Axis Mapping

The Estimation of Surface Orientations and its Applications in Vehicle Localization and Structural Geology

by

Marc J. Gallant

A thesis submitted to the graduate program in Electrical & Computer Engineering in conformity with the requirements for the degree of Doctor of Philosophy

Queen’s University
Kingston, Ontario, Canada
September 2016

Copyright © Marc J. Gallant, 2016
“Every revolutionary idea seems to evoke three stages of reaction. They may be summed up by the phrases: (1) It’s completely impossible. (2) It’s possible, but it’s not worth doing. (3) I said it was a good idea all along.”

— Arthur C. Clarke
Abstract

The map representation of an environment should be selected based on its intended application. For example, a geometrically accurate map describing the Euclidean space of an environment is not necessarily the best choice if only a small subset its features are required. One possible subset is the orientations of the flat surfaces in the environment, represented by a special parameterization of normal vectors called axes. Devoid of positional information, the entries of an axis map form a non-injective relationship with the flat surfaces in the environment, which results in physically distinct flat surfaces being represented by a single axis. This drastically reduces the complexity of the map, but retains important information about the environment that can be used in meaningful applications in both two and three dimensions.

This thesis presents axis mapping, which is an algorithm that accurately and automatically estimates an axis map of an environment based on sensor measurements collected by a mobile platform. Furthermore, two major applications of axis maps are developed and implemented. First, the LiDAR compass is a heading estimation algorithm that compares measurements of axes with an axis map of the environment. Pairing the LiDAR compass with simple translation measurements forms the basis for an accurate two-dimensional localization algorithm. It is shown that this algorithm eliminates the growth of heading error in both indoor and outdoor environments, resulting in accurate localization over long distances. Second, in the context of geotechnical engineering, a three-dimensional axis map is called a stereonet, which is used as
a tool to examine the strength and stability of a rock face. Axis mapping provides a novel approach to create accurate stereonets safely, rapidly, and inexpensively compared to established methods. The non-injective property of axis maps is leveraged to probabilistically describe the relationships between non-sequential measurements of the rock face. The automatic estimation of stereonets was tested in three separate outdoor environments. It is shown that axis mapping can accurately estimate stereonets while improving safety, requiring significantly less time and effort, and lowering costs compared to traditional and current state-of-the-art approaches.
Acknowledgments

Research and development is a team sport, and the body of work presented in this thesis is no exception. First and foremost, I would like to extend my sincerest gratitude to my supervisor, Joshua Marshall, for his unwavering support that started before I considered even a single idea in this thesis and I am sure will continue in the future when I inevitably bother him for advice. Josh’s competency, enthusiasm, and flexibility allowed me to always choose the way forward, even if it meant choosing my ideas over his. I am also indebted to the accommodations he made for me on a personal level, for which my family is also grateful.

I would also like to thank my colleagues in the Mining Systems Laboratory, whose knowledge on a broad range of topics proved to be a useful resource that I frequently and unabashedly used. In particular, thank you to Rob Hewitt and Brian Lynch for fruitful whiteboard discussions, and to Heshan Fernando for helping design some of the hardware I used in my experiments. I would also like to thank James Procopio and Glen Turner for helping with the mundane task of manual data collection, Steve McKinnon for providing consultation on geotechnical matters, and Matthew Lato for permission to use some of his field data. Thanks as well to my parents, Maurice and Donna, for their continued encouragement of my seemingly never-ending education. Lastly, I am forever thankful to my wife, Natalie MacDonald, for her enduring support that saw me through late nights, early mornings, deadlines, and dirty diapers, while still having time to share with me the joy of our beautiful baby boy, Malcolm.
Contents

Abstract i
Acknowledgments iii
List of Tables viii
List of Figures x
List of Symbols xi

Chapter 1: Introduction 1
  1.1 Organization of this Thesis ...................................... 6
  1.2 Publications ....................................................... 7

