

LIE ALGEBRA BASED AUGMENTED STATE EKF DESIGN FOR  
INFORMATION FUSION IN ODOMETRY

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
OF  
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HAKTAN YALÇIN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR  
THE DEGREE OF MASTER OF SCIENCE  
IN  
ELECTRICAL AND ELECTRONICS ENGINEERING

JANUARY 2025



Approval of the thesis:

**LIE ALGEBRA BASED AUGMENTED STATE EKF DESIGN FOR  
INFORMATION FUSION IN ODOMETRY**

submitted by **HAKTAN YALÇIN** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Naci Emre Altun  
Dean, Graduate School of **Natural and Applied Sciences** \_\_\_\_\_

Prof. Dr. İlkey Ulusoy  
Head of Department, **Electrical and Electronics Engineering** \_\_\_\_\_

Prof. Dr. Afşar Saranlı  
Supervisor, **Electrical and Electronics Engineering, METU** \_\_\_\_\_

Assoc. Prof. Dr. M. Mert Ankaralı  
Co-supervisor, **Electrical and Electronics Engineering, METU** \_\_\_\_\_

**Examining Committee Members:**

Prof. Dr. A. Aydın Alatan  
Electrical and Electronics Engineering, METU \_\_\_\_\_

Prof. Dr. Afşar Saranlı  
Electrical and Electronics Engineering, METU \_\_\_\_\_

Assoc. Prof. Dr. M. Mert Ankaralı  
Electrical and Electronics Engineering, METU \_\_\_\_\_

Assoc. Prof. Dr. İsmail Uyanık  
Electrical and Electronics Engineering, Hacettepe University \_\_\_\_\_

Assoc. Prof. Dr. H. Ersin Söken  
Aerospace Engineering, METU \_\_\_\_\_

Date: 10.01.2025

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Surname: Haktan Yalçın

Signature :

## ABSTRACT

### LIE ALGEBRA BASED AUGMENTED STATE EKF DESIGN FOR INFORMATION FUSION IN ODOMETRY

Yalçın, Haktan

M.S., Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. Afşar Saranlı

Co-Supervisor: Assoc. Prof. Dr. M. Mert Ankaralı

January 2025, 87 pages

Rigid transformations or rotation matrices, which inherently do not belong to any vector space, are frequently encountered in state estimation. However, many state estimation frameworks are designed to operate on vector spaces. This thesis explores the application of Lie Algebra to effectively handle rigid transformations within state estimation. Specifically, it examines methods for expressing the uncertainty associated with rigid transformations and for differentiating nonlinear functions with respect to rotation matrices. As a key contribution, an augmented state extended Kalman filter is developed to integrate incremental pose information derived from a stereo camera setup with inertial measurements. The proposed framework is evaluated using the publicly available *KITTI* dataset. Experimental results demonstrate the effectiveness of the fusion in obtaining more accurate trajectory estimation.

Keywords: Lie Algebra, Extended Kalman Filter, State Estimation, Visual Inertial Odometry, Dense Odometry

## ÖZ

### ODOMETRİDE BİLGİ FÜZYONU İÇİN LİE CEBİRİNE DAYALI ÇOKLU DURUM EKİ TASARIMI

Yalçın, Haktan

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Afşar Saranlı

Ortak Tez Yöneticisi: Doç. Dr. M. Mert Ankaralı

Ocak 2025 , 87 sayfa

Konum ve yönelim bilgisi, matematiksel olarak,  $SE(3)$  grubu olarak tanımlanır. Gruplar, çoğu zaman vektörel bir uzay oluşturmazlar. Ancak, geleneksel durum kestirimi yöntemleri, kestirilen durumun vektörel bir uzay oluşturduğu varsayımına dayanarak tasarlanmıştır. Bu tez kapsamında, Lie cebiri kullanarak, konum ve yönelime ait hatayı ve belirsizliği nasıl tanımlayabileceğimizi inceliyorum. Dahası, bu tanım üzerine inşa edilmiş çoklu durum Kalman filtresi tasarlıyorum. Tasarladığım filtresinin amacı, ikili kamera düzeneğinden gelen ardışık görüntüler arası göreceli konum bilgisi ile ataletsel sensörden gelen ölçümleri birleştirerek daha hassas ve gürbüz bir şekilde konum ve yönelim kestirimi sağlamaktır. Öne sürdüğüm yöntemi, *KITTI* verisetinde test ederek, tasarlanan filtresinin amacını yerine getirdiğini gösteriyorum.

Anahtar Kelimeler: Lie Cebiri, Genişletilmiş Kalman Filtresi, Durum Kestirimi, Görsel-Ataletsel Odometri, Yoğun Odometri

Dedicated to children.

## ACKNOWLEDGMENTS

First and foremost, I would like to express my heartfelt gratitude to my advisor, Afşar Saranlı, and my co-advisor, M. Mert Ankaralı, for their support, guidance, and patience throughout my studies. Their expertise and encouragement have been invaluable in shaping both my academic journey and personal growth.

I also extend my sincere thanks to Aydın Alatan and the OGAM (METU Center for Image Analysis) team for providing me with the opportunity to participate in inspiring projects that enriched my learning experience.

A special thanks to my friends and colleagues, from whom I have learned so much. Their support has been a constant source of innovation and motivation.

Finally, I am deeply grateful to my family. Their support has been the foundation that allowed me to pursue my dream of becoming an engineer.



## TABLE OF CONTENTS

ABSTRACT . . . . .	v
ÖZ . . . . .	vi
ACKNOWLEDGMENTS . . . . .	viii
TABLE OF CONTENTS . . . . .	ix
LIST OF TABLES . . . . .	xiii
LIST OF FIGURES . . . . .	xiv
CHAPTERS	
1 INTRODUCTION . . . . .	1
1.1 Structure of the Thesis . . . . .	3
1.2 Notation . . . . .	4
2 LITERATURE REVIEW . . . . .	7
2.1 A Historical Overview of Error State . . . . .	8
2.2 Filter Based Approaches . . . . .	10
2.3 Optimization Based Approaches . . . . .	13
3 MATHEMATICAL BACKGROUND . . . . .	15
3.1 Groups and Vector Spaces . . . . .	15
3.1.1 Groups . . . . .	15
3.1.2 Toy Example . . . . .	16

3.1.3	Differentiation w.r.t. Rotation Matrices . . . . .	17
3.2	Lie Algebra . . . . .	18
3.2.1	Definition of Lie Algebra . . . . .	19
3.2.2	<i>Wedge</i> and <i>Vee</i> Operators . . . . .	20
3.2.3	Capitalized Exponential and Logarithmic Maps . . . . .	20
3.2.4	Left and Right Jacobians . . . . .	21
3.2.5	Adjoint Matrix . . . . .	22
3.3	Inertial Navigation on Inertial Frames . . . . .	23
3.3.1	Attitude Update . . . . .	24
3.3.2	Velocity Update . . . . .	25
3.3.3	Position Update . . . . .	27
3.3.4	Summary . . . . .	28
3.4	Error State Kalman Filter . . . . .	28
3.5	VOLDOR Revisited . . . . .	29
4	METHODOLOGY . . . . .	35
4.1	Overview of the Methodology . . . . .	35
4.2	State Definition . . . . .	36
4.3	Error State Definition . . . . .	37
4.4	State Transition Model . . . . .	38
4.4.1	Error State Dynamics with Noise-Free IMU . . . . .	39
4.4.2	Error State Dynamics with Noisy IMU . . . . .	41
4.4.3	The Full State Transition Matrix . . . . .	45
4.4.4	State Augmentation . . . . .	46

4.5	Measurement Model . . . . .	47
5	EXPERIMENTS . . . . .	49
5.1	Dataset Description . . . . .	49
5.1.1	Test Sequences . . . . .	50
5.1.2	Dataset Visualization . . . . .	51
5.2	Performance Metrics . . . . .	52
5.3	Experimental Results . . . . .	53
5.3.1	<i>Seq1</i> - Error Plots . . . . .	55
5.3.2	<i>Seq2</i> - Error Plots . . . . .	56
5.3.3	<i>Seq3</i> - Error Plots . . . . .	56
5.3.4	<i>Seq4</i> - Error Plots and a Failure Case . . . . .	57
5.4	Comparison of OpenVINS and VOLDOR . . . . .	58
5.4.1	Real Time Performances . . . . .	58
5.5	Conclusions . . . . .	59
	REFERENCES . . . . .	61
	APPENDICES . . . . .	66
A	APPENDIX FOR LIE ALGEBRA . . . . .	67
A.1	Skew Symmetric Matrices . . . . .	67
A.2	Taylor Expansion of $\sin(\cdot)$ and $\cos(\cdot)$ . . . . .	68
A.3	Rotation Matrices and Rigid Transformations - Representing The Pose . . . . .	68
A.4	Matrix Exponential and Matrix Logarithm . . . . .	73
A.5	More on to Rotation Matrices . . . . .	74

A.6	Lie Algebra Derivations . . . . .	76
A.6.1	Rodrigues' Formula . . . . .	76
A.6.2	First Order Integration of Rotation Matrix . . . . .	77
A.6.3	Adjoint Matrix of $SE(3)$ . . . . .	78
A.7	IMU Dynamics . . . . .	80
A.7.1	Properties of $\Phi_t(\mathcal{T})$ . . . . .	80
B	APPENDIX FOR VISUAL SLAM . . . . .	83
B.1	Optical Flow . . . . .	83
B.2	Depth Map . . . . .	86
	CURRICULUM VITAE . . . . .	87

## LIST OF TABLES

### TABLES

Table 5.1	Available Sensors . . . . .	49
Table 5.2	Evaluated Data Sequences (Ordered with Sequence Duration) . . . .	50
Table 5.3	Numerical Comparisons . . . . .	53

## LIST OF FIGURES

### FIGURES

Figure 2.1	Comparison of a Gaussian Distribution and a Constrained Distribution . . . . .	8
Figure 2.2	Filter Based SLAM/Odometry . . . . .	11
Figure 2.3	Optimization Based SLAM/Odometry . . . . .	13
Figure 3.1	An IMU Sensor with Gyroscope and Accelerometer . . . . .	25
Figure 3.2	Frame, Optical Flow and Depth Map . . . . .	30
Figure 3.3	Stereo Frames $\mathcal{I}_{l,1}, \mathcal{I}_{l,2}, \mathcal{I}_{l,3}, \mathcal{I}_{l,4}$ and $\mathcal{I}_{r,1}$ . . . . .	31
Figure 3.4	3D Point Clouds Generated by <i>VOLDOR</i> on <i>KITTI</i> Dataset . . . . .	33
Figure 4.1	Measurement Visualized . . . . .	47
Figure 5.1	Trajectory of Sequence 1 . . . . .	50
Figure 5.2	A Sample Stereo Visual Pair From Sequence 1 . . . . .	51
Figure 5.3	Trajectories Overview . . . . .	54
Figure 5.4	Error Plot For Sequence 1 . . . . .	55
Figure 5.5	Error Plot For Sequence 2 . . . . .	56
Figure 5.6	Error Plot For Sequence 3 . . . . .	56
Figure 5.7	Error Plot For Sequence 4 . . . . .	57

Figure 5.8	Time Intervals Between Successive IMU Measurements . . . . .	57
Figure A.1	Visualization of 2 Distinct Frames ( $\alpha$ and $\beta$ ) . . . . .	69
Figure A.2	Visualization of 3 Distinct Frames ( $\alpha$ , $\beta$ and $\gamma$ ) . . . . .	71
Figure B.1	Visualization of Optical Flow . . . . .	84
Figure B.2	Disparity for Stereo Frames . . . . .	85
Figure B.3	Inverse Depth Map is Visualized . . . . .	86





## CHAPTER 1

### INTRODUCTION

Simultaneous Localization and Mapping (SLAM) and odometry are critical to enabling uninterrupted real-time pose estimation in autonomous systems like unmanned aerial vehicles, self-driving cars, indoor robotics, underwater robots, and racing drones. These algorithms typically utilize sensors such as LiDAR, cameras, and Inertial Measurement Units (IMUs) to gather fast and precise pose information, all without relying on external infrastructure or setups.

In many SLAM and odometry pipelines, the probability distribution function (*pdf*) of the state is evaluated, instead of a single point in the state space. The multivariate Gaussian density remains the gold standard for navigation systems. However, the state in navigation systems often includes rotation matrices or quaternions that do not live in a vector space but rather belong to mathematical groups. We will see that fitting a Gaussian density to a trajectory that evolves on such a manifold is impractical.

In addition to inherent challenges in the definition of the state, the state evolution through time also follows a set of nonlinear equations. Hence, the manipulation of the *pdf*, representing the pose estimation, requires linearizations and approximations. We will also see that such linearizations lead to superior information gain in unobservable state modes.

To address the aforementioned challenges, researchers have incorporated Lie Algebra into state estimation frameworks. Although Lie Algebra dates back to the 1880s, its adoption by the robotics community is relatively recent, beginning in the early 2000s. Even today, some state-of-the-art works in the literature have not fully leveraged the advantages of Lie Algebra yet.

The primary reason for the slow adoption of Lie Algebra is its inherent complexity as a deep research field, requiring a strong mathematical background. However, the aspects of Lie Algebra relevant to navigation and state estimation are relatively compact and focused. The first objective of this thesis is to provide a self-contained and concise discussion of Lie Algebra, tailored specifically to its applications in navigation and state estimation.

In the scope of this thesis, I leverage Lie Algebra to design an augmented state extended Kalman filter. The objective of the filter is to enhance the performance of stereo visual odometry by fusing measurements from an IMU. The proposed filter operates as an error-state Kalman filter, where the error state is defined using the exponential map of Lie Algebra.

The designed filter can be summarized as follows: The state is the pose and velocity of a moving body. The state transition is formulated by integrating IMU measurements between two successive image frames. The measurement in the system is defined as the relative rigid transformation between camera frames, obtained from a readily available stereo visual odometry algorithm, namely *VOLDOR* [35]. Experimental results demonstrate that fusing IMU measurements significantly improves the trajectory estimation accuracy of *VOLDOR* [35].

Notably, the state definition includes historical clones of the active state, which is why the designed filter is referred to as an augmented state filter (or multi-state filter). As a byproduct of this thesis, it will be highlighted that the augmented state formulation enables the effective incorporation of delayed measurements.

As a final remark, the proposed Extended Kalman Filter fuses relative rigid transformations with IMU measurements. As previously mentioned, the relative rigid transformations are derived using *VOLDOR* [35]. While I provide an exploration of how *VOLDOR* [35] estimates these transformations, this discussion is included solely for the sake of completeness and is not central to the content of this thesis.

## 1.1 Structure of the Thesis

Chapter 2 is dedicated to the literature review. First, I discuss the evolution of the error state definition over time. Then, I explore the most influential visual-inertial odometry (VIO) methods in the literature.

Chapter 3 presents the background. I begin with explaining the motivation for utilization of Lie Algebra in robotics applications. Next, I provide a compact and self-contained exploration of Lie Algebra. Finally, I revisit *VOLDOR* [35] to illustrate how the stereo camera setup is used to compute the relative rigid transformations between two camera frames. As highlighted in the introduction, *VOLDOR* [35] is explored for the completeness of the thesis and the reader can skip the discussion without hesitation.

Chapter 4 introduces the proposed Extended Kalman Filter framework. I provide a detailed definition of the state, including the consecutive error state. The state transition and measurement update processes are explained. While the state definition and transition equations are adopted from existing work, the measurement update equations are derived specifically as part of this thesis.

Finally, Chapter 5 presents an extensive set of experiments designed to evaluate the performance and robustness of the proposed method. This chapter details the experimental setup, datasets, evaluation metrics, and results. The experiments are conducted to demonstrate the effectiveness of the proposed framework in real-world scenarios.

## 1.2 Notation

- Basics:

- Bold uppercase characters, denoted as  $\mathbf{A}, \mathbf{B}, \mathbf{\Xi}, \dots$ , are utilized to indicate matrices.
- Bold lowercase characters, denoted as  $\mathbf{a}, \mathbf{b}, \boldsymbol{\xi}, \boldsymbol{\zeta}, \dots$ , are employed to indicate vectors.
- The character  $t$  represents a continuous time instant, whereas  $k$  is used to denote a discrete time index, as exemplified by  $\mathbf{v}_t$  and  $\mathbf{v}_k$ .

- State Items:

- $\mathbf{R}_\alpha^\beta \in \mathbb{R}^{3 \times 3}$ : Rotation Matrix from frame  $\alpha$  to frame  $\beta$ . (Column vectors of  $\mathbf{R}_\alpha^\beta$  constitute the axes of frame  $\alpha$  resolved in frame  $\beta$ )
- $\mathbf{v}_{\beta\alpha}^\gamma \in \mathbb{R}^{3 \times 1}$ : velocity of frame  $\alpha$  with respect to frame  $\beta$  resolved in frame  $\gamma$ .
- $\mathbf{t}_{\beta\alpha}^\gamma \in \mathbb{R}^{3 \times 1}$ : translation vector of frame  $\alpha$  with respect to frame  $\beta$  resolved in frame  $\gamma$ .
- To increase the readability of the notation, we have

$$\mathbf{v}_{\beta\alpha}^\beta = \mathbf{v}_\alpha^\beta \qquad \mathbf{p}_{\beta\alpha}^\beta = \mathbf{p}_\alpha^\beta$$

- $\mathbf{b}_g \in \mathbb{R}^{3 \times 1}$ : Gyroscope biases
- $\mathbf{b}_a \in \mathbb{R}^{3 \times 1}$ : Accelerometer biases

Note that the notation permits to write

$$\mathbf{t}_{\beta\alpha}^\zeta = \mathbf{R}_{\gamma\zeta}^\zeta \mathbf{t}_{\beta\alpha}^\gamma \tag{1.1}$$

$$\mathbf{t}_{\gamma\alpha}^\beta = \mathbf{t}_{\beta\alpha}^\beta - \mathbf{t}_{\beta\gamma}^\beta \tag{1.2}$$

$$\mathbf{t}_{\beta\alpha}^\gamma = -\mathbf{t}_{\alpha\beta}^\gamma \tag{1.3}$$

- Coordinate frames:

- $\mathbf{g}$ : Global frame (or the navigation frame)
- $\mathbf{b}$ : Body frame. For simplicity we assume that the body frame and IMU frame are the same.

–  $\mathbf{c}$ : Camera Frame.

- Skew-Symmetric Matrix Form

$$\boldsymbol{\Omega} \triangleq [\boldsymbol{\omega}] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \quad \text{where} \quad \boldsymbol{\omega} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \in \mathbb{R}^{3 \times 1}$$

- We represent gyroscope measurement with  $\boldsymbol{\omega} \in \mathbb{R}^{3 \times 1}$ .
- We represent accelerometer measurement with  $\mathbf{f} \in \mathbb{R}^{3 \times 1}$ .
- The gravity compensation term is denoted by  $\mathbf{g}$ .



## CHAPTER 2

### LITERATURE REVIEW

Simultaneous Localization and Mapping (SLAM) and odometry are critical for enabling real-time, uninterrupted pose estimation in autonomous systems like unmanned aerial vehicles, self-driving cars, indoor robotics, underwater robots, and racing drones. These algorithms typically utilize sensors such as LiDAR [47], cameras [38], Inertial Measurement Units (IMUs) [27], or a combination of all [53] to gather fast and precise pose information without relying on external infrastructure or setups. Throughout the thesis, I will only consider Visual Inertial Odometry (VIO) methods that utilize camera and IMU.

VIO is broadly categorized into three main approaches: filter-based [25, 23, 9], optimization-based [15, 11, 20, 39], and learning-based [49, 37]. Filter-based methods are known for their computational efficiency while maintaining reasonable accuracy. Optimization-based methods, although computationally demanding, excel in robust estimation performance, making them suitable for applications requiring high reliability. Learning-based approaches, on the other hand, offer the flexibility to learn the sensor noise characteristics and calibration without manual adjustments [2]. Recently, the hybrid models [44, 31, 40, 38], that exploits the geometry through filtering or optimization while boosting the performance through learning, have particularly shown promising results.

The concept of the error state is fundamental to both filter-based and optimization-based approaches, as it facilitates the incorporation of rotation matrices into estimation pipelines. Hence, I will begin with a historical overview of the error state, tracing its evolution through time. Following this, I delve into the fundamentals of filter-based, optimization-based approaches.

## 2.1 A Historical Overview of Error State

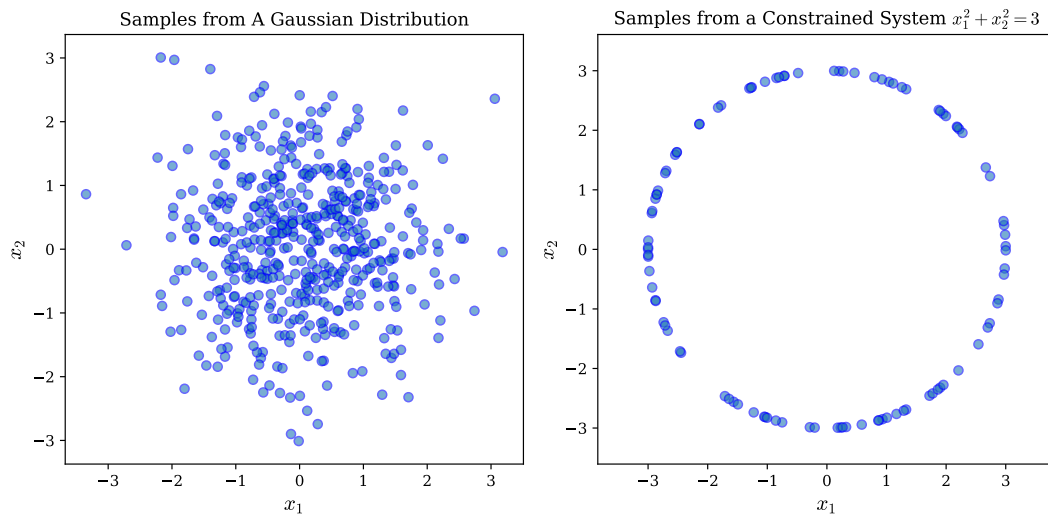


Figure 2.1: Comparison of a Gaussian Distribution and a Constrained Distribution

Both the Kalman filter framework and the optimization framework operate under the assumption that the system's state can be represented as a multivariate Gaussian distribution. At least the residual, defined as the discrepancy between the estimated measurement and the observed measurement, should live in a vector space. However, this assumption breaks down when applied to constrained states, such as rotation matrices, which cannot be accurately represented by a Gaussian distribution. This concept is illustrated in Figure 2.1, where we see that a constrained system cannot be described using a Gaussian model. The left-hand figure illustrates a Gaussian distribution, whereas the right-hand figure shows samples confined to a circular structure. This circular structure can be interpreted as a manifold. The key observation is that distributions constrained to manifolds exhibit fundamentally different properties compared to Gaussian distributions, emphasizing the need for specialized modeling techniques when analyzing data constrained by such geometrical structures.

Differentiating a nonlinear function with respect to constrained representations, such as rotation matrices and quaternions, expands the search space beyond what the problem actually requires. Moreover, during an optimization or filtering framework, the constraints might be violated. Hence, practitioners usually define a constraint-free error term, allowing differentiation to be performed with respect to the error state rather



than the full rotation matrix or quaternion. For example, early attempts such as [27] defined the error state by leveraging the small angle approximation as in Equation 2.1. This approach allowed researchers to confine the search space to  $\mathbb{R}^{3 \times 1}$  by differentiating the residual terms in the original problem with respect to  $\boldsymbol{\psi} \in \mathbb{R}^{3 \times 1}$  instead of  $\mathbf{R} \in SO(3) \subset \mathbb{R}^{3 \times 3}$ .

$$\mathbf{R} \triangleq \hat{\mathbf{R}} \tilde{\mathbf{R}} \approx \hat{\mathbf{R}}(\mathbf{I}_3 + [\boldsymbol{\psi}]) \quad (2.1)$$

In Equation 2.1, it is assumed that the estimation error is sufficiently small to justify the use of the small-angle approximation. Even the original paper of the famous MSCKF [36] uses small angle approximation to define error state. However, this introduces additional linearization errors. On the other hand, with the incorporation of Lie Algebra [41] in robotic applications, researchers defined the error term using the exponential map as in Equation 2.2.

$$\mathbf{R} \triangleq \hat{\mathbf{R}} \tilde{\mathbf{R}} = \hat{\mathbf{R}} \exp([\boldsymbol{\psi}]) = \hat{\mathbf{R}} \sum_{n=0}^{\infty} \frac{[\boldsymbol{\psi}]^n}{n!} \quad (2.2)$$

A comparison between Equation 2.1 and 2.2, illustrates that the small angle approximation in Equation 2.1 is equivalent to first order approximation of the exponential map in Equation 2.2.

Navigation entails the estimation of position and velocity of the body in addition to rotation. Hence, researchers represented the estimated state as  $SO(3) \times \mathbb{R}^{6 \times 1}$ . A separate vector  $\mathbf{x} \in \mathbb{R}^{6 \times 1}$  is utilized to store the position and velocity. As a result, the error state for velocity and position is defined through conventional subtraction operator as in Equation 2.3.

$$\tilde{\mathbf{x}} \triangleq \mathbf{x} - \hat{\mathbf{x}} \quad (2.3)$$

The error state of the overall system is defined through concatenation of  $\boldsymbol{\psi}$  and  $\tilde{\mathbf{x}}$ . Several well-known implementations in the literature, such as OpenVINS[23], and ORB-SLAM3 [11] have used  $SO(3) \times \mathbb{R}^{6 \times 1}$  representation. However, linearization of  $SO(3) \times \mathbb{R}^6$  based methods introduces an observability issue.

More recent studies, including [6] and [7], have demonstrated that linearization errors can be further minimized by employing the Extended Special Euclidean group

$SE_2(3)$ . We borrow the definition of  $SE_2(3)$  from [10] and slightly modify it to be consistent with the work [51]. Throughout the thesis,  $SE_2(3)$  is defined as in Equation 2.4.

$$SE_2(3) \triangleq \left\{ \left[ \begin{array}{c|cc} \mathbf{R} & \mathbf{t} & \mathbf{v} \\ \hline \mathbf{0}_{2 \times 3} & I_2 & \end{array} \right] \mid \mathbf{R} \in SO(3), \mathbf{t}, \mathbf{v} \in \mathbb{R}^{3 \times 1} \right\} \quad (2.4)$$

The key difference between the state representations  $SO(3) \times \mathbb{R}^{6 \times 1}$  and  $SE_2(3)$  lies in the definition of the error state, or residual. I have already defined the error state for  $SO(3) \times \mathbb{R}^{6 \times 1}$ . In contrast, the error state for the  $SE_2(3)$  group is defined through exponential map as in Equation 2.5.

$$\mathcal{T} \triangleq \hat{\mathcal{T}} \tilde{\mathcal{T}} = \hat{\mathcal{T}} \text{Exp}(\zeta^\wedge) \quad (2.5)$$

where  $\mathcal{T} \in SE_2(3)$  and  $\zeta \in \mathbb{R}^{9 \times 1}$ . Moreover, the *wedge* operator  $(\cdot)^\wedge$  for  $SE_2(3)$  is defined as follows

$$\zeta^\wedge \triangleq \left[ \begin{array}{c|cc} [\boldsymbol{\psi}] & \boldsymbol{\rho} & \boldsymbol{\nu} \\ \hline \mathbf{0}_{2 \times 3} & \mathbf{0}_{2 \times 2} & \end{array} \right] \quad (2.6)$$

For a more detailed discussion on  $SE_2(3)$ , you can refer to [10].

