_W(j) - V_W(i) - g \sum_{k=i}^{j-1} \Delta t) \\
 &\quad - (\bar{\mu}(i,j) + \frac{\partial \bar{V}(i,j)}{\partial b^a} \delta b^a(i) + \frac{\partial \bar{V}(i,j)}{\partial b^g} \delta b^g(i)) \\
 r_{\Delta P(i,j)} &= R_{BW}^T(i)(P_W(j) - P_W(i) - \Theta(i,j)) \\
 &\quad - (\bar{\zeta}(i,j) + \frac{\partial \bar{P}(i,j)}{\partial b^a} \delta b^a(i) + \frac{\partial \bar{P}(i,j)}{\partial b^g} \delta b^g(i))
 \end{aligned} \tag{3.87}$$

3.7.1 Jacobian of Residual Errors

In this subsection, analytic expressions are provided for the Jacobian matrices of the residual errors. These Jacobians are crucial when using iterative optimization techniques to minimize the computational cost. According to the lift-solve-retract method, the retraction functions from Equations (3.12) and (3.13) are used to reparameterize Equation (3.87) at each Gauss-Newton iteration. The "solve" step requires the use of first-order Taylor expansion to linearize the resulting cost function around the current estimate. "Lifting" the cost function consists of substituting the following retractions:

$$\left\{ \begin{array}{l} R_{BW}(i) \leftarrow R_{BW}(i)E(\delta\phi(i)) \\ R_{BW}(j) \leftarrow R_{BW}(j)E(\delta\phi(j)) \\ P_W(i) \leftarrow P_W(i) + R_{BW}(i)\delta P(i) \\ P_W(j) \leftarrow P_W(j) + R_{BW}(j)\delta P(j) \\ V_W(i) \leftarrow V_W(i) + \delta V(i) \\ V_W(j) \leftarrow V_W(j) + \delta V(j) \\ \delta b^g(i) \leftarrow \delta b^g(i) + \tilde{\delta}b^g(i) \\ \delta b^a(i) \leftarrow \delta b^a(i) + \tilde{\delta}b^a(i) \end{array} \right. . \quad (3.88)$$

3.7.2 Jacobians of Position Residual Error ($r_{\Delta P(i,j)}$)

$$\begin{aligned} r_{\Delta P(i,j)} &= R_{BW}^T(i)(P_W(j) - P_W(i) - \Theta(i, j)) \\ &\quad - \bar{\zeta}(i, j) - \frac{\partial \zeta(i,j)}{\partial b^g(i)} \delta b^g(i) - \frac{\partial \zeta(i,j)}{\partial b^a(i)} \delta b^a(i) \end{aligned} \quad (3.89)$$

the Jacobians of $r_{\Delta P(i,j)}$ are computed as

$$\boxed{\begin{aligned} \frac{\partial r_{\Delta P(i,j)}}{\partial \delta\phi(i)} &= (R_{BW}^T(i)(P_W(j) - P_W(i) - \Theta(i, j)))^\wedge \\ \frac{\partial r_{\Delta P(i,j)}}{\partial \delta\phi(j)} &= 0_{3 \times 1} \\ \frac{\partial r_{\Delta P(i,j)}}{\partial \delta P(i)} &= -I_{3 \times 1} \\ \frac{\partial r_{\Delta P(i,j)}}{\partial \delta P(j)} &= R_{BW}^T(i)R_{BW}(j) \\ \frac{\partial r_{\Delta P(i,j)}}{\partial \delta V(i)} &= -R_{BW}^T(i) \sum_{k=i}^{j-1} \Delta t \\ \frac{\partial r_{\Delta P(i,j)}}{\partial \delta V(j)} &= 0_{3 \times 1} \\ \frac{\partial r_{\Delta P(i,j)}}{\partial \tilde{\delta}b^a(i)} &= -\frac{\partial \bar{P}(i,j)}{\partial b^a} \\ \frac{\partial r_{\Delta P(i,j)}}{\partial \tilde{\delta}b^g(i)} &= -\frac{\partial \bar{P}(i,j)}{\partial b^g} \end{aligned}} \quad (3.90)$$

For more details of Equation (3.90) derivation, the reader is referred to Appendix E.4.1.

3.7.3 Jacobians of Velocity Residual Error ($r_{\Delta V(i,j)}$)

$$\begin{aligned}
 r_{\Delta V(i,j)} = & R_{BW}^T(i)(V_W(j) - V_W(i) - g \sum_{k=i}^{j-1} \Delta t) \\
 & - \bar{\mu}(i,j) - \frac{\partial \mu(i,j)}{\partial b^g(i)} \delta b^g(i) - \frac{\partial \mu(i,j)}{\partial b^a(i)} \delta b^a(i)
 \end{aligned} \tag{3.91}$$

the Jacobians of $r_{\Delta V(i,j)}$ are computed as

$$\begin{aligned}
 \frac{\partial r_{\Delta V(i,j)}}{\partial \delta \phi(i)} &= (R_{BW}^T(i)(V_W(j) - V_W(i) - g \sum_{k=i}^{j-1} \Delta t))^{\wedge} \\
 \frac{\partial r_{\Delta V(i,j)}}{\partial \delta \phi(j)} &= 0_{3 \times 1} \\
 \frac{\partial r_{\Delta V(i,j)}}{\partial \delta P(i)} &= 0_{3 \times 1} \\
 \frac{\partial r_{\Delta V(i,j)}}{\partial \delta P(j)} &= 0_{3 \times 1} \\
 \frac{\partial r_{\Delta V(i,j)}}{\partial \delta V(i)} &= -R_{BW}^T(i) \\
 \frac{\partial r_{\Delta V(i,j)}}{\partial \delta V(j)} &= R_{BW}^T(i) \\
 \frac{\partial r_{\Delta V(i,j)}}{\partial \delta b^a(i)} &= -\frac{\partial \bar{V}(i,j)}{\partial b^a} \\
 \frac{\partial r_{\Delta V(i,j)}}{\partial \delta b^g(i)} &= -\frac{\partial \bar{V}(i,j)}{\partial b^g}
 \end{aligned} \tag{3.92}$$

For more details of Equation (3.92) derivation, the reader is referred to Appendix E.4.2.

3.7.4 Jacobians of Rotational Residual Error ($r_{\Delta R(i,j)}$)

$$r_{\Delta R(i,j)} = \text{Log}((\bar{F}(i,j) E(\frac{\partial F(i,j)}{\partial b^g(i)} \delta b^g(i)))^T R_{BW}^T(i) R_{BW}(j)) \tag{3.93}$$

the Jacobians of $r_{\Delta R(i,j)}$ are computed as

$$\begin{aligned}
 \frac{\partial r_{\Delta R(i,j)}}{\partial \delta \phi(i)} &= -\Gamma^{-1}(-r_{\Delta R(i,j)})R_{BW}^T(j)R_{BW}(i) \\
 \frac{\partial r_{\Delta R(i,j)}}{\partial \delta \phi(j)} &= \Gamma^{-1}(-r_{\Delta R(i,j)}) \\
 \frac{\partial r_{\Delta R(i,j)}}{\partial \delta P(i)} &= 0_{3 \times 1} \\
 \frac{\partial r_{\Delta R(i,j)}}{\partial \delta P(j)} &= 0_{3 \times 1} \\
 \frac{\partial r_{\Delta R(i,j)}}{\partial \delta V(i)} &= 0_{3 \times 1} \\
 \frac{\partial r_{\Delta R(i,j)}}{\partial \delta V(j)} &= 0_{3 \times 1} \\
 \frac{\partial r_{\Delta R(i,j)}}{\partial \delta b^a(i)} &= 0_{3 \times 1} \\
 \frac{\partial r_{\Delta R(i,j)}}{\partial \delta b^g(i)} &= -\Gamma^{-1}(-r_{\Delta R(i,j)})E^T(r_{\Delta R(i,j)})\Gamma(M)\frac{\partial F(i,j)}{\partial b^g(i)}
 \end{aligned} \tag{3.94}$$

For more details of Equation (3.94) derivation, the reader is referred to Appendix E.4.3.

3.8 A Novel Nonlinear De-skewing for Lidar Inertial Odometry

In this section, *a novel nonlinear de-skewing algorithm* is proposed to correct possible motion distortions caused by the movement of the agent. The proposed algorithm is based on an observation that the IMU has potentials to provide knowledge of the lidar pose during continuous laser ranging due to its high sampling rate. The novel motion integration model which proposed in section 3.3.2 by Henawy *et al.* [168] is adopted to correct the motion distortions caused by the movement of the agent. The corrected scans serve as the input of the LIO and the corresponding transformation matrix is computed by the LIO [56]. The proposed de-skewing algorithm and the LIO are seamlessly integrated together to reduce the motion distortion caused by the movement of the agent. Subsequently, accuracy of the LIO algorithm is further improved. *The major contribution of the proposed algorithm is a motion integration based method to correct the motion distortions caused by the movement of agent.* Moreover, the proposed *IMU factor*,

Figure 3.2: Motion distortions in a sweep of LiDAR. (a) is a given environment. Blue points in (b) stand for a distorted scan, and red points in (b) are a motion corrected scan.

discussed in section 3.7, has been used with the *Lidar factor* by Ye *et al.* [56] for a tightly coupled joint-optimization. Thus, it is a good application to show the effectiveness of the novel proposed IMU factor with different localization systems such as lidar.

3.8.1 Proposed Motion Distortion Solution

In general, The challenges associated with the LIO shall highlight. For simplicity, it is assumed that once the last point in one sweep is scanned, the first point of the subsequent sweep will be scanned immediately. A lidar scan consists of a number of distance measurements because a lidar rangefinder scans point by point in a discrete manner. Each of the point data in one scan is not measured at the same time. Due to the movement of the rangefinder, there are distortions among all the points in a sweep. An example is illustrated in Figure 3.2 [1]. A black line in Figure 3.2(a) stands for an environment, and the movement of the rangefinder is indicated by the arrow. Blue points in Figure 3.2(b) are a raw scan data. Obviously, the set of points is distorted.

The ICP algorithms are used for geometric alignments of three-dimensional data from two 3D sweeps. As shown in Figure 3.2, there are motion distortions among all the points in the two sweeps. To improve the accuracy, all the motion distortions in the

sweeps should be corrected. When there is an IMU in the system, a motion integration model can be adopted to correct the motion distortions in the sweeps. By taking the m^{th} sweep and the $(m + 1)^{th}$ sweep as an example. All the IMU measurements associated with the m^{th} sweep can be used to reduce the motion distortions in the m^{th} sweep. Similarly, the motion distortions in the $(m+1)^{th}$ sweep can be reduced. However, the IMU measurements include noise and there is a drift if only the IMU measurements are used. It is necessary to reduce both the drift and the noise of the IMU measurements by tightly coupled joint-optimization. It is expected that more motion distortions are removed by the corrected IMU measurements. Overall, the problem studied in this research is formulated as follows:

Given a sequence of sweeps and the associated IMU measurements, design a nonlinear de-skewing algorithm such that the ego-motion of the lidar between any pair of sweeps is estimated as accurate as possible.

3.8.2 A Novel Nonlinear De-skewing

The new motion integration model [168] will be used in this part to correct the motion distortion. The novel motion integration model can be expressed from Equation (3.50) as the following:

$$\begin{aligned} P_W(j) &= P_W(i) + \Theta(i, j) + R_{BW}(i)\zeta(i, j) \\ V_W(j) &= V_W(i) + g_W \sum_{k=i}^{j-1} \Delta t + R_{BW}(i)\mu(i, j) \\ R_{BW}(j) &= R_{BW}(i)F(i, j) \end{aligned} \tag{3.95}$$

where the matrix $F(i, j)$ and the vectors $\Theta(i, j)$, $\zeta(i, j)$ and $\mu(i, j)$ are computed as

$$\begin{aligned}
 F(i, j) &= \prod_{k=i}^{j-1} E(\theta_B(k)) \\
 \Theta(i, j) &= V_W(i) \sum_{k=i}^{j-1} \Delta t + \frac{g_W}{2} \left(\sum_{k=i}^{j-1} \Delta t \right)^2 \\
 \zeta(i, j) &= \sum_{k=i}^{j-1} (F(i, k) \Lambda(\theta_B(k)) a_B(k) \Delta t^2 + \mu(i, k) \Delta t) \\
 \mu(i, j) &= \sum_{k=i}^{j-1} F(i, k) \Gamma(\theta_B(k)) a_B(k) \Delta t
 \end{aligned} \tag{3.96}$$

Let $\begin{bmatrix} \tilde{X}_l(m) & \tilde{Y}_l(m) & \tilde{Z}_l(m) \end{bmatrix}^T$ be the l^{th} point in the m^{th} sweep. According to Appendix F, the relationship between $\begin{bmatrix} \tilde{X}_l(m) & \tilde{Y}_l(m) & \tilde{Z}_l(m) \end{bmatrix}^T$ and $\begin{bmatrix} \tilde{X}_l(m+1) & \tilde{Y}_l(m+1) & \tilde{Z}_l(m+1) \end{bmatrix}^T$ is given as

$$\begin{bmatrix} \tilde{X}_l(m+1) \\ \tilde{Y}_l(m+1) \\ \tilde{Z}_l(m+1) \\ 1 \end{bmatrix} = T(i, j) \begin{bmatrix} \tilde{X}_l(m) \\ \tilde{Y}_l(m) \\ \tilde{Z}_l(m) \\ 1 \end{bmatrix} \tag{3.97}$$

where the transformation matrix $T(i, j)$ is

$$T(i, j) = \begin{bmatrix} F^T(i, j) & t(i, j) \\ 0 & 1 \end{bmatrix} \tag{3.98}$$

and the translation vector $t(i, j)$ is

$$\begin{aligned}
 t(i, j) &= R_{WB}(j)(P_W(i) - P_W(j)) \\
 &= -F^T(i, j) R_{WB}(i)(P_W(j) - P_W(i)) \\
 &= -F^T(i, j)(\zeta(i, j) + R_{WB}(i)\Theta(i, j)).
 \end{aligned} \tag{3.99}$$

3.8. A NOVEL NONLINEAR DE-SKEWING FOR LIDAR INERTIAL ODOMETRY

The motion prediction model is adopted to correct possible motion distortions in a sweep of lidar. It is further assumed that the l^{th} point of the m^{th} sweep is captured at $t_{l,m}$ and there are $q(q < n)$ IMU data in the interval $[t_{1,m}, t_{l,m}]$. Let $\begin{bmatrix} \hat{X}_l(m) & \hat{Y}_l(m) & \hat{Z}_l(m) \end{bmatrix}^T$ correspond to the l^{th} corrected point in the m^{th} sweep when the last point is scanned. It can be easily derived that

$$\begin{bmatrix} \hat{X}_l(m) \\ \hat{Y}_l(m) \\ \hat{Z}_l(m) \\ 1 \end{bmatrix} = T^{(r)}(i+q, j) \begin{bmatrix} \tilde{X}_l(m) \\ \tilde{Y}_l(m) \\ \tilde{Z}_l(m) \\ 1 \end{bmatrix}, \quad (3.100)$$

where r is the iteration times and its initial value is 0. The matrix $T^{(r)}(i', j) (i \leq i' \leq j)$ is

$$T^{(r)}(i', j) = \begin{bmatrix} \hat{F}^{(r),T}(i', j) & \hat{t}^{(r)}(i', j) \\ 0 & 1 \end{bmatrix}, \quad (3.101)$$

and the matrix $\hat{F}^{(r),T}(i', j)$ and the vector $\hat{t}^{(r)}(i', j)$ are computed from the Equations (3.98) and (3.99) as

$$\begin{cases} \hat{F}^{(r)}(i', j) = \prod_{k=i'}^{j-1} E(\theta^{(r)}(k)) \\ \hat{t}^{(r)}(i', j) = R_{WB}^{(r)}(j)(P_W^{(r)}(i') - P_W^{(r)}(j)) \end{cases}. \quad (3.102)$$

Using the above equations, all the points in the m^{th} sweep can be aligned with respect to the last point of the m^{th} sweep. As such, all the motion distortions in the m^{th} sweep due to the movement of the lidar are expected to be removed.

Chapter 4

Results and Discussion

In this chapter, the proposed *IMU factor* is integrated with the visual-inertial odometry framework to test its effectiveness as compared to the state-of-the-art. This chapter is divided into three sections. First, in section 4.1, the proposed *IMU factor* is integrated with visual-inertial odometry framework and tested with the most popular and recent state-of-the-art using simulated datasets. Section 4.2 shows the results of the proposed *IMU factor* compared with one of the top benchmark on the market using real-world datasets. Finally, in section 4.3, the proposed IMU factor is integrated with the lidar inertial odometry framework for testing. Also, it shows the effectiveness of the proposed de-skewing methods over the state-of-the-art using indoor datasets.

4.1 Simulated Dataset

In this section, a simulated dataset is first used to compare the proposed model with three state-of-the-art ones. These three state-of-the-art methods include two models called closed-form preintegration theory (CPI-1 and CPI-2) proposed by Eckenhoff *et al.* [188, 189, 167]. CPI assumes that the linear acceleration is constant in the local frame, and it provides two different models for the IMU kinematics. The CPI provides a high accuracy

4.1. SIMULATED DATASET

---

state estimation compared to the discrete methods given by Forster *et al.* and Qin *et al.* [116, 163]. The CPI derives the closed-form solution for mean measurements and Jacobians and uses the closed-form solution in the implementation as a theoretical point of view. However, it uses a numerical method to simplify the computation of the covariance matrix that reduces the accuracy of CPI-1 and CPI-2 algorithms. The third model is a discrete model proposed by Forster *et al.* [116, 117]. An open-source code¹ has been used in the implementation for a fair comparison between the proposed model and the three state-of-the-art models given by Forster *et al.* and Ekenhoff *et al.* [116, 167]. Also, it is worth noting that the proposed *IMU factor* is only implemented in the VIO system, and all other factors are kept the same to test the effectiveness of the proposed model.

Both inertial data and poses for ground truth have been collected using a Gazebo simulator [190]. A micro air vehicle (MAV) simulated data has been generated with different IMU sampling rates. Such method has been widely adopted for testing and validation [167]. The characteristics of the simulated data are the following, camera frame has been recorded at 10 Hz using 80 synthetic stereo feature measurements. The inertial data and ground truth have been recorded at different IMU sampling rates (eg. 100 Hz, 200 Hz, etc). The simulated data follow a predefined path after its takeoff. Both the inertial and camera measurements are corrupted by white noise. For the details of the sensor parameters used in the simulation, the reader is referred to the open source code¹. The GTSAM² framework is used whereby factor graph has been generated, optimized and also marginalized using fixed-lag smoother [191]. For fair comparison, the GTSAM.TANGENT_PREINTEGRATION is switched off so that on-manifold preintegrator by Forster is activated [116]. The system has been initialized using the ground truth data. It is worth noting that the system is a visual-inertial system with no loop closures whereby the features have been marginalized out automatically after three seconds. Figure 4.1 shows the estimated trajectories of the proposed model,

---

¹<https://github.com/rpng/cpi>

²<https://bitbucket.org/gtborg/gtsam>

Figure 4.1: Estimation trajectories against ground-truth (Simulated Dataset)

the closed-form preintegration theory model-1 and model-2 given by Ekenhoff *et al.* [167], and the discrete model given by Forster *et al.* [116] as well as the ground-truth trajectory. Figure 4.2 demonstrates the Monte Carlo simulated results of the Root Mean Square Error (RMSE) on both orientation and position averaged over fifteen runs, while Figure 4.3 demonstrates the analysis on the position error in three dimensions for simulated data. The result shows that the proposed model outperforms the state of the art models and achieves a higher motion accuracy. The proposed model outperforms the discrete model proposed by Forster *et al.* [116] by 15% on mean square error of the position and by 12.92% on mean square error of the orientation when the IMU rate is 100 Hz. The proposed model also outperforms both models of the closed-form preintegration theory in [167] by 5.89% and 12.68% for RMSE on position and by 1.79% and 4.97% for RMSE on orientation, respectively. Even the simulated data has been constrained by the MAV motion, the proposed model shows good performance compared to the state-of-the-art models. This means that the proposed model can give a more accurate estimation in the cases of high dynamics changes or when IMU has a lower rate. The reason, that the

4.1. SIMULATED DATASET

(a) RMSE on Position Error

(a) RMSE on Orientation Error

Figure 4.2: RMSE on position and orientation of the proposed model against closed preintegration theory models in [167] and discrete approach in [116]

proposed model gives a higher results than the discrete model by Forster *et al.* [116], is that the assumption of the proposed model (e.g. the linear acceleration and angular velocity are constant in the body frame) is more reasonable and the kinematics of the MAV between two IMU measurements is well-considered (e.g. the rotational matrix is not constant between two successive IMU measurements) by the proposed model. Moreover, the proposed model computes the mean measurements, covariance matrix and Jacobian as a closed-form solution. On the other hand, the proposed model outperforms both high accuracy models given by Ekenhoff *et al.* [167] because the CPI models use high numerical methods such as RK4 [181, 192] to obtain the covariance matrix which hinders its accuracy, while the proposed model computes the covariance matrix as a closed-form solution. High accuracy motion estimation is based on the covariance matrix which is crucial for the optimization process. For further details of the CPI models, the reader is referred to the technical report given by Ekenhoff *et al.* [193].

Figure 4.3: Analysis on Position Error of the proposed model against continuous preintegration theory models in [167] and discrete approach in [116]

Besides testing the IMU sampling rate at 100 Hz, the case when the IMU sampling rate is 200 Hz is also studied. Tables 4.1 and 4.2 illustrate the effect of IMU sampling rate on the proposed model and the three state-of-the-art models given by Forster *et al.* and Ekenhoff *et al.* [116, 167]. The results show that the improvement percentage decreases when the IMU frequency increases. This is because a higher IMU frequency mitigates the extent of the approximation error. It should be pointed out that a low-cost IMU usually has limited precision and frequency, so the proposed model is useful in such cases which are often encountered in real-time application.

In order to illustrate the computational cost of the proposed model compared to the state-of-the-art models, Table 4.3 shows the comparison on different IMU factors timing. It is worth noting that these results on timing comparison are generated on a workstation

4.1. SIMULATED DATASET

Table 4.1: RMSE On Position and Orientation For Different IMU Sampling Rates Compared to CPI-1 and CPI-2 [167] on Simulated Dataset

Proposed Model			CPI Model-2 [167]				CPI Model-1 [167]			
Frequency [Hz]	[cm]	[deg]	[cm]	[%]	[deg]	[%]	[cm]	[%]	[deg]	[%]
100	8.78	0.2871	8.94	1.79	0.3021	4.97	9.33	5.89	0.3288	12.68
200	4.82	0.1841	4.84	0.41	0.1842	0.05	5.07	4.93	0.2069	11.02

Table 4.2: RMSE On Position and Orientation For Different IMU Sampling Rates Compared to Discrete Model [116] on Simulated Dataset

Proposed Model				Forster [116]			
Frequency [Hz]	[cm]	[deg]	[cm]	[%]	[deg]	[%]	
100	8.78	0.2871	10.33	15.00	0.3297	12.92	
200	4.82	0.1841	5.63	14.39	0.2118	13.08	

with CPU *i7 – 4790K@4.00GHz* $\times 8$ and 32 *GiB* RAM. It shows that the proposed IMU factor gives better computational timing compared with both models of the closed-form preintegration theory. The reason is that, both CPI models are using RK4 to compute the covariance matrix whereby it needs more iterations than the proposed model. Furthermore, the CPI-1 gives a better computational timing compared to CPI-2 because CPI-2 makes the selection between the analytic Jacobian and discrete Jacobian which needs more computational timing. Also, the results show that when the IMU sampling rate increased, the computational timing also increases because it needs to integrate all measurements at a high sampling rate. Thus, lower IMU sampling rate is better for computational cost than high IMU sampling rate. However, accuracy is reduced when the IMU frequency is low. Fortunately, *The proposed model can solve this problem where it gives a higher accuracy when IMU sampling rate is low with efficient computational timing.* On the other hand, the discrete model by Forster *et al.* [116] is the simplest. Again, the

Table 4.3: Comparison On Estimation Timing Of Different IMU Factors

Frequency	Proposed [sec]	CPI-2 [sec]	CPI-1 [sec]	Forster [sec]
100 Hz	0.002534	0.007604	0.003618	0.000703
200 Hz	0.00489	0.015085	0.006588	0.001061
400 Hz	0.009238	0.028834	0.011968	0.001812

key point of VIO literature urges a trade-off between accuracy and computational cost.