I Theory 9

Chapter 2: Axes 10
  2.1 Related Work ..................................................... 11
  2.2 Two-Dimensional Unit Axes .................................... 12
  2.3 Three-Dimensional Unit Axes .................................. 16
  2.4 Manifold Encapsulation of Axes ............................... 22
    2.4.1 Two-Dimensional Axes ..................................... 23
    2.4.2 Three-Dimensional Axes ................................... 24
    2.4.3 Axioms of Manifold Operators ......................... 27
  2.5 Random Axes ..................................................... 27

Chapter 3: Rotations 30
  3.1 Related Work ..................................................... 31
  3.2 Two-Dimensional Unit Rotations ............................... 32
  3.3 Unit Quaternions ................................................ 33
3.4 Manifold Encapsulation of Rotations .............................................. 36
  3.4.1 Two-Dimensional Rotations ................................................. 36
  3.4.2 Three-Dimensional Rotations .............................................. 37
3.5 Random Rotations ................................................................. 39

Chapter 4: Axis Mapping ................................................................. 40
  4.1 Related Work ............................................................................... 42
    4.1.1 Line Mapping ........................................................................ 42
    4.1.2 Plane Mapping ...................................................................... 43
    4.1.3 Regression Analysis on Directional Data ................................. 44
  4.2 Problem Description .................................................................... 45
  4.3 Building the Objective Function .................................................. 49
    4.3.1 Log-Likelihood of a Gaussian Random Variable ....................... 49
    4.3.2 Log-Likelihood of Axis Mapping Observations ......................... 50
  4.4 Maximizing the Objective Function .............................................. 51
    4.4.1 Linearizing the Axis Mapping Observation Errors ..................... 53
    4.4.2 Maximizing the Axis Mapping Objective Function ..................... 57
  4.5 Pseudocode ................................................................................ 61

II Applications ...................................................................................... 63

Chapter 5: LiDAR Compass ................................................................. 64
  5.1 Related Work ............................................................................... 65
  5.2 Problem Description .................................................................... 68
  5.3 Building the $A$ Priori Axis Map .................................................... 70
    5.3.1 Obtaining the Axis Mapping Observations ................................. 71
    5.3.2 Incremental Optimal Estimation of the Design Parameter .......... 73
  5.4 LiDAR Compass Localization ....................................................... 78
    5.4.1 Motion Prediction .................................................................... 78
    5.4.2 Motion Correction with the $A$ Priori Axis Map ......................... 81
    5.4.3 Motion Correction with the Local Axis Map ............................... 83
    5.4.4 Adding and Removing Entries of the Local Axis Map ................. 84
    5.4.5 Merging Entries of the Local Axis Map ..................................... 86
  5.5 Pseudocode ................................................................................ 87
  5.6 Experiments .................................................................................. 88
    5.6.1 Simulation .............................................................................. 88
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.1.2</td>
<td>Tricycle</td>
<td>154</td>
</tr>
<tr>
<td>B.2</td>
<td>Gyroscopes</td>
<td>155</td>
</tr>
<tr>
<td>B.2.1</td>
<td>Calibration</td>
<td>156</td>
</tr>
<tr>
<td>B.3</td>
<td>LiDAR</td>
<td>156</td>
</tr>
</tbody>
</table>

**Appendix C: Data Association for LiDAR Compass Localization**
- C.1 Compatibility | 159
- C.2 Joint Compatibility | 160

**Appendix D: Processing Three-Dimensional Sensor Measurements**
- D.1 Gyroscope | 163
- D.2 Accelerometer | 164
  - D.2.1 Detecting Non-Gravitational Acceleration | 166
- D.3 Magnetometer | 167
  - D.3.1 Calibration | 169
- D.4 LiDAR | 169