## 2.2 Filter Based Approaches

Assume that we have a state  $\mathbf{x}_k$ , which stores all the relevant information regarding to a system. For example, in a SLAM pipeline, the state is composed of body pose, velocity, orientation and the map of the scene. In such systems, we usually do not have a direct access to some of the state items. Hence, our goal is to estimate the whole state  $\mathbf{x}_k$ , using only the partial information available.

State is governed by its partially known dynamics, which determines the evolution of the state through time. Hence, given the *pdf* of  $\mathbf{x}_k$ , we can compute, or at least approximate, the *pdf* of  $\mathbf{x}_{k+1}$ . The *pdf* of the transition from state  $\mathbf{x}_k$  to state  $\mathbf{x}_{k+1}$  is denoted as  $p(\mathbf{x}_{k+1} | \mathbf{x}_k)$ .

In addition to state dynamics, we might have sensors such as camera, GPS and LiDAR. The probability distribution of observing a measurement  $\mathbf{z}_k$  depends on the state. The *pdf* of a measurement, conditioned on the state, is represented as  $p(\mathbf{z}_k | \mathbf{x}_k)$ .

With the definition of the state transition model  $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$  and sensor model  $p(\mathbf{z}_k|\mathbf{x}_k)$ , I visualize a filtering framework in Figure 2.2. In the Figure, I implicitly assume that the Markovian property holds. In other words,

$$p(\mathbf{x}_{k+1}|\mathbf{x}_k) = p(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{x}_{k-1}, \dots, \mathbf{x}_0)$$

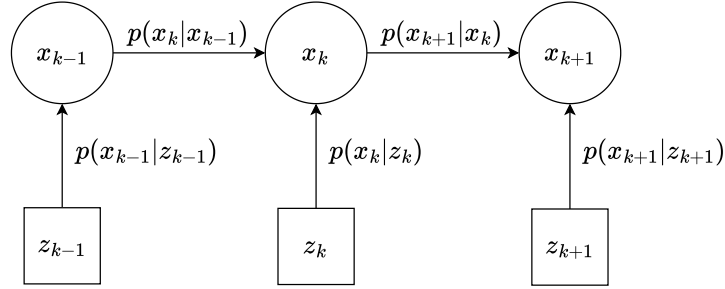


Figure 2.2: Filter Based SLAM/Odometry

The model illustrated in Figure 2.2 enables us to propose a simple and iterative odometry framework. In the proposed framework, a single iteration is composed of two steps. Assume that the initial distribution of the state,  $p(\mathbf{x}_0)$ , is available.

### 1. State Transition:

$$p(\mathbf{x}_k|\mathbf{z}_{0:k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_{0:k-1}) p(\mathbf{x}_{k-1}|\mathbf{z}_{0:k-1}) \quad (2.7)$$

$$= p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{z}_{0:k-1}) \quad (2.8)$$

where we insert that the state transition  $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_{0:k-1})$  only depends on the state itself and we let  $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_{0:k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1})$ .

### 2. Measurement Update:

$$p(\mathbf{x}_k|\mathbf{z}_{0:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{z}_{0:k-1}) p(\mathbf{x}_k|\mathbf{z}_{0:k-1})}{\int p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{z}_{0:k-1}) p(\mathbf{x}_k|\mathbf{z}_{0:k-1}) d\mathbf{x}_k} \quad (2.9)$$

$$= \frac{p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{z}_{0:k-1})}{\int p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{z}_{0:k-1}) d\mathbf{x}_k} \quad (2.10)$$

where we use the Bayes' rule. Note that  $p(\mathbf{z}_k|\mathbf{x}_k)$  is the measurement model and assumed to be available. Moreover, we have the prior distribution of  $p(\mathbf{x}_k|\mathbf{z}_{0:k-1})$  from state transition step.

A Kalman framework can be utilized to efficiently compute  $p(\mathbf{x}_k | \mathbf{z}_{0:k})$ , the posterior distribution of the state  $\mathbf{x}_k$  given all measurements up to time  $k$ . However, the performance of the filter heavily depends on the definition of the state and the available sensors.

The gold standard for filter-based visual-inertial approaches in the literature is the *Multi-State Constraint Kalman Filter* (MSCKF) [36]. Many subsequent works [43, 8, 14] have adopted and extended the state definition introduced in MSCKF [36], demonstrating its robustness and adaptability across various scenarios.

I also adopt the state in MSCKF [36], which I will elaborate on in subsequent sections. However, the state in [36] is defined as  $S^3 \times \mathbb{R}^6$ , where  $S^3$  represents the attitude in quaternion form, and  $\mathbb{R}^6$  stores three components for position and three for velocity. This formulation introduces additional linearization errors during the covariance estimation of the state. It would be beneficial to discuss the issues associated with using  $SO(3) \times \mathbb{R}^6$  or  $S^3 \times \mathbb{R}^6$  and the solutions proposed by the community to address these challenges.

The linearization of  $SO(3) \times \mathbb{R}^6$ -based methods introduces an observability issue, as the rank of the observability matrix for the linearized system exceeds that of the original nonlinear system. This discrepancy results in inconsistencies in Extended Kalman Filter EKF-based VINS estimators due to the gain of spurious information in unobservable directions [13]. To mitigate this issue, First Estimate Jacobian (FEJ) [29, 13, 15] and observability-constrained filters (OCF) [28, 33] have been proposed. FEJ maintains the evaluation of the linearized system state transition matrix and Jacobians at the initial estimate across all time periods, preventing the 4-DOF unobservable VINS subspace from gaining erroneous information [13]. Conversely, OCF enhances the consistency of  $SO(3)$  by artificially adjusting the Jacobian to enforce the system's unobservability along specified directions [48].

Lie group symmetries have proven useful for consistent observer construction [25]. Consequently, equivariant filter design has garnered significant attention in the community, such as the Invariant Extended Kalman Filter (IEKF)[7], Invariant Particle Filter [6], and the Equivariant Filter (EqF) [24]. Equivariant-filter-based VIO algorithms have been proposed to enhance the consistency and accuracy of navigation

further [48, 25].

### 2.3 Optimization Based Approaches

The key distinction between optimization and filtering lies in how the state estimate is determined. Optimization seeks to compute an optimal solution for the current state estimate by processing data within a sliding window, rather than iteratively updating the state.

Optimization-based approaches commonly utilize pose graphs [42, 19, 12]. Figure 2.3 depicts a pose graph where the nodes represent the variables to be estimated. Edges, on the other hand, are the mathematical constraints to be satisfied. The constraints can be obtained through system dynamics between states  $\{\mathbf{x}_n\}_{n=0}^N$  as well as sensor measurements between estimated variables  $\{\mathbf{z}_n\}_{n=0}^M$  and  $\{\mathbf{x}_n\}_{n=0}^N$ .

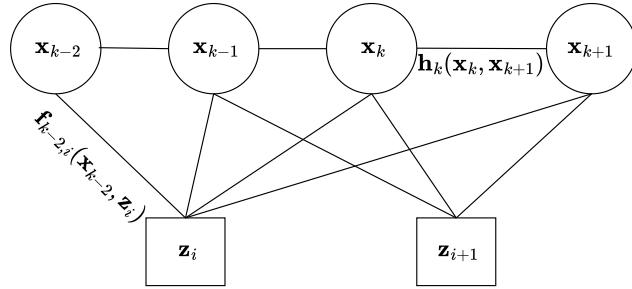


Figure 2.3: Optimization Based SLAM/Odometry

As a toy example, let us consider the connection between state  $\mathbf{x}_k$  and the landmark  $\mathbf{z}_i$ . A sensor measurement at time instant  $k$  creates a constraint in the form of

$$f_{k,i}(\mathbf{x}_k, \mathbf{z}_i) = \mathbf{0}_d \quad (2.11)$$

Due to the sensor noise and non-ideal system dynamics, there is usually no feasible solution  $\{\mathbf{z}_n\}_{n=0}^M$  and  $\{\mathbf{x}_n\}_{n=0}^N$  that satisfy all constraints. But it is possible to obtain a solution  $\{\mathbf{z}_n\}_{n=0}^M$  and  $\{\mathbf{x}_n\}_{n=0}^N$  that minimizes a cost function. Usually, the cost

function is defined as follows

$$C \triangleq \sum_{i,j} \|f_{i,j}(\mathbf{x}_i, \mathbf{z}_j)\|_{\Sigma_{f(i,j)}^{-1}} + \sum_{i,j} \|h_{i,j}(\mathbf{x}_i, \mathbf{x}_j)\|_{\Sigma_{h(i,j)}^{-1}} \quad (2.12)$$

where  $\Sigma_{h(i,j)}$  and  $\Sigma_{f(i,j)}$  are covariances of the corresponding constraints.

Many optimization libraries are available to efficiently solve optimization problems such as [17, 1]. Additionally, there are optimization libraries specifically tailored for pose graph optimization, including [32, 16, 46].

In contrary to filter based approach in Figure 2.2, pose graph permits multiple connections. We immediately recognize that the structure of the pose graph is more suitable for SLAM applications compared to filtering. For example, a landmark  $\mathbf{z}_i$  can be visible through multiple time instants  $\{\dots, k-2, k-1, \dots\}$ . However, the iterative nature of filtering does not permit such connections. To close this gap, multi-state (augmented state) Kalman filters are employed [36]. We will discuss the augmented state in detail in Section 4.2.

A concise and clear summary of graph-based SLAM is provided in [26].

## CHAPTER 3

### MATHEMATICAL BACKGROUND

I will discuss and solidify why we need Lie algebra to handle rotation matrices and rigid transformations. The reader, who is not familiar with definition and properties of rotation matrices and rigid transformations can refer to Appendix A.

The content on this section is adopted from [34] and [3].

#### 3.1 Groups and Vector Spaces

This section is engineered to illustrate the necessity of Lie Algebra, specifically in the context of handling rotation matrices within robotic applications. A formal treatment of Lie Algebra, with the foundational concepts and tools will be rigorously defined and explored in the subsequent section.

##### 3.1.1 Groups

First I provide the definition of *group*, which shows similarities to definition of vector spaces.

Consider a set  $\mathcal{G}$  and a binary operator  $\circ$  defined over the set  $\mathcal{G}$ . The pair  $(\mathcal{G}, \circ)$  is called a *group* if it satisfies the following three axioms.

1. **Closeness Under Binary Operation  $\circ$ :** The set  $\mathcal{G}$  is closed under the binary operation  $\circ$ .

$$g_1 \circ g_2 \in \mathcal{G} \quad \forall g_1, g_2 \in \mathcal{G}$$

2. **Existence of Identity:** There exists an identity element  $e$  such that

$$e \circ g = g \quad \forall g \in \mathcal{G}$$

3. **Existence of Inverse:**  $\forall g \in \mathcal{G}$ , there exists an element  $g^{-1} \in \mathcal{G}$  such that

$$g \circ g^{-1} = e$$

The reader might recognize that the rotation matrices also constitute a group under matrix multiplication.

- **Closeness Under Matrix Multiplication:**

$$\mathbf{R}_3 \stackrel{\Delta}{=} \mathbf{R}_1 \mathbf{R}_2 \in SO(3) \quad \forall \mathbf{R}_1, \mathbf{R}_2 \in SO(3)$$

- **Existence of Identity:**

$$\mathbf{I}_3 \mathbf{R} = \mathbf{R} \quad \forall \mathbf{R} \in SO(3)$$

- **Existence of Inverse**

$$\mathbf{R}^T \mathbf{R} = \mathbf{I}_3 \quad \forall \mathbf{R} \in SO(3)$$

In contrast to vector spaces, groups cannot interact with scalars. For example, we cannot multiply a rotation matrix with an arbitrary scalar to obtain a new rotation matrix.

$$a \mathbf{R} \notin SO(3) \quad \text{unless } a = 1$$

Likewise, we cannot use vector additions for groups. In other words

$$\mathbf{R}_1 + \mathbf{R}_2 \notin SO(3) \quad \text{for } \mathbf{R}_1, \mathbf{R}_2 \in SO(3)$$

### 3.1.2 Toy Example

In the previous discussion, we have seen that the groups are different from vector spaces and they are more challenging to manipulate. For example, consider that



we want to interpolate two rotation matrices,  $\mathbf{R}_1, \mathbf{R}_2 \in SO(3)$ . How do we define  $\mathbf{R}_{interpolated}$ ?

In vector spaces, we can interpolate two vectors by simply computing the average of them.

$$\mathbf{v}_{interpolated} = 0.5 (\mathbf{v}_1 + \mathbf{v}_2)$$

Unfortunately, neither the vector addition nor the scalar multiplication is a valid operation for rotation matrices. On the other hand, we can map rotation matrices into a vector space.

Inspired by Claim 12, we map  $\mathbf{R}_1, \mathbf{R}_2$  into corresponding skew symmetric matrices through matrix logarithm. Let us denote the set of skew symmetric matrices as  $\mathcal{S}$ . Here, we recall that  $\mathcal{S}$  is a vector space under matrix addition. Hence, the interpolation might take place in  $\mathcal{S}$ .

$$\begin{aligned} \mathbf{S}_1 &\triangleq \log_m(\mathbf{R}_1) \in \mathcal{S} \\ \mathbf{S}_2 &\triangleq \log_m(\mathbf{R}_2) \in \mathcal{S} \\ \mathbf{S}_{interpolated} &= 0.5 (\mathbf{S}_1 + \mathbf{S}_2) \\ \mathbf{R}_{interpolated} &= \exp_m(\mathbf{S}_{interpolated}) \end{aligned}$$

In conclusion, I point out that it might be possible to map a *group* into a vector space where robotic applications are handled easily. Throughout the thesis, we will only consider matrix Lie groups where such a mapping is always possible.

### 3.1.3 Differentiation w.r.t. Rotation Matrices

Differentiation of a nonlinear function with respect to a rotation matrix is a fundamental step both in optimization and filter frameworks. To illustrate the concept, I provide a toy example.

$$\arg \min_{\mathbf{R}^*} C = \|\mathbf{R} \mathbf{v}_1 - \mathbf{v}_2\|^2 \quad (3.1)$$

Imagine that we want to find the optimal solution for the problem given in Equation 3.1. One approach would be to compute the differentiation of the cost with respect to each element of the rotation matrix and apply gradient descent algorithm.

$$\nabla_{\mathbf{R}} C \triangleq \begin{bmatrix} \frac{dC}{dr_{11}} & \frac{dC}{dr_{12}} & \frac{dC}{dr_{13}} \\ \frac{dC}{dr_{21}} & \frac{dC}{dr_{22}} & \frac{dC}{dr_{23}} \\ \frac{dC}{dr_{31}} & \frac{dC}{dr_{32}} & \frac{dC}{dr_{33}} \end{bmatrix} \quad (3.2)$$

$$\mathbf{R}^{(k+1)} \leftarrow \mathbf{R}^{(k)} - \eta \nabla_{\mathbf{R}} C \quad (3.3)$$

where subscript  $(\cdot)^{(k)}$  denotes the optimization iteration. However we see that in Equation 3.3, we subtract a matrix from a rotation matrix. Hence, it is not guaranteed that  $\mathbf{R}^{(k+1)}$  will be still a rotation matrix. We can always insert the constraints of a rotation matrix into our optimization framework. However, we see that the search space is expanded to 9 parameters with 6 constraints while the original problem only has 3 degree of freedom.

Another approach for solving Equation 3.1 would be utilization of matrix exponential. Let us express the problem using the skew symmetric representation of the rotation matrix.

$$\arg \min_{\boldsymbol{\psi}^*} C = \|\exp_m([\boldsymbol{\psi}]) \mathbf{v}_1 - \mathbf{v}_2\|^2 \quad (3.4)$$

The new representation of the rotation  $\boldsymbol{\psi} \in \mathbb{R}^{3 \times 1}$  is a constraint free vector. Hence, we can easily apply the gradient descent method. Mathematically,

$$\nabla_{\boldsymbol{\psi}} C \triangleq \begin{bmatrix} \frac{dC}{d\psi_1} \\ \frac{dC}{d\psi_2} \\ \frac{dC}{d\psi_3} \end{bmatrix} \quad (3.5)$$

$$\boldsymbol{\psi}^{(k+1)} \leftarrow \boldsymbol{\psi}^{(k)} - \eta \nabla_{\boldsymbol{\psi}} C \quad (3.6)$$

The optimal rotation matrix  $\mathbf{R}^*$  is written as  $\mathbf{R}^* = \exp_m([\boldsymbol{\psi}^*])$ .

## 3.2 Lie Algebra

This section presents essential definitions and proofs from the literature on Lie Algebra. While a comprehensive and rigorous study of Lie Algebra necessitates a com-

plete book chapter, only a small portion of this extensive work is relevant to the thesis. All relevant definitions are comprehensively addressed within this section.

Even though provided definitions exist for a larger set of groups, we will specifically examine  $SO(3)$  and  $SE(3)$ .

### 3.2.1 Definition of Lie Algebra

Consider a smooth manifold  $\mathcal{M}$  and a point  $\mathcal{X}(t)$  which stays on the manifold for all time instants  $t$ . For smooth manifolds such as rotation matrices and rigid transformations, any velocity vector  $\dot{\mathcal{X}} \triangleq d\mathcal{X}(t)/dt$  lies within the tangent space at that point on the manifold. The tangent space at a specific point  $\mathcal{X}$  on the manifold  $\mathcal{M}$  is denoted as  $\mathcal{T}_{\mathcal{X}}\mathcal{M}$ . Moreover, the tangent space  $\mathcal{T}_{\mathcal{E}}\mathcal{M}$  at the identity element  $\mathcal{E}$  is called the **Lie Algebra** of the manifold and represented by  $\mathfrak{m}$ .

- Lie Algebra of  $SO(3)$  is represented by  $\mathfrak{so}(3)$ .

**Claim 1.**  $\mathfrak{so}(3)$  is equivalent to skew symmetric matrices. In other words

$$\mathfrak{so}(3) = \{[\psi] \mid \psi \in \mathbb{R}^{3 \times 1}\} \quad (3.7)$$

*Proof.*

$$\mathbf{R} \mathbf{R}^T = \mathbf{I}_3 \quad \rightarrow \quad \frac{d}{dt} \mathbf{R} \mathbf{R}^T = \frac{d}{dt} \mathbf{I}_3 \quad \rightarrow \quad \dot{\mathbf{R}} \mathbf{R}^T + \mathbf{R} \dot{\mathbf{R}}^T = \mathbf{0}_{3 \times 3} \quad (3.8)$$

Tangent space is equivalent to the space spanned by all possible  $\dot{\mathbf{R}}$  matrices.

We are interested in the tangent space at identity; hence, insert  $\mathbf{R} = \mathbf{I}_3$ .

$$\dot{\mathbf{R}} \mathbf{R}^T + \mathbf{R} \dot{\mathbf{R}}^T = \mathbf{0}_{3 \times 3} = \mathbf{0}_{3 \times 3} \quad \xrightarrow[\mathbf{R}=\mathbf{I}_3]{\text{at identity}} \quad \dot{\mathbf{R}} + \dot{\mathbf{R}}^T = \mathbf{0}_{3 \times 3} \quad (3.9)$$

$\dot{\mathbf{R}} + \dot{\mathbf{R}}^T = \mathbf{0}_{3 \times 3}$  implies that  $\dot{\mathbf{R}}$  is a skew symmetric matrix. Q.E.D.

- Lie Algebra of  $SE(3)$  is represented by  $\mathfrak{se}(3)$ . Moreover,

$$\mathfrak{se}(3) = \left\{ \left[ \begin{array}{c|c} [\xi_\psi] & \xi_\rho \\ \hline \mathbf{0}_{1 \times 3} & 0 \end{array} \right] \mid \xi_\psi, \xi_\rho \in \mathbb{R}^{3 \times 1} \right\} \quad (3.10)$$

### 3.2.2 Wedge and Vee Operators

Wedge operator  $(\cdot)^\wedge$  maps a vector into Lie algebra of the corresponding manifold.

Vee operator  $(\cdot)^\vee$ , on the other hand, is the inverse of the wedge operator.

- Wedge operator for  $SO(3)$  maps the vector  $\psi \in \mathbb{R}^{3 \times 1}$  into  $\mathfrak{so}(3)$ .

$$\psi^\wedge \triangleq [\psi] \in \mathfrak{so}(3) \quad (3.11)$$

$$[\psi]^\vee \triangleq \psi \in \mathbb{R}^{3 \times 1} \quad (3.12)$$

- Wedge operator for  $SE(3)$  maps the vector  $\xi \in \mathbb{R}^{6 \times 1}$  into  $\mathfrak{se}(3)$ .

$$\xi^\wedge \triangleq \begin{bmatrix} [\xi_\psi] & \xi_\rho \\ \mathbf{0}_{1 \times 3} & 0 \end{bmatrix} \in \mathfrak{se}(3) \quad (3.13)$$

$$\begin{bmatrix} [\xi_\psi] & \xi_\rho \\ \mathbf{0}_{1 \times 3} & 0 \end{bmatrix}^\vee \triangleq \xi \in \mathbb{R}^{6 \times 1} \quad (3.14)$$

$$\text{where } \xi = \begin{bmatrix} \xi_\psi \\ \xi_\rho \end{bmatrix} \text{ and } \xi_\psi, \xi_\rho \in \mathbb{R}^{3 \times 1}$$

### 3.2.3 Capitalized Exponential and Logarithmic Maps

Capitalized exponential and logarithmic map for any matrix Lie group is defined as follows.

$$\text{Exp}(\mathbf{v}) \triangleq \exp_m(\mathbf{v}^\wedge) \quad (3.15)$$

$$\text{Log}(\mathbf{V}) \triangleq \log_m(\mathbf{V})^\vee \quad (3.16)$$

Note that

$$\text{Exp}(\text{Log}(\mathbf{V})) = \mathbf{V} \quad \text{Log}(\text{Exp}(\mathbf{v})) = \mathbf{v} \quad (3.17)$$

- $\text{Exp}_{SO(3)}(\cdot) : \mathbb{R}^{3 \times 1} \rightarrow SO(3)$

$$\text{Exp}_{SO(3)}(\psi) = \exp_m(\psi^\wedge) = \exp_m([\psi]) \in SO(3) \quad (3.18)$$

Moreover, we have closed form expression for  $\text{Exp}_{SO(3)}(\mathbf{v})$ , called the Rodrigues formula.

$$\text{Exp}_{SO(3)}(\mathbf{v}) \triangleq \sum_{n=0}^{\infty} \frac{[\mathbf{v}]^n}{n!} \quad (3.19)$$

$$\xrightarrow{\text{By Claim 13}} = \mathbf{I}_3 + \frac{\sin(\|\mathbf{v}\|)}{\|\mathbf{v}\|} [\mathbf{v}] + \frac{1 - \cos(\|\mathbf{v}\|)}{\|\mathbf{v}\|^2} [\mathbf{v}]^2 \quad (3.20)$$

- $\text{Exp}_{SE(3)}(\cdot) : \mathbb{R}^{6 \times 1} \rightarrow SE(3)$

$$\text{Exp}_{SE(3)}(\boldsymbol{\xi}) = \exp_m(\boldsymbol{\xi}^\wedge) = \exp_m \left( \begin{bmatrix} [\boldsymbol{\xi}_\psi] & \boldsymbol{\xi}_\rho \\ \mathbf{0}_{1 \times 3} & 0 \end{bmatrix} \right) \in SE(3) \quad (3.21)$$

We have the closed form expression

$$\text{Exp}_{SE(3)}(\boldsymbol{\xi}) = \sum_{n=0}^{\infty} \frac{(\boldsymbol{\xi}^\wedge)^n}{n!} \quad (3.22)$$

$$= \begin{bmatrix} \text{Exp}_{SO(3)}(\boldsymbol{\xi}_\psi) & \mathcal{J}_{l_{SO(3)}}(\boldsymbol{\xi}_\psi) \boldsymbol{\xi}_\rho \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (3.23)$$

where  $\mathcal{J}_{l_{SO(3)}}(\cdot) : \mathbb{R}^{3 \times 1} \rightarrow \mathbb{R}^{3 \times 3}$  is called left Jacobian of  $SO(3)$  and defined in the next subsection.

### 3.2.4 Left and Right Jacobians

Left Jacobian of a group  $\mathcal{G}$  is defined as follows:

$$\mathcal{J}_{l_{\mathcal{G}}}(\mathbf{v}) \triangleq \lim_{\tau \rightarrow 0} \frac{\text{Log}_{\mathcal{G}} \left( \text{Exp}_{\mathcal{G}}(\mathbf{v} + \tau) \text{Exp}_{\mathcal{G}}(\mathbf{v})^{-1} \right)}{\tau} \quad (3.24)$$

$$= \frac{\partial}{\partial \tau} \text{Log}_{\mathcal{G}} \left( \text{Exp}_{\mathcal{G}}(\mathbf{v} + \tau) \text{Exp}_{\mathcal{G}}(\mathbf{v})^{-1} \right) \quad (3.25)$$

Right Jacobian of a group  $\mathcal{G}$  is defined as follows:

$$\mathcal{J}_{r_{\mathcal{G}}}(\mathbf{v}) \triangleq \lim_{\tau \rightarrow 0} \frac{\text{Log}_{\mathcal{G}} \left( \text{Exp}_{\mathcal{G}}(\mathbf{v})^{-1} \text{Exp}_{\mathcal{G}}(\mathbf{v} + \tau) \right)}{\tau} \quad (3.26)$$

$$= \frac{\partial}{\partial \tau} \text{Log}_{\mathcal{G}} \left( \text{Exp}_{\mathcal{G}}(\mathbf{v})^{-1} \text{Exp}_{\mathcal{G}}(\mathbf{v} + \tau) \right) \quad (3.27)$$

- Left and right Jacobians for  $SO(3)$ .

$$\mathcal{J}_{l_{SO(3)}}(\boldsymbol{\psi}) \triangleq \lim_{\boldsymbol{\tau} \rightarrow \mathbf{0}} \frac{\text{Log}_{SO(3)}\left(\text{Exp}_{SO(3)}(\boldsymbol{\psi} + \boldsymbol{\tau}) \text{Exp}_{SO(3)}(\boldsymbol{\psi})^{-1}\right)}{\boldsymbol{\tau}} \quad (3.28)$$

$$= \sum_{n=0}^{\infty} \frac{[\boldsymbol{\psi}]^n}{(n+1)!} \quad (3.29)$$

$$\mathcal{J}_{r_{SO(3)}}(\boldsymbol{\psi}) = \mathcal{J}_{l_{SO(3)}}(-\boldsymbol{\psi}) \quad (3.30)$$

Moreover, we have the closed form expression for  $\mathcal{J}_{l_{SO(3)}}(\boldsymbol{\psi})$ .

$$\mathcal{J}_{l_{SO(3)}}(\boldsymbol{\psi}) = \sum_{n=0}^{\infty} \frac{[\boldsymbol{\psi}]^n}{(n+1)!} \quad (3.31)$$

$$= \mathbf{I}_3 - \frac{1 - \cos(\|\boldsymbol{\psi}\|)}{\|\boldsymbol{\psi}\|^2} [\boldsymbol{\psi}] + \frac{\|\boldsymbol{\psi}\| - \sin(\|\boldsymbol{\psi}\|)}{\|\boldsymbol{\psi}\|^3} [\boldsymbol{\psi}]^2 \quad (3.32)$$

We frequently encounter the inverse of the Jacobians of  $SO(3)$  in state estimation. Hence, I also provide the related closed form expressions.

$$\mathcal{J}_{l_{SO(3)}}^{-1}(\boldsymbol{\psi}) \triangleq \sum_{n=0}^{\infty} \frac{B_n}{n!} [\boldsymbol{\psi}]^n \quad (3.33)$$

$$= \mathbf{I}_3 - \frac{1}{2} [\boldsymbol{\psi}] + \left( \frac{1}{\|\boldsymbol{\psi}\|^2} - \frac{1 + \cos(\|\boldsymbol{\psi}\|)}{2 \|\boldsymbol{\psi}\| \sin(\|\boldsymbol{\psi}\|)} \right) [\boldsymbol{\psi}]^2 \quad (3.34)$$

$$\mathcal{J}_{r_{SO(3)}}^{-1}(\boldsymbol{\psi}) = \mathcal{J}_{l_{SO(3)}}^{-1}(-\boldsymbol{\psi}) \quad (3.35)$$

$B_1, B_2, \dots$  are the *Bernoulli numbers*.