In conclusion, *the proposed IMU factor is more accurate than all the state-of-the-art VISLAM systems. Moreover, its accuracy is improved when IMU sampling rate is reduced*

4.2 Real World Dataset (EuRoC Dataset)

The EuRoC dataset has been used in this test where it provides the measurement using a micro aerial vehicle (MAV) in different places with many conditions such as slow/fast motion, lighting conditions and texture. The datasets have been classified as easy, medium and difficult. The measurements have been recorded at 200 Hz for IMU and 20 Hz for images. The dataset has been synchronized, corrected and matched with the IMU measurements output [194].

The proposed *IMU factor* is integrated with visual-inertial odometry framework and compared it with the state-of-the-art open-source codes. It is worth noting that the proposed model has not been compared with one top open-source code, which is, Open Keyframe-based Visual Inertial SLAM (OKVIS)³ [42]. The reason is that, all the state-of-the-art VISLAM systems are compared with OKVIS and all showed a good performance over OKVIS [116, 163, 167]. On the other hand, VINS-Mono⁴ system [163]

---

³<https://github.com/ethz-asl/okvis>

⁴<https://github.com/HKUST-Aerial-Robotics/VINS-Mono>

4.2. REAL WORLD DATASET (EUROC DATASET)

---

is based on a discrete model, and it has the same proposed IMU factor by Forster *et al.* [116]. Thus the proposed is compared and tested *IMU factor* with VINS_Mono algorithm. Another reason the proposed model is compared with VINS_Mono is that, the VINS_Mono considers one of the top benchmarks on VIO research area according to the evaluation studying that has been done by Delmerico *et al.* [195] and Tschopp *et al.* [196]. It should be pointed out that, only the VINS_Mono IMU factor is replaced by the proposed *IMU factor* in this implementation and all other factors are kept the same without changing for fair comparison and evaluation.

Moreover, the closed-form preintegration theory models (CPI-1 and CPI-2) [167] are excluded from this comparison and testing on EuRoC datasets for two reasons. First, the closed-form preintegration theory is not an open-source on real-world datasets. The second reason, being the most important one, is that CPI-1 and CPI-2 models depend on high numerical integration methods such as RK4. Thus the outperforming of the closed-form preintegration models over the discrete model perhaps because they use high numerical integration methods such as RK4. For example, for simplicity, the VISLAM systems use the Euler method in their presentation. However, they use the Midpoint integration method in their implementation because Midpoint gives higher accuracy than the Euler integration method [192]. Thus, this is not a fair comparison. Therefore, CPI-1 and CPI-2 have been excluded from this testing.

This evaluation has been done using a high sampling IMU rate of 200 Hz and the images have been selected, from the left camera as it is a monocular experiment, at 10 Hz. Noted also that the loop closure has been disabled (i.e. the evaluation testing does not keep any kind of map) to show the effectiveness of the proposed *IMU factor* whereby there is a high chance that the evaluation is corrupted by the loop closure effect rather than the proposed model itself. For fair comparison, the proposed *IMU factor* is implemented using the Midpoint integration rather than Euler integration method. However, the whole methodology in Chapter 3 has been illustrated using the Euler method for simplicity.

Figure 4.4: Raw Estimated Trajectory Against Ground-Truth (V1_03_difficult)

Since the estimation output from VINS_Mono benchmark has a different reference from the ground truth reference as shown in Figure 4.4, the estimated raw data has been aligned with the ground truth reference using the trajectory evaluation toolbox⁵ by Zhang *et al.* [197]. The alignment process has been done using the full trajectory states, i.e., by solving the following optimization problem:

$$\arg \min_{s, R, t} \sum_{i=0}^{N-1} \|P_i - sR\hat{P}_i - t\|^2 \quad (4.1)$$

where $\{\hat{P}_i\}_{i=0}^{N-1}$ is the raw estimation data and $\{P_i\}_{i=0}^{N-1}$ is the ground-truth data.

Moreover, there is a high chance that randomness that may occur when the VINS_Mono algorithm is tracking the features, for example, the RANSAC outlier rejection. This is ended by variations in the state estimated data even if it uses the same sequences data. Thus, the experiment has been performed and averaged over five runs. During the five runs, it is observed that the variation is limited to ± 0.02 mm as there is no loop closure. This is a good opportunity to test and conduct a quantitative evaluation of

⁵https://github.com/uzh-rpg/rpg_trajectory_evaluation

the proposed model compared to the state-of-the-art ones. Noted also that the proposed model nodes are running using Robot Operating System (ROS)⁶ simultaneously with the state-of-the-art nodes without affecting their estimation output on the desktop with CPU *i7 – 6700@3.40GHz* $\times 8$ and 64 *GiB* RAM. For details of the ROS nodes diagram, the reader is referred to Appendix I.

Depending on the set parameters, the state estimation is varied. For example, the window size is crucial because the measurements within the window will be optimized. In other words, if the window size is three the system takes into consideration the last three states, and it marginalizes all other previous states out of the optimization. Also, when the window size is increased the accuracy is improved. However, it is more expensive in terms of computation. Thus, selecting a good window size, which achieves good and acceptable performance, is very important especially in the trade-off between accuracy and computation cost. Accordingly, this evaluation has been performed using two window sizes (e.g. window size 10 = "W10" and window size 5 = "W5") to test the proposed model performance under these conditions compared to the state-of-the-art ones.

4.2.1 Window Size 10 (W10)

In this subsection, evaluation test is performed within a window size 10. It is worth noting that it does not keep any kind of map within the window size, and window size is used to maintain some previous state within the optimization process to obtain high accuracy. Noted also that the window size 10 is the default size which has been used by VINS_Mono system.

Difficult Conditions Sequences

V1_03_difficult and V2_03_difficult sequences have been selected from EuRoC datasets to represent the most challenging conditions in terms of lighting, texture-less environment

---

⁶<http://wiki.ros.org/>

(a) V1_03_difficult

(b) V2_03_difficult

Figure 4.5: Estimation Trajectories Against Ground-Truth

and high flight dynamics. In this subsection, the proposed approach has been evaluated on both sequences compared to the-state-of-the-art ones. The estimation trajectories of the proposed approach and VINS_mono model on V1_03_difficult and V2_03_difficult sequence after alignment against ground-truth are shown in Figure 4.5(a) and 4.5(b), respectively.

4.2. REAL WORLD DATASET (EUROC DATASET)

(a) RMSE on Position Error

(b) RMSE on Orientation Error

(c) Absolute Error

Figure 4.6: RMSE and position analysis on V1_03_difficult Sequence for the proposed model and VINS_mono [163]

(a) RMSE on Position Error

(b) RMSE on Orientation Error

(c) Absolute Error

Figure 4.7: RMSE and position analysis on V2.03_difficult Sequence for the proposed model and VINS_mono [163]

4.2. REAL WORLD DATASET (EUROC DATASET)

Table 4.4: RMSE On Position For Different Real Sequences (Window Size 10)

	VINS-Mono [163]	Proposed	Metric Error	Improvement Percentage
Sequence	[cm]	[cm]	[cm]	[%]
V1.01_easy	8.89	7.59	1.3	14.62
V1.02_medium	11.25	9.81	1.44	12.8
V1.03_difficult	18.84	17.21	1.63	8.65
V2.01_easy	8.82	7.61	1.21	13.72
V2.02_medium	16.29	12.59	3.7	22.71
V2.03_difficult	27.84	22.71	5.13	18.43

Figures 4.6 and 4.7 show the Root Mean Square Error (RMSE) on both position and orientation comparison over five runs and also the absolute position errors in the x-direction, y-direction and z-direction on both sequences. The result shows that the proposed approach outperforms the-state-of-the-art model and achieves a higher motion accuracy. The proposed approach outperforms the model in [163] by 18.43% and 8.65% on the mean square error of the position over the full path on V2.03_difficult and V1.03_difficult, respectively. The reason is that the assumption of the proposed approach is more reasonable and the kinematics of the MAV between two IMU measurements is well considered by the proposed model. As such, it can compute the covariance matrix with higher accuracy which is crucial for the optimization process. The RMSE errors of both sequences are illustrated in Table 4.4.

Comparison On Different Sequences

Table 4.4 shows the RMSE on position over the full path for different sequences in flight dynamics, illumination and texture. The results show that the proposed approach outperforms the VINS_mono model in all the sequences, especially when the conditions turn into difficult as demonstrated in both sequences V1.03_difficult and V2.03_difficult

Table 4.5: RMSE On Position For Different Real Sequences (Window Size 5)

	VINS-Mono [163]	Proposed	Metric Error	Improvement Percentage
Sequence	[cm]	[cm]	[cm]	[%]
V1.01_easy	14.79	13.13	1.66	11.22
V1.02_medium	-	-	-	-
V1.03_difficult	27.24	22.01	5.23	19.20
V2.01_easy	11.59	10.34	1.25	10.79
V2.02_medium	28.69	23.97	4.72	16.45
V2.03_difficult	28.68	22.53	6.15	21.44

whereby the metric error is increased proportionally with the difficulty degree of the sequences. This implies that the proposed approach achieves higher accuracy than the-state-of-the-art model, especially in these cases when high dynamic changes in angular velocity or significant illumination changing environment with texture-less area. For the details of the RMSE on position and orientation and absolute error on other sequences, the reader is referred to Appendix G.

4.2.2 Window Size 5 (W5)

In this subsection, the window size is reduced to test and show the effectiveness of the proposed model compared with the state-of-the-art ones. When the window size is reduced, the computational cost also reduces. Thus, it is beneficial to observe and test the *IMU factor* under this condition. Both different window sizes, i.e., W10 and W5, are combined and compared in one figure together on difficult sequences V1.03_difficult and V2.03_difficult as shown in Figure 4.5. The error analysis on position and orientation are shown in Figures 4.6 and 4.7. Moreover, the metric error and improvement percentage compared to the VINS-Mono are illustrated in Table 4.5 for different conditions of sequences.

Figure 4.8: Failed Estimation On (V1.02_medium)

Incontrovertibly, when the window size is reduced, the state estimation accuracy is also reduced. The reason is that the number of measurements, which are optimized within the window size, is reduced. Thus, the number of marginalized states is increased. Accordingly, the accuracy is hindered and decreased. The results in Table 4.5 show that the metric error difference between the proposed model and the state-of-the-art one increases. This means that the proposed model is beneficial and more accurate for state estimation than the state-of-the-art. That is a crucial result because, in the deep tunnel system, there is a high chance for sensor degradation happens. Thus, the proposed model can provide a reliable and efficient estimation using IMU until the measurements from other sensors are observed. Moreover, improvement percentage is increased on the difficult datasets, i.e. V1.03_difficult and V1.03_difficult, by 54.95% and 14.04% respectively when the window size is reduced. It means that the proposed model can handle the most challenging conditions which will be beneficial for deep sewerage tunnel environment.

On the other hand, both the proposed model and the state-of-the-art have failed on V1.02_medium when the window size is reduced. The reason is that the system is unable to perform a Cholesky factorization for a short while, that causes a huge error in estimation and makes a strong spike in state estimation as shown in Figure 4.8.

Table 4.6: Overall Odometry Metric Error

Window 10		Window 5	
VINS-Mono	Proposed	VINS-Mono	Proposed
Distance [m]	[cm]	[cm]	[cm]
7.00	16	27	26
14.00	18	30	27
21.00	23	39	36
28.00	25	44	38
35.00	27	49	43

4.2.3 Overall Evaluation On Odometry Metric Error

The evaluation toolbox by Zhang *et al.* [197] allows us to perform an overall metric odometry error evaluation using all available sequences. The overall evaluation considers all sequence as one big trajectory divided into segments of {7,14,21,28,35} meters. At each segment, the absolute RMSE between the aligned sequences and ground truth is calculated for all available sequences. That is very useful because it allows us to have a clear vision and idea about how much the VIO system drifted over the entire distance travelled. Clearly, Table 4.6 shows that the proposed model has less drift over the state-of-the-art one. Moreover, the results show that when the window size is reduced, the state-of-the-art one drifts rapidly as compared to the proposed model. Besides, it is observed that, with window size 5, the metric error increases rapidly over the travelling distance as compared to the system with window size 10 as shown in Figure 4.9. This emphasizes that, for travelling over long-distance, the full smoothing approach is preferred over the filtering approach. Thus, it is concluded that, for deep tunnel inspection, the full smoothing VIO approach is recommended over filtering approach because of inspection timing and long travelling distance.

4.3. LIDAR INERTIAL ODOMETRY EVALUATION

(a) Window Size 10

(b) Window Size 5

Figure 4.9: Over All Odometry Metric Error

In conclusion, *the proposed model gives superior performance compared to one of the top VIO benchmark (VINS-Mono). In particular, when there are high dynamic changes. Results on simulated data and real-world data validate the proposed model and its efficiency*

4.3 Lidar Inertial Odometry Evaluation

In this section, the proposed IMU factor is integrated with Lidar odometry framework. A quantitative and qualitative experiment has been done to show and evaluate the proposed model performance against the state-of-the-art ones. For fair comparison, the six datasets with different motion speed and movement given by Ye *et al.* [56] have been used in this evaluation. The datasets [56] have been recorded indoor using a Velodyne⁷ VLP-16 Lidar

⁷<https://velodynelidar.com/products/puck/>

and Xsens⁸ MTi-100 IMU. The lidar has a 5 Hz update rate, while the IMU has 400 Hz rate. It is worth noting that, the high IMU sampling rate is very useful when the lidar degradation occurs. The results will be aligned with ground truth from a motion capture system with 100 Hz update rate using the toolbox by Zhang *et al.* [197]. Moreover, this implementation has been done using the open-source code of LIO_mapping⁹ by Ye *et al.* [56]. It is worth noting that, all the illustrated results are averaged over 5 runs due to the variation in the output estimation. Noted also that the proposed model and the state-of-the-art nodes are running using ROS separately. The reason is that, when we have tried to run both nodes simultaneously such as in VIO case, it is found that both nodes are affecting each other. Moreover, the results are obtained using a desktop with CPU *i7 – 6700@3.40GHz* $\times$ 8 and 64 *GiB* RAM. For the details of the ROS nodes diagram, the reader is referred to Appendix I.

This section is organized as follows: First, in subsection 4.3.1, the results show the predicted odometry, i.e., the state estimation from the integrated IMU model. In subsection 4.3.2, the results show the tightly coupled lidar inertial odometry, i.e., joint-optimization between the IMU factor and lidar factor. Finally, in subsection 4.3.3, the mapping results are discussed.

4.3.1 Predicted Odometry

This subsection aims to evaluate the novel integrated IMU model in Equation (3.50) [168] using the switched linear system over the discrete model in Equation (D.7) [116, 163]. The previous evaluation in section 4.2 on the novel IMU model has been done using visual-inertial odometry (VIO) framework by Henawy *et al.* [168], while in this subsection the evaluation is done using lidar inertial odometry (LIO) framework. The advantage of the open-source code by Ye *et al.* [56] is taken to provide this comparison. This

---

⁸<https://www.xsens.com/products/mti-100-series>

⁹<https://github.com/hyye/lio-mapping>

4.3. LIDAR INERTIAL ODOMETRY EVALUATION

Table 4.7: RMSE On Position For Different Real Sequences With Different Motion Speeds (Predicted Odometry)

	Predicted Odometry [56]	Proposed	Metric Error	Percentage
Sequence	[cm]	[cm]	[cm]	[%]
Fast 1	17.41	10.39	7.02	40.32
Fast 2	50.24	50.29	- 0.05	- 0.1
Med 1	31.69	27.69	4.00	12.62
Med 2	64.30	66.76	-2.46	- 3.83
Slow 1	79.27	81.03	-1.76	-2.22
Slow 2	102.71	102.52	-0.19	-0.18

open-source code provides predicted odometry results which essentially is obtained by IMU integrated model. It is worth noting that, LIO_mapping is also using the discrete IMU model as proposed by Forster *et al.* and Qin *et al.* [116, 163].

Table 4.7 shows the RMSE on position over the full trajectory for the predicted odometry on different datasets with different motion speed. The second column in Table 4.7 shows the RMSE on position for the discrete IMU model, while the third column shows the RMSE on position for the proposed IMU model. The fourth and fifth columns show the metric error between the proposed model and the state-of-the-art and the improvement percentage of the proposed model over the discrete model, respectively. The results are obtained using a 100 Hz update rate for both the predicted odometry and ground truth as default from open-source code. Moreover, the predicted state estimation output has been aligned with the ground truth using the first 50 estimated poses using the evaluation trajectory toolbox by Zhang *et al.* [197]. The results show that the state-of-the-art model outperforms the proposed model on Slow 1, Slow 2, Med 2 and Fast 2 sequences. Even though the state-of-the-art one outperforms the proposed model on easy motion conditions, the metric error doesn't have a big difference between of them. For example,

Figure 4.10: Estimation trajectory against ground-truth On Fast 1 Dataset (Predicted Odometry)

the state-of-the-art outperforms the proposed model on the Fast 2 and Slow 2 by 0.5 mm and 1.9 mm only, respectively. The reason is that the high updated IMU sampling rate of 400 Hz mitigates the assumption of the discrete model.

On the contrary, the proposed IMU model outperforms the state-of-the-art one when the motion conditions become difficult in terms of motion speed and movement. This is because the proposed model takes into consideration the high dynamic change between two consecutive IMU measurements even at high IMU sampling rate. Also, the linear acceleration becomes constant in the body frame instead of the world frame in difficult conditions. This is because the rotational matrix is not constant in such a situation. Thus, it can be observed that the proposed model outperforms the state-of-the-art one on Fast 1 and Med 1 sequences by 40 % and 12.62 %, respectively. Figure 4.10 shows the aligned estimation trajectory with ground truth on the Fast 1 sequence, and Figure 4.11 shows the RMSE on position and orientation also the absolute error analysis in X-direction, Y-direction and Z-direction.

4.3. LIDAR INERTIAL ODOMETRY EVALUATION

(a) RMSE on Position Error

(b) RMSE on Orientation Error

(c) Absolute Error

Figure 4.11: RMSE and position analysis on Fast 1 Sequence for the proposed model and LIO_Mapping [56] (Predicted Odometry)

Clearly, the proposed model gives a higher state estimation over the state-of-the-art. Moreover, its accuracy is increased proportionally with the difficulty of the agent motion. These results reach the same conclusion with the work has been done by Henawy *et al.* [168] but both of them have been obtained by using a different framework, i.e, VIO and LIO.

4.3.2 De-skewing And LIO Evaluation

In this subsection, the proposed de-skewing model and the proposed IMU factor have been evaluated using LIO framework. First, the proposed de-skewing model in Equation (3.98) is implemented instead of the state-of-the-art de-skewing model from open-source code [56]. Followed by integrating the proposed IMU factor in Equation (3.87) with lidar factor as a tightly coupled joined-optimization. Thus, the discrete IMU factor by Ye *et al.* [56] is replaced with the novel IMU factor with the closed forms of the mean value, the covariance matrix and Jacobians. All state estimation output has been aligned with the ground truth using the first 50 estimation poses using the toolbox by Zhang *et al.* [197]. Moreover, the lidar inertial odometry has a 5 Hz update rate and all other settings in the experiment are kept the same for fair comparison with LIO model by Ye *et al.* [56].

It is worth noting that there is no mapping in this test, thus the comparison has been done using LIO only. Table 4.8 shows the results on LIO evaluation test. The second column in Table 4.8 shows the RMSE on position over the full trajectory for the state-of-the-art, i.e., LIO model, while the third column shows RMSE on position for the proposed LIO model. The fourth and fifth columns show the metric error between the proposed model and the stat-of-the-art and the improvement percentage of the proposed model over the state-of-the-art, respectively. The results show that the proposed LIO model outperforms the state-of-the-art one on all the sequences with different motion speed. Not only that but also the proposed model gives a higher state estimation compared to the state-of-the-art model when the motion conditions are fast and slow. The proposed model outperforms the

4.3. LIDAR INERTIAL ODOMETRY EVALUATION

Table 4.8: RMSE On Position For Different Real Sequences With Different Motion Speeds Compared to LIO model [56] (Lidar Inertial Odometry Only Without Mapping)

	LIO [56]	Proposed LIO	Metric Error	Percentage
Sequence	[cm]	[cm]	[cm]	[%]
Fast 1	9.24	7.27	1.97	21.32
Fast 2	13.56	12.63	0.93	6.86
Med 1	25.63	25.61	0.02	0.08
Med 2	24.97	24.38	0.59	2.42
Slow 1	18.26	16.80	1.46	8.00
Slow 2	14.67	12.34	2.33	15.88

state-of-the-art by 21.32 % and 15.88 % on Fast 1 and Slow 2 sequences, respectively. The reason is that, for fast motion, the proposed model has good representation for the IMU excitation and takes into consideration the high dynamic change between two consecutive IMU measurements. This conclusion is foreseeable and consistent with the results of the predicted odometry test shown in Table 4.7.

However, in the case of the slow motion, the results show that the proposed model has a slight edge over the state-of-the-art one. The proposed model outperforms the stat-of-the-art by 8.00 % and 15.88 % on Slow 1 and Slow 2 sequences, respectively. This conclusion is unexpected because the state-of-the-art model outperforms the proposed model in the predicted odometry as shown in Table 4.7. The main cause of this improvement estimation is that the proposed model has an accurate transformation matrix (3.98) than the state-of-the-art one. Thus, the proposed model has a de-skewing algorithm more accurate than the state-of-the-art one.

Figure 4.12 shows the aligned estimated trajectories against the ground-truth for Fast 1 and Slow 2 sequences. Clearly, the Fast 1 trajectory is more complicated and has many turning points more than the Slow 2 trajectory. Moreover, the proposed model has well

(a) Fast 1

(b) Slow 2

Figure 4.12: Estimation Trajectories Against Ground-Truth

behaviour on both trajectories. On the other hand, Figures 4.13 and 4.14 show the RMSE on position and orientation, and the absolute error analysis in the three dimensions on Fast 1 and Slow 2 Datasets, respectively. It shows that the proposed model has a good

4.3. LIDAR INERTIAL ODOMETRY EVALUATION

(a) RMSE on Position Error

(b) RMSE on Orientation Error

(c) Absolute Error

Figure 4.13: RMSE and position analysis on Fast 1 Sequence for the proposed model and LIO [56]

(a) RMSE on Position Error

(b) RMSE on Orientation Error

(c) Absolute Error

Figure 4.14: RMSE and position analysis on Slow 2 Sequence for the proposed model and LIO [56]

performance and estimation regarding the RMSE over the full trajectory. Also, it is observed that the drift in state estimation between the proposed model and state-of-the-art model is increased over the travelling time. This concludes that when the travelling time is increased, the proposed model is expected to give higher accuracy than the state-of-the-art one. This is very useful for tunnel inspection application because the inspection methods need a long flying distance. For the details of the RMSE on position and orientation and absolute error on other sequences, the reader is referred to Appendix H.

Obviously, the proposed LIO model outperforms the state-of-the-art one especially when the conditions have a fast motion speed or slow-motion speed. Thus, the proposed model is very useful and applicable to the tunnel inspection application. First, fast motion speed could represent a high chance for lidar degradation inside the tunnel. Second, the slow-motion speed could represent the case of the inspection process. In both cases, the proposed model shows the reliability and robustness of state estimation.

4.3.3 Mapping Evaluation

In this subsection, the mapping effect is studied and discussed. The LIO model has been extended for mapping construction and state estimation. The proposed model is compared with the state-of-the-art model by Ye *et al.* [56] and Lidar Odometry And Mapping (LOAM)¹⁰ [55]. LOAM is considered one of the most accurate state estimation algorithms in lidar odometry and mapping, where it shows a high accuracy estimation on the most popular dataset in lidar odometry field, that is, Kitti datasets¹¹ [198]. However, this dataset has been recorded and obtained using Ground Vehicle (GV). Thus, it is expected that LOAM has a different behaviour when applied for fast motion datasets as the results will show later. It is worth noting that, the state estimation using LOAM is done without any assistance from IMU. So, it is purely lidar odometry and mapping.