**Appendix E: Manifold Encapsulation of Three-Dimensional Directions** | 172
List of Tables

4.1 Axis mapping variables ................................................. 46
4.2 Axis mapping sensor configurations ................................. 47
5.1 Summary of the BMH results ........................................... 98
5.2 Summary of the Queen’s University campus results ............. 102
6.1 Processed 3DAM sensor measurements ............................. 110
6.2 Specifications of the Microsoft Kinect v2 ......................... 121
6.3 Summary of the Brewer Lake results ............................... 128
6.4 Summary of the Ivy Lea results ..................................... 130
6.5 Summary of the Myra-Bellevue results ............................. 133
C.1 Possible associations in joint compatibility example ............ 161
List of Figures

1.1 Axes of two-dimensional line segments ................................ 2
1.2 Four examples of rock faces ............................................. 4
1.3 Discontinuities in a rock face ........................................... 5

2.1 Orientation of a line represented by a two-dimensional axis .......... 13
2.2 Two-dimensional unit axis ................................................. 13
2.3 Orientation of a plane represented by a three-dimensional axis ...... 17
2.4 Three-dimensional unit axis .............................................. 18
2.5 The axis product ............................................................ 22
2.6 A geometric representation of two-dimensional axis vector space . 23
2.7 A geometric representation of three-dimensional axis vector space . 26
2.8 Representing the uncertainty of an axis with a Gaussian distribution 29

3.1 Heading of a vehicle represented by a two-dimensional rotation .... 33
3.2 Orientation of a vehicle represented by a three-dimensional rotation 34
3.3 A geometric representation of two-dimensional rotation vector space 37
3.4 A geometric representation of three-dimensional rotation vector space 38

4.1 Overview of axis mapping .................................................. 41
4.2 Axis mapping in two and three dimensions. .............................. 46
4.3 Axis mapping factor graph configurations ................................ 48

5.1 Components estimated by LiDAR compass localization ............... 69
5.2 Simple determination of axis maps from environments ................. 71
5.3 Importance of initial estimate of design parameter ..................... 74
5.4 Outline of the 2DAM algorithm for obtaining the a priori axis map . 75
5.5 Data association and estimation of the axis map ....................... 77
5.6 LiDAR compass localization .............................................. 79
5.7 Two-dimensional simulated environment ................................ 88
5.8 Taylor-Dunn SS-534 electric vehicle ................................... 91
5.9 Axis maps of the simulated environment ................................ 94
## List of Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Axis map</td>
</tr>
<tr>
<td>$a^T$</td>
<td>Transpose of $a$</td>
</tr>
<tr>
<td>$a^{-1}$</td>
<td>Inverse of $a$</td>
</tr>
<tr>
<td>$a^+$</td>
<td>(Left-hand) compound operator acting on $a$</td>
</tr>
<tr>
<td>$a^-$</td>
<td>(Right-hand) compound operator acting on $a$</td>
</tr>
<tr>
<td>$a^\oplus$</td>
<td>Inverse (left-hand) compound operator acting on $a$</td>
</tr>
<tr>
<td>$a^\ominus$</td>
<td>Inverse (right-hand) compound operator acting on $a$</td>
</tr>
<tr>
<td>$a\times$</td>
<td>Cross product matrix of $a$</td>
</tr>
<tr>
<td>$\hat{a}$</td>
<td>Estimate of $a$</td>
</tr>
<tr>
<td>$\hat{a}^*$</td>
<td>Optimal estimate of $a$</td>
</tr>
<tr>
<td>$|a|$</td>
<td>Vector norm of $a$</td>
</tr>
<tr>
<td>$C(q)$</td>
<td>Rotation matrix parameterization of unit quaternion $q$</td>
</tr>
<tr>
<td>$\text{Tr}(C)$</td>
<td>Trace of square matrix $C$</td>
</tr>
<tr>
<td>$e_{q_i,q_j}$</td>
<td>Rotation vector error between observed and expected rotation from $q_i$ to $q_j$</td>
</tr>
<tr>
<td>$e_{q_i}$</td>
<td>Rotation vector error between observed and expected unit quaternion $q_i$</td>
</tr>
<tr>
<td>$e_{q_i,m_j}$</td>
<td>Axis vector error between observed and expected unit axis $m_j$, observed at $q_i$</td>
</tr>
<tr>
<td>$\mathcal{F}_g$</td>
<td>Global coordinate frame</td>
</tr>
<tr>
<td>$\mathcal{F}_m$</td>
<td>Mobile platform coordinate frame</td>
</tr>
<tr>
<td>$\mathcal{F}_v$</td>
<td>Vehicle coordinate frame</td>
</tr>
<tr>
<td>$h^1$</td>
<td>Topological space of two-dimensional axis vectors</td>
</tr>
<tr>
<td>$h^2$</td>
<td>Topological space of three-dimensional axis vectors</td>
</tr>
<tr>
<td>$h^3$</td>
<td>Topological space of three-dimensional rotation vectors</td>
</tr>
<tr>
<td>$I_n$</td>
<td>Identity matrix of dimension $n$</td>
</tr>
<tr>
<td>$\iota$</td>
<td>Identity unit quaternion</td>
</tr>
<tr>
<td>$o$</td>
<td>Identity unit axis</td>
</tr>
<tr>
<td>$Q$</td>
<td>Orientation path</td>
</tr>
<tr>
<td>$\mathcal{R}$</td>
<td>Topological space of unconstrained axis mapping design parameter</td>
</tr>
<tr>
<td>$\mathcal{S}$</td>
<td>Topological space of axis mapping design parameter</td>
</tr>
</tbody>
</table>
$s^3$ Topological space of two-dimensional rotation vectors