- Left and right Jacobians for  $SE(3)$  turn out to be more difficult to acquire. Hence, closed form expressions are not provided here intentionally.

### 3.2.5 Adjoint Matrix

Consider a manifold  $\mathcal{G}$  with Lie Algebra  $\mathfrak{g}$ . The adjoint matrix on a specific point  $\mathbf{V} \in \mathcal{M}$  satisfy the following equality.

$$\text{Exp}_{\mathcal{G}}(\mathbf{Ad}_{\mathbf{V}} \boldsymbol{\tau}) = \mathbf{V} \text{Exp}_{\mathcal{G}}(\boldsymbol{\tau}) \mathbf{V}^{-1} \quad (3.36)$$

where  $\boldsymbol{\tau}^{\wedge} \in \mathfrak{g}$ ,  $\boldsymbol{\tau} \in \mathbb{R}^{n \times 1}$  and  $\mathbf{Ad}_{\mathbf{V}} \in \mathbb{R}^{n \times n}$ .

Remark: Expression (3.37) is equivalent to (3.36).

$$\text{Exp}_{\mathcal{G}}(\boldsymbol{\tau}) \mathbf{V} = \mathbf{V} \text{Exp}_{\mathcal{G}}(\mathbf{Ad}_{\mathbf{V}^{-1}} \boldsymbol{\tau}) \quad (3.37)$$

- Adjoint matrix for  $SO(3)$ :

**Claim 2.** Adjoint matrix for  $SO(3)$  is simply equal to  $\text{Ad}_{\mathbf{R}} = \mathbf{R}$ . In other words, it holds that

$$\mathbf{R} \text{Exp}_{SO(3)}(\boldsymbol{\psi}) \mathbf{R}^T = \text{Exp}_{SO(3)}(\mathbf{R} \boldsymbol{\psi}), \quad \forall \mathbf{R} \in SO(3), \forall \boldsymbol{\psi} \in \mathfrak{so}(3)$$

*Proof.* We directly use the definition for the proof.

$$\text{Exp}_{SO(3)}(\mathbf{Ad}_{\mathbf{R}} \boldsymbol{\psi}) = \mathbf{R} \text{Exp}_{SO(3)}(\boldsymbol{\psi}) \mathbf{R}^T \quad (3.38)$$

$$= \mathbf{R} \sum_{n=0}^{\infty} \frac{[\boldsymbol{\psi}]^n}{n!} \mathbf{R}^T \quad (3.39)$$

$$= \sum_{n=0}^{\infty} \frac{\mathbf{R} [\boldsymbol{\psi}]^n \mathbf{R}^T}{n!} \quad (3.40)$$

$$= \sum_{n=0}^{\infty} \frac{\mathbf{R} [\boldsymbol{\psi}] \mathbf{R} \mathbf{R}^T [\boldsymbol{\psi}]^{n-1} \mathbf{R}^T}{n!} \quad (3.41)$$

$$= \sum_{n=0}^{\infty} \frac{(\mathbf{R} [\boldsymbol{\psi}] \mathbf{R}^T)^n}{n!} \quad (3.42)$$

$$= \sum_{n=0}^{\infty} \frac{[\mathbf{R} \boldsymbol{\psi}]^n}{n!} \quad (3.43)$$

$$= \text{Exp}_{SO(3)}(\mathbf{R} \boldsymbol{\psi}) \quad (3.44)$$

Q.E.D.

- Adjoint matrix for  $SE(3)$ :

$$\text{Ad}_{\mathbf{T}} = \begin{bmatrix} \mathbf{R} & \mathbf{0}_{3 \times 3} \\ [\mathbf{t}] \mathbf{R} & \mathbf{R} \end{bmatrix} \quad \text{where} \quad \mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (3.45)$$

The proof is provided in Claim 14.

### 3.3 Inertial Navigation on Inertial Frames

Inertial Measurement Units (IMUs) are essential components in navigation systems [27]. Despite their susceptibility to vibrations and drift errors that can quickly accumulate, IMUs can significantly boost the performance of SLAM and odometry when integrated with complementary sensors such as cameras, LiDAR, and GPS. Their

straightforward output and minimal computational demand make them an attractive option.

This section examines the dynamics of IMUs equipped with accelerometers and gyroscopes. Although some IMU designs incorporate magnetometers, their susceptibility to interference from nearby metals poses challenges for creating robust, generalized navigation systems.

For the completeness of the thesis, Figure 3.1 depicts an IMU with gyroscope and accelerometer measurements. The gyroscope measures the angular velocity along each axis, denoted as  $\boldsymbol{\omega}_{\text{gb}}^{\text{b}} = (\omega_x, \omega_y, \omega_z)$ . On the other hand, the accelerometer captures the linear acceleration along each axis, represented as  $\mathbf{f}_{\text{gb}}^{\text{b}} = (f_x, f_y, f_z)$ . Note that the gravity affects the accelerometer measurement. Specifically

$$\mathbf{f}_{\text{gb}}^{\text{g}} = \mathbf{a}_{\text{gb}}^{\text{b}} - \mathbf{R}_{\text{g}}^{\text{b}} \mathbf{g} \quad (3.46)$$

where  $\mathbf{g} = (0, 0, 9.81) \text{ m/sec}^2$  is the gravity term and  $\mathbf{a}_{\text{gb}}^{\text{b}}$  is the acceleration of the body  $\text{b}$  with respect to global frame  $\text{g}$  resolved in body frame  $\text{b}$ . The remaining of this section examines how to relate the IMU measurements  $\boldsymbol{\omega}_{\text{gb}}^{\text{b}}$  and  $\mathbf{f}_{\text{gb}}^{\text{b}}$  with the position, velocity and orientation of the body  $\text{b}$  with respect to global frame  $\text{g}$ .

### 3.3.1 Attitude Update

In Claim 11, we have shown that the differentiation of a rotation matrix can be expressed as

$$\dot{\mathbf{R}}_{\text{b}}^{\text{g}} = \mathbf{R}_{\text{b}}^{\text{g}} \mathbf{S} \quad (3.47)$$

where  $\mathbf{S}$  is a skew symmetric matrix. Now, I further claim that  $\mathbf{S} = [\boldsymbol{\omega}_{\text{gb}}^{\text{b}}]$ , where  $\boldsymbol{\omega}_{\text{gb}}^{\text{b}}$  is the gyroscope measurement. The detailed proof can be found in [27].

In practical applications, IMU measurements are typically recorded at time intervals of  $2\text{ms} - 20\text{ms}$ , necessitating the discretization of the system. At a given time instant  $t_k$ , the body rotation relative to the global frame is represented by  $\mathbf{R}_{\text{b}}^{\text{g}}[k]$  and the gyroscope measurement is denoted by  $\boldsymbol{\omega}_k$ . Assuming that the angular rate  $\boldsymbol{\omega}$  is constant and equal to  $\boldsymbol{\omega}_k$  for time interval  $[t_k, t_{k+1})$ , we compute the body rotation at



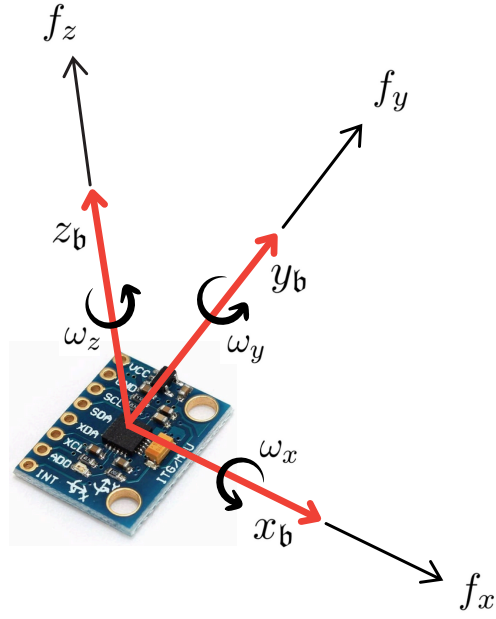


Figure 3.1: An IMU Sensor with Gyroscope and Accelerometer

time instant  $t_{k+1}$  as follows.

$$\mathbf{R}_b^g[k+1] = \mathbf{R}_b^g[k] \exp_m([\boldsymbol{\omega}_k] \Delta t) \quad (3.48)$$

where  $\Delta t = t_{k+1} - t_k$ .

### 3.3.2 Velocity Update

The differentiation of velocity with respect to time is acceleration.

$$\dot{\mathbf{v}}_{gb}^g = \mathbf{a}_{gb}^g \quad (3.49)$$

To relate the IMU measurements to  $\mathbf{a}_{gb}^g$ , it is crucial to note that the IMU measures  $\mathbf{a}_{gb}^b$  with gravity term, rather than directly measuring  $\mathbf{a}_{gb}^g$ . There is a simple conversion.

$$\xrightarrow{\text{Eq 3.46}} \mathbf{f}_{gb}^b = \mathbf{a}_{gb}^b - \mathbf{R}_g^b \mathbf{g} \quad (3.50)$$

$$= \mathbf{R}_g^b \mathbf{a}_{gb}^g - \mathbf{R}_g^b \mathbf{g} \quad (3.51)$$

$$\Rightarrow \mathbf{a}_{gb}^g = \mathbf{R}_g^b \mathbf{f}_{gb}^b + \mathbf{g} \quad (3.52)$$

As the final step, we need to discretize the Equation (3.49).

$$\dot{\mathbf{v}}_{\text{gb}}^{\text{g}}(t) = \mathbf{a}_{\text{gb}}^{\text{g}}(t) \quad (3.53)$$

$$\xrightarrow{\text{Eq. (3.52)}} = \mathbf{R}_{\text{b}}^{\text{g}}(t) \mathbf{f}_{\text{gb}}^{\text{b}}(t) + \mathbf{g} \quad (3.54)$$

$$\Rightarrow \mathbf{v}_{\text{gb}}^{\text{g}}[k+1] = \mathbf{v}_{\text{gb}}^{\text{g}}[k] + \int_{t_k}^{t_{k+1}} \left( \mathbf{R}_{\text{b}}^{\text{g}}(t) \mathbf{f}_{\text{gb}}^{\text{b}}(t) + \mathbf{g} \right) dt \quad (3.55)$$

$$= \mathbf{v}_{\text{gb}}^{\text{g}}[k] + \int_{t_k}^{t_{k+1}} \left( \mathbf{R}_{\text{b}}^{\text{g}}(t) \mathbf{f}_{\text{gb}}^{\text{b}}(t) \right) dt + \mathbf{g} \Delta t \quad (3.56)$$

$$= \mathbf{v}_{\text{gb}}^{\text{g}}[k] + \int_{t_k}^{t_{k+1}} \left( \mathbf{R}_{\text{b}}^{\text{g}}(t) \mathbf{f}_{\text{gb}}^{\text{b}}(t) \right) dt + \mathbf{g} \Delta t \quad (3.57)$$

Assuming that the specific force  $\mathbf{f}_{\text{gb}}^{\text{b}}(t)$  is constant and equal to  $\mathbf{f}_k$  for time interval  $[t_k, t_{k+1})$ , we obtain

$$\mathbf{v}_{\text{gb}}^{\text{g}}[k+1] = \mathbf{v}_{\text{gb}}^{\text{g}}[k] + \int_{t_k}^{t_{k+1}} \mathbf{R}_{\text{b}}^{\text{g}}(t) \mathbf{f}_k dt + \mathbf{g} \Delta t \quad (3.58)$$

$$= \mathbf{v}_{\text{gb}}^{\text{g}}[k] + \int_{t_k}^{t_{k+1}} \mathbf{R}_{\text{b}}^{\text{g}}(t) dt \mathbf{f}_k + \mathbf{g} \Delta t \quad (3.59)$$

For time interval  $[t_k, t_{k+1})$ , we have  $\mathbf{R}_{\text{b}}^{\text{g}}(t) = \mathbf{R}_{\text{b}}^{\text{g}}[k] \exp_m \left( [\boldsymbol{\omega}_k] (t - t_k) \right)$

$$\mathbf{v}_{\text{gb}}^{\text{g}}[k+1] = \mathbf{v}_{\text{gb}}^{\text{g}}[k] + \int_{t_k}^{t_{k+1}} \mathbf{R}_{\text{b}}^{\text{g}}(t) dt \mathbf{f}_k + \mathbf{g} \Delta t \quad (3.60)$$

$$= \mathbf{v}_{\text{gb}}^{\text{g}}[k] + \int_{t_k}^{t_{k+1}} \mathbf{R}_{\text{b}}^{\text{g}}[k] \exp_m \left( [\boldsymbol{\omega}_k] (t - t_k) \right) dt \mathbf{f}_k + \mathbf{g} \Delta t \quad (3.61)$$

$$= \mathbf{v}_{\text{gb}}^{\text{g}}[k] + \mathbf{R}_{\text{b}}^{\text{g}}[k] \int_{t_k}^{t_{k+1}} \exp_m \left( [\boldsymbol{\omega}_k] (t - t_k) \right) dt \mathbf{f}_k + \mathbf{g} \Delta t \quad (3.62)$$

Even though a closed form expression for (3.62) is available in [50], practitioners usually tend to use an approximation of (3.60). The most simple approximation of (3.60) is that  $\mathbf{R}_{\text{b}}^{\text{g}}(t)$  is also assumed to be constant for the small time interval  $[t_k, t_{k+1})$ .

$$\mathbf{v}_{\text{gb}}^{\text{g}}[k+1] = \mathbf{v}_{\text{gb}}^{\text{g}}[k] + \int_{t_k}^{t_{k+1}} \mathbf{R}_{\text{b}}^{\text{g}}(t) dt \mathbf{f}_k + \mathbf{g} \Delta t \quad (3.63)$$

$$\approx \mathbf{v}_{\text{gb}}^{\text{g}}[k] + \int_{t_k}^{t_{k+1}} \mathbf{R}_{\text{b}}^{\text{g}}[k] dt \mathbf{f}_k + \mathbf{g} \Delta t \quad (3.64)$$

$$= \mathbf{v}_{\text{gb}}^{\text{g}}[k] + \mathbf{R}_{\text{b}}^{\text{g}}[k] \mathbf{f}_k \Delta t + \mathbf{g} \Delta t \quad (3.65)$$

One the other hand, [27] uses triangle approximation.

$$\mathbf{v}_{\text{gb}}^{\text{g}}[k+1] = \mathbf{v}_{\text{gb}}^{\text{g}}[k] + \int_{t_k}^{t_{k+1}} \mathbf{R}_{\text{b}}^{\text{g}}(t) dt \mathbf{f}_k + \mathbf{g}\Delta t \quad (3.66)$$

$$\approx \mathbf{v}_{\text{gb}}^{\text{g}}[k] + 0.5 \left( \mathbf{R}_{\text{b}}^{\text{g}}[k] + \mathbf{R}_{\text{b}}^{\text{g}}[k+1] \right) \mathbf{f}_k + \mathbf{g}\Delta t \quad (3.67)$$

In the scope of this thesis, I use the closed form solution provided by [50]. For the convenience of the reader, I explain how to take the integral of a rotation matrix in appendix A.6.2.

### 3.3.3 Position Update

We have the simple expression

$$\dot{\mathbf{t}}_{\text{gb}}^{\text{g}}(t) = \mathbf{v}_{\text{gb}}^{\text{g}}(t) \quad (3.68)$$

Apply discretization

$$\mathbf{t}_{\text{gb}}^{\text{g}}[k+1] = \mathbf{t}_{\text{gb}}^{\text{g}}[k] + \int_{t_k}^{t_{k+1}} \mathbf{v}_{\text{gb}}^{\text{g}}(t) dt \quad (3.69)$$

$$\xrightarrow{\text{By Eq. (3.62)}} = \mathbf{t}_{\text{gb}}^{\text{g}}[k] + \int_{t_k}^{t_{k+1}} \left( \mathbf{v}_{\text{gb}}^{\text{g}}[k] + \mathbf{R}_{\text{b}}^{\text{g}}[k] \int_{t_k}^t \exp_m \left( [\boldsymbol{\omega}_k] (\tau - t_k) \right) d\tau \mathbf{f}_k + \mathbf{g}\Delta t \right) dt \quad (3.70)$$

A few algebraic manipulation leads to

$$\begin{aligned} \mathbf{t}_{\text{gb}}^{\text{g}}[k+1] &= \mathbf{t}_{\text{gb}}^{\text{g}}[k] + \mathbf{v}_{\text{gb}}^{\text{g}}[k] \Delta t + 0.5\Delta t^2 \mathbf{g} + \dots \\ &\dots + \mathbf{R}_{\text{b}}^{\text{g}}[k] \int_{t_k}^{t_{k+1}} \int_{t_k}^t \exp_m \left( [\boldsymbol{\omega}_k] (\tau - t_k) \right) d\tau dt \mathbf{f}_k \end{aligned} \quad (3.71)$$

Again, a closed form expression for the term  $\int_{t_k}^{t_{k+1}} \int_{t_k}^t \exp_m \left( [\boldsymbol{\omega}_k] (\tau - t_k) \right) d\tau dt$  is provided in [50]. The derivation is similar to appendix A.6.2. However, many practitioners prefer to use triangle approximation for solving (3.69), which leads to a simple and intuitive solution.

$$\mathbf{t}_{\text{gb}}^{\text{g}}[k+1] \approx \mathbf{t}_{\text{gb}}^{\text{g}}[k] + 0.5(\mathbf{v}_{\text{gb}}^{\text{g}}[k] + \mathbf{v}_{\text{gb}}^{\text{g}}[k+1]) \quad (3.72)$$

Throughout the thesis, I will employ the closed form expression provided by [50] rather than using the approximations.

### 3.3.4 Summary

IMU dynamics can be summarized with 3 equations

#### 1. Attitude Update

$$\mathbf{R}_b^g[k+1] = \mathbf{R}_b^g[k] \exp_m([\boldsymbol{\omega}_k] \Delta t)$$

#### 2. Velocity Update

$$\begin{aligned} \mathbf{v}_{gb}^g[k+1] &= \mathbf{v}_{gb}^g[k] + \mathbf{R}_b^g[k] \int_{t_k}^{t_{k+1}} \exp_m([\boldsymbol{\omega}_k] (t - t_k)) dt \mathbf{f}_k + \mathbf{g} \Delta t \\ &\approx \mathbf{v}_{gb}^g[k] + 0.5 \left( \mathbf{R}_b^g[k] + \mathbf{R}_b^g[k+1] \right) \mathbf{f}_k + \mathbf{g} \Delta t \end{aligned}$$

#### 3. Position Update

$$\begin{aligned} \mathbf{t}_{gb}^g[k+1] &= \mathbf{t}_{gb}^g[k] + \mathbf{v}_{gb}^g[k] \Delta t + 0.5 \Delta t^2 \mathbf{g} + \dots \\ &\quad \dots + \mathbf{R}_b^g[k] \int_{t_k}^{t_{k+1}} \int_{t_k}^t \exp_m([\boldsymbol{\omega}_k] (\tau - t_k)) d\tau dt \mathbf{f}_k \\ &\approx \mathbf{t}_{gb}^g[k] + 0.5 (\mathbf{v}_{gb}^g[k] + \mathbf{v}_{gb}^g[k+1]) \end{aligned}$$

## 3.4 Error State Kalman Filter

In this section, I provide an overview of error state EKF framework. Assume that the actual state at time instant  $k$  is represented by  $\mathcal{X}_k$ . The actual state might be composed of vectors and groups. On the other hand, the error state is represented by  $\delta \mathbf{x}$ , that lives in a vector space. The error state EKF is summarized as follows

- Prediction step

$$\mathcal{X}_{k+1|k}, \delta \mathbf{x}_+ = f(\mathcal{X}_{k|k}, \delta \mathbf{x} = \mathbf{0}, \boldsymbol{\eta}_q = \mathbf{0}) \quad (3.73)$$

$$\boldsymbol{\Sigma}_{k+1|k} = \mathbf{F}_k \boldsymbol{\Sigma}_{k|k} \mathbf{F}_k^T + \mathbf{G}_k \mathbf{Q} \mathbf{G}_k^T \quad (3.74)$$

I assume that the error state is defined such that  $\delta \mathbf{x}_+ = \mathbf{0}$  though its covariance  $\boldsymbol{\Sigma}_{k+1|k}$  has nonzero entries.

- Measurement Update

$$\mathbf{S}_k = \mathbf{H}_k \Sigma_{k+1|k} \mathbf{H}_k^T + \mathbf{W}_k \mathbf{R} \mathbf{W}_k^T \quad (3.75)$$

$$\mathbf{K}_k = \Sigma_{k+1|k} \mathbf{H}_k^T \mathbf{S}_k^{-1} \quad (3.76)$$

$$\delta \mathbf{x} = \mathbf{K}_k \left( z_k - h(\mathcal{X}_{k+1|k}, \delta \mathbf{x} = \mathbf{0}, \boldsymbol{\eta}_r = \mathbf{0}) \right) \quad (3.77)$$

$$\mathcal{X}_{k+1|k+1} \leftarrow \mathcal{X}_{k+1|k} \oplus \delta \mathbf{x} \quad (3.78)$$

where

$$\mathbf{F}_k \triangleq \frac{\partial \delta \mathbf{x}_+}{\partial \delta \mathbf{x}} \quad \mathbf{G}_k \triangleq \frac{\partial \delta \mathbf{x}_+}{\partial \boldsymbol{\eta}_q} \quad (3.79)$$

$$\mathbf{H}_k \triangleq \frac{\partial \mathbf{h}(\cdot)}{\partial \delta \mathbf{x}} \quad \mathbf{W}_k \triangleq \frac{\partial \mathbf{h}(\cdot)}{\partial \boldsymbol{\eta}_r} \quad (3.80)$$

The sign  $\oplus$  in (3.78) is a special addition operation which depends on the definition of the error state.

It is important to note that both the actual state  $\mathcal{X}$  and the error state  $\delta \mathbf{x}$  evolve according to the same dynamics  $f(\cdot)$  in (3.75). However, the evaluated covariance corresponds to the error state. Consequently, the Jacobians in (3.79) and (3.80) are computed with respect to the error state,  $\delta \mathbf{x}$ . This approach allows us to operate within a vector space, even though the actual state  $\mathcal{X}$  does not belong to any vector space.

### 3.5 VOLDOR Revisited

**Remark:** This section, which is provided for the completeness of the thesis, is not fully integrated into the scope of this work. Hence, the reader can skip without any hesitation. In the proposed method, I solely utilize the output of *VOLDOR* [35] without any intervention in its inner dynamics. The output of *VOLDOR* [35] is discussed in detail in the relevant section.

**Warning:** Readers who are not familiar with *optical flow* and *depth maps* are encouraged to review Appendices B.1 and B.2. In these sections, I provide an exploration of optical flow and depth maps, supplemented with detailed visuals.

*VOLDOR* [35] is a stereo visual odometry algorithm introduced in 2020 at the *International Conference on Robotics and Automation (ICRA)*. Unlike other visual naviga-

tion systems, *VOLDOR* [35] processes optical flow vectors between successive image frames instead of the raw image data. Consequently, *VOLDOR* [35] performs SLAM directly using dense optical flow.

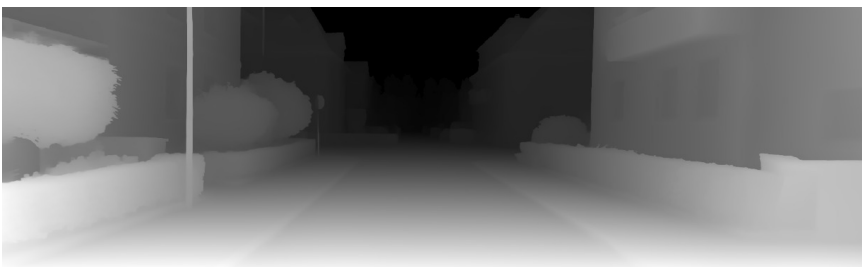
Figure 3.2 illustrates a sample frame from KITTI dataset with optical flow and corresponding inverse depth map.



(a) Frame



(b) Optical Flow



(c) Depth Map

Figure 3.2: Frame, Optical Flow and Depth Map

Consider two image frames  $\mathcal{I}_1 \in \mathbb{R}^{H \times W \times 3}$  and  $\mathcal{I}_2 \in \mathbb{R}^{H \times W \times 3}$ . The relative poses between two camera frames,  $\mathbf{T}_{c_2}^{c_1} \in SE(3)$  is defined such that

$$\mathbf{T}_{c_2}^g = \mathbf{T}_{c_1}^g \mathbf{T}_{c_2}^{c_1}$$

Denote the depth map of  $\mathcal{I}_1$  as  $\mathcal{D}_1 \in \mathbb{R}^{H \times W \times 1}$ . Finally, optical flow vectors from  $\mathcal{I}_1$  to  $\mathcal{I}_2$  are stored in  $\mathcal{F}_{1 \rightarrow 2} \in \mathbb{R}^{H \times W \times 2}$ .