---

¹⁰https://github.com/laboshinl/loam_velodyne

¹¹http://www.cvlibs.net/datasets/kitti/eval_odometry.php

Table 4.9: RMSE On Position For Different Real Sequences With Different Motion Speeds Compared to LOAM [55] and LIO_mapping [56] (Mapping)

Proposed Model		LIO [56]			LOAM [55]		
LIO_Mapping		LIO_mapping					
Sequence	[cm]	[cm]	Error [cm]	[%]	[cm]	Error [cm]	[%]
Fast 1	4.84	4.87	0.03	0.62	155.50	-	-
Fast 2	9.80	9.88	0.08	0.81	152.80	-	-
Med 1	11.28	10.66	-0.62	-5.82	203.80	-	-
Med 2	8.92	9.10	0.18	1.98	158.90	-	-
Slow 1	6.19	6.00	-0.19	-3.17	15.60	9.41	60.32
Slow 2	5.13	5.23	0.1	1.91	9.20	4.07	44.24

Table 4.9 shows the RMSE on position over the full trajectory for the proposed model compared to the state-of-the-art ones, i.e., LIO_mapping, and LOAM. The second column in Table 4.9 shows the RMSE on position for the proposed model mapping, while the third column shows the RMSE on position for the LIO_mapping. The fourth and fifth columns show the metric error between the state-of-the-art and the proposed model and the proposed model improvement percentage over the state-of-the-art, i.e., LIO_mapping, respectively. With the same order, the comparison between the proposed model and LOAM is covered in sixth to the eighth column.

The results show that LOAM fails to find state estimation when the motion condition has a fast or medium speed motion. The reason is that there is no assistance from the IMU, and LOAM fails on these conditions. It can also be expected that LOAM fails for mobile robot localization under insufficient features environments because the lidar has high chance to degrade under these environments. The RMSE of position for the LOAM on the Fast and Med datasets are highlighted using the green color in Table 4.9. The red highlight indicates that LOAM fails to estimation the position of the robot as shown in

Figure 4.15: Loam Estimation trajectory against ground-truth On Fast 1 Dataset

Figure 4.15.

On the other hand, LOAM achieves a good state estimation for datasets with slow motion. The proposed algorithm shows a superior's performance compared to the LOAM on these conditions by 60.32 % on Slow 1 dataset and by 44.24 % on Slow 2 dataset. This is because that, first, the proposed deskewing algorithm does not assume the motion of the agent to be linear during the sweep as required by the LOAM; and second, the proposed motion integration model is integrated with the lidar factor via the tightly coupled joint optimization by Ye *et al.* [56]. It is clear that, the tightly coupled is more accurate than the loosely coupled. It should be pointed out that the accuracy of LOAM will be improved if an IMU is deployed as an additional sensor as mentioned by Zhang *et al.* [178].

The results in Table 4.9 show that the proposed model and the state-of-the-art one by Ye *et al.* [56] are comparable in presence of metric maps. This is because the results highly depend on the metric map with salient features. However, it would be very difficult to update the map in a frequently changed environment. The proposed algorithm is useful in such an environment because it can be adopted to have a higher accuracy LIO without

any metric map. This will be the focus for future work, i.e., the state estimation without any metric map in a frequently changed environment.

In conclusion, *the proposed model gives superior and competitive performance as compared to one of the top lidar odometry and mapping benchmark (LOAM). Moreover, the proposed model outperforms the state-of-the-art (LIO_mapping) when there are high dynamic change and slow-motion speed. The results on real-world data show the validity of the proposed model and its efficiency*

Chapter 5

Experimental Setup

This chapter aims to provide a clear idea about how the platform is selected to perform the inspection task. Also, it aims to show the experimental setup and how the experiment is conducted in the lab. Moreover, it aims to show that the proposed model can be used in real-time and it provides a competitive online state estimation as compared to the state-of-the-art ones. This chapter is organized as follows, the platform selection is discussed briefly in section 5.1. Section 5.2 shows the experimental setup and components connection. Section 5.3 shows the communication scheme that has been used in the experiment. Finally, in section 5.4, the proposed model is integrated with VIO framework and it is compared with one of the top VIO algorithm, i.e., VINS-Mono, in the lab. Also, the conclusion is stated and discussed.

5.1 Platform Selection

Nowadays there are numerous platforms available on the market due to the advanced technology. These platforms have been designed for various tasks (e.g. photography, monitoring, search and rescue etc.). It is necessary to find a suitable platform which meets the specific criteria of the application. For example, the platform frame shall handle the

(a) DJI Phantom

(b) Cheerson CX20

(c) AR Drone

(d) Blacklion 068

Figure 5.1: Some Different Drone Platforms

onboard sensors and battery weight. It should be pointed out that the platform will carry two onboard sensors, i.e., camera and Lidar, companion computer and a heavy battery. The reason, that the platform should have a large capacity battery, is that it should provide a required power supply for all onboard sensors and also for companion computer over long flight duration. Not only the frame should handle the heavyweight, but also it must be adaptable to mount and fit an all onboard sensors.

In the beginning, the most widely used commercial quadrotor platforms on the market were considered. DJI Phantom¹ is one of the top commercial platforms and it has been used by many researchers [199, 200, 201]. It is neat, easy to control and good for photos shooting. Moreover, it has the ability to front and rear obstacles avoidance, and it has a forward, backward and downward vision system. Additionally, the flight duration is about 30 minutes approximately. The DJI Phantom 4 is shown in Figure 5.1(a).

Also, DJI Phantom 4 Pro is considered as a cheap platform which is priced around 2000 *SGD*. Although the DJI Phantom has various features and advantages, the DJI Phantom

---

¹<https://www.dji.com/sg/phantom-4-pro>

is not suitable for our application because of two main reasons. The first reason is that the DJI Phantom is using a commercial controller. Thus, it is a closed source. Although there are many communication algorithms to access the Phantom controller, there are difficulties to gain full access and modification of the controller. The second one is that the DJI Phantom frame is non-adaptable for additional installation sensors such as lidar. Furthermore, the DJI Phantom frame cannot handle payloads weight of the lidar and heavy battery for supplying enough power. For all these reasons, DJI is excluded from the selection base.

On the other hand, Cheerson CX20 has an open-source controller which overcomes the difficulties to access the controller. However, the Cheerson CX20 frame is still not suitable for the application because it cannot install additional sensors. Cheerson CX20 is shown in Figure 5.1(b). Another interesting platform is AR-Drone² which has been used by many researchers recently [202]. Also, it has various open-source implementations that have been developed and implemented by the research community over the past few years. It is designed mainly for indoor application whereby it can localize itself very well using the optical flow techniques. It has been used for navigation [203], tracking [204], SLAM [205] etc. AR-Drone is shown in Figure 5.1(c). However, it is not suitable for DTSS application because it is very light and cannot handle any additional sensor. Thus, Cheerson and AR-Drone are excluded from the selection base as well.

Aerolion³ is a local company which sells a long-endurance quadrotor called Blacklion⁴ platforms where it suitable for industrial application and academic research [206]. It has different versions such as the GPS version and lidar version. The indoor environment needs to lidar version where GPS denied environments. So, it is equipped with a 2D lidar and camera to carry out obstacle avoidance and autonomous navigation for an unknown environment using the Robot Operating System (ROS). Although the Blacklion platform

---

²<https://www.parrot.com/global/drones/parrot-ardrone-20-elite-edition>

³<https://www.aerolion.com>

⁴<https://www.aerolion.com/bl-068>

Figure 5.2: DJI F450 Platform

has almost all the features needed for DTSS application, Blacklion's price is too high which costs around 18000 *SGD*. The Blacklion platform is shown in Figure 5.1(d).

Thus, it has been decided to design and assemble the platform according to DTSS application needs with reasonable price. The priority is to choose a suitable frame that handles the required weight for onboard sensors. Thus, the available kits on the market were checked, and it was found that there were various numbers of kits with different configuration and designs. For example, X525 V3 QUADCOPTER FOLDING FRAME⁵, DOCOOLER NEW HJ MWC MULTICOPTER QUADCOPTER FRAME KIT⁶ and DJI F450 KIT⁷. DJI F450 platform has been selected in the experimental setup because it has the prerequisite conditions that achieve the experiment objectives. Also, it has been used

---

⁵<https://www.amazon.com/X525-QuadCopter-Fiberglass-Folding-Frame/dp/B06XHJVYX5>

⁶<https://www.amazon.com/docooler-x-mode-alien-multicopter-quadcopter/dp/b00xdr64ca>

⁷<https://www.dji.com/sg/flame-wheel-arf/feature>

by many researchers before in their experiments work [207, 208, 209, 210]. The primary elements of the F450 kit are shown in Figure 5.2 where it consists of the mainframe and four brushless motors with four electronic speed control.

5.2 Experimental Setup

In this section, the experimental setup is discussed briefly. It shows how the test rig and components wiring are being assembled. First, Pixhawk⁸ has been selected as a controller to provide low-level control in the experimental test. It is the hardware standard for open source autopilots, and it has been the main autopilot for many types of research over the past few years [211, 212].

The second part is related to the perception sensor and Mynteye⁹ camera has been selected for environmental perception. It is a stereo camera with resolution 752×480 @ 60 FPS. Also, it has a built-in six-axis IMU with different frequencies, i.e, sampling rate. One good advantage of the Mynteye camera is that both camera and IMU are synchronized thus there is no problem in terms of synchronization between the camera and IMU. Moreover, it has an open-source driver¹⁰ and it is easy to connect with the Ubuntu system. Finally, Mynteye has great attention recently by the research community, and it has been used in many SLAM applications and pose estimation algorithms. [213, 214]. The reason is that it is available on the market at a cheap price. Previously, the RealSense¹¹ camera ZR300 was recommended by the research community for the same reasons as that for the Mynteye camera [215, 216]. But recently, it was not available on the market anymore.

Finally, Intel NUC kit¹² with 7th generation intel core processors has been selected in the experiment as a high-level controller, e.g., data fusion, video streaming, etc. It has

---

⁸<https://pixhawk.org>

⁹<https://www.mynteye.com/products/mynt-eye-stereo-camera>

¹⁰<https://github.com/slightech/MYNT-EYE-S-SDK>

¹¹<https://click.intel.com/media/ZR300-Product-Datasheet-Public-002.pdf>

¹²<https://www.intel.sg/content/www/xa/en/products/boards-kits/nuc/kits/nuc7i7bnh.html>

Figure 5.3: Experimental Setup

a CPU $i7 - 7567U@4.00GHz \times 4$ and 32 GiB RAM. All the programming nodes are running onboard using NUC for online-estimation. Also, Futaba T6K has been used in the experiment for the remote control. For power supply, a high capacity Lithium battery has been chosen with 4 cells and 5200 mAh.

It is worth noting that the Mynteye camera is connected with the NUC using USB 3 ports, while Pixhawk is connected with NUC using an FTDI USB cable. The overall experimental setup and platform installation is shown in Figure 5.3.

5.3 Communication Scheme

In this section, the communication scheme is discussed. The main software part is the Robot Operating System (ROS) which organizes the workflow and piping it. The communication scheme starts with environmental perception using Mynteye camera whereby it publishes two topics, images and imu measurements, using ROS wrapping¹³. These two topics represent the feature extraction node's input in the proposed model and VINS_Mono. The pose estimation has been done separately using each algorithm. The output is provided by two ROS topic, then each ROS topic is recorded and compared with the ground truth. The proposed model and VINS_Mono ROS graph representation are shown in Appendix I.

¹³<https://github.com/slightech/MYNT-EYE-ROS-Wrapper>

5.3. COMMUNICATION SCHEME

Figure 5.4: Communication scheme

To control the drone, the first onboard control mode will be used, i.e., the drone will be controlled using RC directly. It could be an open-loop control without any feedback or could be a closed-loop control with feedback. In the case of feedback control, the feedback pose is provided by optitrack system. The optitrack system publishes its pose estimation to the mavros¹⁴ link which it communicates with pixhawk using ROS and it sends feedback to the pixhawk controller. It is worth noting that pixhawk uses the classical PID control system.

On the other hand, to control the drone using off-board control mode, i.e., the drone will follow a predefined trajectory. First, the off-board control mode will be activated using the RC switch. Once the off-board control mode is activated, a predefined trajectory node will start using ROS. The path planning algorithm will send the predefined trajectory to mavros link which sends it to pixhawk. Then the optitrack system will send the feedback pose estimation continuously for a feedback control system. The communication scheme is shown in Figure 5.4. It is worth noting that all the control architecture is running onboard using NUC (companion computer) to be sure there is no delay in controlling loop. Also, it

¹⁴<https://github.com/mavlink/mavros>

should be pointed out that the main purpose of the experiment is to show the effectiveness of the proposed model as compared to the state-of-the-art one. Therefore, the feedback control is provided by optitrack system. In the future work, the pose estimation using the proposed model will be provided as feedback measurements to the controller.

5.4 Experimental Results

In this section, an indoor environment experiment is performed to evaluate the proposed model. First, both of camera and camera-IMU calibration have been calibrated using Kalibr¹⁵ toolbox [217, 218, 219]. Also, it should be pointed out that the pinhole camera model has been used in this calibration [220]. The measurements have been recorded at 200Hz for IMU and 10Hz for images. As both models are evaluated based on monocular visual-inertial odometry, the images from the left camera have been used only. The proposed approach and the-state-of-the-art model are running simultaneously on-board using i7 Intel NUC with CPU $i7 - 7567U@4.00GHz \times 4$ and 32 GiB RAM for fair comparison. The estimated motion of both algorithms is aligned first with ground truth using evaluation trajectory toolbox by Zhang *et al.* [197]. It is worth noting that the alignment process has been done using the full trajectory states. Also, it should be pointed out that the ground truth has a 100 Hz update rate. Moreover, the data have been recorded and rerun again for five times off-line to show the reliability of the proposed model compared to the state-of-the-art one.

5.4.1 Arbitrary Trajectory

In this subsection, a free flight was done using Remote Control (RC). An arbitrary trajectory was generated with high speed and with high dynamic changing rate. Fig. 5.5 shows the aligned estimation trajectories of both the proposed model and VINS_mono

---

¹⁵<https://github.com/ethz-asl/kalibr>

Figure 5.5: Estimation Trajectories Against Ground-Truth (Arbitrary Trajectory)

against the ground-truth, while Fig. 5.6 shows the RMSE on position, orientation and absolute error analysis in X-direction, Y-direction and Z-direction. The results show that the proposed approach gives a superior performance as compared to the state-of-the-art one.

To evaluate the performance of both models, Table 5.1 shows the results on RMSE on position and orientation over the full traversed path. In Table 5.1, the second and third columns show the RMSE on position and orientation for the proposed model, respectively. The fourth and fifth columns show the RMSE on position and orientation for the state-of-the-art one. The sixth and seventh columns show the metric error on RMSE on position and the improvement percentage between the proposed model and the state-of-the-art, respectively. To give in depth details of each sequence, the total travelled distance and the linear velocity are mentioned in seventh and eighth columns, respectively. Moreover, the best value on RMSE is highlighted in bold with light-yellow colouring.

The results show that the proposed model outperforms the-state-of-the-art model by 38.15% on RMSE over the full traversed path with an average linear velocity of 0.8108 m/s .

(a) RMSE on Position Error

(b) RMSE on Orientation Error

(c) Absolute Error

Figure 5.6: RMSE and Position Analysis on Arbitrary Trajectory for the Proposed Model and VINS_Mono [163]

The reason is that the proposed model has been built on more reasonable assumptions, i.e., the linear acceleration and angular velocity are constant in body frame, instead of the state-of-the-art model which assumes that the linear acceleration is constant in the world frame. These assumptions make the proposed model well suited for quadrotor dynamic. Therefore, the rotational matrix between two successive IMU measurements has been taken into consideration. Consequently, the proposed model takes into consideration the dynamic changes between measurements even when there is a high IMU sampling rate. Moreover, the proposed model allows us to find and calculate the covariance matrix and Jacobians with a closed-form solution. For the aforementioned reasons, the proposed model outperforms one of the top benchmarks for state estimation with superior performance and improvement.

The results also show that the VINS_Mono outperforms the proposed model on RMSE on orientation. However, the difference is very small, and it is considered that both algorithms give the same performance on orientation. Also, it is observed that during the experimental period the state-of-the-art model sometimes fails for initialization as compared to the proposed model. This implies that the proposed model is more reliable and robust for state estimation as compared to the state-of-the-art one.

5.4.2 Eight Trajectory

In this subsection, the drone followed a pre-defined trajectory in the shape of the digit "eight". This trajectory represents medium challenging conditions to evaluate the proposed model. It is worth noting that the predefined trajectory has been tested first on Gazebo¹⁶ simulator [190]. The reason is to ensure that the drone will follow the predefined trajectory so that it does not lose control. Also, it should be pointed out that, the system is first armed using RC before switching from onboard control mode to off-board control mode.

---

¹⁶<https://dev.px4.io/v1.9.0/en/simulation/gazebo.html>

(a) Eight Trajectory

(b) Square Trajectory

Figure 5.7: Estimation Trajectories Against Ground-Truth

5.4. EXPERIMENTAL RESULTS

(a) RMSE on Position Error

(b) RMSE on Orientation Error

(c) Absolute Error

Figure 5.8: RMSE and Position Analysis on Eight Trajectory for the Proposed Model and VINS_Mono [163]

(a) RMSE on Position Error

(b) RMSE on Orientation Error

(c) Absolute Error

Figure 5.9: RMSE and Position Analysis on Square Trajectory for the Proposed Model and VINS_Mono [163]

5.4. EXPERIMENTAL RESULTS

Table 5.1: RMSE On Position and Orientation For Different Trajectories (Indoor Experiments)

	Proposed Model		VINS-Mono [163]		Metric Error	Percentage	Total	Linear
					On Position	On Position	Distance	Velocity
Sequence	[cm]	[deg]	[cm]	[deg]	[cm]	[%]	[m]	[m/s]
Arbitrary Trajectory	23.77	3.0719	38.43	3.0656	14.66	38.15	44.25	0.8108
Eight Trajectory	18.2	3.0879	24.29	3.0817	6.09	25.07	37.73	0.1937
Square Trajectory	7.83	3.1182	8.97	3.1125	1.14	12.71	26.96	0.0770

In this test, there is no high dynamic change and the linear velocity is 0.1937 m/s . Figure 5.7(a) shows the aligned estimation trajectories of both the proposed approach and VINS-Mono against the ground-truth, while Figure 5.8 shows the RMSE on position, orientation and absolute error analysis in X-direction, Y-direction and Z-direction. The results show that the proposed approach outperforms the-state-of-the-art by 25.07% on RMSE over the full traversed path as shown in Table 5.1.

5.4.3 Square Trajectory

For easy condition, the drone followed a square pre-defined trajectory. The linear velocity is reduced to 0.077 m/s . As shown in Table 5.1, the proposed model still outperforms the-state-of-the-art one by 12.71%. However, both the accuracy improvement and metric error are smaller compared to the difficult condition such as in cases of arbitrary and Eight trajectories. The reason is that the high IMU sampling rate mitigate the effect of discretization of the-state-of-art one. The aligned estimation trajectories against the ground-truth are shown in Figure 5.7(b), while the RMSE on position, orientation and error analysis in X-direction, Y-direction and Z-direction are shown in Figure 5.9. They show that, in the easy conditions, the drone followed the predefined trajectory very well and gives a better estimation compared to the difficult conditions.

Also, it is worth noting that, all of the state-of-the-art models [116, 163, 167] are compared

Table 5.2: Analysis on Computational Cost

VINS-mono				Proposed		
Sequence	CPU [%]	Memory [GB]	Time [ms]	CPU [%]	Memory [GB]	Time [ms]
Arbitrary	25.91	3.31	0.0455	26.04	3.25	0.0683
Eight	37.06	3.87	0.0376	37.44	3.81	0.0522
Square	33.56	3.6	0.0505	34.57	3.54	0.0699

with OKVIS [42] and all of them outperform the OKVIS model. However, the literature review shows that there is no comparison between the state-of-the-art by Eckenhoff *et al.* [167] and VINS_Mono [163]. The reason is that VINS_Mono is considered as one of the top benchmarks for state estimation according to the evaluation studying which has been done by Delmerico *et al.* and Tschopp *et al.* [195, 196]. Thus, the proposed model has a superior and competitive performance as compared to one of the top VIO algorithm, i.e., VINS_Mono.

5.4.4 Analysis on Computational Cost

In this subsection, a full analysis on computational cost has been done to show the usefulness of the proposed model. Table 5.2 shows the CPU and memory consumption. The proposed model has almost the same consumption compared to the state-of-the-art. The reason is the proposed model require only discrete operation even though it has a closed-form solution. This allow the proposed model to have both high accuracy estimations without increase in the computational cost. However, the execution time of the proposed IMU factor is about 0.02 *ms* higher than the state-of-the-art as the proposed model needs to calculate additional matrices such as $\Gamma(\theta)$ and $\Lambda(\theta)$.

5.4. EXPERIMENTAL RESULTS

In conclusion, *the proposed model gives superior and competitive performance as compared to one of the top VIO algorithm, i.e., VINS_Mono. Moreover, the proposed model outperforms the state-of-the-art one on the difficult conditions, i.e., there is a high dynamic motion change up to 38 %. Not only the proposed model outperforms the state-of-the-art one on difficult conditions, but also it outperforms the state-of-the-art on easy conditions up to 12 %. The reason is that the IMU kinematic has been represented very well using the proposed model. Moreover, the proposed model offers a closed-form solution for mean value, covariance matrix and Jacobians discretely, i.e., it updates their value without recalculating them from scratch. Therefore, the proposed IMU factor gives an accurate state estimation as compared to the state-of-the-art ones and is efficient for real-time applications.*

Chapter 6

Conclusion

In this thesis, an aerial robot was proposed that is able to perform inspection task in the deep tunnel sewerage system. To successfully perform the task, the aerial robot should have a good control system and should be able to localize itself with respect to the features inside the deep tunnel sewerage system (DTSS) environment. The aerial robot localization in such environment is very challenging that has not been fully solved before. The DTSS has a dark, wet surface, texture-less and structure-less environment. Moreover, the GPS signal is not available. Thus, there is no readily available or unique localization technique in such a situation. This necessitates the use of multiple sensing system to solve the localization challenge.

Visual and ranging sensing systems were recommended in such an environment. However, the visual system will be affected by the poor lighting condition and texture-less environment while the ranging system will be affected by wet surfaces and structure-less environment. Therefore, an inertial navigation system was recommended as an additional sensing system. The inertial navigation system improved the reliability and robustness of both visual and ranging systems. Employing an inertial navigation system with a visual system is called visual inertial odometry (VIO), while employing it with the ranging system is called lidar inertial odometry (LIO).

For inspection purposes, the mobile robot should have an accurate state estimation or pose estimation algorithm that is also known as mobile robot localization. Achieving accurate state estimation implies achieving a minimum drift over the entire navigation trajectory. Thus, the accuracy of VIO and LIO is dependent on data fusion between different sensing systems. Data fusion can be classified into two major paradigms: fusion paradigm and optimization paradigm.

The data fusion paradigm can be divided into two major fusion types: loosely coupled and tightly coupled. The tightly coupled is more accurate than the loosely coupled fusion paradigm. However, the loosely coupled is less computationally expensive than the tightly coupled paradigm. For the optimization paradigm, it can be classified into three major approaches, namely filtering approach, fixed-lag approach and full smoothing approach.

An extensive literature review on the VIO and LIO systems concluded that the most accurate and reliable state estimation approach can be obtained by tightly coupled fusion paradigm along with using full smoothing approach. To the best of our knowledge, the only fully autonomous tunnel inspection system using an aerial robot was proposed by University of Pennsylvania. The study showed that for 40 *m* flight, the drift recorded was 1 *m* from the actual state, which is an enormous drift. This large drift is attributed to their use of the filtering approach.

In this thesis, the tightly coupled fusion paradigm along with full smoothing approach was used to obtain the most accurate state estimation. Both of VIO and LIO problems were represented graphically using the factor graph. Both of tightly coupled VIO and LIO problem consists of two joint-optimization factors. Whereby VIO was represented by IMU factor and visual factor, while LIO was represented by IMU factor and lidar factor. As both VIO and LIO systems share the same IMU factor, the research focus was on the IMU factor and motion distortion caused by the robot's movement while the laser rangefinder performs scanning in the LIO system.