**Claim 3.** Given  $\mathcal{F}_{1 \rightarrow 2}$  and  $\mathbf{T}_{c_2}^{c_1}$ , we can compute  $\mathcal{D}_1$ .

**Claim 4.** Given  $\mathcal{F}_{1 \rightarrow 2}$  and  $\mathcal{D}_1$ , we can compute  $\mathbf{T}_{c_2}^{c_1}$ .

Extraction of optical flow from image pairs is a well-studied problem in literature [18, 30, 54] and  $\mathcal{F}_{1 \rightarrow 2}$  can be easily extracted. During the experiments, RAFT [45] is used to extract optical flow vectors. However, the difficulty in visual SLAM usually arises from the fact that we neither have  $\mathbf{T}_{c_2}^{c_1}$  between successive frames nor the depth map  $\mathcal{D}_1$ .

*VOLDOR* [35] employs a calibrated stereo camera setup to solve the unknown  $\mathbf{T}_{c,k+1}^{c,k}$  and  $\mathcal{D}_1$  dilemma. Let us denote the successive frames belonging to left camera as  $\mathcal{I}_{l,1}, \mathcal{I}_{l,2}, \mathcal{I}_{l,3} \in \mathbb{R}^{H \times W \times 3}$ . Similarly, frames belonging to right camera is denoted as  $\mathcal{I}_{r,1}, \mathcal{I}_{r,2}, \dots$ . Frames  $\mathcal{I}_{l,1}, \mathcal{I}_{l,2}, \mathcal{I}_{l,3}, \mathcal{I}_{l,4}$  and  $\mathcal{I}_{r,1}$  are visualized in Figure 3.3.



Figure 3.3: Stereo Frames  $\mathcal{I}_{l,1}, \mathcal{I}_{l,2}, \mathcal{I}_{l,3}, \mathcal{I}_{l,4}$  and  $\mathcal{I}_{r,1}$

We have the relative poses of the right and left camera,  $\mathbf{T}_{c,l}^{c,r}$ . Then, by Claim 3, we can compute the depth map  $\mathcal{D}_{l,1}$  for the image frame  $\mathcal{I}_{l,1}$ . Now, we have  $\mathcal{D}_{l,1}$ ; hence, we can compute  $\mathbf{T}_{c,l_2}^{c,l_1}$  by the Claim 4. Doing the same operation for each arrived image pair, we can obtain the full trajectory.

$$\mathbf{T}_{c,l_n}^{c,g} = \mathbf{T}_{c,l_0}^{c,g} \mathbf{T}_{c,l_1}^{c,l_0} \mathbf{T}_{c,l_2}^{c,l_1} \cdots \mathbf{T}_{c,l_n}^{c,l_{n-1}} \quad (3.81)$$

This simple, yet effective, approach enables us to execute a dense odometry. A sample point cloud obtained by *VOLDOR* [35] is depicted in Figure 3.4. The curious reader is encouraged to read [35] for details.

For the sake of notation simplicity, the incremental pose is defined only for the left camera and is denoted as  $\mathbf{T}_{c(n+1)}^{c,n}$ . For the remainder of the thesis, I will assume that the incremental pose is readily available.



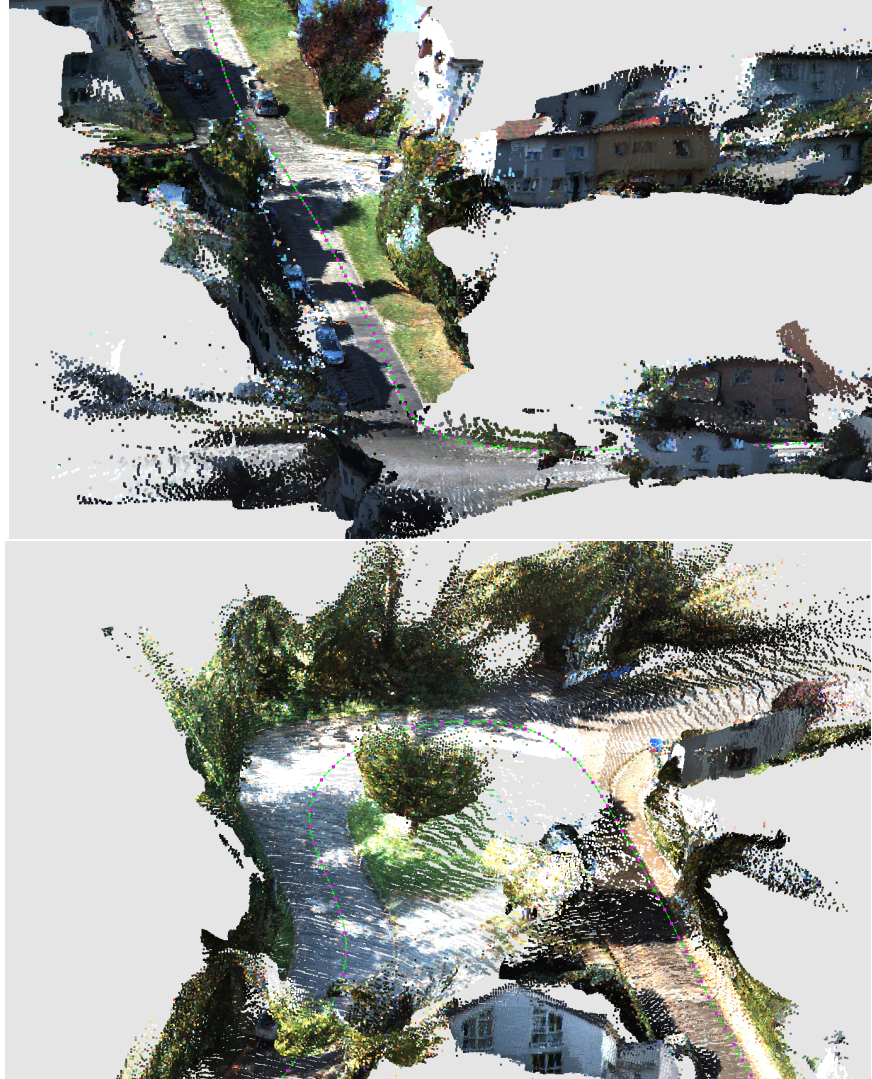


Figure 3.4: 3D Point Clouds Generated by *VOLDOR* on *KITTI* Dataset



## CHAPTER 4

### METHODOLOGY

In the previous Chapter, we first explored the Special Euclidean Group  $SE(3)$  and Extended Special Euclidean Group  $SE_2(3)$ . We discussed how to represent and manipulate the error term regarding to these groups through Lie Algebra.

Moreover, we have examined the geometric relations of the IMU measurements and the body state. However, we have not discussed how to propagate the error state of the body, which lives in  $\mathfrak{se}_2(3)$ , through IMU measurements.

The previous chapter, finally examined a visual SLAM system, demonstrating that there is a geometric relation between the depth map of the image, successive optical flow vectors and the motion of the camera. Moreover, I provided an overview of *VOLDOR* [35], which exploits these geometric constraints to construct a Stereo Visual Odometry pipeline.

In this Chapter, on the other hand, I will examine how the error state evolves given the successive IMU measurements. Moreover, I will propose an extended Kalman Filter framework to fuse IMU measurements to enhance the performance of *VOLDOR* [35].

#### 4.1 Overview of the Methodology

In order to implement a Kalman filter, we have to define 3 core components

1. State Definition (including the actual state and error state)
2. State Transition (Time propagation)
3. Measurement Update Equation

I will define the state in Section 4.2 with the corresponding error state explored in Section 4.3. Then, the state transition will be defined through IMU measurements in Section 4.4. The dynamics of the error state with respect to IMU measurements will be examined in this section.

Finally, we will examine the measurement model in Section 4.5. Here, the measurement is defined as the iterative (or incremental) camera poses obtained by *VOLDOR* [35]. In this section, I assume that incremental camera poses are available without any reference to *VOLDOR* [35]. The focus will be fusion of these incremental poses with IMU measurements.

I remark the fact that the state definition and state transition proposed in this thesis is adopted from the existing work. However, necessity for fusing incremental pose with IMU measurements is indicated within the scope of this thesis and the resulting Kalman measurement update equations are derived as a contribution to the state estimation community.

## 4.2 State Definition

The state definition adopted in this work follows the structure introduced in MSCKF [36], that consists of the active IMU state and its historical clones. Specifically, the full state is represented as:

$$\mathbf{x} = \left[ \mathbf{x}_{IMU} \quad \mathbf{x}_{clone,1} \quad \mathbf{x}_{clone,2} \quad \cdots \quad \mathbf{x}_{clone,N} \right] \quad (4.1)$$

This structure is also called the augmented state [5]. Active IMU state, denoted as  $\mathbf{x}_{IMU}$ , stores the current up-to-date estimations of rotation, position and velocity of the body  $\mathfrak{b}$  with respect to global reference frame  $\mathfrak{g}$ . In addition, the active IMU state also includes the estimated biases for the gyroscope and accelerometer, represented as  $\mathbf{b}_g$  and  $\mathbf{b}_a$ , respectively.

$$\mathbf{x}_{IMU} = \left[ \mathbf{R}_b^g \quad \mathbf{t}_b^g \quad \mathbf{v}_b^g, \quad \mathbf{b}_g \quad \mathbf{b}_a \right] \quad (4.2)$$

On the other hand, each historical clone stores the pose of the body at a specific previous time instant.

$$\mathbf{x}_{clone,i} = \left[ \mathbf{R}_{clone,i}^g \quad \mathbf{t}_{clone,i}^g \right] \quad (4.3)$$

Depending on the application, we can initialize a new historical clone for each IMU measurement or we can initialize clones at specific time instants. For example, [36] and [23] initialize a new clone for each arrived image frame. To maintain a bounded computational load, a sliding window approach is employed. In other words, the oldest historical clones are removed from the state as new clones are initialized.

As a final remark, it is important to emphasize that the historical clones and the active IMU state are correlated variables. Consequently, when a new measurement is received for one of the historical clones, the active IMU state is also updated during the measurement update phase of the MSCKF framework. As a result, the presence of historical clones allows the system to incorporate measurements from previous time instants, thereby refining the active IMU state and improving the accuracy of the current state estimation.

### 4.3 Error State Definition

An EKF framework operates under the assumption that the system's state can be modeled as a multivariate Gaussian distribution. However, this assumption breaks down when applied to constrained states, such as rotation matrices, which cannot be accurately represented by a Gaussian distribution. To overcome this technical difficulty, researchers have proposed the error state. In this section, we discuss the error state for the proposed framework.

The active IMU state is represented as  $SE_2(3) \times \mathbb{R}^6$ , where  $SE_2(3)$  is called the extended special Euclidean group [10].  $SE_2(3)$  stores the rotation, position and velocity while  $\mathbb{R}^6$  stores the gyroscope and accelerometer biases. Error state for  $SE_2(3)$  is defined as a perturbation term, which is adopted from [10].

$$\mathcal{T}_b^g = \left[ \begin{array}{c|cc} \mathbf{R}_b^g & \mathbf{t}_b^g & \mathbf{v}_b^g \\ \hline 0_{2 \times 3} & & I_2 \end{array} \right] \quad (4.4)$$

$$= \hat{\mathcal{T}}_b^g \text{Exp}_{SE_2(3)}(\zeta) \quad (4.5)$$

Where  $\mathcal{T}_b^g$  is the actual state,  $\hat{\mathcal{T}}_b^g$  is the estimated state and  $\zeta \in \mathbb{R}^{9 \times 1}$  is the perturbation term.

The error states for the gyroscope and accelerometer biases are defined through conventional subtraction, specifically as  $\tilde{\mathbf{b}}_g \triangleq \mathbf{b}_g - \hat{\mathbf{b}}_g$  and  $\tilde{\mathbf{b}}_a \triangleq \mathbf{b}_a - \hat{\mathbf{b}}_a$ . Similarly, we define the error state for clones.

$$\mathbf{T}_{\text{clone},i}^g = \hat{\mathbf{T}}_{\text{clone},i}^g \text{Exp}_{SE(3)}(\boldsymbol{\xi}_{\text{clone},i}) \quad (4.6)$$

$\boldsymbol{\xi}_{\text{clone},i} \in \mathbb{R}^6$  is the error state for clone  $i$ . The overall error state, represented by  $\tilde{\mathbf{x}}$ , is formed by concatenating individual error states.

$$\tilde{\mathbf{x}} \triangleq \begin{bmatrix} \zeta \\ \tilde{\mathbf{b}}_g \\ \tilde{\mathbf{b}}_a \\ \boldsymbol{\xi}_{\text{clone},1} \\ \vdots \\ \boldsymbol{\xi}_{\text{clone},n} \end{bmatrix} \quad (4.7)$$

#### 4.4 State Transition Model

This section examines the state transition equations derived from the integration of IMU measurements. All the derivations in this section are adopted and borrowed from [10] and [51], which are the extended versions of [4]. Note that we will examine the evolution of the error state rather than the pose of the system. We will see that while the mean of the error remains zero, its covariance evolves during state transitions.

We previously analyzed IMU dynamics in Section 3.3 and presented three key equations as a summary in Subsection 3.3.4. Recall that, within the state, we have defined bias terms  $\mathbf{b}_g$  and  $\mathbf{b}_a$ , which also should be included in state transition.

The overall state transition is examined in two steps.

1. First assume that the IMU is noise-free. Obtain the state-to-state transition.
2. Include IMU noise, examine imu noise-to-state transition.

#### 4.4.1 Error State Dynamics with Noise-Free IMU

Recall that we employ the extended special Euclidean group  $SE(3)$ .

$$\mathcal{T}_{b_t}^g = \left[ \begin{array}{c|c} \mathbf{R}_{b_t}^g & \mathbf{t}_{b_t}^g \quad \mathbf{v}_{b_t}^i \\ \hline 0_{2 \times 3} & I_2 \end{array} \right] \quad (4.8)$$

The notation  $\mathcal{T}_{b_t}^g$  stands for the extended pose of body  $b$  with respect to global frame  $g$  at time instant  $t$ .

First, [4] examined the error state dynamics of  $SE(3)$ . Later, [10] extended this work to include velocity in the analysis. As highlighted by these studies, the IMU dynamics can be expressed in the form of matrix multiplication.

$$\mathcal{T}_{b_t}^g = \Gamma_t \Phi_t(\mathcal{T}_{b_0}^g) \Upsilon_t \quad (4.9)$$

where

$$\Gamma_t \triangleq \left[ \begin{array}{c|c} \mathbf{I}_3 & 0.5\mathbf{g}t^2 \quad \mathbf{g}t \\ \hline 0_{2 \times 3} & I_2 \end{array} \right] \quad \Phi_t(\mathcal{T}_{b_0}^g) \triangleq \left[ \begin{array}{c|c} \mathbf{R}_b^g & \mathbf{t}_b^g + \mathbf{v}_b^g t \quad \mathbf{v}_b^g \\ \hline 0_{2 \times 3} & I_2 \end{array} \right] \quad (4.10)$$

And  $\Upsilon_t$  is the integrated IMU measurements

$$\Upsilon_t \triangleq \left[ \begin{array}{c|c} \Delta_{\mathbf{R}}(t) & \Delta_{\mathbf{t}}(t) \quad \Delta_{\mathbf{v}}(t) \\ \hline 0_{2 \times 3} & I_2 \end{array} \right] \quad (4.11)$$

and we have

$$\Delta_{\mathbf{R}}(t) = \text{Exp}\left(\int_0^t \boldsymbol{\omega}_{ib}^b d\tau\right) \quad \Delta_{\mathbf{v}}(t) = \int_0^t \Delta_{\mathbf{R}}(\tau) \mathbf{f}(\tau) d\tau \quad \Delta_{\mathbf{t}}(t) = \int_0^t \Delta_{\mathbf{v}}(\tau) d\tau \quad (4.12)$$

Discretization of (4.9) leads to

$$\mathcal{T}_{b_{k+1}}^g = \Gamma_k \Phi_k(\mathcal{T}_{b_k}^g) \Upsilon_k \quad (4.13)$$

Now, we are ready to examine the error dynamics. Decompose  $\mathcal{T}_{b_{k+1}}^g$  into state and error state.

$$\mathcal{T}_{b_k}^g = \hat{\mathcal{T}}_{b_k}^g \text{Exp}(\boldsymbol{\zeta}_k) \quad (4.14)$$

Insert (4.14) into (4.13).

$$\hat{\mathcal{T}}_{\mathfrak{b}_{k+1}}^{\mathfrak{g}} \text{Exp}(\zeta_{k+1}) = \Gamma_k \Phi_t \left( \hat{\mathcal{T}}_{\mathfrak{b}_k}^{\mathfrak{g}} \text{Exp}(\zeta_k) \right) \Upsilon_k \quad (4.15)$$

$$\xrightarrow[16]{\text{By Claim}} = \Gamma_k \Phi_t \left( \hat{\mathcal{T}}_{\mathfrak{b}_k}^{\mathfrak{g}} \right) \Phi_t \left( \text{Exp}(\zeta_k) \right) \Upsilon_k \quad (4.16)$$

$$\xrightarrow[15]{\text{By Claim}} = \Gamma_k \Phi_t \left( \hat{\mathcal{T}}_{\mathfrak{b}_k}^{\mathfrak{g}} \right) \text{Exp}(\mathbf{F} \zeta_k) \Upsilon_k \quad (4.17)$$

$$\xrightarrow[3.37]{\text{By Equation}} = \Gamma_k \Phi_t \left( \hat{\mathcal{T}}_{\mathfrak{b}_k}^{\mathfrak{g}} \right) \Upsilon_k \text{Exp}(\text{Ad}_{\Upsilon_k^{-1}} \mathbf{F} \zeta_k) \quad (4.18)$$

where

$$\mathbf{F} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{0 \times 3} & \mathbf{0}_{0 \times 3} \\ \mathbf{0}_{0 \times 3} & \mathbf{I}_3 & \mathbf{I}_3 t \\ \mathbf{0}_{0 \times 3} & \mathbf{0}_{0 \times 3} & \mathbf{I}_3 \end{bmatrix}$$

In summary, we have

$$\hat{\mathcal{T}}_{\mathfrak{b}_{k+1}}^{\mathfrak{g}} \text{Exp}(\zeta_{k+1}) = \Gamma_k \Phi_t \left( \hat{\mathcal{T}}_{\mathfrak{b}_k}^{\mathfrak{g}} \right) \Upsilon_k \text{Exp}(\text{Ad}_{\Upsilon_k^{-1}} \mathbf{F} \zeta_k) \quad (4.19)$$

Moreover, the estimated state also obeys Equation (4.13).

$$\hat{\mathcal{T}}_{\mathfrak{b}_{k+1}}^{\mathfrak{g}} = \Gamma_k \Phi_t \left( \hat{\mathcal{T}}_{\mathfrak{b}_k}^{\mathfrak{g}} \right) \Upsilon_k \quad (4.20)$$

Insert (4.20) into (4.19)

$$\hat{\mathcal{T}}_{\mathfrak{b}_{k+1}}^{\mathfrak{g}} \text{Exp}(\zeta_{k+1}) = \hat{\mathcal{T}}_{\mathfrak{b}_{k+1}}^{\mathfrak{g}} \text{Exp}(\text{Ad}_{\Upsilon_k^{-1}} \mathbf{F} \zeta_k) \quad (4.21)$$

$$\iff \text{Exp}(\zeta_{k+1}) = \text{Exp}(\text{Ad}_{\Upsilon_k^{-1}} \mathbf{F} \zeta_k) \quad (4.22)$$

Unfortunately, the map  $\text{Exp}(\cdot)$  is not *one-to-one*. Hence, we cannot directly claim that

$$\text{Exp}(\zeta_{k+1}) = \text{Exp}(\text{Ad}_{\Upsilon_k^{-1}} \mathbf{F} \zeta_k) \iff \zeta_{k+1} = \text{Ad}_{\Upsilon_k^{-1}} \mathbf{F} \zeta_k$$

However, for errors, where the magnitude of  $\zeta_{\psi}$  is smaller than  $\pi$ , we can safely assume that

$$\zeta_{k+1} = \text{Ad}_{\Upsilon_k^{-1}} \mathbf{F} \zeta_k \quad (4.23)$$

**It turns out that, even though the original state dynamics are highly nonlinear, the error state evolves through a linear system given in (4.23).**



#### 4.4.2 Error State Dynamics with Noisy IMU

In the previous part, we analyzed the state transition assuming a noise-free IMU. Now, we extend this analysis to include IMU noise. Recall the equation (4.17).

$$\hat{\mathcal{T}}_{b_{k+1}}^g \text{Exp}(\zeta_{k+1}) = \Gamma_k \Phi_t(\hat{\mathcal{T}}_{b_k}^g) \text{Exp}(\mathbf{F} \zeta_k) \Upsilon_k$$

Here, the IMU measurements are represented by  $\Upsilon_k$  where noise-related errors are inherently included. To account for this, we decompose  $\Upsilon_k$  into two components:

$$\Upsilon_k = \hat{\Upsilon}_k \tilde{\Upsilon}_k \quad (4.24)$$

where  $\hat{\Upsilon}_k$  denotes the measurement, and  $\tilde{\Upsilon}_k$  is the error term. In this section, we will derive  $\tilde{\Upsilon}_k$ .

#### Derivation of $\tilde{\Upsilon}$

Let us recall the definition of  $\tilde{\Upsilon}$ .

$$\Upsilon_k \triangleq \left[ \begin{array}{c|cc} \Delta_{\mathbf{R}}[k] & \Delta_{\mathbf{t}}[k] & \Delta_{\mathbf{v}}[k] \\ \hline 0_{2 \times 3} & & I_2 \end{array} \right]$$

with

$$\Delta_{\mathbf{R}}[k] = \text{Exp}\left(\int_{t_k}^{t_{k+1}} \boldsymbol{\omega}_{\text{ib}}^b dt\right) \quad \Delta_{\mathbf{v}}[k] = \int_{t_k}^{t_{k+1}} \Delta_{\mathbf{R}}(t) \mathbf{f}(t) dt \quad \Delta_{\mathbf{t}}[k] = \int_{t_k}^{t_{k+1}} \Delta_{\mathbf{v}}(t) dt$$

Here we will represent the arrived gyroscope and accelerometer measurements by  $\boldsymbol{\omega}_m$  and  $\mathbf{f}_m$ , respectively. Recall that we have estimates of the bias terms. Then, the estimated angular velocity and linear specific force are

$$\hat{\boldsymbol{\omega}} \triangleq \boldsymbol{\omega} - \hat{\mathbf{b}}_g \quad (4.25)$$

$$\hat{\mathbf{f}} \triangleq \mathbf{f}_m - \hat{\mathbf{b}}_a \quad (4.26)$$

In such a case, the bias error states,  $\tilde{\mathbf{b}}_g$  and  $\tilde{\mathbf{b}}_a$ , as well as instantaneous IMU noises,  $\boldsymbol{\eta}_g$  and  $\boldsymbol{\eta}_a$ , appears as the source of the error. Let us examine the noise within each term.

## 1. IMU Noise in Orientation Estimation

$$\Delta_{\mathbf{R}} = \text{Exp}\left((\hat{\boldsymbol{\omega}} - \tilde{\mathbf{b}}_g - \boldsymbol{\eta}_g)\Delta t\right) \quad (4.27)$$

$$\approx \underbrace{\text{Exp}\left(\hat{\boldsymbol{\omega}}\Delta t\right)}_{=\hat{\Delta}_{\mathbf{R}}} \underbrace{\text{Exp}\left(-\mathcal{J}_r(\hat{\boldsymbol{\omega}}\Delta t)(\tilde{\mathbf{b}}_g + \boldsymbol{\eta}_g)\Delta t\right)}_{\hat{\Delta}_{\mathbf{R}}} \quad (4.28)$$

## 2. IMU Noise in Velocity Estimation

$$\Delta_{\mathbf{v}} = \int_0^{\Delta t} \Delta_{\mathbf{R}}(\tau) d\tau (\hat{\mathbf{f}} - \tilde{\mathbf{b}}_a - \boldsymbol{\eta}_a) \quad (4.29)$$

$$= \int_0^{\Delta t} \text{Exp}\left((\hat{\boldsymbol{\omega}} - \tilde{\mathbf{b}}_g - \boldsymbol{\eta}_g)\tau\right) d\tau (\hat{\mathbf{f}} - \tilde{\mathbf{b}}_a - \boldsymbol{\eta}_a) \quad (4.30)$$

$$\approx \int_0^{\Delta t} \text{Exp}\left(\hat{\boldsymbol{\omega}}\tau\right) \text{Exp}\left(-\mathcal{J}_r(\hat{\boldsymbol{\omega}}\tau)(\tilde{\mathbf{b}}_g + \boldsymbol{\eta}_g)\tau\right) d\tau (\hat{\mathbf{f}} - \tilde{\mathbf{b}}_a - \boldsymbol{\eta}_a) \quad (4.31)$$

$$\approx \int_0^{\Delta t} \text{Exp}\left(\hat{\boldsymbol{\omega}}\tau\right) \left(\mathcal{I}_3 - [\mathcal{J}_r(\hat{\boldsymbol{\omega}}\tau)(\tilde{\mathbf{b}}_g + \boldsymbol{\eta}_g)\tau]\right) d\tau (\hat{\mathbf{f}} - \tilde{\mathbf{b}}_a - \boldsymbol{\eta}_a) \quad (4.32)$$

Ignore second order term  $(\tilde{\mathbf{b}}_g + \boldsymbol{\eta}_g)(\tilde{\mathbf{b}}_a + \boldsymbol{\eta}_a)$ . Also use the property  $[\mathbf{a}]\mathbf{b} = -[\mathbf{b}]\mathbf{a}$

$$\Delta_{\mathbf{v}} \approx \underbrace{\int_0^{\Delta t} \text{Exp}\left(\hat{\boldsymbol{\omega}}\tau\right) d\tau}_{=\hat{\Delta}_{\mathbf{v}}} \hat{\mathbf{f}} + \underbrace{\int_0^{\Delta t} \text{Exp}\left(\hat{\boldsymbol{\omega}}\tau\right) [\hat{\mathbf{f}}] \mathcal{J}_r(\hat{\boldsymbol{\omega}}\tau)\tau d\tau}_{\hat{\Delta}_{\mathbf{v}\mathbf{g}}} (\tilde{\mathbf{b}}_g + \boldsymbol{\eta}_g) \quad (4.33)$$

$$- \underbrace{\int_0^{\Delta t} \text{Exp}\left(\hat{\boldsymbol{\omega}}\tau\right) d\tau}_{\hat{\Delta}_{\mathbf{v}\mathbf{a}}} (\tilde{\mathbf{b}}_a + \boldsymbol{\eta}_a) \quad (4.34)$$

Then

$$\tilde{\Delta}_{\mathbf{v}} = \hat{\Delta}_{\mathbf{v}\mathbf{g}}(\tilde{\mathbf{b}}_g + \boldsymbol{\eta}_g) + \hat{\Delta}_{\mathbf{v}\mathbf{a}}(\tilde{\mathbf{b}}_a + \boldsymbol{\eta}_a) \quad (4.35)$$

### 3. IMU Noise in Position Estimation

$$\Delta_{\mathbf{t}} = \int_0^{\Delta t} \Delta_{\mathbf{v}}(\tau) \, d\tau \quad (4.36)$$

$$= \int_0^{\Delta t} \left( \int_0^{\tau} \Delta_{\mathbf{R}}(\delta) \, d\delta (\hat{\mathbf{f}} - \tilde{\mathbf{b}}_a - \boldsymbol{\eta}_a) \right) d\tau \quad (4.37)$$

$$= \int_0^{\Delta t} \left( \int_0^{\tau} \Delta_{\mathbf{R}}(\delta) \, d\delta \right) d\tau (\hat{\mathbf{f}} - \tilde{\mathbf{b}}_a - \boldsymbol{\eta}_a) \quad (4.38)$$

$$= \left( \int_0^{\Delta t} \int_0^{\tau} \Delta_{\mathbf{R}}(\delta) \, d\delta \, d\tau \right) (\hat{\mathbf{f}} - \tilde{\mathbf{b}}_a - \boldsymbol{\eta}_a) \quad (4.39)$$

Let us examine the term  $\int_0^{\Delta t} \int_0^{\tau} \Delta_{\mathbf{R}}(\delta) \, d\delta \, d\tau$ .

$$\begin{aligned} \int_0^{\Delta t} \int_0^{\tau} \Delta_{\mathbf{R}}(\delta) \, d\delta \, d\tau &\approx \\ &\approx \int_0^{\Delta t} \int_0^{\tau} \text{Exp}(\hat{\boldsymbol{\omega}}\delta) \text{Exp}\left(-\mathcal{J}_r(\hat{\boldsymbol{\omega}}\delta)(\tilde{\mathbf{b}}_g + \boldsymbol{\eta}_g)\delta\right) \, d\delta \, d\tau \end{aligned} \quad (4.40)$$

$$\approx \int_0^{\Delta t} \int_0^{\tau} \text{Exp}(\hat{\boldsymbol{\omega}}\delta) \left( \mathbf{I}_3 - \lfloor \mathcal{J}_r(\hat{\boldsymbol{\omega}}\delta)(\tilde{\mathbf{b}}_g + \boldsymbol{\eta}_g)\delta \rfloor \right) \, d\delta \, d\tau \quad (4.41)$$

$$\begin{aligned} &\approx \int_0^{\Delta t} \int_0^{\tau} \text{Exp}(\hat{\boldsymbol{\omega}}\delta) \, d\delta \, d\tau - \\ &\quad - \int_0^{\Delta t} \int_0^{\tau} \text{Exp}(\hat{\boldsymbol{\omega}}\delta) \lfloor \mathcal{J}_r(\hat{\boldsymbol{\omega}}\delta)(\tilde{\mathbf{b}}_g + \boldsymbol{\eta}_g)\delta \rfloor \, d\delta \, d\tau \end{aligned} \quad (4.42)$$

Insert (4.42) into (4.39) and ignore the second order term  $(\tilde{\mathbf{b}}_a + \boldsymbol{\eta}_a)(\tilde{\mathbf{b}}_g + \boldsymbol{\eta}_g)$ .