For the IMU factor, the problem arises as the IMU has a much higher sampling rate than

the other sensors used in the VIO and LIO systems. This causes bottleneck for the online state estimation using graph-based optimization approach as the trajectory grows rapidly causing the online estimation not feasible. This can be solved by IMU preintegration where the IMU measurements can be integrated into a single equivalent value between two keyframes, thus reducing the complexity of graph-based optimization. However, the IMU preintegration theory is based on discrete IMU motion integration model. This discrete model assumes that the linear acceleration in the world frame and the angular velocity in the body frame are constant between two consecutive IMU measurements. Consequently, the robot orientation has also been assumed to be constant for each integration interval between two successive IMU measurements. However, this assumption is not always true. For example, the linear acceleration of a multi-copter system has a higher chance to be constant in the body frame rather than in the world frame between two consecutive IMU measurement. This arises when there is a high dynamic change of the aerial robot motion or when the robot is operating at high speed or when the IMU sampling rate is lower compared to the speed of motion. This hinders the accuracy of the state estimation.

Moreover, the discrete IMU factor is based on a discrete mean value, i.e., the motion integration between two consecutive keyframes has been obtained discretely. Also, it depends on a discrete covariance matrix which is crucial for the optimization process as well as on the Jacobians. Therefore, the discrete IMU factor limits the accuracy of the state estimation even though the full smoothing tightly coupled paradigms is used.

For motion distortion problem, this arises because the rangefinder is also moving while scanning a sweep due to the movement of the aerial robot. This causes motion distortion among all the points in a sweep. The state-of-the-art method uses linear de-skewing algorithms to compensate this motion distortion. The linear de-skewing causes inaccurate motion estimation.

Accordingly, this work solved two key problems required to obtain an accurate state estimation needed for DTSS inspection: provide an accurate IMU factor for VIO and

LIO systems and accurate de-skewing for LIO system.

6.1 Contributions

In this section, the contribution of this work is concluded. These contributions aim to solve the problems highlighted previously.

6.1.1 A Novel IMU Motion Integration Model

A novel IMU motion integration model was proposed using a switched linear system. The novel IMU model assumes that the linear acceleration and angular velocity are constant in the body frame between two consecutive IMU measurements. It does not require the aerial robot orientation between two consecutive IMU measurements to be constant anymore and thus, able to take into account the dynamic change between two successive IMU measurements. A simple discrete linear system was first developed to represent the kinematic between two successive IMU measurements. Since both the linear acceleration and the angular velocity between two successive visual frames are usually time-varying, the overall system between two successive visual frames is a switched linear system. The novel IMU motion integration model was derived using the linear control theory. The novel IMU model provides an analytical solution instead of the discrete solution as obtained by the state-of-the-art models.

6.1.2 A Novel IMU Preintegration Model

Even though the novel IMU motion integration model provides an accurate state estimation, it has its drawback. First, consider that the IMU model integrates the IMU measurements between two keyframes: i and j . The drawback is that the IMU motion integration in Equation (6.1) needs to repeat its integration whenever the system is linearized at t_i . To avoid this problem, the IMU model was reformulated in such a way

that the right-hand side is independent of position, velocity and rotational matrix at t_i such as in Equation (6.2). Thus, the left-hand side is called the IMU preintegration model or relative motion increments. This IMU preintegration model is equivalent to the motion integration between two keyframes. Unfortunately, the IMU measurements are affected by noise. This complicates the MAP estimation because MAP estimation requires an accurate densities (and their log-likelihood) of the measurement. Novel noise isolation method was proposed for the IMU measurements. The preintegration measurements are considered as the mean value for the least square error minimization process. Moreover, all the preintegrated measurements were obtained as a closed-form solution using the novel preintegration model.

$$\begin{aligned}
 P_W(j) &= P_W(i) + \Theta(i, j) + R_{BW}(i)\zeta(i, j) \\
 V_W(j) &= V_W(i) + g_W \sum_{k=i}^{j-1} \Delta t + R_{BW}(i)\mu(i, j) \\
 R_{BW}(j) &= R_{BW}(i)F(i, j)
 \end{aligned} \tag{6.1}$$

$$\begin{aligned}
 \Delta R_{BW}(i, j) &\doteq R_{BW}^T(i)R_{BW}(j) = F(i, j) \\
 \Delta V_W(i, j) &\doteq R_{BW}^T(i)(V_W(j) - V_W(i) - g \sum_{k=i}^{j-1} \Delta t) = \mu(i, j) \\
 \Delta P_W(i, j) &\doteq R_{BW}^T(i)(P_W(j) - P_W(i) - \Theta(i, j)) = \zeta(i, j)
 \end{aligned} \tag{6.2}$$

where the matrix $F(i, j)$ and the vectors $\Theta(i, j)$, $\zeta(i, j)$ and $\mu(i, j)$ are computed in (3.51)

6.1.3 A Novel Iterative Noise Propagation

A novel propagated covariance matrix was proposed which is crucial and has a strong influence on the MAP estimation. The novel covariance matrix is calculated as a closed form solution based on the novel preintegrated noise propagation. The advantage of computing the covariance matrix in an iterative manner is that, when a new IMU measurement is received, the covariance matrix is updated immediately rather than

it is calculated from scratch, thus reducing the computational load. The iterative form is therefore considered suitable for online estimation. Thus, a novel propagated covariance matrix was represented and calculated as a closed form solution in a discrete manner. Clearly, the proposed model has a good balance between the accuracy and the computational cost.

6.1.4 A Novel Analytic Bias Update

A novel analytic bias update was proposed. In practice, the bias will be changed by a small amount. Thus, an expression must be found to update the preintegration measurements when there are changes in the bias estimation. The main idea is to find a small correction plus the old preintegration measurements. Accordingly, an analytical way was proposed to find these small bias corrections, i.e., the Jacobians. The proposed model provides a closed-form solution to obtain the Jacobians whenever the bias is changed.

6.1.5 A Novel Preintegrated IMU Factor

A novel preintegrated IMU factor was proposed in this work. The proposed IMU factor is based on the novel preintegration measurements, i.e., the iterative closed-form covariance matrix, and the novel analytic bias update. The proposed IMU factor has a better state estimation rather than the discrete IMU factor which was proposed by the state-of-the-art model. The reason is that the proposed IMU factor is expressed in an analytic form solution. Moreover, the Jacobians of the residual errors were expressed in such an analytic form. These Jacobians are crucial when using iterative optimization techniques to minimize the cost function.

6.1.6 A Novel Nonlinear De-skewing for LIO

A novel nonlinear de-skewing for LIO system was proposed in this work. The proposed model adopts the novel preintegration IMU model to correct the motion distortion caused by the robot movement. Thus, a closed-form translation matrix was proposed to correct the motion distortion.

6.2 Findings and Limitations

To evaluate the proposed IMU factor, it was integrated with both VIO and LIO frameworks. For the VIO framework, the proposed IMU factor was integrated with visual factor as joint-optimization. The proposed VIO system was evaluated using a simulated dataset, real-world dataset and finally an indoor real-world experiment. For the LIO framework, the proposed IMU factor was integrated with the lidar factor as joint-optimization. Also, the nonlinear de-skewing was implemented. Finally, the proposed LIO system was evaluated and tested using real-world datasets.

6.2.1 A Novel IMU Factor For VIO System

In this subsection, the finding of the proposed model is concluded, when the proposed model is integrated with VIO framework as tightly coupled joint-optimization.

Simulated Datasets: The proposed model was evaluated using a simulated dataset generated by the Gazebo simulation to simulate the real-world situation. In this test, the proposed model was compared with three state-of-the-art models. The first two state-of-the-art models are based on a closed-form solution and high accuracy IMU motion integration model. The third state-of-the-art model is based on the discrete IMU model. The proposed model outperforms three state-of-the-art models on RMSE on position. First, the proposed model outperforms the two models of closed-form preintegration

6.2. FINDINGS AND LIMITATIONS

---

theory, i.e., CPI-1 and CPI-2, by 5.89% and 1.79%, respectively when the IMU sampling rate is set at 100 Hz,. Also, it outperforms the same algorithms when the IMU sampling rate is set at 200 Hz by 4.93% and 0.41%. Second, the proposed model outperforms the discrete state-of-the-art by 15.00% when the IMU sampling rate is set at 100 Hz and by 14.39% when the IMU sampling rate is set at 200 Hz.

The reason for the higher accuracy is that the proposed model provides a closed-form solution for the preintegration mean value, covariance matrix and the Jacobians. In comparison, the state-of-the-art models use high integration numerical methods to calculate the covariance matrix instead of using a closed-form solution. The proposed model outperforms the discrete state-of-the-art model because the discretization hinders the state estimation accuracy in the discrete model. This validates the claim of higher accuracy of the proposed model as compared to all the state-of-the-art models. Moreover, it shows that the improvement percentage of RMSE on position is increased when the IMU sampling rate is reduced. Also, the proposed model was evaluated and compared to the three state-of-the-art models in terms of computational cost. It shows that the simplest model is the discrete model, while the proposed model comes in second after the discrete model. Clearly, the proposed model offers a high accuracy state estimation with a good computational cost.

Real World Datasets: The proposed model was compared with one of the top benchmarks on VIO system, i.e., VINS_Mono, using one of the most popular real-world datasets, i.e., EuRoC dataset. The proposed model outperforms the VINS_Mono algorithm by up to 22.71% on RMSE over the full traversed path. Also, it is noted that the metric error between the proposed model and VINS_Mono on RMSE on position is increased proportionally with the level of difficulty of the dataset. Obviously, the proposed model shows superior performance on V1.03_difficult and V2.03_difficult datasets. This shows that the proposed model is suitable for DTSS application because the deep tunnel environment has challenging conditions in terms of lighting and texture-less. Moreover, it

was observed that when the window size is decreased, the proposed model is recommended over the state-of-the-art one because it is more accurate as compared to the state-of-the-art one under these conditions. The results show that, when the window size is reduced from 10 to 5, the improvement in percentage is increased by 54.95% and 14.04% on V1_03_difficult and V2_03_difficult datasets, respectively. Also, the overall odometry drift was stated using both models. The proposed model has less drift over the travelled distance compared to the state-of-the-art model.

The reason is that the high dynamic change in the aerial robot movement has been taken into consideration using the proposed model. Also, the proposed model shows lower accuracy improvement for the easy conditions. The reason is that the high IMU sampling rate is able to mitigate the effect of the discretization. Moreover, when the window size is reduced, the smoothing effect is also reduced. Again, the proposed model shows superior performance compared to the state-of-the-art one. Finally, it can be concluded that the proposed model shows a competitive and superior performance compared to one of the top VIO algorithm, i.e, VINS_Mono.

Indoor Experiment: The proposed model was tested and evaluated using actual indoor experiment. First, a customized platform was designed and assembled that could be suitable for DTSS application. A free flight was done using remote control (RC). The drone follows an arbitrary trajectory which was generated with high speed and with high dynamic changes. Also, The drone follows two predefined trajectories: eight shape trajectory and square shape trajectory. An arbitrary trajectory represents the difficult condition whereby there is a high dynamic motion change with high motion speed. Eight trajectory represents medium challenging condition where is no high dynamic change but with high motion speed. Finally, the square shape trajectory represents the easy condition.

The proposed model achieves a superior performance compared to the top VIO algorithm, i.e., VINS_Mono. Moreover, the proposed model outperforms the state-of-the-art on the

6.2. FINDINGS AND LIMITATIONS

Table 6.1: Conclusion Of The Novel IMU Factor For VIO (Finding)
The Improvement Percentage on RMSE on Position

Simulated Data	Frequency	CPI-1 [167]	CPI-2 [167]	Forster [116]
	100 Hz	up to 5%	up to 4%	up to 15%
	200 Hz	up to 2%	up to 1%	up to 14%
Real-World Data	VINS_Mono [163]			
	up to 22%			
Indoor Experiment	Arbitrary Trajectory		Eight Trajectory	
	up to 38%		up to 25%	

difficult condition by up to 38 % on RMSE on position over the full path. Not only the proposed model outperforms the state-of-the-art model on the difficult condition, but it also outperforms the state-of-the-art model on medium and easy conditions by up to 25 % and 12 %, respectively on RMSE on position over the full path. The reason is that the IMU kinematics have been represented well using the proposed model. Moreover, the proposed model offers a closed-form solution for mean value, covariance matrix and Jacobians. Therefore, the IMU factor gives an accurate state estimation as compared to the state-of-the-art models. Also, during the experimental tests, it was observed that the proposed model is more reliable and robust for state estimation than the state-of-the-art one. The findings of this work on the proposed model for VIO systems are concluded in Table 6.1.

6.2.2 A Novel IMU Factor For LIO System

In this subsection, the finding of the proposed model integrated with LIO framework as tightly coupled joint-optimization is provided.

Real World Datasets: The proposed model was evaluated and compared with one of the most accurate LIO algorithm, i.e, lio_mapping. First, the proposed model

Table 6.2: Conclusion Of The Novel IMU Factor For LIO (Finding)
The Improvement Percentage on RMSE on Position

Real-World Data	LIO [56]
up to 21%	
Real-World Data	LIO_mapping [56] LOAM [55]
up to 2%	up to 60%

was compared with the state-of-the-art one on the predicted odometry. The predicted odometry represents the motion estimation using the preintegration IMU model only. The proposed model shows superior performance on fast motion datasets whereby it outperforms the state-of-the-art by 40% on RMSE on position over the full traversed path. It is clear that the proposed model has good representation for the high dynamic changes by the robot's motion. Also, it is noted that, the state-of-the-art one outperforms the proposed model by up to 3% only on slow and medium motion speed. The reason is that the high IMU sampling rate mitigates the effect of the discretization of the state-of-the-art model which samples at 400 Hz.

The proposed model was further compared with the state-of-the-art one as full lidar inertial odometry framework. The proposed model shows a better performance than the state-of-the-art ones, achieving 21% better accuracy in position on fast speed motion dataset and up to 15% on slow speed motion. For fast-speed motion dataset, high dynamic change is taken into consideration by the proposed model. However, on the slow speed motion dataset, the proposed model has an accurate de-skewing algorithm whereby the transformation matrix is represented as a closed-form solution.

It is worth noting that the fast speed motion dataset could resemble the higher likelihood of the lidar sensor degradation inside the DTSS, while the slow speed motion dataset could resemble the case of the inspection process. Therefore, surely, the proposed model provides an accurate state estimation suitable for DTSS inspection application.

The proposed model was also evaluated in terms of mapping. The proposed model and the state-of-the-art one are comparable in presence of metric maps. This is because the results highly depend on the metric map with salient features. However, both models show superior performance compared to the lidar odometry and mapping (LOAM) which is considered as one of the top lidar odometry and mapping algorithm on the market. The results show that the proposed model outperforms LOAM by up to 60% on RMSE on position over the full traversed path on slow speed motion datasets, while LOAM fails to find the state estimation on fast motion dataset. The reason is that employing IMU as an additional sensor to the lidar odometry improves its reliability and its robustness. The findings of this work on the proposed model for LIO systems are tabulated in Table 6.2.

6.3 Significance of The Study

In this section, the significance of this thesis for academic and engineering community is concluded.

6.3.1 Academic Significance

This research contributes to the academic community by proposing a novel IMU motion integration model. This model assumes that both linear acceleration and angular velocity in the body are constant. This assumption is more reasonable, especially for the aerial robot, where rapid changes in direction is common. The proposed model represents the motion integration between two IMU measurements as a closed-form solution. Thus, the proposed model has represented the kinematics of the IMU more accurately and is closer to real-world representation.

In addition, this research contributes to the academic community by proposing accurate IMU factor for VIO and LIO. The proposed IMU factor represents the mean value, the

covariance matrix and Jacobian as a closed-form solution. This allows the proposed model to achieve better accuracy with lower computational cost compared to the state-of-the-arts.

Finally, this research contributes to the academic community by proposing an accurate de-skewing for LIO. The proposed model compensates motion distortion caused by mobile robot accurately.

6.3.2 Engineering Significance

This research is significant for engineering application that requires accurate motion estimation such as path planning, mission planning, autonomous navigation and virtual reality. Moreover, this research is significant for mobile robots that travel at high speed with high dynamic motion change. Also, the proposed approach is more suitable for deploying autonomous mobile robot in difficult environments such as in dark, structureless and featureless environment.

6.4 Future Directions

In this section, the future directions is stated.

6.4.1 On The Way To Underground

This work aims to provide an accurate localization method for an autonomous aerial robot to perform inspection task in the underground environment. The proposed work is shown to achieve a superior performance compared to the other state-of-the-art models [168] in both the VIO and LIO benchmarks. Separately, solution for the robust control of the aerial robot with manipulated arm has been proposed by Kocer *et al.* [26, 27, 28]. However, the control system and indoor localization algorithms have not been combined and tested together and deployed in an aerial robot and tested in the lab. This is the

most immediate next step required to allow the aerial robot to operate in an underground tunnel.

6.4.2 Multi-Sensors Fusion

Lidar inertial odometry is a robust method for pose estimation problem with high accuracy as the error does not grow over the measured distance. However, it is suffering from the structure-less environment such as along the tunnel axis. On the other hand, the visual odometry is mature, and its robustness is increased by employing the inertial measurements. Moreover, accuracy is improved by using graph-based optimization. Nevertheless, it is suffering from the texture-less environment, lighting changes and low light condition of the tunnel. So, when the lidar or camera are non-functional, this is known as sensor degradation problem [221].

The sensor degradation problem is considered as a data fusion problem when lidar or camera is non-functional, whereby the IMU, camera and lidar have different sampling rates. Therefore, the existing methods, which solve the sensor degradation problem, usually solve the VIO or LIO problem independently [222, 221]. For example, the visual output will be used to compensate the motion error estimation from lidar rangefinder. However, if the visual output is not observed, the pose estimation algorithm fails. Thus, Zhang *et al.* has developed an algorithm that uses VIO when the rangefinder is not reliable and uses LIO when the camera is non-functional. When both sensors are available, the algorithm divides the error compensation between both algorithms VIO and LIO to reduce the computational cost [223]. As such, state estimation approach that simultaneously optimizes output from multiple sensing systems is still an open research and will impact many applications when successful [224].

6.4.3 High Dynamic Environment

One of the most difficult challenges that face the robotic community is the frequently changing or highly dynamic environment. The state-of-the-art models assume a static environment i.e. without any changes. However, this assumption is not practical as the real world is rarely static. Thus, developing a SLAM algorithm that is able to deal with a highly dynamic environment is still an open research [225, 81]. It is worth noting that the proposed work on lidar inertial odometry gives acceptable state estimation compared to the lidar odometry and mapping, and may be considered as the first step towards localization without mapping. This might be useful in solving localization in the highly dynamic environment.

Appendix A

Least-Square Error Minimization

In this appendix, we will provide a summary of the least squares error minimization based on graph-based optimization problem for robot state estimation.

First, let $\mathbf{X}$ be the state vector which represents the robot pose with respect to the environment. The actual measurement (noisy measurements) is represented by $\mathbf{Z}_i$ where $\mathbf{Z}_i$ is the set of observations or measurements from $\{i = 1 : n\}$. The predicted measurements is represented by $\hat{\mathbf{Z}}_i$ which is the function $f_i(\mathbf{X})$ that maps $\mathbf{X}$ to the predicted measurements.

The problem can be defined as the following; given a system which is described by a set of the measurements functions $\{\hat{\mathbf{Z}}_i = f_i(\mathbf{X})\}_{i=1:n}$ and given actual noisy measurements $\mathbf{Z}_i$. We would like to estimate the state of the robot $\mathbf{X}$ which can be the best explanation and representation of the actual measurements $\mathbf{Z}_i$. The graphical representation of the problem is shown in Fig. [A.1](#).

The error is the difference between the real measurements and the predicted measurements. The error vector can be expressed by:

$$\mathbf{e}_i(\mathbf{X}) = \mathbf{Z}_i - f_i(\mathbf{X}) \quad (\text{A.1})$$

The error vector assumed to be a normal distribution with information matrix Ω_i (the

Figure A.1: Graphical Problem

information matrix is the inverse of the covariance matrix). Thus, the squared error can be expressed by:

$$e_i = \mathbf{e}_i(\mathbf{X})^T \Omega_i \mathbf{e}_i(\mathbf{X}) \quad (\text{A.2})$$

The squared error function in Equation (A.2) is a scalar and nonlinear function. The goal of the least square error minimization is to calculate the state estimation $\mathbf{X}^*$ that minimizes the summation of the squared error of all the measurements. In general Equation (A.3) has not closed form solution and it is complex. Thus, the solution typically is found using the numerical methods.

$$\begin{aligned} \mathbf{X}^* &= \arg \min_{\mathbf{X}} F(\mathbf{X}) \\ &= \arg \min_{\mathbf{X}} \sum_i e_i \\ &= \arg \min_{\mathbf{X}} \sum_i \mathbf{e}_i(\mathbf{X})^T \Omega_i \mathbf{e}_i(\mathbf{X}) \end{aligned} \quad (\text{A.3})$$

We can assume that the error function is smooth with good initial value or guess. The

---

solution can be solved by iterative local linearizations. First, Taylor expansion [49] will be used to linearize the error function around the initial value by following expression:

$$\mathbf{e}_i(\mathbf{X} + \Delta\mathbf{X}) \simeq \mathbf{e}_i(\mathbf{X}) + \mathbf{J}_i(\mathbf{X})\Delta\mathbf{X} \quad (\text{A.4})$$

Where $\mathbf{J}_i(\mathbf{X})$ is the Jacobian matrix and can be expressed by:

$$\mathbf{J}_i(\mathbf{X}) = \begin{pmatrix} \frac{\partial f_1(x)}{\partial x_1} & \frac{\partial f_1(x)}{\partial x_2} & \dots & \frac{\partial f_1(x)}{\partial x_n} \\ \frac{\partial f_2(x)}{\partial x_1} & \frac{\partial f_2(x)}{\partial x_2} & \dots & \frac{\partial f_2(x)}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_n(x)}{\partial x_1} & \frac{\partial f_n(x)}{\partial x_2} & \dots & \frac{\partial f_n(x)}{\partial x_n} \end{pmatrix} \quad (\text{A.5})$$

By substituting Equation (A.4) in Equation (A.2) it can obtain the linearized error function as the following:

$$\begin{aligned} \mathbf{e}_i(\mathbf{X} + \Delta\mathbf{X}) &\simeq (\mathbf{e}_i(\mathbf{X}) + \mathbf{J}_i(\mathbf{X})\Delta\mathbf{X})^T \Omega_i (\mathbf{e}_i(\mathbf{X}) + \mathbf{J}_i(\mathbf{X})\Delta\mathbf{X}) \\ &= \mathbf{e}_i^T(\mathbf{X}) \Omega_i \mathbf{e}_i(\mathbf{X}) + \Delta\mathbf{X}^T \mathbf{J}_i^T(\mathbf{X}) \Omega_i \mathbf{e}_i(\mathbf{X}) + \mathbf{e}_i^T(\mathbf{X}) \Omega_i \mathbf{J}_i(\mathbf{X}) \Delta\mathbf{X} \\ &+ \Delta\mathbf{X}^T \mathbf{J}_i^T(\mathbf{X}) \Omega_i \mathbf{J}_i(\mathbf{X}) \Delta\mathbf{X} \\ &= \overbrace{\mathbf{e}_i^T(\mathbf{X}) \Omega_i \mathbf{e}_i(\mathbf{X})}^{=\mathbf{C}_i} + \Delta\mathbf{X}^T \overbrace{\mathbf{J}_i^T(\mathbf{X}) \Omega_i \mathbf{J}_i(\mathbf{X})}^{=\mathbf{H}_i} \Delta\mathbf{X} + 2 \overbrace{\mathbf{e}_i^T(\mathbf{X}) \Omega_i \mathbf{J}_i(\mathbf{X})}^{=\mathbf{b}_i^T} \Delta\mathbf{X} \\ &= \mathbf{C}_i + 2\mathbf{b}_i^T \Delta\mathbf{X} + \Delta\mathbf{X}^T \mathbf{H}_i \Delta\mathbf{X} \end{aligned} \quad (\text{A.6})$$

By summation of Equation (A.6) we can obtain the global error function as the following:

$$\begin{aligned}
 \mathbf{F}(\mathbf{X} + \Delta\mathbf{X}) &= \sum_i \mathbf{e}_i(\mathbf{X} + \Delta\mathbf{X}) \\
 &\simeq \sum_i (\mathbf{C}_i + 2\mathbf{b}_i^T \Delta\mathbf{X} + \Delta\mathbf{X}^T \mathbf{H}_i \Delta\mathbf{X}) \\
 &= \overbrace{\left(\sum_i \mathbf{C}_i\right)}^{\mathbf{C}} + 2 \overbrace{\left(\sum_i \mathbf{b}_i^T\right)}^{\mathbf{b}^T} \Delta\mathbf{X} + \Delta\mathbf{X}^T \overbrace{\left(\sum_i \mathbf{H}_i\right)}^{\mathbf{H}} \Delta\mathbf{X} \\
 &= \mathbf{C} + 2\mathbf{b}^T \Delta\mathbf{X} + \Delta\mathbf{X}^T \mathbf{H} \Delta\mathbf{X}
 \end{aligned} \tag{A.7}$$

Where $\mathbf{b}^T$ and $\mathbf{H}$ are

$$\begin{aligned}
 \mathbf{b}^T &= \sum_i \mathbf{e}_i^T(\mathbf{X}) \Omega_i \mathbf{J}_i(\mathbf{X}) \\
 \mathbf{H} &= \sum_i \mathbf{J}_i^T(\mathbf{X}) \Omega_i \mathbf{J}_i(\mathbf{X})
 \end{aligned} \tag{A.8}$$

Equation (A.7) is a quadratic form. From [226], if the quadratic form is

$$f(x) = x^T H x + b^T x \tag{A.9}$$

Then, the first derivative with respect to x is

$$\frac{\partial f(x)}{\partial x} = (H + H^T)x + b \tag{A.10}$$

According to Equations (A.9) and (A.10), we can find the first derivative of the Equation (A.7) as the following:

$$\frac{\partial \mathbf{F}(\mathbf{X} + \Delta\mathbf{X})}{\partial \Delta\mathbf{X}} \simeq 2\mathbf{b} + 2\mathbf{H}\Delta\mathbf{X} \tag{A.11}$$

To find the minimum of the global error function, we set the first derivative to zero. Thus,

---

Equation (A.11) is equal to zero.

$$0 = 2\mathbf{b} + 2\mathbf{H}\Delta\mathbf{X} \quad (\text{A.12})$$

The system in Equation (A.12) can be solved as the following which the increment $\Delta\mathbf{X}^*$ is

$$\Delta\mathbf{X}^* = -\mathbf{H}^{-1}\mathbf{b} \quad (\text{A.13})$$

Theoretically, the linear system in Equation (A.12) can be solved by matrix inversion as shown in Equation (A.13). However, the matrix inversion is computationally expensive. Thus, in practice, QR decomposition [227] or Cholesky factorization [228, 229] can be used.

In conclusion, we can summarize local iterative method to solve the global error function as the following:

1. The global error will be linearized around initial guess or good initial value using the Taylor expansion.
2. The linear system components in Equation (A.8) will be computed $\mathbf{b}^T$ and $\mathbf{H}$.
3. In theory, the linear system will be solved by matrix inversion (A.13).
4. Update the state estimation by $\mathbf{X} = \mathbf{X} + \Delta\mathbf{X}^*$.

For more details of this method, the reader is referred to [51]. Moreover, we can observe that the accuracy of solution depends on mean value (initial guess), the Jacobians and the covariance matrices.

Appendix B

Cubic Skew Matrix

$$\begin{aligned}
 (\omega_B^\wedge)^3 &= \begin{bmatrix} \omega_1\omega_2\omega_3 - \omega_1\omega_2\omega_3 & \omega_1^2\omega_3 + \omega_2^2\omega_3 + \omega_3^3 & -\omega_1^2\omega_2 - \omega_2^3 - \omega_2\omega_3^2 \\ -\omega_1^2\omega_3 - \omega_2^2\omega_3 - \omega_3^3 & \omega_1\omega_2\omega_3 - \omega_1\omega_2\omega_3 & \omega_1^3 + \omega_1\omega_2^2 + \omega_1\omega_3^2 \\ \omega_1^2\omega_2 + \omega_2^3 + \omega_2\omega_3^2 & -\omega_1^3 - \omega_1\omega_2^2 - \omega_1\omega_3^2 & \omega_1\omega_2\omega_3 - \omega_1\omega_2\omega_3 \end{bmatrix} \\
 (\omega_B^\wedge)^3 &= \begin{bmatrix} 0 & \omega_3(\omega_1^2 + \omega_2^2 + \omega_3^2) & -\omega_2(\omega_1^2 + \omega_2^2 + \omega_3^2) \\ -\omega_3(\omega_1^2 + \omega_2^2 + \omega_3^2) & 0 & \omega_1(\omega_1^2 + \omega_2^2 + \omega_3^2) \\ \omega_2(\omega_1^2 + \omega_2^2 + \omega_3^2) & -\omega_1(\omega_1^2 + \omega_2^2 + \omega_3^2) & 0 \end{bmatrix} \quad (B.1) \\
 (\omega_B^\wedge)^3 &= (-1)\|\omega_B\|^2\omega_B^\wedge
 \end{aligned}$$

where $\|\omega_B\|^2$ is $(\omega_1^2 + \omega_2^2 + \omega_3^2)$.

Appendix C

Maximum A Posteriori (MAP)

The *joint probability*, $p(X, Y)$, can be factored into a *conditional probability*, $p(Y|X)$, and *prior factor*, $p(X)$. Thus, the joint probability can be represented as follows:

$$p(X, Y) = p(X|Y)p(Y) = p(Y|X)p(X) \quad (\text{C.1})$$

The *Bayes' rule* is given by

$$p(X|Y) = p(X) \frac{p(Y|X)}{p(Y)} \quad (\text{C.2})$$

Where $p(X|Y)$ is the posterior, $p(X)$ is prior and $p(Y|X)$ is likelihood.

$$\overbrace{p(X|Y)}^{\text{posterior}} \propto \overbrace{p(Y|X)}^{\text{likelihood}} \quad (\text{C.3})$$

The MAP estimation is to find the set of parameters X that maximize the likelihood of a set of measurements or observations Y as follows:

$$\begin{aligned}
 X^* &= \arg \max_X \overbrace{p(X|Y)}^{\textit{posterior}} \\
 &= \arg \max_X \overbrace{p(Y|X)}^{\textit{likelihood}} \\
 &= \arg \max_X \prod_i p(y_i|X) \\
 &= \arg \max_X \log \prod_i p(y_i|X) \\
 &= \arg \max_X \sum_i \overbrace{\log\{p(y_i|X)\}}^{\textit{loglikelihood}} \\
 &= \arg \min_X - \sum_i \log\{p(y_i|X)\}
 \end{aligned} \tag{C.4}$$

For more details of MAP, the reader is referred to [230, 231, 51].

Appendix D

State-of-The-Art

The state-of-the-art has been introduced in [163, 116] which assumes that *the linear acceleration in the world frame and the angular velocity in the body frame are constant in a small time interval*. An IMU provides both linear acceleration and angular velocity in the body frame in the form of 3-axis of the accelerometer and 3-axis gyroscope data denoted by a_B and ω_B . The body frame will be expressed by the suffix B notation while the world frame will be expressed by the suffix W notation. The linear acceleration vector and the angular velocity vector will be transformed into the world frame using the rotational matrix from the body frame to the world frame. The gravity vector in the world frame is denoted by g_W .

Let $P_W = \begin{bmatrix} P_1 & P_2 & P_3 \end{bmatrix}^T$ and $V_W = \begin{bmatrix} V_1 & V_2 & V_3 \end{bmatrix}^T$ be the position of the camera centre and the linear velocity of the agent having the camera in the world reference frame and R_{BW} be the rotation matrix from the body frame to the world reference one. The motion pre-integration can be obtained by introducing the IMU kinematic model as [116, 156]:

$$\begin{cases} \dot{P}_W = V_W \\ \dot{V}_W = a_W \\ \dot{R}_{BW} = R_{BW}\omega_B^\wedge \end{cases}, \quad (\text{D.1})$$

where $\omega_B = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 \end{bmatrix}^T$ is the angular velocity of the agent measured by a gyroscope. The matrix $\omega_B^\wedge$ is

$$\omega_B^\wedge = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \quad (\text{D.2})$$

The state estimation is obtained by the integration of (D.1) at $t + \Delta t$:

$$\begin{cases} P_W(t + \Delta t) = P_W(t) + \int_t^{t+\Delta t} V_W(\tau) d\tau + \int_t^{t+\Delta t} \int_t^\tau a_W(\tau) d\tau \\ V_W(t + \Delta t) = V_W(t) + \int_t^{t+\Delta t} a_W(\tau) d\tau \\ R_{BW}(t + \Delta t) = R_{BW}(t) \exp\left(\int_t^{t+\Delta t} \omega_B(\tau) d\tau\right) \end{cases}. \quad (\text{D.3})$$

Generally, equation (D.3) does not have an exact solution for measurements with non-zero rotation rates. By assuming that *the linear acceleration in the world frame a_W and the angular velocity in the body frame ω_B are constant in a small time interval $[k\Delta t, (k+1)\Delta t]$* [116], the kinematic model can be expressed by

$$\begin{cases} P_W(k+1) = P_W(k) + V_W(k)\Delta t + \frac{1}{2}a_W(k)\Delta t^2 \\ V_W(k+1) = V_W(k) + a_W(k)\Delta t \\ R_{BW}(k+1) = R_{BW}(k)\exp(\omega_B(k)\Delta t) \end{cases}, \quad (\text{D.4})$$

Notice that a_W can be expressed by:

$$a_W(k) = R_{BW}(k)a_B(k) + g_W, \quad (\text{D.5})$$

where $a_B = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}^T$ is the linear acceleration in the body frame that is measured with an acceleromete. $g_W = \begin{bmatrix} 0 & 0 & -9.81 \end{bmatrix}^T$ with g being the gravity acceleration in the world frame.

Substituting (D.5) in (D.4), the IMU kinematic model becomes

$$\begin{cases} P_W(k+1) = P_W(k) + V_W(k)\Delta t + \frac{1}{2}(g_W + R_{BW}(k)a_B(k))\Delta t^2 \\ V_W(k+1) = V_W(k) + (g_W + R_{BW}(k)a_B(k))\Delta t \\ R_{BW}(k+1) = R_{BW}(k)\exp(\omega_B(t)\Delta t) \end{cases} \quad (\text{D.6})$$

The principle of motion integration theory is to integrate all IMU measurements between two visual keyframes into a single compound measurement. Suppose that two video frames are captured at the time instances $k = i$ and $k = j$ as shown in Figure D.1. The IMU integrated model can be obtained by integration of (D.6) between the two frames as shown in the following equation:

Figure D.1: Different sampling rates [116]

$$\begin{aligned}
 P_W(j) &= P_W(i) + \sum_{k=i}^{j-1} [V_W(k)\Delta t + \frac{1}{2}(g_W + R_{BW}(k)a_B(k))\Delta t^2] \\
 V_W(j) &= V_W(i) + \sum_{k=i}^{j-1} (g_W + R_{BW}(k)a_B(k))\Delta t \\
 R_{BW}(j) &= R_{BW}(i) \prod_{k=i}^{j-1} \exp(\omega_B(k)\Delta t)
 \end{aligned}
 \tag{D.7}$$

Appendix E

The Detailed Derivation of Chapter 3

E.1 The Proposed IMU Integration Model

$$\begin{aligned}
\Psi_{13} &= s\hat{A}_1(a) \left[\frac{I_{3 \times 3}}{2!} + \frac{1}{3!}(-\omega_B^\wedge) + \frac{1}{4!}(-\omega_B^\wedge)^2 \right. \\
&\quad + \frac{1}{5!}(-1)\|\omega_B\|^2(-\omega_B^\wedge) + \frac{1}{6!}(-1)\|\omega_B\|^2(-\omega_B^\wedge)^2 \\
&\quad \left. + \frac{1}{7!}(-1)^2\|\omega_B\|^4(-\omega_B^\wedge) + \frac{1}{8!}(-1)^2\|\omega_B\|^4(-\omega_B^\wedge)^2 + \dots \right] \\
&= s\hat{A}_1(a) \left[\frac{I_{3 \times 3}}{2!} + \sum_{m=1}^{\infty} \frac{(-1)^{m-1}}{(2m+1)!} \|\omega_B\|^{(2m-2)} (-\omega_B^\wedge) \right. \\
&\quad \left. + \sum_{m=2}^{\infty} \frac{(-1)^{m-2}}{(2m)!} \|\omega_B\|^{(2m-4)} (-\omega_B^\wedge)^2 \right] \\
&= s\hat{A}_1(a) \left[\frac{I_{3 \times 3}}{2!} + \frac{(-1)}{(-1)} \frac{\|\omega_B\|^3}{\|\omega_B\|^3} \sum_{m=1}^{\infty} \frac{(-1)^{m-1}}{(2m+1)!} \|\omega_B\|^{(2m-2)} (-\omega_B^\wedge) \right. \\
&\quad \left. + \frac{(-1)^2}{(-1)^2} \frac{\|\omega_B\|^4}{\|\omega_B\|^4} \sum_{m=2}^{\infty} \frac{(-1)^{m-2}}{(2m)!} \|\omega_B\|^{(2m-4)} (-\omega_B^\wedge)^2 \right] \\
&= s\hat{A}_1(a) \left[\frac{I_{3 \times 3}}{2!} + \frac{1}{\|\omega_B\|^3} \sum_{m=1}^{\infty} \frac{(-1)^m}{(2m+1)!} \|\omega_B\|^{(2m+1)} \omega_B^\wedge \right. \\
&\quad \left. + \frac{1}{\|\omega_B\|^4} \sum_{m=2}^{\infty} \frac{(-1)^m}{(2m)!} \|\omega_B\|^{(2m)} (\omega_B^\wedge)^2 \right]. \tag{E.1}
\end{aligned}$$

$$\begin{aligned}
 \Psi_{23} &= \hat{A}_1(a) \left[I_{3 \times 3} + \frac{1}{2!}(-\omega_B^\wedge) + \frac{1}{3!}(-\omega_B^\wedge)^2 \right. \\
 &\quad + \frac{1}{4!}(-1)\|\omega_B\|^2(-\omega_B^\wedge) + \frac{1}{5!}(-1)\|\omega_B\|^2(-\omega_B^\wedge)^2 \\
 &\quad \left. + \frac{1}{6!}(-1)^2\|\omega_B\|^4(-\omega_B^\wedge) + \frac{1}{7!}(-1)^2\|\omega_B\|^4(-\omega_B^\wedge)^2 + \dots \right] \\
 &= \hat{A}_1(a) \left[I_{3 \times 3} + \sum_{m=1}^{\infty} \frac{(-1)^{m-1}}{(2m)!} \|\omega_B\|^{(2m-2)} (-\omega_B^\wedge) \right. \\
 &\quad \left. + \sum_{m=1}^{\infty} \frac{(-1)^{m-1}}{(2m+1)!} \|\omega_B\|^{(2m-2)} (-\omega_B^\wedge)^2 \right] \\
 &= \hat{A}_1(a) \left[I_{3 \times 3} + \frac{(-1)}{(-1)} \frac{\|\omega_B\|^2}{\|\omega_B\|^2} \sum_{m=1}^{\infty} \frac{(-1)^{m-1}}{(2m)!} \|\omega_B\|^{(2m-2)} (-\omega_B^\wedge) \right. \\
 &\quad \left. + \frac{(-1)}{(-1)} \frac{\|\omega_B\|^3}{\|\omega_B\|^3} \sum_{m=1}^{\infty} \frac{(-1)^{m-1}}{(2m+1)!} \|\omega_B\|^{(2m-2)} (-\omega_B^\wedge)^2 \right] \\
 &= \hat{A}_1(a) \left[I_{3 \times 3} + \frac{1}{\|\omega_B\|^2} \sum_{m=1}^{\infty} \frac{(-1)^m}{(2m)!} \|\omega_B\|^{(2m)} \omega_B^\wedge \right. \\
 &\quad \left. - \frac{1}{\|\omega_B\|^3} \sum_{m=1}^{\infty} \frac{(-1)^m}{(2m+1)!} \|\omega_B\|^{(2m+1)} (\omega_B^\wedge)^2 \right]. \tag{E.2}
 \end{aligned}$$

$$\begin{aligned}
 \Psi_{33} &= I_{3 \times 3} + (-\omega_B^\wedge) + \frac{1}{2!}(-\omega_B^\wedge)^2 + \frac{1}{3!}(-1)\|\omega_B\|^2(-\omega_B^\wedge) \\
 &+ \frac{1}{4!}(-1)\|\omega_B\|^2(-\omega_B^\wedge)^2 + \frac{1}{5!}(-1)^2\|\omega_B\|^4(-\omega_B^\wedge) \\
 &+ \frac{1}{6!}(-1)^2\|\omega_B\|^4(-\omega_B^\wedge)^2 + \dots \\
 &= I_{3 \times 3} + \sum_{m=0}^{\infty} \frac{(-1)^m}{(2m+1)!} \|\omega_B\|^{(2m)} (-\omega_B^\wedge) \\
 &+ \sum_{m=1}^{\infty} \frac{(-1)^{m-1}}{(2m)!} \|\omega_B\|^{(2m-2)} (-\omega_B^\wedge)^2 \\
 &= I_{3 \times 3} + \frac{\|\omega_B\|}{\|\omega_B\|} \sum_{m=0}^{\infty} \frac{(-1)^m}{(2m+1)!} \|\omega_B\|^{(2m)} (-\omega_B^\wedge) \\
 &+ \frac{(-1)}{(-1)} \frac{\|\omega_B\|^2}{\|\omega_B\|^2} \sum_{m=1}^{\infty} \frac{(-1)^{m-1}}{(2m)!} \|\omega_B\|^{(2m-2)} (-\omega_B^\wedge)^2 \\
 &= I_{3 \times 3} - \frac{1}{\|\omega_B\|} \sum_{m=0}^{\infty} \frac{(-1)^m}{(2m+1)!} \|\omega_B\|^{(2m+1)} \omega_B^\wedge \\
 &- \frac{1}{\|\omega\|^2} \sum_{m=1}^{\infty} \frac{(-1)^m}{(2m)!} \|\omega_B\|^{(2m)} (\omega_B^\wedge)^2.
 \end{aligned} \tag{E.3}$$

E.2 A Novel Iterative Noise Propagation

E.2.1 Preintegrated Rotational Noise

$$\begin{aligned}
 \delta\phi_{BW}^\eta(i, j+1) &\doteq \sum_{k=i}^j \bar{F}^T(k+1, j+1) \Gamma(-\bar{\theta}_B(k)) \eta^g(k) \Delta t \\
 &= \sum_{k=i}^{j-1} \bar{F}^T(k+1, j+1) \Gamma(-\bar{\theta}_B(k)) \eta^g(k) \Delta t + \overbrace{\bar{F}^T(j+1, j+1)}^{=I_{3 \times 3}} \Gamma(-\bar{\theta}_B(j)) \eta^g(j) \Delta t \\
 &= \sum_{k=i}^{j-1} \overbrace{(\bar{F}(k+1, j) \bar{F}(j, j+1))^T}^{\bar{F}^T(k+1, j+1)} \Gamma(-\bar{\theta}_B(k)) \eta^g(k) \Delta t + \Gamma(-\bar{\theta}_B(j)) \eta^g(j) \Delta t \\
 &= \bar{F}^T(j, j+1) \sum_{k=i}^{j-1} \bar{F}^T(k+1, j) \Gamma(-\bar{\theta}_B(k)) \eta^g(k) \Delta t + \Gamma(-\bar{\theta}_B(j)) \eta^g(j) \Delta t \\
 &= \bar{F}^T(j, j+1) \underline{\delta\phi_{BW}^\eta(i, j)} + \Gamma(-\bar{\theta}_B(j)) \Delta t \underline{\eta^g(j)} \tag{E.4}
 \end{aligned}$$

E.2.2 Preintegrated Velocity Noise

$$\begin{aligned}
 \delta V_W^\eta(i, j+1) &\doteq - \sum_{k=i}^j \bar{F}(i, k) (\Gamma(\bar{\theta}_B(k)) \bar{a}_B(k))^\wedge \delta \phi_{BW}^\eta(i, k) \Delta t \\
 &- \frac{1}{2} \sum_{k=i}^j \bar{F}(i, k) \Gamma(\bar{\theta}_B(k)) (\bar{a}_B(k))^\wedge \Lambda(-\bar{\theta}_B(k)) \eta^g(k) \Delta t^2 \\
 &+ \sum_{k=i}^j \bar{F}(i, k) \Gamma(\bar{\theta}_B(k)) \eta^a(k) \Delta t \\
 &= - \sum_{k=i}^{j-1} \bar{F}(i, k) (\Gamma(\bar{\theta}_B(k)) \bar{a}_B(k))^\wedge \delta \phi_{BW}^\eta(i, k) \Delta t \\
 &- \frac{1}{2} \sum_{k=i}^{j-1} \bar{F}(i, k) \Gamma(\bar{\theta}_B(k)) (\bar{a}_B(k))^\wedge \Lambda(-\bar{\theta}_B(k)) \eta^g(k) \Delta t^2 \\
 &+ \sum_{k=i}^{j-1} \bar{F}(i, k) \Gamma(\bar{\theta}_B(k)) \eta^a(k) \Delta t \\
 &- \bar{F}(i, j) (\Gamma(\bar{\theta}_B(j)) \bar{a}_B(j))^\wedge \delta \phi_{BW}^\eta(i, j) \Delta t \\
 &- \frac{1}{2} \bar{F}(i, j) \Gamma(\bar{\theta}_B(j)) (\bar{a}_B(j))^\wedge \Lambda(-\bar{\theta}_B(j)) \eta^g(j) \Delta t^2 \\
 &+ \bar{F}(i, j) \Gamma(\bar{\theta}_B(j)) \eta^a(j) \Delta t \\
 &= \underline{\delta V_W^\eta(i, j)} - \bar{F}(i, j) (\Gamma(\bar{\theta}_B(j)) \bar{a}_B(j))^\wedge \Delta t \underline{\delta \phi_{BW}^\eta(i, j)} \\
 &- \frac{1}{2} \bar{F}(i, j) \Gamma(\bar{\theta}_B(j)) (\bar{a}_B(j))^\wedge \Lambda(-\bar{\theta}_B(j)) \Delta t^2 \underline{\eta^g(j)} \\
 &+ \bar{F}(i, j) \Gamma(\bar{\theta}_B(j)) \Delta t \underline{\eta^a(j)}
 \end{aligned} \tag{E.5}$$

E.2.3 Preintegrated Position Noise

$$\begin{aligned}
 \delta P_W^\eta(i, j+1) &\doteq - \sum_{k=i}^j \bar{F}(i, k) (\Lambda(\bar{\theta}_B(k)) \bar{a}_B(k))^\wedge \delta \phi_{BW}^\eta(i, k) \Delta t^2 \\
 &- \frac{1}{3} \sum_{k=i}^j \bar{F}(i, k) \Lambda(\bar{\theta}_B(k)) (\bar{a}_B(k))^\wedge \Phi(-\bar{\theta}_B(k)) \eta^g(k) \Delta t^3 \\
 &+ \sum_{k=i}^j \bar{F}(i, k) \Lambda(\bar{\theta}_B(k)) \eta^a(k) \Delta t^2 + \sum_{k=i}^j \delta V_W^\eta(i, k) \Delta t \\
 &= - \sum_{k=i}^{j-1} \bar{F}(i, k) (\Lambda(\bar{\theta}_B(k)) \bar{a}_B(k))^\wedge \delta \phi_{BW}^\eta(i, k) \Delta t^2 \\
 &- \frac{1}{3} \sum_{k=i}^{j-1} \bar{F}(i, k) \Lambda(\bar{\theta}_B(k)) (\bar{a}_B(k))^\wedge \Phi(-\bar{\theta}_B(k)) \eta^g(k) \Delta t^3 \\
 &+ \sum_{k=i}^{j-1} \bar{F}(i, k) \Lambda(\bar{\theta}_B(k)) \eta^a(k) \Delta t^2 + \sum_{k=i}^{j-1} \delta V_W^\eta(i, k) \Delta t \\
 &- \bar{F}(i, j) (\Lambda(\bar{\theta}_B(j)) \bar{a}_B(j))^\wedge \delta \phi_{BW}^\eta(i, j) \Delta t^2 \\
 &- \frac{1}{3} \bar{F}(i, j) \Lambda(\bar{\theta}_B(j)) (\bar{a}_B(j))^\wedge \Phi(-\bar{\theta}_B(j)) \eta^g(j) \Delta t^3 \\
 &+ \bar{F}(i, j) \Lambda(\bar{\theta}_B(j)) \eta^a(j) \Delta t^2 + \delta V_W^\eta(i, j) \Delta t \\
 &= \underline{\delta P_W^\eta(i, j)} + \Delta t \underline{\delta V_W^\eta(i, j)} \\
 &- \bar{F}(i, j) (\Lambda(\bar{\theta}_B(j)) \bar{a}_B(j))^\wedge \Delta t^2 \underline{\delta \phi_{BW}^\eta(i, j)} \\
 &- \frac{1}{3} \bar{F}(i, j) \Lambda(\bar{\theta}_B(j)) (\bar{a}_B(j))^\wedge \Phi(-\bar{\theta}_B(j)) \Delta t^3 \underline{\eta^g(j)} \\
 &+ \bar{F}(i, j) \Lambda(\bar{\theta}_B(j)) \Delta t^2 \underline{\eta^a(j)}
 \end{aligned} \tag{E.6}$$