Then, we obtain

$$\Delta_{\mathbf{t}} \approx \underbrace{\int_0^{\Delta t} \int_0^{\tau} \text{Exp}(\hat{\boldsymbol{\omega}}\delta) \, d\delta \, d\tau}_{\triangleq \hat{\Delta}_{\mathbf{t}}} \hat{\mathbf{f}} + \underbrace{\tilde{\mathcal{J}}_{\mathbf{t}g}(\tilde{\mathbf{b}}_g + \boldsymbol{\eta}_g) + \tilde{\mathcal{J}}_{\mathbf{t}a}(\tilde{\mathbf{b}}_a + \boldsymbol{\eta}_a)}_{\triangleq \hat{\Delta}_{\mathbf{t}}} \quad (4.43)$$

where

$$\tilde{\mathcal{J}}_{\mathbf{t}g} = \int_0^{\Delta t} \int_0^{\tau} \left( \text{Exp}(\hat{\boldsymbol{\omega}}\delta) \lfloor \hat{\mathbf{f}} \rfloor \mathcal{J}_r(\hat{\boldsymbol{\omega}}\delta)\delta \right) \, d\delta \, d\tau \quad (4.44)$$

$$\tilde{\mathcal{J}}_{\mathbf{t}a} = - \int_0^{\Delta t} \int_0^{\tau} \text{Exp}(\hat{\boldsymbol{\omega}}\delta) \, d\delta \, d\tau \quad (4.45)$$

Note that the closed form expressions (analytical solutions) for  $\tilde{\mathcal{J}}_{\text{vg}}$ ,  $\tilde{\mathcal{J}}_{\text{va}}$ ,  $\tilde{\mathcal{J}}_{\text{tg}}$ ,  $\tilde{\mathcal{J}}_{\text{ta}}$  are available at [51]. Now, put everything in matrix form.

$$\mathbf{\Upsilon} \triangleq \left[ \begin{array}{c|cc} \Delta_{\mathbf{R}} & \Delta_{\mathbf{t}} & \Delta_{\mathbf{v}} \\ \hline \mathbf{0}_{2 \times 3} & \mathbf{I}_2 & \end{array} \right] = \left[ \begin{array}{c|cc} \hat{\Delta}_{\mathbf{R}} \tilde{\Delta}_{\mathbf{R}} & \hat{\Delta}_{\mathbf{t}} + \tilde{\Delta}_{\mathbf{t}} & \hat{\Delta}_{\mathbf{v}} + \tilde{\Delta}_{\mathbf{v}} \\ \hline \mathbf{0}_{2 \times 3} & \mathbf{I}_2 & \end{array} \right] \quad (4.46)$$

$$= \underbrace{\left[ \begin{array}{c|cc} \hat{\Delta}_{\mathbf{R}} & \hat{\Delta}_{\mathbf{t}} & \hat{\Delta}_{\mathbf{v}} \\ \hline \mathbf{0}_{2 \times 3} & \mathbf{I}_2 & \end{array} \right]}_{=\tilde{\Upsilon}} \underbrace{\left[ \begin{array}{c|cc} \tilde{\Delta}_{\mathbf{R}} & \hat{\Delta}_{\mathbf{R}}^T \tilde{\Delta}_{\mathbf{t}} & \hat{\Delta}_{\mathbf{R}}^T \tilde{\Delta}_{\mathbf{v}} \\ \hline \mathbf{0}_{2 \times 3} & \mathbf{I}_2 & \end{array} \right]}_{=\tilde{\Upsilon}} \quad (4.47)$$

Even though we have managed to obtain an expression for the term  $\tilde{\Upsilon}$ , it remains necessary to express  $\tilde{\Upsilon}$  in the Lie algebra of the corresponding manifold,  $\mathfrak{so}_2(3)$ . This ensures that both the IMU noise representation and the state noise representation are consistently expressed in the same space,  $\mathfrak{so}_2(3)$ .

In other words, we express  $\tilde{\Upsilon}$  as a matrix exponential.

$$\tilde{\Upsilon} = \left[ \begin{array}{c|cc} \tilde{\Delta}_{\mathbf{R}} & \hat{\Delta}_{\mathbf{R}}^T \tilde{\Delta}_{\mathbf{t}} & \hat{\Delta}_{\mathbf{R}}^T \tilde{\Delta}_{\mathbf{v}} \\ \hline \mathbf{0}_{2 \times 3} & \mathbf{I}_2 & \end{array} \right] \quad (4.48)$$

$$= \text{Exp} \left( \left[ \begin{array}{c} \text{Log}(\tilde{\Delta}_{\mathbf{R}}) \\ \mathcal{J}_l^{-1}(\text{Log}(\tilde{\Delta}_{\mathbf{R}})) \hat{\Delta}_{\mathbf{R}}^T \tilde{\Delta}_{\mathbf{t}} \\ \mathcal{J}_l^{-1}(\text{Log}(\tilde{\Delta}_{\mathbf{R}})) \hat{\Delta}_{\mathbf{R}}^T \tilde{\Delta}_{\mathbf{v}} \end{array} \right] \right) \quad (4.49)$$

$$\approx \text{Exp} \left( \left[ \begin{array}{c} \text{Log}(\tilde{\Delta}_{\mathbf{R}}) \\ \hat{\Delta}_{\mathbf{R}}^T \tilde{\Delta}_{\mathbf{t}} \\ \hat{\Delta}_{\mathbf{R}}^T \tilde{\Delta}_{\mathbf{v}} \end{array} \right] \right) \quad (4.50)$$

$$= \text{Exp} \left( \left[ \begin{array}{cc|c} -\mathcal{J}_r(\hat{\omega} \Delta t)(\tilde{\mathbf{b}}_g + \boldsymbol{\eta}_g) \Delta t & \mathbf{0}_{3 \times 6} & \\ & \hat{\Delta}_{\mathbf{R}}^T \tilde{\mathcal{J}}_{\text{tg}} & \hat{\Delta}_{\mathbf{R}}^T \tilde{\mathcal{J}}_{\text{ta}} \\ & \hat{\Delta}_{\mathbf{R}}^T \tilde{\mathcal{J}}_{\text{vg}} & \hat{\Delta}_{\mathbf{R}}^T \tilde{\mathcal{J}}_{\text{va}} \end{array} \right] \begin{bmatrix} \tilde{\mathbf{b}}_g + \boldsymbol{\eta}_g \\ \tilde{\mathbf{b}}_a + \boldsymbol{\eta}_a \end{bmatrix} \right) \quad (4.51)$$

$$\triangleq \text{Exp}(\tilde{\mathbf{v}}_k) \quad (4.52)$$

### 4.4.3 The Full State Transition Matrix

In this section, we will put everything into matrix form. The objective is to linearize state transition matrix. Recall the equation (4.17).

$$\hat{\mathcal{T}}_{b_{k+1}}^g \text{Exp}(\zeta_{k+1}) = \Gamma_k \Phi_t \left( \hat{\mathcal{T}}_{b_k}^g \right) \text{Exp}(\mathbf{F} \zeta_k) \Upsilon_k$$

We further decomposed the term  $\Upsilon_k$  into measurement  $\hat{\Upsilon}_k$  and error term  $\text{Exp}(\tilde{v}_k)$ . As a result, we have,

$$\hat{\mathcal{T}}_{b_{k+1}}^g \text{Exp}(\zeta_{k+1}) = \Gamma_k \Phi_t \left( \hat{\mathcal{T}}_{b_k}^g \right) \text{Exp}(\mathbf{F} \zeta_k) \hat{\Upsilon}_k \text{Exp}(\tilde{v}_k) \quad (4.53)$$

$$\xrightarrow[3.37]{\text{By Equation}} = \underbrace{\Gamma_k \Phi_t \left( \hat{\mathcal{T}}_{b_k}^g \right) \hat{\Upsilon}_k}_{\triangleq \hat{\mathcal{T}}_{b_{k+1}}^g} \text{Exp}(\text{Ad}_{\hat{\Upsilon}_k^{-1}} \mathbf{F} \zeta_k) \text{Exp}(\tilde{v}_k) \quad (4.54)$$

It is obvious that

$$\text{Exp}(\zeta_{k+1}) = \text{Exp}(\text{Ad}_{\hat{\Upsilon}_k^{-1}} \mathbf{F} \zeta_k) \text{Exp}(\tilde{v}_k) \quad (4.55)$$

We **cannot** directly deduce that  $\zeta_{k+1} = \text{Ad}_{\hat{\Upsilon}_k^{-1}} \mathbf{F} \zeta_k + \tilde{v}_k$ . However, the linearization around the estimated error state  $\zeta_k = \mathbf{0}$ ,  $\tilde{b}_g = \mathbf{0}$ ,  $\tilde{b}_a = \mathbf{0}$ , leads to

$$\zeta_{k+1} \approx \text{Ad}_{\hat{\Upsilon}_k^{-1}} \mathbf{F} \zeta_k + \tilde{v}_k \quad (4.56)$$

Put everything into matrix form,

$$\begin{bmatrix} \zeta_{k+1} \\ \tilde{b}_g[k+1] \\ \tilde{b}_a[k+1] \end{bmatrix} = \Phi_k \begin{bmatrix} \zeta_k \\ \tilde{b}_g[k] \\ \tilde{b}_a[k] \end{bmatrix} + \mathbf{G}_k \begin{bmatrix} \eta_g \\ \eta_a \\ \eta_{bg} \\ \eta_{ba} \end{bmatrix} \quad (4.57)$$

where  $\eta_{bg}$  and  $\eta_{ba}$  are the random walk parameters for the bias terms. The random walk in bias is modeled as

$$\tilde{b}_g[k+1] = \tilde{b}_g[k] + \eta_{bg} \Delta t \quad (4.58)$$

$$\tilde{b}_a[k+1] = \tilde{b}_a[k] + \eta_{ba} \Delta t \quad (4.59)$$

Finally, we obtain

$$\Phi_k = \left[ \begin{array}{c|cc} & -\mathcal{J}_r(\hat{\omega}\Delta t)\Delta t & \mathbf{0}_{3\times 3} \\ \mathbf{Ad}_{\hat{\mathbf{Y}}_k^{-1}}\mathbf{F} & \hat{\Delta}_{\mathbf{R}}^T\tilde{\mathcal{J}}_{\mathbf{t}g} & \hat{\Delta}_{\mathbf{R}}^T\tilde{\mathcal{J}}_{\mathbf{t}a} \\ & \hat{\Delta}_{\mathbf{R}}^T\tilde{\mathcal{J}}_{\mathbf{v}g} & \hat{\Delta}_{\mathbf{R}}^T\tilde{\mathcal{J}}_{\mathbf{v}a} \\ \hline \mathbf{0}_{6\times 9} & & \mathbf{I}_6 \end{array} \right]$$

$$\mathbf{G}_k = \left[ \begin{array}{cc|c} -\mathcal{J}_r(\hat{\omega}\Delta t)\Delta t & \mathbf{0}_{3\times 3} & \\ \hat{\Delta}_{\mathbf{R}}^T\tilde{\mathcal{J}}_{\mathbf{t}g} & \hat{\Delta}_{\mathbf{R}}^T\tilde{\mathcal{J}}_{\mathbf{t}a} & \mathbf{0}_{9\times 6} \\ \hat{\Delta}_{\mathbf{R}}^T\tilde{\mathcal{J}}_{\mathbf{v}g} & \hat{\Delta}_{\mathbf{R}}^T\tilde{\mathcal{J}}_{\mathbf{v}a} & \\ \hline & \mathbf{0}_{6\times 6} & \mathbf{I}_6 \Delta t \end{array} \right]$$

#### 4.4.4 State Augmentation

In the previous section, we have explored how to propagate the noise given the IMU measurements. Now, we will examine how to initialize a new historical clone. Assume that at a specific time instant  $t_k$ , we desire to store the current state as a clone. At the given time stamp, we have the state

$$\left[ \mathbf{x}_{IMU} \quad \mathbf{x}_{\text{clone},1} \quad \mathbf{x}_{\text{clone},2} \quad \cdots \quad \mathbf{x}_{\text{clone},N} \right]$$

The error state is represented by

$$\tilde{\mathbf{x}} = \left[ \tilde{\mathbf{x}}_{IMU}^T \quad \boldsymbol{\xi}_{\text{clone},1}^T \quad \boldsymbol{\xi}_{\text{clone},2}^T \quad \cdots \quad \boldsymbol{\xi}_{\text{clone},N}^T \right]^T \in \mathbb{R}^{15+6N}$$

with the covariance  $\Sigma \in \mathbb{R}^{(15+6N)\times(15+6N)}$ . Creating a new clone is equivalent to a simple matrix multiplication. Only the first 6 items of the error is stored in the new clone.

$$\tilde{\mathbf{x}}_{\text{augmented}} = \begin{bmatrix} \tilde{\mathbf{x}} \\ \boldsymbol{\xi}_{\text{clone},N+1} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{I}_{15+6N} \\ \mathbf{I}_6 \quad \mathbf{0}_{9+6N} \end{bmatrix}}_{\hat{\mathbf{c}}} \tilde{\mathbf{x}} \quad (4.60)$$

We also need to compute the covariance of the augmented state,  $\Sigma_{\text{augmented}}$ .

$$\Sigma_{\text{augmented}} = \hat{\mathbf{c}} \Sigma \hat{\mathbf{c}}^T \in \mathbb{R}^{(15+6(N+1))\times(15+6(N+1))} \quad (4.61)$$

## 4.5 Measurement Model

The base algorithm, *VOLDOR* [35], provides estimates of the relative rigid transformation between two successive frames, referred to as the incremental pose and denoted as  $\mathbf{Z}_{c_2}^{c_1}$ . In other words,

$$\mathbf{T}_{c_2}^g = \mathbf{T}_{c_1}^g \mathbf{Z}_{c_2}^{c_1} \quad (4.62)$$

where  $\mathbf{T}_{c,n}^g$  is the camera pose at time instant  $n$ .

For simplicity, I assume that the incremental poses between camera frames are readily available. Readers unfamiliar with *VOLDOR* [35] may assume that this incremental pose information comes from an external source. The objective is to fuse this information into the IMU-propagated state.

It is important to note that the incremental pose is defined between the camera frames, while the estimation is performed for the IMU frame. The extrinsic calibration between the IMU and the camera,  $\mathbf{T}_b^c$ , is assumed to be known. The measurement is visualized in Figure 4.1.

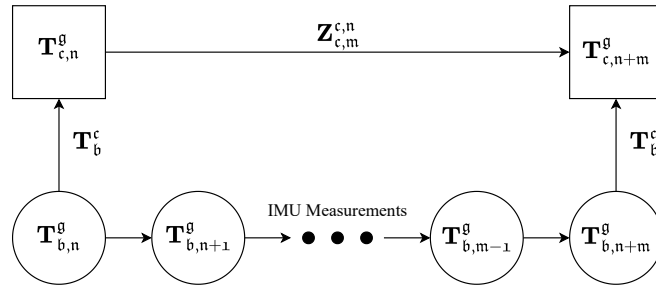


Figure 4.1: Measurement Visualized

The frequency of IMU is higher than the camera. Measurements arrive only between camera frames. Hence, historical clones of the active state are stored only for the instants when a new image frame arrives.

To update the state within the Kalman framework, it is first necessary to express the measurement in terms of the estimated state parameters. Here, I remind you that the estimated state is  $\tilde{\mathbf{x}}$ , which is declared in (4.7).

The measurement only depends on the error states of the corresponding clones,  $\xi_1$  and  $\xi_2$ .

$$\mathbf{Z}_{c_2}^{c_1} \triangleq \mathbf{T}_{c_2}^{c_1} \text{Exp}(\boldsymbol{\eta}_z) \quad (4.63)$$

$$= \mathbf{T}_g^{c_1} \mathbf{T}_{c_2}^g \text{Exp}(\boldsymbol{\eta}_z) \quad (4.64)$$

$$= (\mathbf{T}_{c_1}^g)^{-1} \mathbf{T}_{c_2}^g \text{Exp}(\boldsymbol{\eta}_z) \quad (4.65)$$

$$= (\mathbf{T}_{b_1}^g \mathbf{T}_c^b)^{-1} \mathbf{T}_{b_2}^g \mathbf{T}_c^b \text{Exp}(\boldsymbol{\eta}_z) \quad (4.66)$$

$$= \mathbf{T}_b^c (\mathbf{T}_{b_1}^g)^{-1} \mathbf{T}_{b_2}^g \mathbf{T}_c^b \text{Exp}(\boldsymbol{\eta}_z) \quad (4.67)$$

$$= \mathbf{T}_b^c \left( \hat{\mathbf{T}}_{b_1}^g \text{Exp}(\xi_1) \right)^{-1} \hat{\mathbf{T}}_{b_2}^g \text{Exp}(\xi_2) \mathbf{T}_c^b \text{Exp}(\boldsymbol{\eta}_z) \quad (4.68)$$

$$= \mathbf{T}_b^c \left( \text{Exp}(\xi_1) \right)^{-1} \left( \hat{\mathbf{T}}_{b_1}^g \right)^{-1} \hat{\mathbf{T}}_{b_2}^g \text{Exp}(\xi_2) \mathbf{T}_c^b \text{Exp}(\boldsymbol{\eta}_z) \quad (4.69)$$

$$= \mathbf{T}_b^c \left( \text{Exp}(\xi_1) \right)^{-1} \hat{\mathbf{T}}_g^{b_1} \hat{\mathbf{T}}_{b_2}^g \text{Exp}(\xi_2) \mathbf{T}_c^b \text{Exp}(\boldsymbol{\eta}_z) \quad (4.70)$$

$$= \mathbf{T}_b^c \text{Exp}(\xi_1^{-1}) \hat{\mathbf{T}}_{b_2}^{b_1} \text{Exp}(\xi_2) \mathbf{T}_c^b \text{Exp}(\boldsymbol{\eta}_z) \quad (4.71)$$

$$\xrightarrow{\text{By 3.37}} = \mathbf{T}_b^c \hat{\mathbf{T}}_{b_2}^{b_1} \text{Exp}(\text{Ad}_{\hat{\mathbf{T}}_{b_1}^{b_2}} \xi_1^{-1}) \text{Exp}(\xi_2) \text{Exp}(\text{Ad}_{\mathbf{T}_c^b} \boldsymbol{\eta}_z) \mathbf{T}_c^b \quad (4.72)$$

In (4.71), with an abuse of notation, I declare  $\xi_1^{-1}$  such that

$$\text{Exp}(\xi_1^{-1}) \triangleq \left( \text{Exp}(\xi_1) \right)^{-1} \quad (4.73)$$

Note that the measurement  $\mathbf{Z}_{c_2}^{c_1}$  does not belong to any vector space. Hence, the measurement is mapped to the Lie Algebra of  $SE(3)$  through logarithmic map.

$$\mathbf{z} \triangleq \text{Log} \left( \hat{\mathbf{T}}_{b_1}^{b_2} \mathbf{T}_c^b \mathbf{Z}_{c_2}^{c_1} \mathbf{T}_b^c \right) \quad (4.74)$$

$$= \text{Log} \left( \text{Exp}(\text{Ad}_{\hat{\mathbf{T}}_{b_1}^{b_2}} \xi_1^{-1}) \text{Exp}(\xi_2) \text{Exp}(\text{Ad}_{\mathbf{T}_c^b} \boldsymbol{\eta}_z) \right) \quad (4.75)$$

where the measurement model is simplified by eliminating the terms  $\hat{\mathbf{T}}_{b_1}^{b_2}$  and  $\mathbf{T}_c^b$ . In order to implement an EKF, we need the differentiation of the measurement with respect to error state. It is trivial to show that

$$\left. \frac{\partial \mathbf{z}}{\partial \xi_1} \right|_{\xi_1=0, \xi_2=0, \boldsymbol{\eta}_z=0} = -\text{Ad}_{\hat{\mathbf{T}}_{b_1}^{b_2}} \quad (4.76)$$

$$\left. \frac{\partial \mathbf{z}}{\partial \xi_2} \right|_{\xi_1=0, \xi_2=0, \boldsymbol{\eta}_z=0} = \mathbf{I}_6 \quad (4.77)$$

$$\left. \frac{\partial \mathbf{z}}{\partial \boldsymbol{\eta}_z} \right|_{\xi_1=0, \xi_2=0, \boldsymbol{\eta}_z=0} = \text{Ad}_{\mathbf{T}_c^b} \quad (4.78)$$



## CHAPTER 5

### EXPERIMENTS

#### 5.1 Dataset Description

The *KITTI* Vision Benchmark Suite [21, 22] is a publicly available dataset collected especially for stereo, optical flow, visual odometry, 3D object detection and 3D tracking. An accurate ground-truth is provided through a velodyne laser scanner and a GPS localization system. I summarize the sensor setup in Table 5.1.

Table 5.1: Available Sensors

Sensor	Model	Resolution	FPS
2 Color Cameras	Point Grey Flea 2 (FL2-14S3M-C)	$1382 \times 512$	10 FPS
2 Grayscale Cameras	Point Grey Flea 2 (FL2-14S3C-C)	$1382 \times 512$	10 FPS
GPS/IMU	OXTS RT 3003	—	100 FPS
Laser Scanner	Velodyne HDL-64E	$\approx 100k$ per scan	10 FPS

During the tests, I only utilize the stereo color camera frames and IMU measurements. Both the image frames and IMU measurements are timestamped with synchronized clocks. GPS is employed as the ground-truth.

Extrinsic calibrations between sensors are available. Moreover, the radial distortion in the camera frames are corrected and the rectified frames are shared along with their corresponding calibration parameters. Through the experiments, I only utilize the rectified image frames with raw GPS/IMU data.

As a final remark, calibration is essential for achieving precise odometry. *KITTI* provides a well calibrated sensor setup.

### 5.1.1 Test Sequences

The *KITTI* dataset includes sequences of varying durations. To emphasize the effectiveness of the proposed fusion, the longest sequences are specifically selected, and the proposed framework is evaluated on these sequences. A list of sequences with sequence durations are provided in Table 5.2.

Table 5.2: Evaluated Data Sequences (Ordered with Sequence Duration)

Data Sequence	Official Sequence Name	Sequence Duration
<i>Seq1</i>	2011_09_30_drive_0028	530 secs
<i>Seq2</i>	2011_10_03_drive_0034	310 secs
<i>Seq3</i>	2011_09_30_drive_0033	164 secs
<i>Seq4</i>	2011_09_30_drive_0018	150 secs

The trajectory of **Sequence 1** is shown in Figure 5.1. It can be observed that the ground truth is corrupted in the *down* channel at  $t = 100sec$ . Points where the GPS data is definitively corrupted will be excluded during the evaluation.

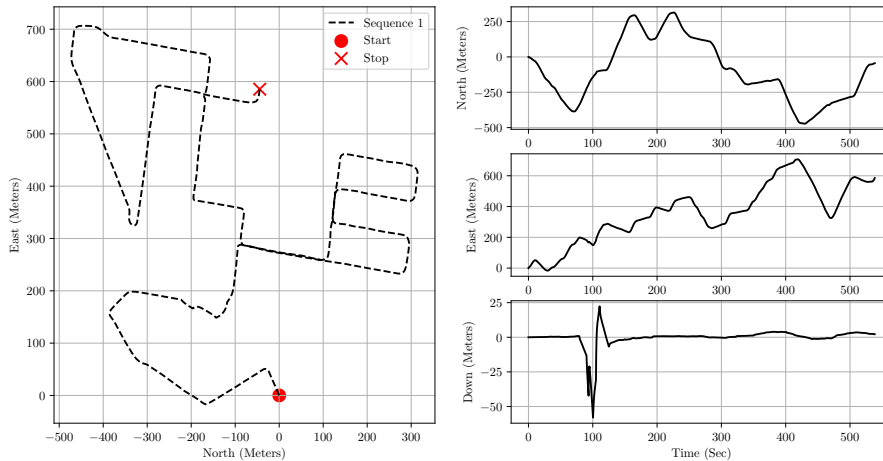


Figure 5.1: Trajectory of Sequence 1

### 5.1.2 Dataset Visualization

In this section, to visualize the dataset, I present a sample stereo image pair from the KITTI dataset in Figure 5.2. Moreover, the disparity between the frames are illustrated for the convenience of the reader.

In Figure 5.2c, I put the stereo image pair on top of each other using weighted sum of the images. Then, the optical flow vectors (disparity vectors) are illustrated.