E.3 A Novel Analytic Biases Update

E.3.1 Preintegrated Rotational Measurements via Bias Correction

$$\begin{aligned}
 \Delta R_{BW}(i, j) &\stackrel{\text{eq. (3.81)}}{=} \prod_{k=i}^{j-1} E((\tilde{\omega}_B(k) - \bar{b}^g(i) - \delta b^g(i))\Delta t) \\
 &\stackrel{\text{eq. (3.61)}}{\approx} \prod_{k=i}^{j-1} E((\tilde{\omega}_B(k) - \bar{b}^g(i))\Delta t) E(-\Gamma(-\bar{\theta}_B(k))\delta b^g(i)\Delta t) \\
 &\stackrel{\text{eq. (3.59)}}{=} \bar{F}(i, j) \prod_{k=i}^{j-1} E(-\bar{F}^T(k+1, j)\Gamma(-\bar{\theta}_B(k))\delta b^g(i)\Delta t) \\
 &\stackrel{\text{eq. (3.63)}}{\approx} \bar{F}(i, j) E\left(-\sum_{k=i}^{j-1} \bar{F}^T(k+1, j)\Gamma(-\bar{\theta}_B(k))\Delta t \delta b^g(i)\right) \\
 &= \bar{F}(i, j) E\left(\frac{\partial \bar{F}(i, j)}{\partial b^g} \delta b^g(i)\right) \\
 &\doteq \bar{F}(i, j) E(\delta \phi_{BW}^b(i, j))
 \end{aligned} \tag{E.7}$$

E.3.2 Preintegrated Velocity Measurements via Bias Correction

$$\begin{aligned}
 \Delta V_W(i, j) &\stackrel{\text{eq. (3.81), (3.82)}}{=} \sum_{k=i}^{j-1} \bar{F}(i, k) E(\delta \phi_{BW}^b(i, k)) \Gamma(\bar{\theta}_B(k) - \delta b^g(i) \Delta t) \\
 &\quad (\bar{a}_B(k) - \delta b^a(i)) \Delta t \\
 &\stackrel{\text{eq. (3.61)}}{\approx} \sum_{k=i}^{j-1} \bar{F}(i, k) E(\delta \phi_{BW}^b(i, k)) \Gamma(\bar{\theta}_B(k)) \Gamma(-\Lambda(-\bar{\theta}_B(k)) \delta b^g(i) \Delta t) \\
 &\quad (\bar{a}_B(k) - \delta b^a(i)) \Delta t \\
 &\stackrel{\text{eq. (3.56)}}{\approx} \sum_{k=i}^{j-1} \bar{F}(i, k) (I + \delta \phi_{BW}^b(i, k)^\wedge) \Gamma(\bar{\theta}_B(k)) (I - \frac{1}{2} (\Lambda(-\bar{\theta}_B(k)) \delta b^g(i) \Delta t)^\wedge) \\
 &\quad (\bar{a}_B(k) - \delta b^a(i)) \Delta t \\
 &\stackrel{\text{eq. (3.60)}}{\approx} \sum_{k=i}^{j-1} \bar{F}(i, k) \Gamma(\bar{\theta}_B(k)) \bar{a}_B(k) \Delta t \\
 &\quad - \sum_{k=i}^{j-1} \bar{F}(i, k) \Gamma(\bar{\theta}_B(k)) \Delta t \delta b^a(i) \\
 &\quad - \sum_{k=i}^{j-1} \bar{F}(i, k) (\Gamma(\bar{\theta}_B(k)) \bar{a}_B(k))^\wedge \Delta t \delta \phi_{BW}^b(i, k) \\
 &\quad + \frac{1}{2} \sum_{k=i}^{j-1} \bar{F}(i, k) \Gamma(\bar{\theta}_B(k)) (\bar{a}_B(k))^\wedge \Lambda(-\bar{\theta}_B(k)) \Delta t^2 \delta b^g(i) \\
 &\doteq \bar{\mu}(i, j) + \frac{\partial \bar{V}(i, j)}{\partial b^a} \delta b^a(i) + \frac{\partial \bar{V}(i, j)}{\partial b^g} \delta b^g(i) \\
 &\doteq \bar{\mu}(i, j) + \delta V_W^b(i, j). \tag{E.8}
 \end{aligned}$$

E.3.3 Preintegrated Position Measurements via Bias Correction

$$\begin{aligned}
 \Delta P_W(i, j) &\stackrel{\text{eq. (3.81), (3.83)}}{=} \sum_{k=i}^{j-1} [\bar{F}(i, k) E(\delta \phi_{BW}^b(i, k)) \Lambda(\bar{\theta}_B(k) - \delta b^g(i) \Delta t) \\
 &\quad (\bar{a}_B(k) - \delta b^a(i)) \Delta t^2 + (\bar{\mu}(i, k) + \delta V_W^b(i, k)) \Delta t] \\
 &\stackrel{\text{eq. (3.61)}}{\approx} \sum_{k=i}^{j-1} [\bar{F}(i, k) E(\delta \phi_{BW}^b(i, k)) 2\Lambda(\bar{\theta}_B(k)) \Lambda(-\Phi(-\bar{\theta}_B(k)) \delta b^g(i) \Delta t) \\
 &\quad (\bar{a}_B(k) - \delta b^a(i)) \Delta t^2 + (\bar{\mu}(i, k) + \delta V_W^b(i, k)) \Delta t] \\
 &\stackrel{\text{eq. (3.56)}}{\approx} \sum_{k=i}^{j-1} \left[\bar{F}(i, k) (I + \delta \phi_{BW}^b(i, k)^\wedge) \Lambda(\bar{\theta}_B(k)) (I - \frac{1}{3} (\Phi(-\bar{\theta}_B(k)) \delta b^g(i) \Delta t)^\wedge) \right. \\
 &\quad \left. (\bar{a}_B(k) - \delta b^a(i)) \Delta t^2 + (\bar{\mu}(i, k) + \delta V_W^b(i, k)) \Delta t \right] \\
 &\approx \sum_{k=i}^{j-1} \bar{F}(i, k) \Lambda(\bar{\theta}_B(k)) \bar{a}_B(k) \Delta t^2 + \sum_{k=i}^{j-1} \bar{\mu}(i, k) \Delta t \\
 &\quad - \sum_{k=i}^{j-1} \bar{F}(i, k) \Lambda(\bar{\theta}_B(k)) \Delta t^2 \underline{\delta b^a(i)} \\
 &\quad - \sum_{k=i}^{j-1} \bar{F}(i, k) (\Lambda(\bar{\theta}_B(k)) \bar{a}_B(k))^\wedge \Delta t^2 \underline{\delta \phi_{BW}^b(i, k)} \\
 &\quad + \frac{1}{3} \sum_{k=i}^{j-1} \bar{F}(i, k) \Lambda(\bar{\theta}_B(k)) (\bar{a}_B(k))^\wedge \Phi(-\bar{\theta}_B(k)) \Delta t^3 \underline{\delta b^g(i)} \\
 &\quad + \sum_{k=i}^{j-1} \Delta t \underline{\delta V_W^b(i, k)} \\
 &\doteq \bar{\zeta}(i, j) + \frac{\partial \bar{P}(i, j)}{\partial b^a} \delta b^a(i) + \frac{\partial \bar{P}(i, j)}{\partial b^g} \delta b^g(i) \\
 &\doteq \bar{\zeta}(i, j) + \delta P_W^b(i, j)
 \end{aligned} \tag{E.9}$$

E.4 A Novel Preintegrated IMU Factor

E.4.1 Jacobians of Position Residual Error ($r_{\Delta P(i,j)}$)

$$\begin{aligned} r_{\Delta P(i,j)} &= R_{BW}^T(i)(P_W(j) - P_W(i) - \Theta(i,j)) \\ &\quad - \bar{\zeta}(i,j) - \frac{\partial \zeta(i,j)}{\partial b^g(i)} \delta b^g(i) - \frac{\partial \zeta(i,j)}{\partial b^a(i)} \delta b^a(i) \end{aligned} \quad (\text{E.10})$$

Both $R_{BW}(j)$ and $V_W(j)$ do not exist in $r_{\Delta P(i,j)}$ in (3.89), accordingly the jacobians w.r.t $\delta \phi(j)$ and $\delta V(j)$ are zero. On the other hand, $r_{\Delta P(i,j)}$ is a linear function in both $\delta b^g(i)$ and $\delta b^a(i)$, hence the jacobians w.r.t $\tilde{\delta} b^g(i)$ and $\tilde{\delta} b^a(i)$ are the same coefficients of $\delta b^g(i)$ and $\delta b^a(i)$ in (3.86).

For jacobians w.r.t $\delta P(i)$

$$\begin{aligned} r_{\Delta P(i,j)}(P_W(i) + R_{BW}(i)\delta P(i)) &= R_{BW}^T(i)(P_W(j) - P_W(i) - R_{BW}(i)\delta P(i) - \Theta(i,j)) - C \\ &= r_{\Delta P(i,j)}(P_W(i)) - I_{3 \times 1} \delta P(i) \end{aligned} \quad (\text{E.11})$$

Where $C \doteq \bar{\zeta}(i,j) + \frac{\partial \zeta(i,j)}{\partial b^g(i)} \delta b^g(i) + \frac{\partial \zeta(i,j)}{\partial b^a(i)} \delta b^a(i)$

For jacobians w.r.t $\delta P(j)$

$$\begin{aligned} r_{\Delta P(i,j)}(P_W(j) + R_{BW}(j)\delta P(j)) &= R_{BW}^T(i)(P_W(j) + R_{BW}(j)\delta P(j) - P_W(i) - \Theta(i,j)) - C \\ &= r_{\Delta P(i,j)}(P_W(j)) - R_{BW}^T(i)R_{BW}(j)\delta P(j) \end{aligned} \quad (\text{E.12})$$

For jacobians w.r.t $\delta V(i)$

$$\begin{aligned}
 r_{\Delta P(i,j)}(V_W(i) + \delta V(i)) &= R_{BW}^T(i) \left(P_W(j) - P_W(i) - V_W(i) \sum_{k=i}^{j-1} \Delta t - \delta V(i) \sum_{k=i}^{j-1} \Delta t \right. \\
 &\quad \left. - \frac{g_W}{2} \left(\sum_{k=i}^{j-1} \Delta t \right)^2 \right) - C \\
 &= r_{\Delta P(i,j)}(V_W(i)) - R_{BW}^T(i) \sum_{k=i}^{j-1} \Delta t \delta V(i)
 \end{aligned} \tag{E.13}$$

For jacobians w.r.t $\delta\phi(i)$

$$\begin{aligned}
 r_{\Delta P(i,j)}(R_{BW}(i)E(\delta\phi(i))) &= (R_{BW}(i)E(\delta\phi(i))^T(P_W(j) - P_W(i) - \Theta(i,j)) - C \\
 &\stackrel{\text{eq. (3.56)}}{\approx} (I - \delta\phi(i)^\wedge) R_{BW}^T(i) (P_W(j) - P_W(i) - \Theta(i,j)) - C \\
 &\stackrel{\text{eq. (3.60)}}{=} r_{\Delta P(i,j)}(R_{BW}(i)) + (R_{BW}^T(i) (P_W(j) - P_W(i) - \Theta(i,j)))^\wedge \delta\phi(i)
 \end{aligned} \tag{E.14}$$

the Jacobians of $r_{\Delta P(i,j)}$ are computed as

$$\begin{aligned}
 \frac{\partial r_{\Delta P(i,j)}}{\partial \delta\phi(i)} &= (R_{BW}^T(i) (P_W(j) - P_W(i) - \Theta(i,j)))^\wedge \\
 \frac{\partial r_{\Delta P(i,j)}}{\partial \delta\phi(j)} &= 0_{3 \times 1} \\
 \frac{\partial r_{\Delta P(i,j)}}{\partial \delta P(i)} &= -I_{3 \times 1} \\
 \frac{\partial r_{\Delta P(i,j)}}{\partial \delta P(j)} &= R_{BW}^T(i) R_{BW}(j) \\
 \frac{\partial r_{\Delta P(i,j)}}{\partial \delta V(i)} &= -R_{BW}^T(i) \sum_{k=i}^{j-1} \Delta t \\
 \frac{\partial r_{\Delta P(i,j)}}{\partial \delta V(j)} &= 0_{3 \times 1} \\
 \frac{\partial r_{\Delta P(i,j)}}{\partial \delta b^a(i)} &= -\frac{\partial \bar{P}(i,j)}{\partial b^a} \\
 \frac{\partial r_{\Delta P(i,j)}}{\partial \delta b^g(i)} &= -\frac{\partial \bar{P}(i,j)}{\partial b^g}
 \end{aligned}$$

(E.15)

E.4.2 Jacobians of Velocity Residual Error ($r_{\Delta V(i,j)}$)

$$\begin{aligned}
 r_{\Delta V(i,j)} &= R_{BW}^T(i)(V_W(j) - V_W(i) - g \sum_{k=i}^{j-1} \Delta t) \\
 &\quad - \bar{\mu}(i,j) - \frac{\partial \mu(i,j)}{\partial b^g(i)} \delta b^g(i) - \frac{\partial \mu(i,j)}{\partial b^a(i)} \delta b^a(i)
 \end{aligned} \tag{E.16}$$

Both $R_{BW}(j)$, $P_W(j)$ and $P_W(i)$ do not exist in $r_{\Delta V(i,j)}$ in (3.91), accordingly the jacobians w.r.t $\delta \phi(j)$, $\delta p(j)$ and $\delta p(i)$ are zero. On the other hand, $r_{\Delta V(i,j)}$ is a linear function in both $\delta b^g(i)$ and $\delta b^a(i)$, hence the jacobians w.r.t $\tilde{\delta} b^g(i)$ and $\tilde{\delta} b^a(i)$ are the same coefficients of $\delta b^g(i)$ and $\delta b^a(i)$ in (3.86).

For jacobians w.r.t $\delta V(i)$

$$\begin{aligned}
 r_{\Delta V(i,j)}(V_W(i) + \delta V(i)) &= R_{BW}^T(i)(V_W(j) - V_W(i) - \delta V(i) - g \sum_{k=i}^{j-1} \Delta t) - D \\
 &= r_{\Delta V(i,j)}(V_W(i)) - R_{BW}^T(i) \delta V(i)
 \end{aligned} \tag{E.17}$$

Where $D \doteq \bar{\mu}(i,j) + \frac{\partial \mu(i,j)}{\partial b^g(i)} \delta b^g(i) + \frac{\partial \mu(i,j)}{\partial b^a(i)} \delta b^a(i)$

For jacobians w.r.t $\delta V(j)$

$$\begin{aligned}
 r_{\Delta V(i,j)}(V_W(j) + \delta V(j)) &= R_{BW}^T(i)(V_W(j) + \delta V(j) - V_W(i) - g \sum_{k=i}^{j-1} \Delta t) - D \\
 &= r_{\Delta V(i,j)}(V_W(j)) + R_{BW}^T(i) \delta V(j)
 \end{aligned} \tag{E.18}$$

For jacobians w.r.t $\delta \phi(i)$

$$\begin{aligned}
 r_{\Delta V(i,j)}(R_{BW}(i)E(\delta\phi(i))) &= (R_{BW}(i)E(\delta\phi(i))^T(V_W(j) - V_W(i) - g\sum_{k=i}^{j-1}\Delta t) - D \\
 &\stackrel{\text{eq. (3.56)}}{\approx} (I - \delta\phi(i)^\wedge)R_{BW}^T(i)(V_W(j) - V_W(i) - g\sum_{k=i}^{j-1}\Delta t) - D \\
 &\stackrel{\text{eq. (3.60)}}{=} r_{\Delta V(i,j)}(R_{BW}(i)) + \left(R_{BW}^T(i)(V_W(j) - V_W(i) - g\sum_{k=i}^{j-1}\Delta t)\right)^\wedge \delta\phi(i)
 \end{aligned} \tag{E.19}$$

the Jacobians of $r_{\Delta V(i,j)}$ are computed as

$$\begin{aligned}
 \frac{\partial r_{\Delta V(i,j)}}{\partial \delta\phi(i)} &= (R_{BW}^T(i)(V_W(j) - V_W(i) - g\sum_{k=i}^{j-1}\Delta t))^\wedge \\
 \frac{\partial r_{\Delta V(i,j)}}{\partial \delta\phi(j)} &= 0_{3 \times 1} \\
 \frac{\partial r_{\Delta V(i,j)}}{\partial \delta P(i)} &= 0_{3 \times 1} \\
 \frac{\partial r_{\Delta V(i,j)}}{\partial \delta P(j)} &= 0_{3 \times 1} \\
 \frac{\partial r_{\Delta V(i,j)}}{\partial \delta V(i)} &= -R_{BW}^T(i) \\
 \frac{\partial r_{\Delta V(i,j)}}{\partial \delta V(j)} &= R_{BW}^T(i) \\
 \frac{\partial r_{\Delta V(i,j)}}{\partial \tilde{\delta}b^a(i)} &= -\frac{\partial \bar{V}(i,j)}{\partial b^a} \\
 \frac{\partial r_{\Delta V(i,j)}}{\partial \tilde{\delta}b^g(i)} &= -\frac{\partial \bar{V}(i,j)}{\partial b^g}
 \end{aligned} \tag{E.20}$$

E.4.3 Jacobians of Rotational Residual Error ($r_{\Delta R(i,j)}$)

$$r_{\Delta R(i,j)} = \text{Log}((\bar{F}(i,j)E(\frac{\partial F(i,j)}{\partial b^g(i)}\delta b^g(i)))^T R_{BW}^T(i)R_{BW}(j)) \tag{E.21}$$

Both of $P_W(i)$, $P_W(j)$, $V_W(i)$, $V_W(j)$ and $\delta b^a(i)$ do not exist in $r_{\Delta R(i,j)}$ in (3.93), accordingly the jacobians w.r.t $\delta p(i)$, $\delta p(j)$, $\delta V(i)$, $\delta V(j)$ and $\tilde{\delta}b^a(i)$ are zero.

For jacobians w.r.t $\delta\phi(i)$

$$\begin{aligned}
 r_{\Delta R(i,j)}(R_{BW}(i)E(\delta\phi(i))) &= \text{Log}((\bar{F}(i,j)U)^T(R_{BW}(i)E(\delta\phi(i)))^T R_{BW}(j)) \\
 &= \text{Log}((\bar{F}(i,j)U)^T E(-\delta\phi(i))R_{BW}^T(i)R_{BW}(j)) \\
 &\stackrel{\text{eq. (3.59)}}{=} \text{Log}((\bar{F}(i,j)U)^T R_{BW}^T(i)R_{BW}(j)E(-R_{BW}^T(j)R_{BW}(i)\delta\phi(i))) \\
 &\stackrel{\text{eq. (3.63)}}{\approx} r_{\Delta R(i,j)} - \Gamma^{-1}(-r_{\Delta R(i,j)})R_{BW}^T(j)R_{BW}(i)\delta\phi(i)
 \end{aligned} \tag{E.22}$$

Where $U \doteq E(\frac{\partial F(i,j)}{\partial b^g(i)}\delta b^g(i))$

For jacobians w.r.t $\delta\phi(j)$

$$\begin{aligned}
 r_{\Delta R(i,j)}(R_{BW}(j)E(\delta\phi(j))) &= \text{Log}((\bar{F}(i,j)U)^T(R_{BW}^T(i)R_{BW}(j)E(\delta\phi(j)))) \\
 &\stackrel{\text{eq. (3.63)}}{\approx} r_{\Delta R(i,j)} + \Gamma^{-1}(-r_{\Delta R(i,j)})\delta\phi(i)
 \end{aligned} \tag{E.23}$$

For jacobians w.r.t $\tilde{\delta}b^g(i)$

$$\begin{aligned}
 r_{\Delta R(i,j)}(\delta b^g(i) + \tilde{\delta}b^g(i)) &= \text{Log}\left(\left(\bar{F}(i,j)E\left(\frac{\partial F(i,j)}{\partial b^g(i)}(\delta b^g(i) + \tilde{\delta}b^g(i))\right)\right)^T R_{BW}^T(i)R_{BW}(j)\right) \\
 &\stackrel{\text{eq. (3.61)}}{\approx} \text{Log}\left(\left(\bar{F}(i,j)UE\left(\Gamma(-M)\frac{\partial F(i,j)}{\partial b^g(i)}\tilde{\delta}b^g(i)\right)\right)^T R_{BW}^T(i)R_{BW}(j)\right) \\
 &= \text{Log}\left(E\left(\Gamma(M)\frac{\partial F(i,j)}{\partial b^g(i)}\tilde{\delta}b^g(i)\right)(\bar{F}(i,j)U)^T R_{BW}^T(i)R_{BW}(j)\right) \\
 &\stackrel{\text{eq. (3.59)}}{=} \text{Log}\left(E(r_{\Delta R(i,j)})E\left(E(r_{\Delta R(i,j)})^T\left(\Gamma(M)\frac{\partial F(i,j)}{\partial b^g(i)}\tilde{\delta}b^g(i)\right)\right)\right) \\
 &\stackrel{\text{eq. (3.63)}}{=} r_{\Delta R(i,j)} - \Gamma^{-1}(-r_{\Delta R(i,j)})E(r_{\Delta R(i,j)})^T\left(\Gamma(M)\frac{\partial F(i,j)}{\partial b^g(i)}\tilde{\delta}b^g(i)\right)
 \end{aligned} \tag{E.24}$$

E.4. A NOVEL PREINTEGRATED IMU FACTOR

Where $M \doteq \frac{\partial F(i,j)}{\partial b^g(i)} \delta b^g(i)$ and the Jacobians of $r_{\Delta R(i,j)}$ are computed as

$$\begin{aligned}
 \frac{\partial r_{\Delta R(i,j)}}{\partial \delta \phi(i)} &= -\Gamma^{-1}(-r_{\Delta R(i,j)}) R_{BW}^T(j) R_{BW}(i) \\
 \frac{\partial r_{\Delta R(i,j)}}{\partial \delta \phi(j)} &= \Gamma^{-1}(-r_{\Delta R(i,j)}) \\
 \frac{\partial r_{\Delta R(i,j)}}{\partial \delta P(i)} &= 0_{3 \times 1} \\
 \frac{\partial r_{\Delta R(i,j)}}{\partial \delta P(j)} &= 0_{3 \times 1} \\
 \frac{\partial r_{\Delta R(i,j)}}{\partial \delta V(i)} &= 0_{3 \times 1} \\
 \frac{\partial r_{\Delta R(i,j)}}{\partial \delta V(j)} &= 0_{3 \times 1} \\
 \frac{\partial r_{\Delta R(i,j)}}{\partial \delta b^a(i)} &= 0_{3 \times 1} \\
 \frac{\partial r_{\Delta R(i,j)}}{\partial \delta b^g(i)} &= -\Gamma^{-1}(-r_{\Delta R(i,j)}) E^T(r_{\Delta R(i,j)}) \Gamma(M) \frac{\partial F(i,j)}{\partial b^g(i)}
 \end{aligned} \tag{E.25}$$

Appendix F

The Relationship Between Two Consecutive Points

Let $\begin{bmatrix} X_l & Y_l & Z_l \end{bmatrix}^T$ be the coordinate of the l th point in the world reference frame. The relationship between $\begin{bmatrix} \tilde{X}_l(m) & \tilde{Y}_l(m) & \tilde{Z}_l(m) \end{bmatrix}^T$ and $\begin{bmatrix} X_l & Y_l & Z_l \end{bmatrix}^T$ is represented by

$$\begin{bmatrix} \tilde{X}_l(m) \\ \tilde{Y}_l(m) \\ \tilde{Z}_l(m) \end{bmatrix} = R_{WB}(i) \begin{bmatrix} X_l - P_1(i) \\ Y_l - P_2(i) \\ Z_l - P_3(i) \end{bmatrix}. \quad (\text{F.1})$$

Similarly, it can be obtained that

$$\begin{bmatrix} \tilde{X}_l(m+1) \\ \tilde{Y}_l(m+1) \\ \tilde{Z}_l(m+1) \end{bmatrix} = R_{WB}(j) \begin{bmatrix} X_l - P_1(j) \\ Y_l - P_2(j) \\ Z_l - P_3(j) \end{bmatrix}. \quad (\text{F.2})$$

Using the equations (3.50), (3.51) and the above two equations, it can be derived that

---

$$\begin{aligned}
\begin{bmatrix} \tilde{X}_l(m+1) \\ \tilde{Y}_l(m+1) \\ \tilde{Z}_l(m+1) \end{bmatrix} &= R_{WB}(j) \begin{bmatrix} X_l - P_1(i) \\ Y_l - P_2(i) \\ Z_l - P_3(i) \end{bmatrix} + R_{WB}(j)(P_W(i) - P_W(j)) \\
&= F^T(i, j) R_{WB}(i) \begin{bmatrix} X_l - P_1(i) \\ Y_l - P_2(i) \\ Z_l - P_3(i) \end{bmatrix} + R_{WB}(j)(P_W(i) - P_W(j)) \\
&= F^T(i, j) \begin{bmatrix} \tilde{X}_l(m) \\ \tilde{Y}_l(m) \\ \tilde{Z}_l(m) \end{bmatrix} + R_{WB}(j)(P_W(i) - P_W(j))
\end{aligned} \tag{F.3}$$

Therefore, the equations (3.97)-(3.99) hold.