(a) Left Camera From Sequence 1



(b) Right Camera From Sequence 1



(c) Disparity Vectors Between Stereo Frames

Figure 5.2: A Sample Stereo Visual Pair From Sequence 1

## 5.2 Performance Metrics

For  $M$  Monte Carlo runs, each with  $N$  samples (time instants), the performance metrics are defined as follows:

- **Orientation:** Orientation error is converted into an angle-axis representation, providing an intuitive measure of rotational discrepancy.

$$\mathbf{R}_{err} \triangleq \hat{\mathbf{R}}_g^b \mathbf{R}_b^g \quad (5.1)$$

$$\mathbf{s}_{err} = \text{Log}_{SO(3)}(\mathbf{R}_{err}) \quad (5.2)$$

where  $\hat{\mathbf{R}}_g^b$  is the estimated state, and  $\mathbf{R}_b^g$  is the ground-truth.  $\mathbf{s}_{err}$  represents the angle-axis form of the rotation error, where its direction indicates the rotation axis, and its magnitude corresponds to the rotation angle. The root mean square error (RMSE) for orientation is defined as follows.

$$RMSE_{ori} = \sqrt{\frac{1}{M \cdot N} \sum_{m=1}^M \sum_{n=1}^N \|\mathbf{s}_{err}\|^2} \quad (5.3)$$

- **Velocity and Position:** For velocity and position, errors are assessed using the conventional RMSE metric.

$$RMSE_{vel} = \sqrt{\frac{1}{M \cdot N} \sum_{m=1}^M \sum_{n=1}^N \|\mathbf{v}_b^g - \hat{\mathbf{v}}_b^g\|^2} \quad (5.4)$$

$$RMSE_{pos} = \sqrt{\frac{1}{M \cdot N} \sum_{m=1}^M \sum_{n=1}^N \|\mathbf{t}_b^g - \hat{\mathbf{t}}_b^g\|^2} \quad (5.5)$$

Table 5.3: Numerical Comparisons

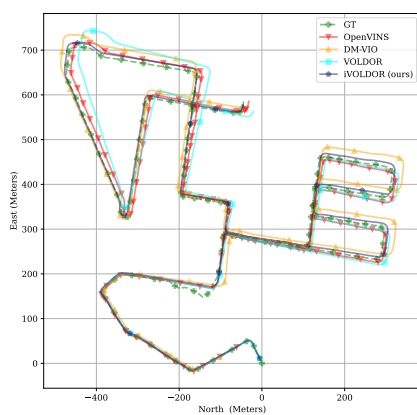
		<b>Open VINS</b>	<b>DM-VIO</b>	<b>VOLDOR (Base Method)</b>	<b>iVOLDOR (ours)</b>
<b>Sensor Setup</b>		Stereo Cam + IMU	Mono Cam + IMU	Stereo Cam	Stereo Cam + IMU
<b>Initialization</b>		Initialized with GT	Automatically Initialized	Not Required	Initialized with GT
<b>Alignment with GT prior to evaluation</b>		Not Required	Umeyma’s Method [52]	Analytically Computed	Not Required
Seq1	$RMSE_{ori}$	0.02792	<b>0.0151</b>	0.0658	0.0160
	$RMSE_{vel}$	0.3427	–	–	<b>0.2941</b>
	$RMSE_{pos}$	10.55	20.76	26.57	<b>8.70</b>
Seq2	$RMSE_{ori}$	<b>0.0145</b>	0.0152	0.0443	0.0162
	$RMSE_{vel}$	<b>0.6822</b>	–	–	0.7315
	$RMSE_{pos}$	<b>7.17</b>	23.46	16.35	10.00
Seq3	$RMSE_{ori}$	0.01542	0.0138	0.0230	<b>0.0125</b>
	$RMSE_{vel}$	<b>0.8208</b>	–	–	0.8324
	$RMSE_{pos}$	7.90	16.22	13.01	<b>7.4736</b>
Seq4	$RMSE_{ori}$	0.0121	<b>0.0080</b>	0.0206	0.0128
	$RMSE_{vel}$	<b>0.2160</b>	–	–	0.2472
	$RMSE_{pos}$	<b>2.35</b>	16.38	3.51	4.36

### 5.3 Experimental Results

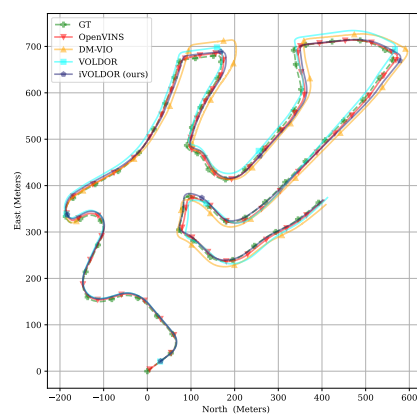
The primary objective of the tests is to demonstrate the effectiveness of the proposed fusion of IMU and incremental pose. Initially, the *VOLDOR* [35] algorithm is evaluated as the baseline. Subsequently, by integrating the IMU measurements with the incremental poses of *VOLDOR* [35], an inertial-aided version, referred to as *iVOLDOR*, is obtained. For completeness, I also evaluated two state-of-the-art odometry algorithms, OpenVINS and DM-VIO.

Each test sequence is analyzed individually in the following sections. However, the overall results are first presented in Table 5.3. As an initial observation, iVOLDOR demonstrates performance improvements over *VOLDOR* [35]. Furthermore, the proposed method competes closely with OpenVINS in all sequences. Although DM-VIO is a promising algorithm, its reliance on a monocular camera limits its performance, resulting in less satisfactory results.

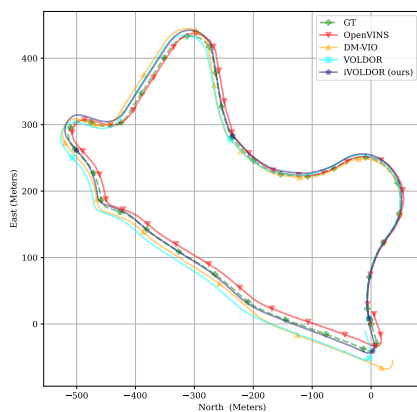
Finally, Figure 5.3 presents an overview of the estimated trajectories for each sequence, showing the estimated paths alongside the actual trajectory.



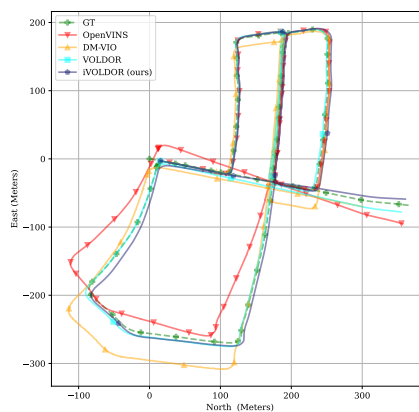
(a) *Seq1* Trajectories



(b) *Seq2* Trajectories



(c) *Seq3* Trajectories



(d) *Seq4* Trajectories

Figure 5.3: Trajectories Overview

### 5.3.1 Seq1 - Error Plots

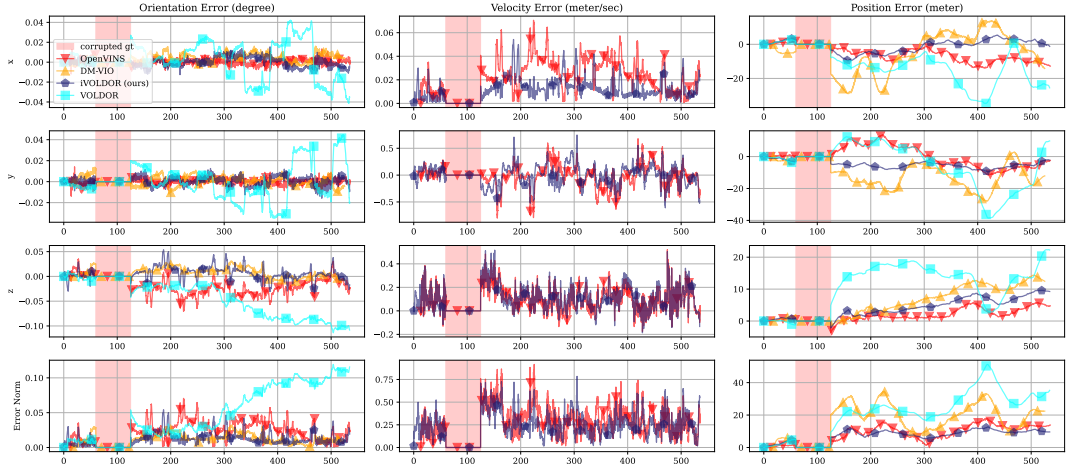


Figure 5.4: Error Plot For Sequence 1

Figure 5.4 demonstrates the orientation, velocity and position errors for each  $x$ ,  $y$  and  $z$  channel. For the convenience of the reader, I also provide the  $l_2$  norm of the error in the last row. As outlined before, for a small portion of the trajectory, the ground-truth is corrupted. The corrupted region is highlighted in the Figure and not included in RMSE evaluation in Table 5.3.

Figure 5.4 reveals that even though the performance of *VOLDOR* [35] is less satisfactory compared to OpenVINS and DM-VIO, the integration of inertial data significantly enhances its performance, enabling iVOLDOR to achieve superior results.

### 5.3.2 Seq2 - Error Plots

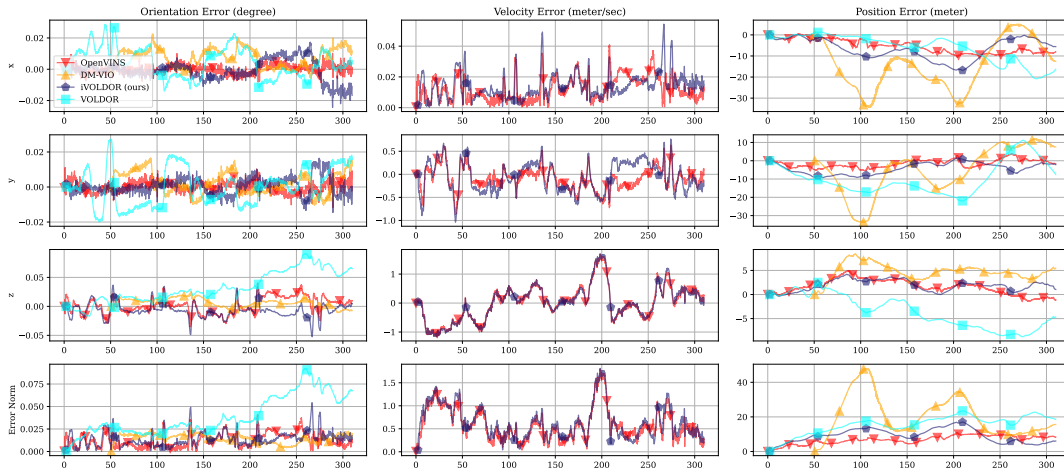


Figure 5.5: Error Plot For Sequence 2

### 5.3.3 Seq3 - Error Plots

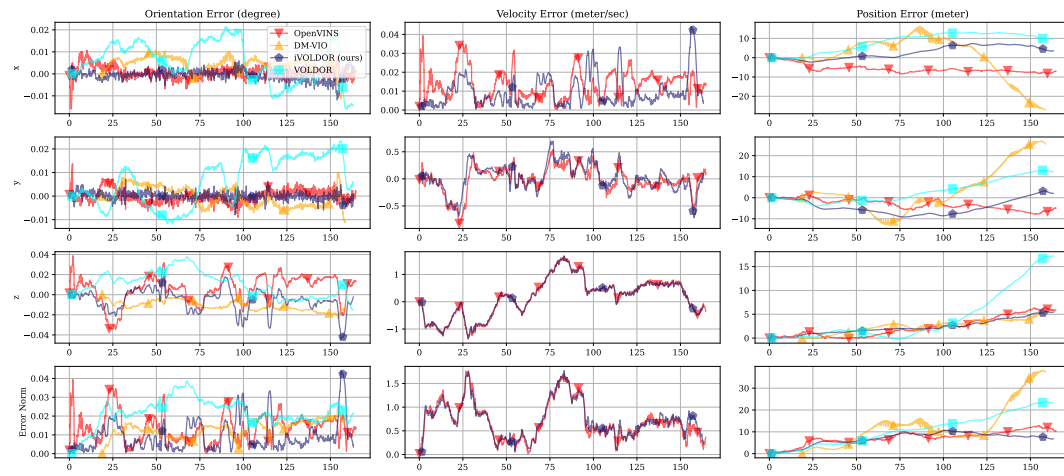


Figure 5.6: Error Plot For Sequence 3



### 5.3.4 Seq4 - Error Plots and a Failure Case

This test sequence reveals a very fundamental failure case for conventional odometry algorithms. First, examine Figure 5.7. First of all, the dataset duration is nearly 300 seconds while only the first 150 seconds is employed during the RMSE evaluation for Table 5.3.

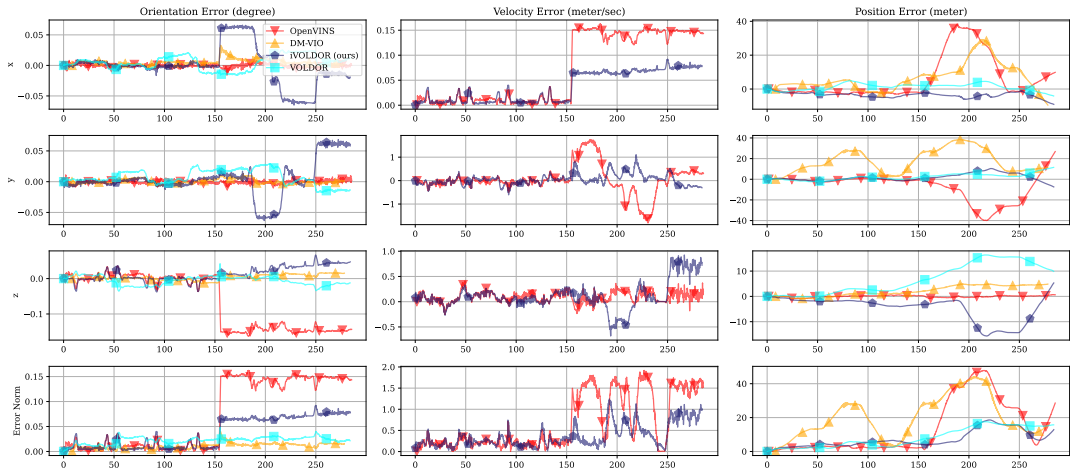


Figure 5.7: Error Plot For Sequence 4

In Figure 5.7, it is observed that the error sharply increases for both OpenVINS and DM-VIO at approximately  $t = 150$ s. Up to that point, OpenVINS demonstrated the best performance among all methods.

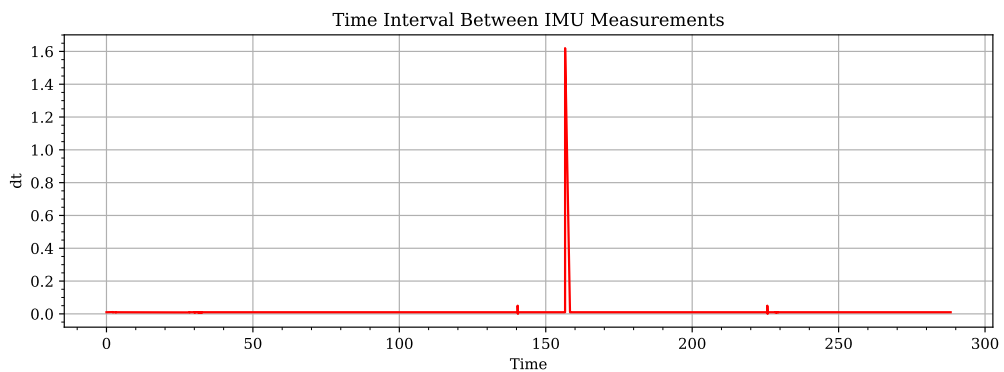


Figure 5.8: Time Intervals Between Successive IMU Measurements

To understand the underlying cause of the abrupt performance degradation, we refer to Figure 5.8, which illustrates the time intervals between successive IMU measure-

ments. While the IMU is expected to output measurements at 100 Hz, it experiences a delay of approximately 1.6 second around  $t = 150$ . This highlights a critical limitation of conventional visual-inertial navigation methods, which assume regular IMU measurement intervals and utilize a zero-order holder. Consequently, a single delayed measurement leads to catastrophic performance degradation. When this pattern is repeated, the algorithms even tend to diverge.

## 5.4 Comparison of OpenVINS and VOLDOR

The experiments demonstrate that OpenVINS and iVOLDOR achieve competitive performance despite their fundamentally different approaches. OpenVINS is a lightweight, feature-based system that extracts sparse features using the Harris Corner detector, discarding the rest of the image. This results in a minimal and computationally efficient image processing pipeline that runs on a CPU. In contrast, iVOLDOR employs a dense methodology, using a deep neural network to compute optical flow vectors and generate a depth map for the entire image. This approach utilizes every pixel for estimation but requires significantly higher computational resources, including a GPU.

OpenVINS delivers competitive results due to two key factors. First, its sensor-level information fusion integrates visual and IMU data at an early stage, enabling better exploitation of complementary data. iVOLDOR, on the other hand, relies on decision-level fusion, where visual and IMU information is combined at a later stage, limiting the benefits of early integration. Second, OpenVINS employs a robust feature-tracking mechanism, preserving long-term dependencies by tracking sparse features across frames. In contrast, iVOLDOR uses dense optical flow without feature tracking, leading to a loss of long-term dependencies during application.

### 5.4.1 Real Time Performances

Both iVOLDOR and OpenVINS can achieve real-time operation, but their computational requirements depend on the selected parameters and system configuration. For OpenVINS, the number of historical clones is a critical factor. Typically, 10-20 clones

are used for effective operation, but some applications may demand longer clone sequences. When exceeding 50 clones, the state dimension grows significantly, potentially hindering real-time performance. In such cases, additional hardware and further parallelization may be necessary. Despite this, OpenVINS remains lightweight and energy-efficient as it is fully implemented on a CPU.

In contrast, iVOLDOR inherently requires a GPU alongside a CPU for real-time performance. The image resolution is a key parameter; as every pixel contributes to the estimation process, resizing the frames is often necessary to maintain real-time operation. Furthermore, the optical flow computation relies on a deep neural network, and depending on the selected model, the user may require two GPUs. One GPU can handle optical flow estimation, while the second manages depth map computations and SLAM tasks. Consequently, iVOLDOR demands more computational resources and hardware for real-time application compared to OpenVINS.

## 5.5 Conclusions

This thesis primarily explores the application of Lie Algebra in state estimation. It begins by introducing the challenges posed by rotation matrices and rigid transformations in representing error and uncertainty. Subsequently, it demonstrates how Lie Algebra effectively addresses these challenges. The main objective is to provide a compact and concise review of Lie Algebra specifically within the context of state estimation, avoiding unnecessary in-depth discussions that would require multiple book chapters.

To illustrate the practical utilization of Lie Algebra, a simple extended Kalman filter is proposed, which fuses IMU measurements with incremental poses. Experimental results show that combining these two sources of information yields more accurate trajectory estimates. Furthermore, the proposed method, termed iVOLDOR, proves to be competitive with existing approaches.

However, it is important to emphasize that the proposed fusion method operates at higher levels. The visual SLAM system has already determined incremental poses, and these processed data are then fused with IMU measurements. Fusion at such

higher levels often leads to inferior performance. In contrast, OpenVINS fuses information at lower levels, resulting in better performance across different sequences, despite being computationally much more lightweight. These experiments underscore the importance of lower-level fusion.

Additionally, as a byproduct, the experiments reveal that most conventional odometry methods heavily depend on the assumption that IMU data or camera frames arrive at regular intervals. Consequently, a single delayed IMU measurement—by just one second—can cause the entire system to diverge. This issue highlights the need for practical systems to handle such delays effectively.

## REFERENCES

- [1]Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. *Ceres Solver*. Version 2.2. Oct. 2023. URL: <https://github.com/ceres-solver/ceres-solver>.
- [2]Ahmet Akman, Abdullah Aydın Alatan, and YUNUS BİLGE KURT. “Causal Transformer for Fusion and Pose Estimation in Deep Visual Inertial Odometry”. In: *ECCV 2024 Workshop on Vision-Centric Autonomous Driving (VCAD)*, 2024. URL: <https://vcad-workshop.github.io/>.
- [3]Timothy D. Barfoot. *State Estimation for Robotics*. Cambridge, UK: Cambridge University Press, 2024. ISBN: 978-1108849830.
- [4]Timothy D. Barfoot and Paul Furgale. “Associating uncertainty with three-dimensional poses for use in estimation problems”. In: *IEEE Transactions on Robotics* 30.3 (2014), pp. 679–693. DOI: 10.1109/tro.2014.2298059.
- [5]Axel Barrau and Silvère Bonnabel. “Invariant filtering for Pose EKF-SLAM aided by an IMU”. In: *2015 54th IEEE Conference on Decision and Control (CDC)*. 2015, pp. 2133–2138. DOI: 10.1109/CDC.2015.7402522.
- [6]Axel Barrau and Silvère Bonnabel. “Invariant particle filtering with application to localization”. In: *53rd IEEE Conference on Decision and Control*. 2014, pp. 5599–5605. DOI: 10.1109/CDC.2014.7040265.
- [7]Axel Barrau and Silvère Bonnabel. “The Invariant Extended Kalman Filter as a Stable Observer”. In: *IEEE Transactions on Automatic Control* 62.4 (2017), pp. 1797–1812. DOI: 10.1109/TAC.2016.2594085.
- [8]Martin Brossard, Silvère Bonnabel, and Axel Barrau. “Unscented Kalman Filter on Lie Groups for Visual Inertial Odometry”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018, pp. 649–655. DOI: 10.1109/IROS.2018.8593627.
- [9]Martin Brossard, Silvère Bonnabel, and Axel Barrau. “Invariant Kalman Filtering for Visual Inertial SLAM”. In: *2018 21st International Conference on Informa-*

- tion Fusion (FUSION)*. 2018, pp. 2021–2028. DOI: 10.23919/ICIF.2018.8455807.
- [10]Martin Brossard et al. “Associating Uncertainty to Extended Poses for on Lie Group IMU Preintegration With Rotating Earth”. In: *IEEE Transactions on Robotics* 38.2 (2022), pp. 998–1015. DOI: 10.1109/TRO.2021.3100156.
- [11]Carlos Campos et al. “ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM”. In: *IEEE Transactions on Robotics* 37.6 (2021), pp. 1874–1890.
- [12]Samuel Cerezo and Javier Civera. “Camera Motion Estimation from RGB-D-Inertial Scene Flow”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2024, pp. 841–849.
- [13]Chuchu Chen et al. “FEJ2: A Consistent Visual-Inertial State Estimator Design”. In: *2022 International Conference on Robotics and Automation (ICRA)*. 2022, pp. 9506–9512. DOI: 10.1109/ICRA46639.2022.9811831.
- [14]Chuchu Chen et al. “Monocular Visual-Inertial Odometry with Planar Regularities”. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 6224–6231. DOI: 10.1109/ICRA48891.2023.10160620.
- [15]Chuchu Chen et al. “Optimization-Based VINS: Consistency, Marginalization, and FEJ”. In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2023, pp. 1517–1524. DOI: 10.1109/IROS55552.2023.10341637.
- [16]Frank Dellaert and GTSAM Contributors. *borglab/gtsam*. Version 4.2a8. May 2022. DOI: 10.5281/zenodo.5794541. URL: <https://github.com/borglab/gtsam>.
- [17]Steven Diamond and Stephen Boyd. “CVXPY: A Python-embedded modeling language for convex optimization”. In: *Journal of Machine Learning Research* 17.83 (2016), pp. 1–5.
- [18]Alexey Dosovitskiy et al. “FlowNet: Learning Optical Flow With Convolutional Networks”. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Dec. 2015.
- [19]Jakob Engel, Vladlen Koltun, and Daniel Cremers. “Direct Sparse Odometry”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.3 (2018), pp. 611–625. DOI: 10.1109/TPAMI.2017.2658577.

- [20]Jakob Engel, Jörg Stückler, and Daniel Cremers. “Large-scale direct SLAM with stereo cameras”. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015, pp. 1935–1942. DOI: 10.1109/IROS.2015.7353631.
- [21]Andreas Geiger, Philip Lenz, and Raquel Urtasun. “Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.
- [22]Andreas Geiger et al. “Vision meets Robotics: The KITTI Dataset”. In: *International Journal of Robotics Research (IJRR)* (2013).
- [23]Patrick Geneva et al. “OpenVINS: A Research Platform for Visual-Inertial Estimation”. In: *Proc. of the IEEE International Conference on Robotics and Automation*. Paris, France, 2020. URL: %5Curl%7Bhttps://github.com/rpng/open%5C\_vins%7D.
- [24]Pieter van Goor, Tarek Hamel, and Robert Mahony. “Equivariant Filter (EqF): A General Filter Design for Systems on Homogeneous Spaces”. In: *2020 59th IEEE Conference on Decision and Control (CDC)*. 2020, pp. 5401–5408. DOI: 10.1109/CDC42340.2020.9303813.
- [25]Pieter van Goor and Robert Mahony. “EqVIO: An Equivariant Filter for Visual-Inertial Odometry”. In: *IEEE Transactions on Robotics* 39.5 (2023), pp. 3567–3585. DOI: 10.1109/TRO.2023.3289587.
- [26]Giorgio Grisetti et al. “A Tutorial on Graph-Based SLAM”. In: *IEEE Intelligent Transportation Systems Magazine* 2.4 (2010), pp. 31–43. DOI: 10.1109/MITS.2010.939925.
- [27]Paul D Groves. *Principles of GNSS, inertial, and multisensor integrated navigation systems*. Artech House, 2013.
- [28]Joel A. Hesch et al. “Consistency Analysis and Improvement of Vision-aided Inertial Navigation”. In: *IEEE Transactions on Robotics* 30.1 (2014), pp. 158–176. DOI: 10.1109/TRO.2013.2277549.
- [29]Guoquan P. Huang, Anastasios I. Mourikis, and Stergios I. Roumeliotis. “A First-Estimates Jacobian EKF for Improving SLAM Consistency”. In: *Experimental Robotics*. Ed. by Raja Chatila and Rudolf Mester. Springer, 2009, pp. 373–382. DOI: 10.1007/978-3-642-00196-3\_43.

- [30]Eddy Ilg et al. “FlowNet 2.0: Evolution of Optical Flow Estimation With Deep Networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.
- [31]Saimouli Katragadda et al. “NeRF-VINS: A Real-time Neural Radiance Field Map-based Visual-Inertial Navigation System”. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. 2024, pp. 10230–10237. DOI: 10.1109/ICRA57147.2024.10610051.
- [32]Rainer Kümmerle et al. “G2o: A general framework for graph optimization”. In: *2011 IEEE International Conference on Robotics and Automation*. 2011, pp. 3607–3613. DOI: 10.1109/ICRA.2011.5979949.
- [33]Maani Li and Anastasios I. Mourikis. “High-Precision, Consistent EKF-Based Visual-Inertial Odometry”. In: *The International Journal of Robotics Research* 32.6 (2013), pp. 690–711.
- [34]Willis K. Miller and Ronny M. Wells. *Stochastic Models, Information Theory, and Lie Groups, Volume 2*. 1st. Applied and Numerical Harmonic Analysis. Springer, 2011. ISBN: 978-1441995035. DOI: 10.1007/978-1-4419-9504-2.
- [35]Zhixiang Min and Enrique Dunn. “VOLDOR+SLAM: For the times when feature-based or direct methods are not good enough”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 13813–13819. DOI: 10.1109/ICRA48506.2021.9561230.
- [36]Anastasios I. Mourikis and Stergios I. Roumeliotis. “A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation”. In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. 2007, pp. 3565–3572. DOI: 10.1109/ROBOT.2007.364024.
- [37]Antoni Rosinol, John J. Leonard, and Luca Carlone. *NeRF-SLAM: Real-Time Dense Monocular SLAM with Neural Radiance Fields*. 2022. arXiv: 2210.13641 [cs.CV]. URL: <https://arxiv.org/abs/2210.13641>.
- [38]Furkan Aykut Sarıkamış and Abdullah Aydın Alatan. “IG-SLAM: Instant Gaussian SLAM”. In: *ECCV 2024 Workshop on Dense Neural SLAM (NeuSLAM)*, 2024. URL: <https://sites.google.com/view/neuslam/program>.
- [39]Hochang Seok and Jongwoo Lim. “ROVINS: Robust Omnidirectional Visual Inertial Navigation System”. In: *IEEE Robotics and Automation Letters* 5.4 (2020), pp. 6225–6232. DOI: 10.1109/LRA.2020.3010457.



- [40]Mo Shan, Qiaojun Feng, and Nikolay Atanasov. “OrcVIO: Object residual constrained Visual-Inertial Odometry”. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020, pp. 5104–5111. DOI: 10.1109/IROS45743.2020.9341660.
- [41]Joan Solà, Jeremie Deray, and Dinesh Atchuthan. *A micro Lie theory for state estimation in robotics*. 2021. arXiv: 1812.01537 [cs.RO].
- [42]Lukas von Stumberg and Daniel Cremers. “DM-VIO: Delayed Marginalization Visual-Inertial Odometry”. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 1408–1415. DOI: 10.1109/LRA.2021.3140129.
- [43]Ke Sun et al. “Robust Stereo Visual Inertial Odometry for Fast Autonomous Flight”. In: *IEEE Robotics and Automation Letters* 3.2 (2018), pp. 965–972. DOI: 10.1109/LRA.2018.2793349.
- [44]Zachary Teed and Jia Deng. “DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras”. In: *Advances in neural information processing systems* (2021).
- [45]Zachary Teed and Jia Deng. *RAFT: Recurrent All-Pairs Field Transforms for Optical Flow*. 2020. arXiv: 2003.12039 [cs.CV]. URL: <https://arxiv.org/abs/2003.12039>.
- [46]Chen Wang et al. “PyPose: A Library for Robot Learning with Physics-based Optimization”. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 22024–22034. DOI: 10.1109/CVPR52729.2023.02109.
- [47]Han Wang et al. “F-LOAM : Fast LiDAR Odometry and Mapping”. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021, pp. 4390–4396. DOI: 10.1109/IROS51168.2021.9636655.
- [48]Kanzhi Wu et al. “An invariant-EKF VINS algorithm for improving consistency”. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 1578–1585. DOI: 10.1109/IROS.2017.8205965.
- [49]Mingyu Yang, Yu Chen, and Hun-Seok Kim. “Efficient Deep Visual and Inertial Odometry with Adaptive Visual Modality Selection”. In: *arXiv preprint arXiv:2205.06187* (2022).
- [50]Yulin Yang et al. “Analytic Combined IMU Integration (ACI2) For Visual Inertial Navigation”. In: *2020 IEEE International Conference on Robotics and Au-*

- tomation (ICRA)*. 2020, pp. 4680–4686. DOI: 10.1109/ICRA40945.2020.9197280.
- [51]Yulin Yang et al. “Decoupled Right Invariant Error States for Consistent Visual-Inertial Navigation”. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 1627–1634. DOI: 10.1109/LRA.2021.3140054.
- [52]Zichao Zhang and Davide Scaramuzza. “A Tutorial on Quantitative Trajectory Evaluation for Visual(-Inertial) Odometry”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018, pp. 7244–7251. DOI: 10.1109/IROS.2018.8593941.
- [53]Shibo Zhao et al. “Super Odometry: IMU-centric LiDAR-Visual-Inertial Estimator for Challenging Environments”. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021, pp. 8729–8736. DOI: 10.1109/IROS51168.2021.9635862.
- [54]Zihua Zheng et al. “DIP: Deep Inverse Patchmatch for High-Resolution Optical Flow”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 8925–8934.

## Appendix A

### APPENDIX FOR LIE ALGEBRA

#### A.1 Skew Symmetric Matrices

Skew-symmetric matrices play a crucial role in handling rotation matrices. Therefore, I present several useful properties of skew-symmetric matrices that will be employed in various proofs.

- Skew Symmetric Matrix Operator :  $[\cdot]$

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \in \mathbb{R}^{3 \times 1} \quad [\boldsymbol{\omega}] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \in \mathbb{R}^{3 \times 3} \quad (\text{A.1})$$

- Let  $\mathbf{R} \in SO(3)$  be a rotation matrix and  $\mathbf{v} \in \mathbb{R}^{3 \times 1}$  be any vector. Then,

$$\mathbf{R} [\mathbf{v}] \mathbf{R}^T = [\mathbf{R} \mathbf{v}] \quad (\text{A.2})$$

Equivalently

$$\mathbf{R} [\mathbf{v}] = [\mathbf{R} \mathbf{v}] \mathbf{R} \quad (\text{A.3})$$

- $\forall \mathbf{a}, \mathbf{b} \in \mathbb{R}^{3 \times 1}$

$$[\mathbf{a}] \mathbf{b} = -[\mathbf{b}] \mathbf{a} \quad (\text{A.4})$$

- For a unit norm vector  $\mathbf{a} \in \mathbb{R}^{3 \times 1}$  where  $\|\mathbf{a}\|^2 = 1$ , it is given that

$$[\mathbf{a}] [\mathbf{a}] = -\mathbf{I}_3 + \mathbf{a} \mathbf{a}^T \quad (\text{A.5})$$

$$[\mathbf{a}] [\mathbf{a}] [\mathbf{a}] = -[\mathbf{a}] \quad (\text{A.6})$$

**Claim 5.** For a unit norm vector  $\mathbf{a} \in \mathbb{R}^{3 \times 1}$  where  $\|\mathbf{a}\|^2 = 1$ ,  $[\mathbf{a}]^k$  can be expressed in terms of  $[\mathbf{a}]$  or  $[\mathbf{a}]^2$  for any positive integer  $k$ . Specifically,

$$[\mathbf{a}]^{2k} = \begin{cases} [\mathbf{a}][\mathbf{a}], & \text{if } k \text{ is an odd number} \\ -[\mathbf{a}][\mathbf{a}], & \text{if } k \text{ is an even number} \end{cases} \quad (\text{A.7})$$

$$[\mathbf{a}]^{2k+1} = \begin{cases} -[\mathbf{a}], & \text{if } k \text{ is an odd number} \\ [\mathbf{a}], & \text{if } k \text{ is an even number} \end{cases} \quad (\text{A.8})$$

*Proof.*

$$\xrightarrow{\text{By A.6}} [\mathbf{a}]^3 = -[\mathbf{a}] \quad (\text{A.9})$$

$$[\mathbf{a}]^4 = -[\mathbf{a}][\mathbf{a}] \quad (\text{A.10})$$

$$[\mathbf{a}]^5 = -[\mathbf{a}][\mathbf{a}][\mathbf{a}] = [\mathbf{a}] \quad (\text{A.11})$$

$$[\mathbf{a}]^6 = [\mathbf{a}][\mathbf{a}] \quad (\text{A.12})$$

$$[\mathbf{a}]^7 = [\mathbf{a}][\mathbf{a}][\mathbf{a}] = -[\mathbf{a}] \quad (\text{A.13})$$

$$[\mathbf{a}]^8 = -[\mathbf{a}][\mathbf{a}] \quad (\text{A.14})$$

$$[\mathbf{a}]^9 = -[\mathbf{a}][\mathbf{a}][\mathbf{a}] = [\mathbf{a}] \quad (\text{A.15})$$

⋮

Q.E.D.

## A.2 Taylor Expansion of $\sin(\cdot)$ and $\cos(\cdot)$

Within the thesis, I will need the Taylor expansion of  $\sin(\cdot)$  and  $\cos(\cdot)$ .

$$\sin(\theta) = \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \frac{\theta^7}{7!} + \dots = \sum_{n=0}^{\infty} \frac{(-1)^n \theta^{2n+1}}{(2n+1)!} \quad (\text{A.16})$$

$$\cos(\theta) = 1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \frac{\theta^6}{6!} + \dots = \sum_{n=0}^{\infty} \frac{(-1)^n \theta^{2n}}{(2n)!} \quad (\text{A.17})$$

## A.3 Rotation Matrices and Rigid Transformations - Representing The Pose

Navigation aims to estimate both the rotation and position of a body relative to a predefined coordinate frame. The combined term for rotation and position is referred

to as "pose". This section focuses on explaining the representation and manipulation of pose in navigation systems.

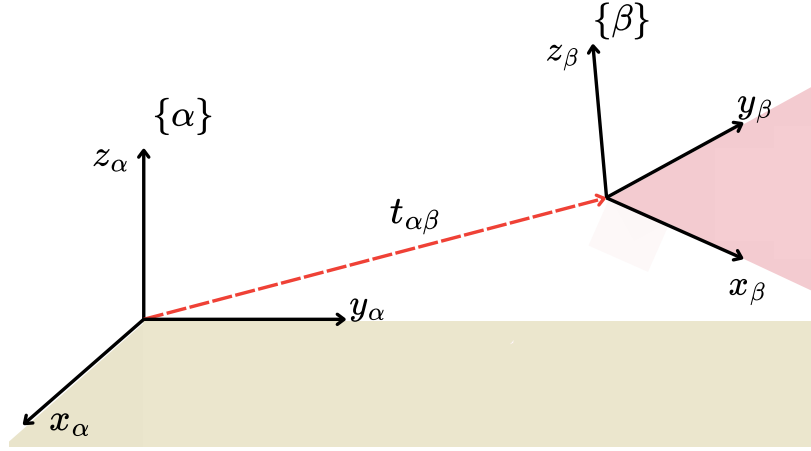


Figure A.1: Visualization of 2 Distinct Frames ( $\alpha$  and  $\beta$ )

In Figure A.1, two distinct frames are depicted. The origins of these frames are separated by the vector  $\mathbf{t}_{\alpha\beta}$ , and their axes are not aligned. For example, assume that frame  $\alpha$  is the global navigation frame, fixed relative to the Earth, while frame  $\beta$  represents the body of a drone navigating outdoors.

The most intuitive way to express the rotation (attitude) of frame  $\beta$  with respect to frame  $\alpha$  is by representing each axis of frame  $\beta$  in terms of frame  $\alpha$ . This involves resolving the directions of the axes  $x_\beta$ ,  $y_\beta$ , and  $z_\beta$  with respect to the  $\alpha$  frame, denoted as  $\mathbf{x}_\beta^\alpha \in \mathbb{R}^{3 \times 1}$ ,  $\mathbf{y}_\beta^\alpha \in \mathbb{R}^{3 \times 1}$ , and  $\mathbf{z}_\beta^\alpha \in \mathbb{R}^{3 \times 1}$ , respectively. In order to provide a compact representation, we define the rotation matrix  $\mathbf{R}_\beta^\alpha \in \mathbb{R}^{3 \times 3}$ .

$$\mathbf{R}_\beta^\alpha \triangleq \begin{bmatrix} \mathbf{x}_\beta^\alpha & \mathbf{y}_\beta^\alpha & \mathbf{z}_\beta^\alpha \end{bmatrix} \quad (\text{A.18})$$

We can express an arbitrary vector  $\mathbf{v} \in \mathbb{R}^3$  with respect to any coordinate frame  $\alpha$ . Superscript in  $\mathbf{v}^\alpha$  indicates that the resolving frame is  $\alpha$ .

**Claim 6.** We can change the resolving frame of an arbitrary vector  $\mathbf{v}^\beta \in \mathbb{R}^3$  by multiplying it on the left by the rotation matrix  $\mathbf{R}_\beta^\alpha$ . In other words,

$$\mathbf{v}^\alpha = \mathbf{R}_\beta^\alpha \mathbf{v}^\beta \quad (\text{A.19})$$

*Proof.* Let us express the vector  $\mathbf{v}^\beta$  in terms of the basis vectors of the coordinate

frame  $\beta$  by decomposing it as a linear combination of the basis vectors  $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$

$$\mathbf{v}^\beta = v_1 \mathbf{e}_1 + v_2 \mathbf{e}_2 + v_3 \mathbf{e}_3 \quad (\text{A.20})$$

where

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (\text{A.21})$$

To express the same vector  $\mathbf{v}$  in  $\alpha$  frame, we can represent the basis vectors of the  $\beta$  frame in Equation A.20 with respect to the  $\alpha$  frame. In other words, the vector  $\mathbf{v}^\alpha$  can be written as:

$$\mathbf{v}^\alpha = v_1 \mathbf{x}_\beta^\alpha + v_2 \mathbf{y}_\beta^\alpha + v_3 \mathbf{z}_\beta^\alpha \quad (\text{A.22})$$

Recall the definition of  $\mathbf{R}_\beta^\alpha$  in Equation A.18. One can write

$$\mathbf{v}^\alpha = v_1 \mathbf{x}_\beta^\alpha + v_2 \mathbf{y}_\beta^\alpha + v_3 \mathbf{z}_\beta^\alpha \quad (\text{A.23})$$

$$= v_1 \mathbf{R}_\beta^\alpha \mathbf{e}_1 + v_2 \mathbf{R}_\beta^\alpha \mathbf{e}_2 + v_3 \mathbf{R}_\beta^\alpha \mathbf{e}_3 \quad (\text{A.24})$$

$$= \mathbf{R}_\beta^\alpha \left( v_1 \mathbf{e}_1 + v_2 \mathbf{e}_2 + v_3 \mathbf{e}_3 \right) \quad (\text{A.25})$$

$$\mathbf{v}^\alpha = \mathbf{R}_\beta^\alpha \mathbf{v}^\beta \quad (\text{A.26})$$

Q.E.D.

**Claim 7.** *We have the following relation between rotation matrices.*

$$\mathbf{R}_\gamma^\alpha = \mathbf{R}_\beta^\alpha \mathbf{R}_\gamma^\beta \quad (\text{A.27})$$

*Proof.*

$$\mathbf{R}_\beta^\alpha \mathbf{R}_\gamma^\beta = \mathbf{R}_\beta^\alpha \begin{bmatrix} \mathbf{x}_\gamma^\beta & \mathbf{y}_\gamma^\beta & \mathbf{z}_\gamma^\beta \end{bmatrix} \quad (\text{A.28})$$

$$= \begin{bmatrix} \mathbf{R}_\beta^\alpha \mathbf{x}_\gamma^\beta & \mathbf{R}_\beta^\alpha \mathbf{y}_\gamma^\beta & \mathbf{R}_\beta^\alpha \mathbf{z}_\gamma^\beta \end{bmatrix} \quad (\text{A.29})$$

$$= \begin{bmatrix} \mathbf{x}_\gamma^\alpha & \mathbf{y}_\gamma^\alpha & \mathbf{z}_\gamma^\alpha \end{bmatrix} \quad (\text{A.30})$$

$$= \mathbf{R}_\gamma^\alpha \quad (\text{A.31})$$

Q.E.D.

It is important to note that we are considering vector quantities. Therefore, the specific starting point of the vector is irrelevant in this context.

In contrary to vectors, the position (or coordinates) of a point cannot be transformed by simple multiplication with a rotation matrix. Let us examine Figure A.2 which illustrates the fact that

$$\mathbf{t}_{\alpha\gamma}^{\alpha} = \mathbf{t}_{\alpha\beta}^{\alpha} + \mathbf{t}_{\beta\gamma}^{\alpha} \quad (\text{A.32})$$

$$= \mathbf{t}_{\alpha\beta}^{\alpha} + \mathbf{R}_{\beta}^{\alpha} \mathbf{t}_{\beta\gamma}^{\beta} \quad (\text{A.33})$$

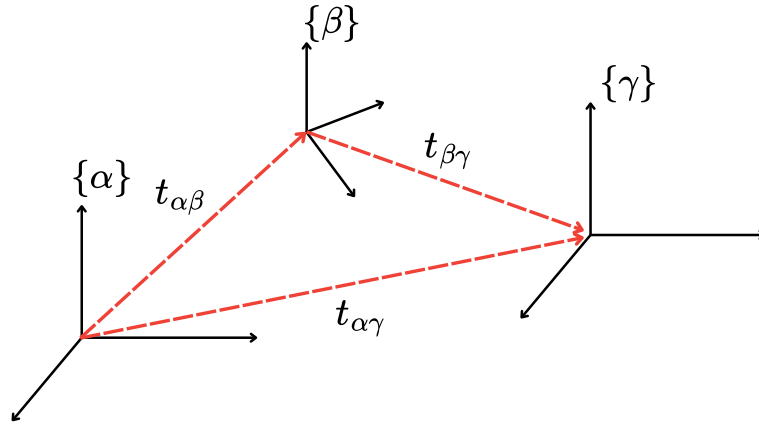


Figure A.2: Visualization of 3 Distinct Frames ( $\alpha$ ,  $\beta$  and  $\gamma$ )

Now, we are ready to introduce the Special Euclidean Group  $SE(3)$  to represent the **pose** of a frame  $\beta$  with respect to frame  $\alpha$ .  $\mathbf{T}_{\beta}^{\alpha} \in SE(3) \subset \mathbb{R}^{4 \times 4}$  is defined as follows

$$\mathbf{T}_{\beta}^{\alpha} \triangleq \begin{bmatrix} \mathbf{R}_{\beta}^{\alpha} & \mathbf{t}_{\alpha\beta}^{\alpha} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (\text{A.34})$$

**Claim 8.** *We have the following relation between frames*

$$\mathbf{T}_{\gamma}^{\alpha} = \mathbf{T}_{\beta}^{\alpha} \mathbf{T}_{\gamma}^{\beta} \quad (\text{A.35})$$

*Proof.*

$$\mathbf{T}_\beta^\alpha \mathbf{T}_\gamma^\beta = \begin{bmatrix} \mathbf{R}_\beta^\alpha & \mathbf{t}_{\alpha\beta}^\alpha \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_\gamma^\beta & \mathbf{t}_{\beta\gamma}^\beta \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (\text{A.36})$$

$$= \begin{bmatrix} \mathbf{R}_\beta^\alpha \mathbf{R}_\gamma^\beta & \mathbf{R}_\beta^\alpha \mathbf{t}_{\beta\gamma}^\beta + \mathbf{t}_{\alpha\beta}^\alpha \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (\text{A.37})$$

$$\xrightarrow{\text{By Eq. A.27 and Eq. A.33}} = \begin{bmatrix} \mathbf{R}_\gamma^\alpha & \mathbf{t}_{\alpha\gamma}^\alpha \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (\text{A.38})$$

$$\xrightarrow[\text{given in Eq. A.34}]{\text{By definition}} = \mathbf{T}_\gamma^\alpha \quad (\text{A.39})$$

Q.E.D.

Finally, we define the homogeneous coordinates of frame  $\beta$  with respect to frame  $\alpha$ .

$$\mathbf{t}_\beta^\alpha \triangleq \begin{bmatrix} \mathbf{t}_{\alpha\beta}^\alpha \\ 1 \end{bmatrix} \in \mathbb{R}^{4 \times 1} \quad (\text{A.40})$$



**Claim 9.** *We have the following relation*

$$\mathbf{t}_\gamma^\alpha = \mathbf{T}_\beta^\alpha \mathbf{t}_\gamma^\beta \quad (\text{A.41})$$

*Proof.*

$$\mathbf{T}_\beta^\alpha \mathbf{t}_\gamma^\beta = \begin{bmatrix} \mathbf{R}_\beta^\alpha & \mathbf{t}_{\alpha\beta}^\alpha \\ \mathbf{0}_{1 \times 3} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{t}_{\beta\gamma}^\beta \\ 1 \end{bmatrix} \quad (\text{A.42})$$

$$= \begin{bmatrix} \mathbf{R}_\beta^\alpha \mathbf{t}_{\beta\gamma}^\beta + \mathbf{t}_{\alpha\beta}^\alpha \\ 1 \end{bmatrix} \quad (\text{A.43})$$

$$\xrightarrow{\text{By Eq. A.33}} = \begin{bmatrix} \mathbf{t}_{\alpha\gamma}^\alpha \\ 1 \end{bmatrix} \quad (\text{A.44})$$

$$\xrightarrow[\text{given in Eq. A.40}]{\text{By definition}} = \mathbf{t}_\gamma^\alpha \quad (\text{A.45})$$

Q.E.D.

#### A.4 Matrix Exponential and Matrix Logarithm

Throughout the thesis, we will frequently encounter matrix exponential and matrix logarithm functions. We will see that these functions play crucial role in Lie Algebra, which will enable us to take the derivative of a function with respect to a rotation matrix.

The exponential of a square matrix  $\mathbf{A} \in \mathbb{R}^{d \times d}$  is defined through Taylor expansion.

$$\exp_m(\mathbf{A}) \triangleq \sum_{n=0}^{\infty} \frac{\mathbf{A}^n}{n!} \quad (\text{A.46})$$

Matrix logarithm  $\log_m(\cdot)$  is defined as the inverse of  $\exp_m(\cdot)$ .

**Claim 10.** *If we have  $\mathbf{A}\mathbf{B} = \mathbf{B}\mathbf{A}$  where  $\mathbf{A} \in \mathbb{R}^{d \times d}$  and  $\mathbf{B} \in \mathbb{R}^{d \times d}$ , then the following relation holds*

$$\exp_m(\mathbf{A} + \mathbf{B}) = \exp_m(\mathbf{A}) \exp_m(\mathbf{B}) \quad (\text{A.47})$$

*Proof.*

$$\exp_m(\mathbf{A} + \mathbf{B}) = \sum_{n=0}^{\infty} \frac{(\mathbf{A} + \mathbf{B})^n}{n!} \quad (\text{A.48})$$

$$\xrightarrow[\text{Expansion}]{\text{Binomial}} = \sum_{n=0}^{\infty} \sum_{k=0}^n \frac{1}{n!} \binom{n}{k} \mathbf{A}^k \mathbf{B}^{n-k} \quad (\text{A.49})$$

$$= \sum_{n=0}^{\infty} \sum_{k=0}^n \frac{1}{n!} \frac{n!}{(n-k)! k!} \mathbf{A}^k \mathbf{B}^{n-k} \quad (\text{A.50})$$

$$= \sum_{n=0}^{\infty} \sum_{k=0}^n \frac{1}{(n-k)! k!} \mathbf{A}^k \mathbf{B}^{n-k} \quad (\text{A.51})$$

$$= \sum_{k=0}^{\infty} \sum_{n=k}^{\infty} \frac{1}{(n-k)! k!} \mathbf{A}^k \mathbf{B}^{n-k} \quad (\text{A.52})$$

$$\xrightarrow[\substack{m \triangleq n-k}]{\text{Change of Variable}} = \sum_{k=0}^{\infty} \sum_{m=0}^{\infty} \frac{1}{m! k!} \mathbf{A}^k \mathbf{B}^m \quad (\text{A.53})$$

$$= \sum_{k=0}^{\infty} \frac{\mathbf{A}^k}{k!} \sum_{m=0}^{\infty} \frac{\mathbf{B}^m}{m!} \quad (\text{A.54})$$

$$= \exp_m(\mathbf{A}) \exp_m(\mathbf{B}) \quad (\text{A.55})$$

Note that the Binomial expansion in Equation A.49 is guaranteed to be valid only if we have  $\mathbf{AB} = \mathbf{BA}$ . In other words,

$$\mathbf{AB} = \mathbf{BA} \Rightarrow (\mathbf{A} + \mathbf{B})^n = \sum_{k=0}^n \binom{n}{k} \mathbf{A}^k \mathbf{B}^{n-k} \quad (\text{A.56})$$

Q.E.D.

## A.5 More on to Rotation Matrices

**Claim 11.** *The differentiation of a rotation matrix is expressed as*

$$\dot{\mathbf{R}} = \mathbf{R} \mathbf{S}$$

where  $\mathbf{S}$  is a skew-symmetric matrix.

*Proof.*

$$\mathbf{R}^T \mathbf{R} = \mathbf{I}_3 \quad (\text{A.57})$$

$$\rightarrow \dot{\mathbf{R}}^T \mathbf{R} + \mathbf{R}^T \dot{\mathbf{R}} = \mathbf{0}_{3 \times 3} \quad (\text{A.58})$$

$$\rightarrow \dot{\mathbf{R}}^T \mathbf{R} = -\mathbf{R}^T \dot{\mathbf{R}} = \mathbf{0}_{3 \times 3} \quad (\text{A.59})$$

$$\rightarrow \dot{\mathbf{R}}^T \mathbf{R} = -(\dot{\mathbf{R}}^T \mathbf{R})^T \quad (\text{A.60})$$

Equation A.60 indicates that  $\dot{\mathbf{R}}^T \mathbf{R}$  is a skew symmetric matrix. Let  $\mathbf{S} \triangleq -\dot{\mathbf{R}}^T \mathbf{R}$  and insert to Equation A.58.

$$-\mathbf{S} + \mathbf{R}^T \dot{\mathbf{R}} = \mathbf{0}_{3 \times 3} \quad (\text{A.61})$$

$$\mathbf{R}^T \dot{\mathbf{R}} = \mathbf{S} \quad (\text{A.62})$$

$$\dot{\mathbf{R}} = \mathbf{R} \mathbf{S} \quad (\text{A.63})$$

Q.E.D.

**Claim 12.** A rotation matrix  $\mathbf{R}$  can be expressed as a matrix exponential of a skew symmetric matrix  $\Psi$ . Mathematically,  $\mathbf{R} = \exp_m(\Psi)$

*Proof.* We have shown that  $\dot{\mathbf{R}} = \mathbf{R} \mathbf{S}$  where  $\mathbf{S}$  is a skew symmetric matrix. Then

$$\mathbf{R}_t = \mathbf{R}_0 \exp_m \left( \int_0^t \mathbf{S}_\tau d\tau \right) \quad (\text{A.64})$$

Let  $\mathbf{R}_0 = \mathbf{I}_3$ . Moreover, integration of skew symmetric matrices leads to a new skew symmetric matrix. Recall that skew symmetric matrices constitute a vector space.

$$\mathbf{R}_t = \exp_m \left( \Psi_t \right) \quad (\text{A.65})$$

Q.E.D.