Appendix G

VIO Error Analysis On Different Sequences

G.1 V1_01_easy

Figure G.1: Estimation Trajectories Against Ground-Truth (V1_01_easy)

(a) RMSE on Position Error

(b) RMSE on Orientation Error

(c) Absolute Error

Figure G.2: RMSE and position analysis on V1_01_easy Sequence for the proposed model and VINS_mono [163]

G.2 V1_02_medium

Figure G.3: Estimation Trajectories Against Ground-Truth (V1_02_medium)

(a) RMSE on Position Error

(b) RMSE on Orientation Error

(c) Absolute Error

Figure G.4: RMSE and position analysis on V1.02_medium Sequence for the proposed model and VINS_mono [163]

G.3 V2_01_easy

Figure G.5: Estimation Trajectories Against Ground-Truth (V2_01_easy)

(a) RMSE on Position Error

(b) RMSE on Orientation Error

(c) Absolute Error

Figure G.6: RMSE and position analysis on V2_01_easy Sequence for the proposed model and VINS_mono [163]

G.4 V2_02_medium

Figure G.7: Estimation Trajectories Against Ground-Truth (V2_02_medium)

(a) RMSE on Position Error

(b) RMSE on Orientation Error

(c) Absolute Error

Figure G.8: RMSE and position analysis on V2.02_medium Sequence for the proposed model and VINS_mono [163]

Appendix H

LIO Error Analysis On Different Sequences

H.1 Fast 2

Figure H.1: Estimation Trajectories Against Ground-Truth (Fast 2)

H.1. FAST 2

(a) RMSE on Position Error

(b) RMSE on Orientation Error

(c) Absolute Error

Figure H.2: RMSE and position analysis on Fast 2 Sequence for the proposed model and Lio [56]

H.2 Med 1

Figure H.3: Estimation Trajectories Against Ground-Truth (Med 1)

(a) RMSE on Position Error

(b) RMSE on Orientation Error

(c) Absolute Error

Figure H.4: RMSE and position analysis on Med 1 Sequence for the proposed model and Lio[56]

H.3 Med 2

Figure H.5: Estimation Trajectories Against Ground-Truth (Med 2)

(a) RMSE on Position Error

(b) RMSE on Orientation Error

(c) Absolute Error

Figure H.6: RMSE and position analysis on Med 2 Sequence for the proposed model and Lio[56]

H.4 Slow 1

Figure H.7: Estimation Trajectories Against Ground-Truth (Slow 1)

H.4. SLOW 1

(a) RMSE on Position Error

(b) RMSE on Orientation Error

(c) Absolute Error

Figure H.8: RMSE and position analysis on Slow 1 Sequence for the proposed model and Lio[56]

Appendix I

ROS Node Graph

I.1 VINS_Mono

Figure I.1: Two VIO Systems Running Simultaneously Without Affecting Each Other

I.2 CPI-Forster-Proposed

Figure I.2: CPI ROS Nodes

I.4 LIO_Mapping

Figure I.4: LIO_Mapping ROS Nodes

Appendix J

Publication

J.1 Journal Publications

- i) J. Henawy, Z. Li, W. Y. Yau, and G. Seet, “Accurate imu factor using switched linear systems for vio,” IEEE Transactions on Industrial Electronics, 2020.

J.2 Conference Publications

- i) J. Henawy, Z. Li, W. Y. Yau, G. Seet, and K. W.Wan, ”Accurate imu preintegration using switched linear systems for autonomous systems” in 2019 IEEE Intelligent Transportation Systems Conference (ITSC). IEEE, 2019, pp. 3839-3844.

J.3 Workshop Publications

- i) J. F. Chow, B. B. Kocer, J. Henawy, G. Seet, Z. Li, W. Y. Yau, and M. Pratama, ”Toward underground localization: Lidar inertial odometry enabled aerial robot navigation”, In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2019 workshop, Challenges in Vision-based Drones Navigation, Macau, China, Nov 2019.

- ii) L. J. Chen, J. Henawy, B. B. Kocer, and G. G. L. Seet, "Aerial robots on the way to underground: An experimental evaluation of vins-mono on visual inertial odometry camera," in 2019 International Conference on Data Mining Workshops (ICDMW). IEEE, 2019, pp. 91-96.

Bibliography

- [1] S. Hong, H. Ko, and J. Kim, “Vicp: Velocity updating iterative closest point algorithm,” in *Robotics and Automation (ICRA), 2010 IEEE International Conference on.* IEEE, 2010, pp. 1893–1898.
- [2] E. D. Negro, a. Boscaro, D. Michelis, C. Campinoti, and D. Nebbia, “Ground conditioning : STEP Abu Dhabi sewer project,” pp. 1925–1932, 2013.
- [3] R. E. Lawrence, E. F. Matthew, F. S. Michael, and F. T. Bradford, *North American Tunneling*, 2004, no. April.
- [4] H. Wu, G. Huang, Q. Meng, M. Zhang, and L. Li, “Deep tunnel for regulating combined sewer overflow pollution and flood disaster: A case study in guangzhou city, china,” *Water*, vol. 8, no. 8, p. 329, 2016.
- [5] Y. W. Hin, “Overview of Singapore’s Deep Tunnel Sewerage System,” vol. 2, no. Dtss 2, 2015.
- [6] S. Keisam, R. Kamei, L. B. Chanu, H. Nabachandra *et al.*, “Sewer gas poisoning: A report of two cases,” *Journal of Indian Academy of Forensic Medicine*, vol. 37, no. 3, pp. 317–319, 2015.
- [7] C. Balaguer, R. Montero, J. Victores, S. Martínez, and A. Jardón, “Towards fully automated tunnel inspection: A survey and future trends,” in *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*,

BIBLIOGRAPHY

- vol. 31. Vilnius Gediminas Technical University, Department of Construction Economics & Property, 2014, p. 1.
- [8] k. N.Aruna and P. Subhajit, “Impact of tunnels in environment,” *International Journal of Engineering Trends and Technology (IJETT)*, vol. 4, no. 8, pp. 3429–3433, 2013.
- [9] W. Bergeson and S. Ernst, “Tunnel operations, maintenance, inspection and evaluation (tomie) manual,” 2015.
- [10] S. Park, Y. Shin, Y. Oh, and S. Ahn, “A guideline on condition assessment of existing old railway tunnels,” *Tunnelling and Underground Space Technology*, vol. 21, no. 3-4, 2006.
- [11] T. C. S. No, “Guidebook on non-destructive testing of concrete structures,” 2002.
- [12] Y. Kawaguchi, I. Yoshida, H. Kurumatani, T. Kikuta, and Y. Yamada, “Internal pipe inspection robot,” in *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, vol. 1. IEEE, 1995, pp. 857–862.
- [13] S. Y. Lee, S. H. Lee, D. I. Shin, Y. K. Son, and C. S. Han, “Development of an inspection system for cracks in a concrete tunnel lining,” *Canadian Journal of Civil Engineering*, vol. 34, no. 8, pp. 966–975, 2007.
- [14] S.-N. Yu, J.-H. Jang, and C.-S. Han, “Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel,” *Automation in Construction*, vol. 16, no. 3, pp. 255–261, 2007.
- [15] H. Toutanji, “Ultrasonic wave velocity signal interpretation of simulated concrete bridge decks,” *Materials and Structures*, vol. 33, no. 3, p. 207, 2000.
- [16] J. Blitz and G. Simpson, *Ultrasonic methods of non-destructive testing*. Springer Science & Business Media, 1995, vol. 2.

BIBLIOGRAPHY

- [17] H. K. D. S. Department, *Sewerage Manual: Key planning issues and gravity collection system*. The Department, 1995, no. pt. 1. [Online]. Available: https://books.google.com.sg/books?id=a_ilQgAACAAJ
- [18] T. Asakura and Y. Kojima, “Tunnel maintenance in japan,” *Tunnelling and Underground Space Technology*, vol. 18, no. 2, pp. 161–169, 2003.
- [19] H. Mashimo and T. Ishimura, “State of the art and future prospect of maintenance and operation of road tunnel,” in *23th International Symposium on Automation and Robotics in Construction ISARC*, 2006, pp. 299–302.
- [20] K. Loupos, A. Amditis, C. Stentoumis, P. Chrobocinski, J. Victores, M. Wietek, P. Panetsos, A. Roncaglia, S. Camarinopoulos, V. Kalidromitis *et al.*, “Robotic intelligent vision and control for tunnel inspection and evaluation-the robinspect ec project,” in *Robotic and Sensors Environments (ROSE), 2014 IEEE International Symposium on*. IEEE, 2014, pp. 72–77.
- [21] A. Ahrary, “Sewer Robotics,” *Production*, 2002.
- [22] M. Liu, G. K. Egan, and F. Santoso, “Modeling, autopilot design, and field tuning of a uav with minimum control surfaces,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 6, pp. 2353–2360, 2015.
- [23] F. Santoso, M. Liu, and G. K. Egan, “Robust μ -synthesis loop shaping for altitude flight dynamics of a flying-wing airframe,” *Journal of Intelligent & Robotic Systems*, vol. 79, no. 2, p. 259, 2015.
- [24] O. E. Mahmoud, M. R. Roman, and J. F. Nasry, “Linear and nonlinear stabilizing control of quadrotor uav,” in *Engineering and Technology (ICET), 2014 International Conference on*. IEEE, 2014, pp. 1–8.

BIBLIOGRAPHY

- [25] T. Özaslan, S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, “Inspection of penstocks and featureless tunnel-like environments using micro uavs,” in *Field and Service Robotics*. Springer, 2015, pp. 123–136.
- [26] B. B. Kocer, T. Tjahjowidodo, and G. G. L. Seet, “Centralized predictive ceiling interaction control of quadrotor vtol uav,” *Aerospace Science and Technology*, vol. 76, pp. 455 – 465, 2018.
- [27] —, “Model predictive uav-tool interaction control enhanced by external forces,” *Mechatronics*, vol. 58, pp. 47–57, 2019.
- [28] B. B. Kocer, T. Tjahjowidodo, M. Pratama, and G. G. L. Seet, “Inspection-while-flying: An autonomous contact-based nondestructive test using uav-tools,” *Automation in Construction*, vol. 106, p. 102895, 2019.
- [29] D. Titterton and J. L. Weston, *Strapdown inertial navigation technology*. IET, 2004, vol. 17.
- [30] O. J. Woodman, “An introduction to inertial navigation,” University of Cambridge, Computer Laboratory, Tech. Rep. UCAM-CL-TR-696, Aug. 2007. [Online]. Available: <http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.pdf>
- [31] A. Nemra and N. Aouf, “Robust ins/gps sensor fusion for uav localization using sdre nonlinear filtering,” *IEEE Sensors Journal*, vol. 10, no. 4, pp. 789–798, 2010.
- [32] B. B. Ready and C. N. Taylor, “Inertially aided visual odometry for miniature air vehicles in gps-denied environments,” *Journal of Intelligent and Robotic Systems*, vol. 55, no. 2-3, pp. 203–221, 2009.
- [33] T. Özaslan, K. Mohta, J. Keller, Y. Mulgaonkar, C. J. Taylor, V. Kumar, J. M. Wozencraft, and T. Hood, “Towards fully autonomous visual inspection of dark

BIBLIOGRAPHY

- featureless dam penstocks using mavs,” in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 4998–5005.
- [34] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, “Robust visual inertial odometry using a direct ekf-based approach,” in *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2015, pp. 298–304.
- [35] M. Bryson and S. Sukkarieh, “Building a robust implementation of bearing-only inertial slam for a uav,” *Journal of Field Robotics*, vol. 24, no. 1-2, pp. 113–143, 2007.
- [36] X. Kong, *Inertial navigation system algorithms for low cost IMU*. Department of Mechanical and Mechatronic Engineering, Graduate School of Engineering, University of Sydney, 2000.
- [37] T. Lupton and S. Sukkarieh, “Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions,” *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 61–76, 2012.
- [38] C. Chen, H. Zhu, M. Li, and S. You, “A review of visual-inertial simultaneous localization and mapping from filtering-based and optimization-based perspectives,” *Robotics*, vol. 7, no. 3, p. 45, 2018.
- [39] G. Falco, M. Pini, and G. Marucco, “Loose and tight gnss/ins integrations: Comparison of performance assessed in real urban scenarios,” *Sensors*, vol. 17, no. 2, p. 255, 2017.
- [40] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, “A robust and modular multi-sensor fusion approach applied to mav navigation,” in *2013 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2013, pp. 3923–3929.

BIBLIOGRAPHY

- [41] A. Soloviev, D. Bates, and F. Van Graas, “Tight coupling of laser scanner and inertial measurements for a fully autonomous relative navigation solution,” *Navigation*, vol. 54, no. 3, pp. 189–205, 2007.
- [42] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, “Keyframe-based visual–inertial odometry using nonlinear optimization,” *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.
- [43] J. Gui, D. Gu, S. Wang, and H. Hu, “A review of visual inertial odometry from filtering and optimisation perspectives,” *Advanced Robotics*, vol. 29, no. 20, pp. 1289–1301, 2015.
- [44] Z. Yang and S. Shen, “Tightly-coupled visual–inertial sensor fusion based on imu pre-integration,” *Hong Kong University of Science and Technology, Tech. Rep*, 2016.
- [45] F. Lu and E. Milios, “Globally consistent range scan alignment for environment mapping,” *Autonomous robots*, vol. 4, no. 4, pp. 333–349, 1997.
- [46] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Transactions on information theory*, vol. 47, no. 2, pp. 498–519, 2001.
- [47] E. B. Olson, “Robust and efficient robotic mapping,” 2008.
- [48] R. T. Birge, “The calculation of errors by the method of least squares,” *Physical Review*, vol. 40, no. 2, p. 207, 1932.
- [49] T. S. Schei, “A finite-difference method for linearization in nonlinear estimation algorithms,” *Automatica*, vol. 33, no. 11, pp. 2053–2058, 1997.
- [50] F. Fraundorfer and D. Scaramuzza, “Visual odometry: Part ii: Matching, robustness, optimization, and applications,” *IEEE Robotics & Automation Magazine*, vol. 19, no. 2, pp. 78–90, 2012.

BIBLIOGRAPHY

- [51] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, “A tutorial on graph-based slam,” *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.
- [52] F. Lu and E. Milios, “Robot pose estimation in unknown environments by matching 2d range scans,” *Journal of Intelligent and Robotic systems*, vol. 18, no. 3, pp. 249–275, 1997.
- [53] S. Rusinkiewicz and M. Levoy, “Efficient variants of the icp algorithm,” in *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*. IEEE, 2001, pp. 145–152.
- [54] J. Mingeuz, L. Montesano, and F. Lamiroux, “Metric-based iterative closest point scan matching for sensor displacement estimation,” *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 1047–1054, 2006.
- [55] J. Zhang and S. Singh, “Loam: Lidar odometry and mapping in real-time.” in *Robotics: Science and Systems*, vol. 2, 2014.
- [56] H. Ye, Y. Chen, and M. Liu, “Tightly coupled 3d lidar inertial odometry and mapping,” in *2019 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2019.
- [57] R. C. Smith and P. Cheeseman, “On the representation and estimation of spatial uncertainty,” *The international journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1986.
- [58] R. Smith, M. Self, and P. Cheeseman, “Estimating uncertain spatial relationships in robotics,” in *Machine intelligence and pattern recognition*. Elsevier, 1988, vol. 5, pp. 435–461.

BIBLIOGRAPHY

- [59] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: part i,” *IEEE robotics & automation magazine*, vol. 13, no. 2, pp. 99–110, 2006.
- [60] T. Bailey and H. Durrant-Whyte, “Simultaneous localization and mapping (slam): Part ii,” *IEEE robotics & automation magazine*, vol. 13, no. 3, pp. 108–117, 2006.
- [61] A. Nouredin, T. B. Karamat, and J. Georgy, *Fundamentals of inertial navigation, satellite-based positioning and their integration*. Springer Science & Business Media, 2012.
- [62] R. Gonzalez, F. Rodriguez, J. L. Guzman, C. Pradalier, and R. Siegwart, “Combined visual odometry and visual compass for off-road mobile robots localization,” *Robotica*, vol. 30, no. 6, pp. 865–878, 2012.
- [63] D. Fernandez and A. Price, “Visual odometry for an outdoor mobile robot,” in *Robotics, Automation and Mechatronics, 2004 IEEE Conference on*, vol. 2. IEEE, 2004, pp. 816–821.
- [64] J. Shen, D. Tick, and N. Gans, “Localization through fusion of discrete and continuous epipolar geometry with wheel and imu odometry,” in *Proceedings of the 2011 American Control Conference*. IEEE, 2011, pp. 1292–1298.
- [65] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe, “Mobile robot positioning-sensors and techniques,” NAVAL COMMAND CONTROL AND OCEAN SURVEILLANCE CENTER RDT AND E DIV SAN DIEGO CA, Tech. Rep., 1997.
- [66] X. Dai, H. Zhang, and Y. Shi, “Autonomous navigation for wheeled mobile robots-a survey,” in *Second International Conference on Innovative Computing, Information and Control (ICICIC 2007)*. IEEE, 2007, pp. 551–551.

BIBLIOGRAPHY

- [67] O. J. Woodman, “An introduction to inertial navigation,” University of Cambridge, Computer Laboratory, Tech. Rep., 2007.
- [68] J. Farrell, *Aided navigation: GPS with high rate sensors*. McGraw-Hill, Inc., 2008.
- [69] J. Borenstein, H. Everett, L. Feng *et al.*, “Where am i? sensors and methods for mobile robot positioning,” *University of Michigan*, vol. 119, no. 120, p. 15, 1996.
- [70] A. Jiménez and F. Seco, “Ultrasonic localization methods for accurate positioning,” *Instituto de Automatica Industrial, Madrid*, 2005.
- [71] B. Kreczmer, “Objects localization and differentiation using ultrasonic sensors,” in *Robot Localization and Map Building*. InTech, 2010.
- [72] T. Shan and B. Englot, “Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4758–4765.
- [73] Y. Balazadegan Sarvrood, S. Hosseinyalamdary, and Y. Gao, “Visual-lidar odometry aided by reduced imu,” *ISPRS International Journal of Geo-Information*, vol. 5, no. 1, p. 3, 2016.
- [74] S. Hening, C. A. Ippolito, K. S. Krishnakumar, V. Stepanyan, and M. Teodorescu, “3d lidar slam integration with gps/ins for uavs in urban gps-degraded environments,” in *AIAA Information Systems-AIAA Infotech@ Aerospace*, 2017, p. 0448.
- [75] B. Schwarz, “Lidar: Mapping the world in 3d,” *Nature Photonics*, vol. 4, no. 7, p. 429, 2010.
- [76] T. Takahashi, “2d localization of outdoor mobile robots using 3d laser range data,” Ph.D. dissertation, Carnegie Mellon University, 2007.

BIBLIOGRAPHY

- [77] K. Lingemann, A. Nüchter, J. Hertzberg, and H. Surmann, “High-speed laser localization for mobile robots,” *Robotics and autonomous systems*, vol. 51, no. 4, pp. 275–296, 2005.
- [78] D. Nistér, O. Naroditsky, and J. Bergen, “Visual odometry,” in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1. Ieee, 2004, pp. I–I.
- [79] ———, “Visual odometry for ground vehicle applications,” *Journal of Field Robotics*, vol. 23, no. 1, pp. 3–20, 2006.
- [80] H. Koyasu, J. Miura, and Y. Shirai, “Real-time omnidirectional stereo for obstacle detection and tracking in dynamic environments,” in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, vol. 1. IEEE, 2001, pp. 31–36.
- [81] D. Zou and P. Tan, “Coslam: Collaborative visual slam in dynamic environments,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 2, pp. 354–366, 2012.
- [82] E. Frontoni, *Vision based mobile robotics: mobile robot localization using vision sensors and active probabilistic approaches*. Lulu. com, 2012.
- [83] W. Rone and P. Ben-Tzvi, “Mapping, localization and motion planning in mobile multi-robotic systems,” *Robotica*, vol. 31, no. 1, pp. 1–23, 2013.
- [84] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

BIBLIOGRAPHY

- [85] M. J. Milford, G. F. Wyeth, and D. Prasser, “Ratslam: a hippocampal model for simultaneous localization and mapping,” in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 1. IEEE, 2004, pp. 403–408.
- [86] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, “Monoslam: Real-time single camera slam,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.
- [87] G. Klein and D. Murray, “Parallel tracking and mapping for small ar workspaces,” in *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*. IEEE, 2007, pp. 225–234.
- [88] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, “Dtam: Dense tracking and mapping in real-time,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2320–2327.
- [89] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon, “Kinectfusion: Real-time dense surface mapping and tracking.” in *ISMAR*, vol. 11, no. 2011, 2011, pp. 127–136.
- [90] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “Orb-slam: a versatile and accurate monocular slam system,” *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [91] M. O. Aqel, M. H. Marhaban, M. I. Saripan, and N. B. Ismail, “Review of visual odometry: types, approaches, challenges, and applications,” *SpringerPlus*, vol. 5, no. 1, p. 1897, 2016.

BIBLIOGRAPHY

- [92] M. Maimone, Y. Cheng, and L. Matthies, “Two years of visual odometry on the mars exploration rovers,” *Journal of Field Robotics*, vol. 24, no. 3, pp. 169–186, 2007.
- [93] B. Zhao, T. Hu, and L. Shen, “Visual odometry-a review of approaches,” in *2015 IEEE International Conference on Information and Automation*. IEEE, 2015, pp. 2569–2573.
- [94] D. Nistér, “An efficient solution to the five-point relative pose problem,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 6, pp. 756–770, 2004.
- [95] F. Moreno-Noguer, V. Lepetit, and P. Fua, “Accurate non-iterative $O(n)$ solution to the pnp problem,” in *Computer vision, 2007. ICCV 2007. IEEE 11th international conference on*. IEEE, 2007, pp. 1–8.
- [96] P. H. Torr and A. Zisserman, “Feature based methods for structure and motion estimation,” in *International workshop on vision algorithms*. Springer, 1999, pp. 278–294.
- [97] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [98] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *European conference on computer vision*. Springer, 2006, pp. 404–417.
- [99] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (surf),” *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.
- [100] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” in *Readings in computer vision*. Elsevier, 1987, pp. 726–740.