## A.6 Lie Algebra Derivations

### A.6.1 Rodrigues' Formula

In Claim 12, we have seen that every rotation matrix can be written as a matrix exponential of a skew symmetric matrix.

$$\mathbf{R} = \exp_m([\boldsymbol{\psi}]) = \sum_{n=0}^{\infty} \frac{[\boldsymbol{\psi}]^n}{n!} \quad (\text{A.66})$$

However, practically, we cannot evaluate infinite summation whenever we need to evaluate the term  $\exp_m([\boldsymbol{\psi}])$ . Rather, we have a closed-form expression.

**Claim 13.**  $\exp_m([\boldsymbol{\psi}])$  has a closed form solution given as

$$\exp_m([\boldsymbol{\psi}]) = \mathbf{I}_3 + \frac{\sin(\|\boldsymbol{\psi}\|)}{\|\boldsymbol{\psi}\|} [\boldsymbol{\psi}] + \frac{1 - \cos(\|\boldsymbol{\psi}\|)}{\|\boldsymbol{\psi}\|^2} [\boldsymbol{\psi}]^2 \quad (\text{A.67})$$

where  $\|\cdot\|$  is the  $l_2$  norm.

*Proof.* Skew symmetric matrix form of unit vectors has useful properties. Hence, express  $\boldsymbol{\psi}$  in terms of corresponding unit vector and its magnitude.

Let  $\|\boldsymbol{\psi}\| \triangleq \theta$  and  $\mathbf{a} = \frac{\boldsymbol{\psi}}{\|\boldsymbol{\psi}\|} = \frac{\boldsymbol{\psi}}{\theta}$ . Then we have

$$\boldsymbol{\psi} = \theta \mathbf{a} \quad (\text{A.68})$$

Insert (A.68) into (A.67).

$$\exp_m([\boldsymbol{\psi}]) = \sum_{n=0}^{\infty} \frac{[\boldsymbol{\psi}]^n}{n!} = \sum_{n=0}^{\infty} \frac{\theta^n}{n!} [\mathbf{a}]^n \quad (\text{A.69})$$

Recall the Claim 5. Then,

$$\exp_m([\boldsymbol{\psi}]) = \sum_{n=0}^{\infty} \frac{\theta^n}{n!} [\mathbf{a}]^n = \mathbf{I}_3 + \frac{\theta}{1} [\mathbf{a}]^1 + \frac{\theta^2}{2!} [\mathbf{a}]^2 + \frac{\theta^3}{3!} [\mathbf{a}]^3 + \dots \quad (\text{A.70})$$

$$= \mathbf{I}_3 + \left( \frac{\theta}{1} - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} + \dots \right) [\mathbf{a}] + \left( \frac{\theta^2}{2!} - \frac{\theta^4}{4!} + \frac{\theta^6}{6!} + \dots \right) [\mathbf{a}]^2 \quad (\text{A.71})$$

$$\xrightarrow[\text{(A.16, A.17)}]{\text{By}} = \mathbf{I}_3 + \sin(\theta) [\mathbf{a}] + (1 - \cos(\theta)) [\mathbf{a}]^2 \quad (\text{A.72})$$

$$= \mathbf{I}_3 + \frac{\sin(\theta)}{\theta} [\theta \mathbf{a}] + \frac{1 - \cos(\theta)}{\theta^2} [\theta \mathbf{a}]^2 \quad (\text{A.73})$$

$$= \mathbf{I}_3 + \frac{\sin(\|\boldsymbol{\psi}\|)}{\|\boldsymbol{\psi}\|} [\boldsymbol{\psi}] + \frac{1 - \cos(\|\boldsymbol{\psi}\|)}{\|\boldsymbol{\psi}\|^2} [\boldsymbol{\psi}]^2 \quad (\text{A.74})$$

Q.E.D.

## A.6.2 First Order Integration of Rotation Matrix

We have examined the differentiation of a rotation matrix in Claim 11. Now, let us see how to take the integral of a rotation matrix.

$$\int_{t_k}^{t_{k+1}} \mathbf{R}(t) dt = \int_{t_k}^{t_{k+1}} \mathbf{R}[k] \exp_m(\boldsymbol{\omega}_k t) dt = \mathbf{R}[k] \int_{t_k}^{t_{k+1}} \exp_m(\boldsymbol{\omega}_k t) dt \quad (\text{A.75})$$

where we assume constant angular velocity for the time interval  $[t_k, t_{k+1})$ . Also, we have closed form expression for the exponential map from Claim 13.

$$\begin{aligned} \exp_m([\boldsymbol{\omega}_k] \tau) &= \mathbf{I}_3 + \frac{\sin(\|\boldsymbol{\omega}_k \tau\|)}{\|\boldsymbol{\omega}_k \tau\|} [\boldsymbol{\omega}_k \tau] + \frac{1 - \cos(\|\boldsymbol{\omega}_k \tau\|)}{\|\boldsymbol{\omega}_k \tau\|^2} [\boldsymbol{\omega}_k \tau]^2 \\ &= \mathbf{I}_3 + \frac{\sin(\|\boldsymbol{\omega}_k\| \tau)}{\|\boldsymbol{\omega}_k\|} [\boldsymbol{\omega}_k] + \frac{1 - \cos(\|\boldsymbol{\omega}_k\| \tau)}{\|\boldsymbol{\omega}_k\|^2} [\boldsymbol{\omega}_k]^2 \end{aligned}$$

We can now evaluate the integral term.

$$\begin{aligned} &\int_0^{\Delta t} \left( \mathbf{I}_3 + \frac{\sin(\|\boldsymbol{\omega}_k\| \tau)}{\|\boldsymbol{\omega}_k\|} [\boldsymbol{\omega}_k] + \frac{1 - \cos(\|\boldsymbol{\omega}_k\| \tau)}{\|\boldsymbol{\omega}_k\|^2} [\boldsymbol{\omega}_k]^2 \right) d\tau \\ &= \left( \mathbf{I}_3 + \frac{-\cos(\|\boldsymbol{\omega}_k\| \tau)}{\|\boldsymbol{\omega}_k\|^2} [\boldsymbol{\omega}_k] + \frac{\|\boldsymbol{\omega}_k\| \tau + \sin(\|\boldsymbol{\omega}_k\| \tau)}{\|\boldsymbol{\omega}_k\|^3} [\boldsymbol{\omega}_k]^2 \right) \Bigg|_{\tau=0}^{\Delta t} \\ &= \mathbf{I} \Delta t + \frac{1 - \cos(\|\boldsymbol{\omega}_k\| \Delta t)}{\|\boldsymbol{\omega}_k\|^2} [\boldsymbol{\omega}_k] + \frac{\|\boldsymbol{\omega}_k\| \Delta t + \sin(\|\boldsymbol{\omega}_k\| \Delta t)}{\|\boldsymbol{\omega}_k\|^3} [\boldsymbol{\omega}_k]^2 \end{aligned}$$

### A.6.3 Adjoint Matrix of $SE(3)$

**Lemma 1.**  $\mathbf{R} \mathcal{J}_{l_{SO(3)}}(\boldsymbol{\psi}) = \mathcal{J}_{l_{SO(3)}}(\mathbf{R} \boldsymbol{\psi}) \mathbf{R}$

*Proof.*

$$\mathbf{R} \mathcal{J}_{l_{SO(3)}}(\boldsymbol{\psi}) = \mathbf{R} \sum_{n=0}^{\infty} \frac{[\boldsymbol{\psi}]^n}{(n+1)!} = \sum_{n=0}^{\infty} \frac{\mathbf{R} [\boldsymbol{\psi}]^n}{(n+1)!} \quad (\text{A.76})$$

$$= \sum_{n=0}^{\infty} \frac{\mathbf{R} [\boldsymbol{\psi}] [\boldsymbol{\psi}]^{n-1}}{(n+1)!} \quad (\text{A.77})$$

$$\xrightarrow[\text{A.3}]{\text{By Equation}} = \sum_{n=0}^{\infty} \frac{[\mathbf{R} \boldsymbol{\psi}] \mathbf{R} [\boldsymbol{\psi}]^{n-1}}{(n+1)!} \quad (\text{A.78})$$

$$\xrightarrow[\text{A.3}]{\text{By Equation}} = \sum_{n=0}^{\infty} \frac{[\mathbf{R} \boldsymbol{\psi}]^2 \mathbf{R} [\boldsymbol{\psi}]^{n-2}}{(n+1)!} \quad (\text{A.79})$$

$$= \sum_{n=0}^{\infty} \frac{[\mathbf{R} \boldsymbol{\psi}]^n \mathbf{R}}{(n+1)!} = \sum_{n=0}^{\infty} \frac{[\mathbf{R} \boldsymbol{\psi}]^n}{(n+1)!} \mathbf{R} \quad (\text{A.80})$$

$$= \mathcal{J}_{l_{SO(3)}}(\mathbf{R} \boldsymbol{\psi}) \mathbf{R} \quad (\text{A.81})$$

Q.E.D.

**Lemma 2.**  $\mathbf{I}_3 - \text{Exp}_{SO(3)}(\boldsymbol{\psi}) = -\mathcal{J}_{l_{SO(3)}}(\boldsymbol{\psi}) [\boldsymbol{\psi}]$ , where  $\boldsymbol{\psi} \in \mathfrak{so}(3)$

*Proof.*

$$\begin{aligned} \mathbf{I}_3 - \text{Exp}_{SO(3)}(\boldsymbol{\psi}) &= \mathbf{I}_3 - \sum_{n=0}^{\infty} \frac{[\boldsymbol{\psi}]^n}{n!} \\ &= \mathbf{I}_3 - \mathbf{I}_3 - \sum_{n=1}^{\infty} \frac{[\boldsymbol{\psi}]^n}{n!} \\ &= - \sum_{n=1}^{\infty} \frac{[\boldsymbol{\psi}]^n}{n!} = - \sum_{n=1}^{\infty} \frac{[\boldsymbol{\psi}]^{n-1}}{n!} [\boldsymbol{\psi}] \\ &\xrightarrow[\text{Let } n=m+1]{\text{Let}} = - \sum_{m=0}^{\infty} \frac{[\boldsymbol{\psi}]^m}{(m+1)!} [\boldsymbol{\psi}] \\ &= -\mathcal{J}_{l_{SO(3)}}(\boldsymbol{\psi}) [\boldsymbol{\psi}] \end{aligned}$$

Q.E.D.

**Claim 14.** *The adjoint matrix for  $SE(3)$  is expressed as follows.*

$$\text{Ad}_{\mathbf{T}} = \begin{bmatrix} \mathbf{R} & \mathbf{0}_{3 \times 3} \\ [\mathbf{t}]_{\mathbf{R}} & \mathbf{R} \end{bmatrix} \quad \text{where} \quad \mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (\text{A.82})$$

*Proof.* We will use the definition of the adjoint matrix for the proof.

$$\begin{aligned} \text{Exp}_{SE(3)}(\text{Ad}_{\mathbf{T}} \boldsymbol{\xi}) &= \mathbf{T} \text{Exp}_{SE(3)}(\boldsymbol{\xi}) \mathbf{T}^{-1} \\ &= \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} \text{Exp}_{SO(3)}(\boldsymbol{\xi}_{\psi}) & \mathcal{J}_{l_{SO(3)}}(\boldsymbol{\xi}_{\psi}) \boldsymbol{\xi}_{\rho} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T \mathbf{t} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R} \text{Exp}_{SO(3)}(\boldsymbol{\xi}_{\psi}) & \mathbf{R} \mathcal{J}_{l_{SO(3)}}(\boldsymbol{\xi}_{\psi}) \boldsymbol{\xi}_{\rho} + \mathbf{t} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T \mathbf{t} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R} \text{Exp}_{SO(3)}(\boldsymbol{\xi}_{\psi}) \mathbf{R}^T & -\mathbf{R} \text{Exp}_{SO(3)}(\boldsymbol{\xi}_{\psi}) \mathbf{R}^T \mathbf{t} + \mathbf{R} \mathcal{J}_{l_{SO(3)}}(\boldsymbol{\xi}_{\psi}) \boldsymbol{\xi}_{\rho} + \mathbf{t} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \\ &\xrightarrow[2]{\text{By Claim}} \begin{bmatrix} \text{Exp}_{SO(3)}(\mathbf{R} \boldsymbol{\xi}_{\psi}) & -\text{Exp}_{SO(3)}(\mathbf{R} \boldsymbol{\xi}_{\psi}) \mathbf{t} + \mathbf{R} \mathcal{J}_{l_{SO(3)}}(\boldsymbol{\xi}_{\psi}) \boldsymbol{\xi}_{\rho} + \mathbf{t} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \\ &\xrightarrow[1]{\text{By Lemma}} \begin{bmatrix} \text{Exp}_{SO(3)}(\mathbf{R} \boldsymbol{\xi}_{\psi}) & -\text{Exp}_{SO(3)}(\mathbf{R} \boldsymbol{\xi}_{\psi}) \mathbf{t} + \mathcal{J}_{l_{SO(3)}}(\mathbf{R} \boldsymbol{\xi}_{\psi}) \mathbf{R} \boldsymbol{\xi}_{\rho} + \mathbf{t} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \\ &= \begin{bmatrix} \text{Exp}_{SO(3)}(\mathbf{R} \boldsymbol{\xi}_{\psi}) & (\mathbf{I}_3 - \text{Exp}_{SO(3)}(\mathbf{R} \boldsymbol{\xi}_{\psi})) \mathbf{t} + \mathcal{J}_{l_{SO(3)}}(\mathbf{R} \boldsymbol{\xi}_{\psi}) \mathbf{R} \boldsymbol{\xi}_{\rho} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \\ &\xrightarrow[2]{\text{By Lemma}} \begin{bmatrix} \text{Exp}_{SO(3)}(\mathbf{R} \boldsymbol{\xi}_{\psi}) & -\mathcal{J}_{l_{SO(3)}}(\mathbf{R} \boldsymbol{\xi}_{\psi}) [\mathbf{R} \boldsymbol{\xi}_{\psi}] \mathbf{t} + \mathcal{J}_{l_{SO(3)}}(\mathbf{R} \boldsymbol{\xi}_{\psi}) \mathbf{R} \boldsymbol{\xi}_{\rho} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \\ &\xrightarrow[\text{A.4}]{\text{BY Equation}} \begin{bmatrix} \text{Exp}_{SO(3)}(\mathbf{R} \boldsymbol{\xi}_{\psi}) & \mathcal{J}_{l_{SO(3)}}(\mathbf{R} \boldsymbol{\xi}_{\psi}) [\mathbf{t}]_{\mathbf{R}} \boldsymbol{\xi}_{\psi} + \mathcal{J}_{l_{SO(3)}}(\mathbf{R} \boldsymbol{\xi}_{\psi}) \mathbf{R} \boldsymbol{\xi}_{\rho} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \\ &= \begin{bmatrix} \text{Exp}_{SO(3)}(\mathbf{R} \boldsymbol{\xi}_{\psi}) & \mathcal{J}_{l_{SO(3)}}(\mathbf{R} \boldsymbol{\xi}_{\psi}) ([\mathbf{t}]_{\mathbf{R}} \boldsymbol{\xi}_{\psi} + \mathbf{R} \boldsymbol{\xi}_{\rho}) \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \end{aligned}$$

In summary, we have

$$\text{Exp}_{SE(3)}(\text{Ad}_{\mathbf{T}} \boldsymbol{\xi}) = \begin{bmatrix} \text{Exp}_{SO(3)}(\mathbf{R} \boldsymbol{\xi}_{\psi}) & \mathcal{J}_{l_{SO(3)}}(\mathbf{R} \boldsymbol{\xi}_{\psi}) ([\mathbf{t}]_{\mathbf{R}} \boldsymbol{\xi}_{\psi} + \mathbf{R} \boldsymbol{\xi}_{\rho}) \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}$$

It is trivial to show that

$$\begin{aligned}\text{Ad}_{\mathbf{T}} \boldsymbol{\xi} &= \begin{bmatrix} \mathbf{R} \boldsymbol{\xi}_\psi \\ [\mathbf{t}] \mathbf{R} \boldsymbol{\xi}_\psi + \mathbf{R} \boldsymbol{\xi}_\rho \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R} & \mathbf{0}_{3 \times 3} \\ [\mathbf{t}] \mathbf{R} & \mathbf{R} \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi}_\psi \\ \boldsymbol{\xi}_\rho \end{bmatrix}\end{aligned}$$

Q.E.D.

## A.7 IMU Dynamics

### A.7.1 Properties of $\Phi_t(\mathcal{T})$

$\Phi_t(\mathcal{T}) : SE_2(3) \rightarrow SE_2(3)$  is defined as follows

$$\Phi_t(\mathcal{T}) \triangleq \left[ \begin{array}{c|cc} \mathbf{R} & \mathbf{t} + \mathbf{v}t & \mathbf{v} \\ \hline \mathbf{0}_{2 \times 3} & \mathbf{I}_2 & \end{array} \right] \quad (\text{A.83})$$

**Claim 15.** *We have the following relation*

$$\Phi_t(\text{Exp}(\boldsymbol{\zeta})) = \text{Exp}(\mathbf{F} \boldsymbol{\zeta}) \quad \text{where} \quad \mathbf{F} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{0 \times 3} & \mathbf{0}_{0 \times 3} \\ \mathbf{0}_{0 \times 3} & \mathbf{I}_3 & \mathbf{I}_3 t \\ \mathbf{0}_{0 \times 3} & \mathbf{0}_{0 \times 3} & \mathbf{I}_3 \end{bmatrix} \quad (\text{A.84})$$

*Proof.* Note that

$$\boldsymbol{\zeta} = \begin{bmatrix} \boldsymbol{\zeta}_\psi \\ \boldsymbol{\zeta}_\rho \\ \boldsymbol{\zeta}_\nu \end{bmatrix} \quad \mathbf{F} \boldsymbol{\zeta} = \begin{bmatrix} \boldsymbol{\zeta}_\psi \\ \boldsymbol{\zeta}_\rho + t \boldsymbol{\zeta}_\nu \\ \boldsymbol{\zeta}_\nu \end{bmatrix} \quad (\text{A.85})$$

Recall the definition of capitalized exponential matrix for  $SE_2(3)$ .

$$\text{Exp}(\boldsymbol{\zeta}) = \left[ \begin{array}{c|cc} \text{Exp}_{SO(3)}(\boldsymbol{\zeta}_\psi) & \mathcal{J}_{SO(3)}(\boldsymbol{\zeta}_\psi) \boldsymbol{\zeta}_\rho & \mathcal{J}_{SO(3)}(\boldsymbol{\zeta}_\psi) \boldsymbol{\zeta}_\nu \\ \hline \mathbf{0}_{2 \times 3} & \mathbf{I}_2 & \end{array} \right] \quad (\text{A.86})$$

$$= \left[ \begin{array}{c|cc} \mathbf{R}_\zeta & \mathbf{t}_\zeta & \mathbf{v}_\zeta \\ \hline \mathbf{0}_{2 \times 3} & \mathbf{I}_2 & \end{array} \right] \quad (\text{A.87})$$



Then we have

$$\text{Exp}(\mathbf{F} \zeta) = \left[ \begin{array}{c|c} \text{Exp}_{SO(3)}(\zeta_\psi) & \mathcal{J}_{SO(3)}(\zeta_\psi) (\zeta_\rho + t \zeta_\nu) \quad \mathcal{J}_{SO(3)}(\zeta_\psi) \zeta_\nu \\ \hline \mathbf{0}_{2 \times 3} & \mathbf{I}_2 \end{array} \right] \quad (\text{A.88})$$

$$= \left[ \begin{array}{c|c} \mathbf{R}_\zeta & \mathbf{t}_\zeta + \mathbf{v}_\zeta t \quad \mathbf{v}_\zeta \\ \hline \mathbf{0}_{2 \times 3} & \mathbf{I}_2 \end{array} \right] \quad (\text{A.89})$$

$$= \Phi_t(\text{Exp}(\zeta)) \quad (\text{A.90})$$

Q.E.D.

**Claim 16.**

$$\Phi_t(\mathcal{T}_1 \mathcal{T}_2) = \Phi_t(\mathcal{T}_1) \Phi_t(\mathcal{T}_2) \quad (\text{A.91})$$

*Proof.*

$$\mathcal{T}_1 \mathcal{T}_2 = \left[ \begin{array}{c|c} \mathbf{R}_1 & \mathbf{t}_1 \quad \mathbf{v}_1 \\ \hline \mathbf{0}_{2 \times 3} & \mathbf{I}_2 \end{array} \right] \left[ \begin{array}{c|c} \mathbf{R}_2 & \mathbf{t}_2 \quad \mathbf{v}_2 \\ \hline \mathbf{0}_{2 \times 3} & \mathbf{I}_2 \end{array} \right] \quad (\text{A.92})$$

$$= \left[ \begin{array}{c|c} \mathbf{R}_1 \mathbf{R}_2 & \mathbf{R}_1 \mathbf{t}_2 + \mathbf{t}_1 \quad \mathbf{R}_1 \mathbf{v}_2 + \mathbf{v}_1 \\ \hline \mathbf{0}_{2 \times 3} & \mathbf{I}_2 \end{array} \right] \quad (\text{A.93})$$

$$\rightarrow \Phi_t(\mathcal{T}_1 \mathcal{T}_2) = \left[ \begin{array}{c|c} \mathbf{R}_1 \mathbf{R}_2 & \mathbf{R}_1 \mathbf{t}_2 + \mathbf{t}_1 + t(\mathbf{R}_1 \mathbf{v}_2 + \mathbf{v}_1) \quad \mathbf{R}_1 \mathbf{v}_2 + \mathbf{v}_1 \\ \hline \mathbf{0}_{2 \times 3} & \mathbf{I}_2 \end{array} \right] \quad (\text{A.94})$$

$$= \left[ \begin{array}{c|c} \mathbf{R}_1 \mathbf{R}_2 & \mathbf{R}_1(\mathbf{t}_2 + t \mathbf{v}_2) + \mathbf{t}_1 + t \mathbf{v}_1 \quad \mathbf{R}_1 \mathbf{v}_2 + \mathbf{v}_1 \\ \hline \mathbf{0}_{2 \times 3} & \mathbf{I}_2 \end{array} \right] \quad (\text{A.95})$$

$$= \left[ \begin{array}{c|c} \mathbf{R}_1 & \mathbf{t}_1 + t \mathbf{v}_1 \quad \mathbf{v}_1 \\ \hline \mathbf{0}_{2 \times 3} & \mathbf{I}_2 \end{array} \right] \left[ \begin{array}{c|c} \mathbf{R}_2 & \mathbf{t}_2 + t \mathbf{v}_2 \quad \mathbf{v}_2 \\ \hline \mathbf{0}_{2 \times 3} & \mathbf{I}_2 \end{array} \right] \quad (\text{A.96})$$

$$= \Phi_t(\mathcal{T}_1) \Phi_t(\mathcal{T}_2) \quad (\text{A.97})$$

Q.E.D.



## Appendix B

### APPENDIX FOR VISUAL SLAM

#### B.1 Optical Flow

Optical flow is a critical concept in visual SLAM. In this section, I explain and visualize the concept to provide better insight.

Consider two successive frames, as shown in Figures B.1a and B.1b. Upon close inspection, you may notice that some pixels shift slightly between the frames.

I combine Figures B.1a and B.1b by applying a weighted sum. Then, I represent the motion of pixels through vectors, as shown in Figure B.1c. The saturation of the vector colors indicates the magnitude, while the color hue represents the direction of the vectors.

By visualizing the optical flow for every pixel, we obtain Figure B.1d. Upon close inspection, you will notice that objects closer to the camera exhibit a higher optical flow magnitude.

It is important to note that optical flow can also be computed for a stereo image pair. Consider two cameras positioned side by side. The resulting optical flow, often referred to as disparity, is visualized in Figure B.2.

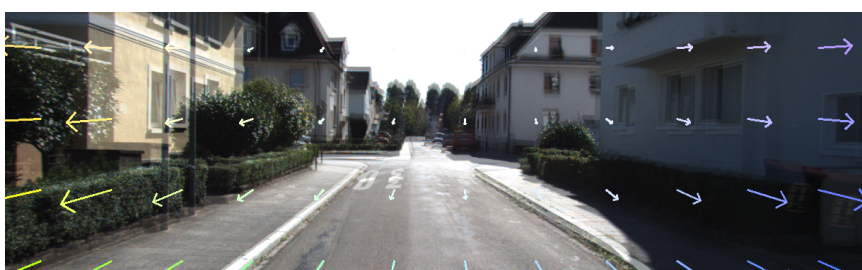
Figures B.1 and B.2 demonstrate that optical flow is influenced by both the structure of the scene and the motion of the camera.



(a) Frame 320



(b) Frame 321



(c) Sparse Optical Flow Vectors



(d) Dense Optical Flow Vectors

Figure B.1: Visualization of Optical Flow



(a) Left Frame 320



(b) Right Frame 320



(c) Sparse Optical Flow Vectors



(d) Dense Optical Flow Vectors

Figure B.2: Disparity for Stereo Frames

## B.2 Depth Map

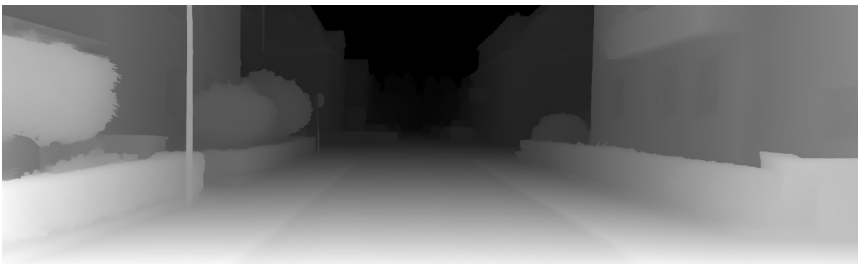
Depth map is another important concept in visual SLAM. The depth of a pixel represents the distance between the observed object and the camera frame. When the depth is computed for each pixel in the image, the result is a depth map.

It is also a common practice to store the inverse depth  $d^{-1}$  rather than the depth  $d$ .

Figure B.3 illustrates an example of an *inverse depth map*, where each pixel's value represents the inverse of the relative distance ( $d^{-1}$ ) to the corresponding object in the scene. Brighter colors indicate a higher  $d^{-1}$ , meaning the actual depth ( $d$ ) is lower.



(a) Left Frame 320



(b) Depth Map

Figure B.3: Inverse Depth Map is Visualized

## CURRICULUM VITAE

### PERSONAL INFORMATION

**Surname, Name:** YALÇIN, Haktan

**Nationality:** Turkish (TC)

### EDUCATION

<b>Degree</b>	<b>Institution</b>	<b>Year of Graduation</b>
B.S.	METU Electrical Electronics	2022
High School	Erzurum Science High Schools	2017

### PROFESSIONAL EXPERIENCE

<b>Year</b>	<b>Place</b>	<b>Enrollment</b>
2	Odtü Görüntü Analiz Merkezi	Bilimsel Araştırma Uzmanı

### PUBLICATIONS

#### International Conference Publications

None