BIBLIOGRAPHY

- [101] K. Yousif, A. Bab-Hadiashar, and R. Hoseinnezhad, “An overview to visual odometry and visual slam: Applications to mobile robotics,” *Intelligent Industrial Systems*, vol. 1, no. 4, pp. 289–311, 2015.
- [102] M. Irani and P. Anandan, “About direct methods,” in *International Workshop on Vision Algorithms*. Springer, 1999, pp. 267–277.
- [103] S. Lovegrove, A. J. Davison, and J. Ibanez-Guzmán, “Accurate visual odometry from a rear parking camera,” in *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011, pp. 788–793.
- [104] J. Engel, T. Schöps, and D. Cremers, “Lsd-slam: Large-scale direct monocular slam,” in *European Conference on Computer Vision*. Springer, 2014, pp. 834–849.
- [105] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, “Svo: Semidirect visual odometry for monocular and multicamera systems,” *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, 2017.
- [106] C. Forster, M. Pizzoli, and D. Scaramuzza, “Svo: Fast semi-direct monocular visual odometry,” in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 15–22.
- [107] A. Martinelli, “Vision and imu data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination,” *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 44–60, 2011.
- [108] D. Scaramuzza and Z. Zhang, “Visual-inertial odometry of aerial robots,” *arXiv preprint arXiv:1906.03289*, 2019.
- [109] P. Corke, J. Lobo, and J. Dias, “An introduction to inertial and visual sensing,” 2007.

BIBLIOGRAPHY

- [110] S. Weiss and R. Siegwart, “Real-time metric state estimation for modular vision-inertial systems,” in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 4531–4537.
- [111] S. Weiss, D. Scaramuzza, and R. Siegwart, “Monocular-slam-based navigation for autonomous micro helicopters in gps-denied environments,” *Journal of Field Robotics*, vol. 28, no. 6, pp. 854–874, 2011.
- [112] K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, and V. Kumar, “Robust stereo visual inertial odometry for fast autonomous flight,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 965–972, 2018.
- [113] M. Li and A. I. Mourikis, “Improving the accuracy of ekf-based visual-inertial odometry,” in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 828–835.
- [114] A. Nuchter, K. Lingemann, J. Hertzberg, and H. Surmann, “6d slam with approximate data association,” in *ICAR’05. Proceedings., 12th International Conference on Advanced Robotics, 2005*. IEEE, 2005, pp. 242–249.
- [115] J. Neira and J. D. Tardós, “Data association in stochastic mapping using the joint compatibility test,” *IEEE Transactions on robotics and automation*, vol. 17, no. 6, pp. 890–897, 2001.
- [116] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, “On-manifold preintegration for real-time visual-inertial odometry,” *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, Feb 2017.
- [117] —, “Supplementary material to: Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation,” Georgia Institute of Technology, Tech. Rep., 2015.

BIBLIOGRAPHY

- [118] D. Simon, “Optimal state estimation kalman, h and nonlinear system,” 2006.
- [119] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.
- [120] F. Daum, “Nonlinear filters: beyond the kalman filter,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 20, no. 8, pp. 57–69, 2005.
- [121] E. S. Jones and S. Soatto, “Visual-inertial navigation, mapping and localization: A scalable real-time causal approach,” *The International Journal of Robotics Research*, vol. 30, no. 4, pp. 407–430, 2011.
- [122] A. I. Mourikis and S. I. Roumeliotis, “A multi-state constraint kalman filter for vision-aided inertial navigation,” in *Robotics and automation, 2007 IEEE international conference on*. IEEE, 2007, pp. 3565–3572.
- [123] J.-P. Tardif, M. George, M. Laverne, A. Kelly, and A. Stentz, “A new approach to vision-aided inertial navigation,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 4161–4168.
- [124] M. Veth and J. Raquet, “Fusion of low-cost imaging and inertial sensors for navigation,” AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH, Tech. Rep., 2007.
- [125] J. Kelly and G. S. Sukhatme, “Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration,” *The International Journal of Robotics Research*, vol. 30, no. 1, pp. 56–79, 2011.
- [126] M. Achtelik, M. Achtelik, S. Weiss, and R. Siegwart, “Onboard imu and monocular vision based control for mavs in unknown in-and outdoor environments,” in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 3056–3063.

BIBLIOGRAPHY

- [127] S. M. Weiss, “Vision based navigation for micro helicopters,” Ph.D. dissertation, ETH Zurich, 2012.
- [128] M. Li and A. I. Mourikis, “High-precision, consistent ekf-based visual-inertial odometry,” *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.
- [129] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, “Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments,” in *2012 IEEE international conference on robotics and automation*. IEEE, 2012, pp. 957–964.
- [130] C. X. Guo and S. I. Roumeliotis, “Imu-rgbd camera 3d pose estimation and extrinsic calibration: Observability analysis and consistency improvement,” in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 2935–2942.
- [131] I. Sa, H. He, V. Huynh, and P. Corke, “Monocular vision based autonomous navigation for a cost-effective mav in gps-denied environments,” in *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. IEEE, 2013, pp. 1355–1360.
- [132] C. X. Guo, D. G. Kottas, R. DuToit, A. Ahmed, R. Li, and S. I. Roumeliotis, “Efficient visual-inertial navigation using a rolling-shutter camera with inaccurate timestamps.” in *Robotics: Science and Systems*. Citeseer, 2014.
- [133] E. Asadi and C. L. Bottasso, “Tightly-coupled stereo vision-aided inertial navigation using feature-based motion sensors,” *Advanced Robotics*, vol. 28, no. 11, pp. 717–729, 2014.
- [134] R. Munguía, E. Nuño, C. I. Aldana, and S. Urzua, “A visual-aided inertial navigation and mapping system,” *International Journal of Advanced Robotic Systems*, vol. 13, no. 3, p. 94, 2016.

BIBLIOGRAPHY

- [135] K. Wu, A. Ahmed, G. A. Georgiou, and S. I. Roumeliotis, “A square root inverse filter for efficient vision-aided inertial navigation on mobile devices.” in *Robotics: Science and Systems*, vol. 2, 2015.
- [136] N. de Palézieux, T. Nägele, and O. Hilliges, “Duo-vio: Fast, light-weight, stereo inertial odometry,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 2237–2242.
- [137] P. Tanskanen, T. Naegeli, M. Pollefeys, and O. Hilliges, “Semi-direct ekf-based monocular visual-inertial odometry,” in *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2015, pp. 6073–6078.
- [138] L. E. Clement, V. Peretroukhin, J. Lambert, and J. Kelly, “The battle for filter supremacy: A comparative study of the multi-state constraint kalman filter and the sliding window filter,” in *2015 12th Conference on Computer and Robot Vision*. IEEE, 2015, pp. 23–30.
- [139] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, “Iterated extended kalman filter based visual-inertial odometry using direct photometric feedback,” *The International Journal of Robotics Research*, vol. 36, no. 10, pp. 1053–1072, 2017.
- [140] I. Sa, M. Kamel, M. Burri, M. Bloesch, R. Khanna, M. Popović, J. Nieto, and R. Siegwart, “Build your own visual-inertial drone: A cost-effective and open-source autonomous drone,” *IEEE Robotics & Automation Magazine*, vol. 25, no. 1, pp. 89–103, 2017.
- [141] M. Brossard, S. Bonnabel, and A. Barrau, “Unscented kalman filter on lie groups for visual inertial odometry,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 649–655.
- [142] Z. Zhang, S. Liu, G. Tsai, H. Hu, C.-C. Chu, and F. Zheng, “Pirvs: An advanced visual-inertial slam system with flexible sensor fusion and hardware co-design,” in

BIBLIOGRAPHY

- 2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–7.
- [143] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, “Camera-imu-based localization: Observability analysis and consistency improvement,” *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 182–201, 2014.
- [144] T. Özaslan, G. Loianno, J. Keller, C. J. Taylor, V. Kumar, J. M. Wozencraft, and T. Hood, “Autonomous navigation and mapping for inspection of penstocks and tunnels with mavs,” *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1740–1747, 2017.
- [145] T.-C. Dong-Si and A. I. Mourikis, “Motion tracking with fixed-lag smoothing: Algorithm and consistency analysis,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5655–5662.
- [146] M. Bryson, M. Johnson-Roberson, and S. Sukkarieh, “Airborne smoothing and mapping using vision and inertial sensors,” in *Robotics and Automation, 2009. ICRA’09. IEEE International Conference on*. IEEE, 2009, pp. 2037–2042.
- [147] S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart, “Keyframe-based visual-inertial slam using nonlinear optimization,” *Proceedings of Robotis Science and Systems (RSS) 2013*, 2013.
- [148] N. Brunetto, S. Salti, N. Fioraio, T. Cavallari, and L. Stefano, “Fusion of inertial and visual measurements for rgb-d slam on mobile devices,” in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 1–9.
- [149] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, “Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation.” Georgia Institute of Technology, 2015.

BIBLIOGRAPHY

- [150] J. M. Falquez, M. Kasper, and G. Sibley, “Inertial aided dense & semi-dense methods for robust direct visual odometry,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 3601–3607.
- [151] M. Burri, H. Oleynikova, M. W. Achtelik, and R. Siegwart, “Real-time visual-inertial mapping, re-localization and planning onboard mavs in unknown environments,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 1872–1878.
- [152] J. Huai, C. K. Toth, and D. A. Grejner-Brzezinska, “Stereo-inertial odometry using nonlinear optimization,” in *International Technical Meeting of the Satellite Division of the Institute of Navigation*, 2015.
- [153] N. Keivan, A. Patron-Perez, and G. Sibley, “Asynchronous adaptive conditioning for visual-inertial slam,” in *Experimental Robotics*. Springer, 2016, pp. 309–321.
- [154] A. Concha, G. Loianno, V. Kumar, and J. Civera, “Visual-inertial direct slam,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1331–1338.
- [155] V. Usenko, J. Engel, J. Stückler, and D. Cremers, “Direct visual-inertial odometry with stereo cameras,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1885–1892.
- [156] R. Mur-Artal and J. D. Tardós, “Visual-inertial monocular slam with map reuse,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.
- [157] J.-C. Piao and S.-D. Kim, “Adaptive monocular visual-inertial slam for real-time augmented reality applications in mobile devices,” *Sensors*, vol. 17, no. 11, p. 2567, 2017.

BIBLIOGRAPHY

- [158] Y. Liu, Z. Chen, W. Zheng, H. Wang, and J. Liu, “Monocular visual-inertial slam: Continuous preintegration and reliable initialization,” *Sensors*, vol. 17, no. 11, p. 2613, 2017.
- [159] Z. Yang, F. Gao, and S. Shen, “Real-time monocular dense mapping on aerial robots using visual-inertial fusion,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4552–4559.
- [160] A. Kasyanov, F. Engelmann, J. Stückler, and B. Leibe, “Keyframe-based visual-inertial online slam with relocalization,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 6662–6669.
- [161] T. Laidlow, M. Bloesch, W. Li, and S. Leutenegger, “Dense rgb-d-inertial slam with map deformations,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 6741–6748.
- [162] P. Li, T. Qin, B. Hu, F. Zhu, and S. Shen, “Monocular visual-inertial state estimation for mobile augmented reality,” in *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2017, pp. 11–21.
- [163] T. Qin, P. Li, and S. Shen, “Vins-mono: A robust and versatile monocular visual-inertial state estimator,” *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [164] Y. Lin, F. Gao, T. Qin, W. Gao, T. Liu, W. Wu, Z. Yang, and S. Shen, “Autonomous aerial navigation using monocular visual-inertial fusion,” *Journal of Field Robotics*, vol. 35, no. 1, pp. 23–51, 2018.
- [165] C. Chen and H. Zhu, “Visual-inertial slam method based on optical flow in a gps-denied environment,” *Industrial Robot: An International Journal*, vol. 45, no. 3, pp. 401–406, 2018.

BIBLIOGRAPHY

- [166] H. Liu, M. Chen, G. Zhang, H. Bao, and Y. Bao, “Ice-ba: Incremental, consistent and efficient bundle adjustment for visual-inertial slam,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1974–1982.
- [167] K. Eickenhoff, P. Geneva, and G. Huang, “Closed-form preintegration methods for graph-based visual-inertial navigation,” *The International Journal of Robotics Research*, vol. 38, no. 5, pp. 563–586, 2019.
- [168] J. Henawy, Z. Li, W. Y. Yau, G. Seet, and K. W. Wan, “Accurate imu preintegration using switched linear systems for autonomous systems,” in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 3839–3844.
- [169] H. Strasdat, J. Montiel, and A. J. Davison, “Real-time monocular slam: Why filter?” in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 2657–2664.
- [170] A. I. Mourikis and S. I. Roumeliotis, “A dual-layer estimator architecture for long-term localization,” in *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW’08. IEEE Computer Society Conference on*. IEEE, 2008, pp. 1–8.
- [171] V. Indelman, S. Williams, M. Kaess, and F. Dellaert, “Information fusion in navigation systems via factor graph based incremental smoothing,” *Robotics and Autonomous Systems*, vol. 61, no. 8, pp. 721–738, 2013.
- [172] M. Kaess, A. Ranganathan, and F. Dellaert, “isam: Incremental smoothing and mapping,” *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.
- [173] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, “isam2: Incremental smoothing and mapping using the bayes tree,” *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

BIBLIOGRAPHY

- [174] P. J. Besl and N. D. McKay, “Method for registration of 3-d shapes,” in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. International Society for Optics and Photonics, 1992, pp. 586–606.
- [175] Y. Chen and G. Medioni, “Object modelling by registration of multiple range images,” *Image and vision computing*, vol. 10, no. 3, pp. 145–155, 1992.
- [176] A. Diosi and L. Kleeman, “Fast laser scan matching using polar coordinates,” *The International Journal of Robotics Research*, vol. 26, no. 10, pp. 1125–1153, 2007.
- [177] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, “Comparing icp variants on real-world data sets,” *Autonomous Robots*, vol. 34, no. 3, pp. 133–148, 2013.
- [178] J. Zhang and S. Singh, “Low-drift and real-time lidar odometry and mapping,” *Autonomous Robots*, vol. 41, no. 2, pp. 401–416, 2017.
- [179] J. Tang, Y. Chen, X. Niu, L. Wang, L. Chen, J. Liu, C. Shi, and J. Hyppä, “Lidar scan matching aided inertial navigation system in gnss-denied environments,” *Sensors*, vol. 15, no. 7, pp. 16 710–16 728, 2015.
- [180] G. Hemann, S. Singh, and M. Kaess, “Long-range gps-denied aerial inertial navigation with lidar localization,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1659–1666.
- [181] H. Munthe-Kaas, “High order runge-kutta methods on manifolds,” *Applied Numerical Mathematics*, vol. 29, no. 1, pp. 115–127, 1999.
- [182] G. Cooper and J. Butcher, “An iteration scheme for implicit runge—kutta methods,” *IMA journal of numerical analysis*, vol. 3, no. 2, pp. 127–140, 1983.
- [183] Z. Li, Y. Soh, and C. Wen, *Switched and impulsive systems: Analysis, design and applications*. Springer Science & Business Media, 2005, vol. 313.

BIBLIOGRAPHY

- [184] T. Hamel and R. Mahony, “Visual servoing of an under-actuated dynamic rigid-body system: an image-based approach,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 2, pp. 187–198, 2002.
- [185] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.
- [186] T. Kailath, *Linear systems*. Prentice-Hall Englewood Cliffs, NJ, 1980, vol. 156.
- [187] Z. Li, K. W. Wan, J. Zheng, Z. Zhu, Q. Ren, and Y. W. Yun, “Real time localization of uav via fusion of visual signal and imu data,” TD2015101, Tech. Rep., 2015.
- [188] K. Eickenhoff, P. Geneva, and G. Huang, “High-accuracy preintegration for visual inertial navigation,” in *International Workshop on the Algorithmic Foundations of Robotics [to appear]*, 2016.
- [189] —, “Direct visual-inertial navigation with analytical preintegration,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1429–1435.
- [190] N. Koenig and A. Howard, “Design and use paradigms for gazebo, an open-source multi-robot simulator,” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2149–2154.
- [191] F. Dellaert, “Factor graphs and gtsam: A hands-on introduction,” Georgia Institute of Technology, Tech. Rep., 2012.
- [192] J. Sola, “Quaternion kinematics for the error-state kalman filter,” *arXiv preprint arXiv:1711.02508*, 2017.
- [193] K. Eickenhoff, P. Geneva, and G. Huang, “Closed-form preintegration methods for graph-based visual-inertial navigation,” Tech. Rep., 2018. [Online]. Available: http://udel.edu/~ghuang/papers/tr_cpi.pdf

BIBLIOGRAPHY

- [194] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, “The euroc micro aerial vehicle datasets,” *The International Journal of Robotics Research*, 2016. [Online]. Available: <http://ijr.sagepub.com/content/early/2016/01/21/0278364915620033.abstract>
- [195] J. Delmerico and D. Scaramuzza, “A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2502–2509.
- [196] F. Tschopp, T. Schneider, A. W. Palmer, N. Nourani-Vatani, C. Cadena, R. Siegwart, and J. Nieto, “Experimental comparison of visual-aided odometry methods for rail vehicles,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1815–1822, 2019.
- [197] Z. Zhang and D. Scaramuzza, “A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 7244–7251.
- [198] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [199] A. Schroder, M. Renker, U. Aulenbacher, A. Murk, U. Boniger, R. Oechslein, and P. Wellig, “Numerical and experimental radar cross section analysis of the quadcopter dji phantom 2,” in *2015 IEEE Radar Conference*. IEEE, 2015, pp. 463–468.
- [200] M. Maarse, L. Sangers, J. van Ginkel, and M. Pouw, “Digital forensics on a dji phantom 2 vision+ uav,” *University of Amsterdam*, vol. 1, p. 22, 2016.
- [201] R. Hill *et al.*, “Phantom network surveillance uav/drone,” in *DEFCON Conference*, 2013.

BIBLIOGRAPHY

- [202] T. Krajník, V. Vonásek, D. Fišer, and J. Faigl, “Ar-drone as a platform for robotic research and education,” in *International conference on research and education in robotics*. Springer, 2011, pp. 172–186.
- [203] P.-J. Bristeau, F. Callou, D. Vissiere, and N. Petit, “The navigation and control technology inside the ar. drone micro uav,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 1477–1484, 2011.
- [204] L. V. Santana, A. S. Brandao, M. Sarcinelli-Filho, and R. Carelli, “A trajectory tracking and 3d positioning controller for the ar. drone quadrotor,” in *2014 international conference on unmanned aircraft systems (ICUAS)*. IEEE, 2014, pp. 756–767.
- [205] N. Dijkshoorn, “Simultaneous localization and mapping with the ar. drone,” *PhD diss., Masters thesis, Universiteit van Amsterdam*, 2012.
- [206] K. Wang, Y. Ke, M. Shan, X. Li, F. Wang, and B. M. Chen, “Development of autonomous quadrotor system for vertical replenishment,” in *2016 12th IEEE International Conference on Control and Automation (ICCA)*. IEEE, 2016, pp. 37–43.
- [207] S. Jung, S. Cho, D. Lee, H. Lee, and D. H. Shim, “A direct visual servoing-based framework for the 2016 iros autonomous drone racing challenge,” *Journal of Field Robotics*, vol. 35, no. 1, pp. 146–166, 2018.
- [208] M. Laurenzis, S. Hengy, A. Hommes, F. Kloeppel, A. Shoykhetbrod, T. Geibig, W. Johannes, P. Naz, and F. Christnacher, “Multi-sensor field trials for detection and tracking of multiple small unmanned aerial vehicles flying at low altitude,” in *Signal Processing, Sensor/Information Fusion, and Target Recognition XXVI*, vol. 10200. International Society for Optics and Photonics, 2017, p. 102001A.

BIBLIOGRAPHY

- [209] S. Jin, J. Zhang, L. Shen, and T. Li, “On-board vision autonomous landing techniques for quadrotor: A survey,” in *2016 35th Chinese Control Conference (CCC)*. IEEE, 2016, pp. 10 284–10 289.
- [210] D. Cavaliere, S. Senatore, M. Vento, and V. Loia, “Towards semantic context-aware drones for aerial scenes understanding,” in *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2016, pp. 115–121.
- [211] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, “Pixhawk: A system for autonomous flight using onboard computer vision,” in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 2992–2997.
- [212] L. Meier, P. Tanskanen, L. Heng, G. H. Lee, F. Fraundorfer, and M. Pollefeys, “Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision,” *Autonomous Robots*, vol. 33, no. 1-2, pp. 21–39, 2012.
- [213] M. Ferrera, A. Boulch, and J. Moras, “Fast stereo disparity maps refinement by fusion of data-based and model-based estimations,” in *2019 International Conference on 3D Vision (3DV)*. IEEE, 2019, pp. 9–17.
- [214] M. Chiaberge and R. Cappellaro, “Comparison of stereo visual inertial odometry algorithms for unmanned ground vehicles,” 2019.
- [215] L. Keselman, J. Iselin Woodfill, A. Grunnet-Jepsen, and A. Bhowmik, “Intel realsense stereoscopic depth cameras,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 1–10.
- [216] I. Sa, M. Kamel, M. Burri, M. Bloesch, R. Khanna, M. Popović, J. Nieto, and R. Siegwart, “Build your own visual-inertial drone: A cost-effective and open-source autonomous drone,” *IEEE Robotics Automation Magazine*, vol. 25, no. 1, pp. 89–103, March 2018.

BIBLIOGRAPHY

- [217] P. Furgale, T. D. Barfoot, and G. Sibley, “Continuous-time batch estimation using temporal basis functions,” in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 2088–2095.
- [218] P. Furgale, J. Rehder, and R. Siegwart, “Unified temporal and spatial calibration for multi-sensor systems,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1280–1286.
- [219] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, “Extending kalibr: Calibrating the extrinsics of multiple imus and of individual axes,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 4304–4311.
- [220] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, 2000.
- [221] D. Droschel, J. Stückler, and S. Behnke, “Local multi-resolution representation for 6d motion estimation and mapping with a continuously rotating 3d laser scanner,” in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5221–5226.
- [222] D. Holz and S. Behnke, “Mapping with micro aerial vehicles by registration of sparse 3d laser scans,” in *Intelligent Autonomous Systems 13*. Springer, 2016, pp. 1583–1599.
- [223] J. Zhang and S. Singh, “Enabling aggressive motion estimation at low-drift and accurate mapping in real-time,” in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5051–5058.
- [224] K. Ahmad Yousef, B. Mohd, K. Al-Widyan, and T. Hayajneh, “Extrinsic calibration of camera and 2d laser sensors without overlap,” *Sensors*, vol. 17, no. 10, p. 2346, 2017.

BIBLIOGRAPHY

- [225] J. P. Saarinen, H. Andreasson, T. Stoyanov, and A. J. Lilienthal, “3d normal distributions transform occupancy maps: An efficient representation for mapping in dynamic environments,” *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1627–1644, 2013.
- [226] K. B. Petersen, M. S. Pedersen *et al.*, “The matrix cookbook,” *Technical University of Denmark*, vol. 7, no. 15, p. 510, 2008.
- [227] A. Bunse-Gerstner, “Matrix factorizations for symplectic qr-like methods,” *Linear Algebra and its Applications*, vol. 83, pp. 49–77, 1986.
- [228] P. E. Gill and W. Murray, “Newton-type methods for unconstrained and linearly constrained optimization,” *Mathematical Programming*, vol. 7, no. 1, pp. 311–350, 1974.
- [229] R. B. Schnabel and E. Eskow, “A revised modified cholesky factorization algorithm,” *SIAM Journal on optimization*, vol. 9, no. 4, pp. 1135–1148, 1999.
- [230] T. D. Barfoot, *State estimation for robotics*. Cambridge University Press, 2017.
- [231] F. Dellaert, M. Kaess *et al.*, “Factor graphs for robot perception,” *Foundations and Trends® in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017.