Manifold addition operator (“box plus”)

Manifold subtraction operator (“box minus”)

$z_{q_i}$ Observation of unit quaternion $q_i$

$z_{q_i,m_j}$ Observation of unit axis $m_j$ from unit quaternion orientation $q_i$

$z_{q_i,q_j}$ Observation of rotation between unit quaternions $q_i$ and $q_j$
Chapter 1

Introduction

The map representation of an environment should be compatible with the tools and methodology that are available to make the map, while at the same time adequately represent the environment in a way that is suitable for its intended application. This is particularly true for automated mapping, where the types of sensors and methods of estimation may have limitations. For example, a set of straight line segments describing the walls and other surfaces inside a building can provide a human interpretable two-dimensional map of the layout of the building, while at the same time being relatively simple to create with a small robotic vehicle equipped with a laser scanner. However, if the purpose of this map is to calculate the areas of individual rooms, limiting the map representation to only straight line segments introduces an approximation in the areas of the environment that are curved. Furthermore, a less obvious problem is that constructing a global map of the entire building is unnecessary, and will actually introduce uncertainty that can be mitigated if instead the map consisted of local submaps of individual rooms. This misapplication of map representation is often a result of forcing the use of established tools and methods without consideration for the subtleties of the application.

This thesis introduces a map representation called an axis map that consists of a
set of the axes of flat surfaces in the environment. Formally, an axis is an unordered pair of opposing directions. For example, the orientation of a highway is well described by an axis, where in this case the unordered pair of opposing directions is the two opposing directions of travel. Another example is the orientation of a line (in two dimensions), or a plane (in three dimensions), such as a flat wall in a building. A natural way to parameterize the orientation of a wall is with a normal vector, whose opposing direction represents the same orientation. This latter example describes the entries of an axis map, which contains the orientations of flat surfaces in the environment (lines or planes, depending on the dimensionality), parameterized as a set of axes. It is important to note that axis maps contain no positional information about the environment. As a result, physically distinct but similarly oriented flat surfaces are represented by a single entry in the axis map. Put differently, the relationship between the flat surfaces in the environment and the entries of an axis map is not injective, which is illustrated in Figure 1.1. It is this property of axis maps in particular that make them a suitable map representation for the specialized applications explored in this thesis.

In two dimensions, axis maps of the orientations of the dominant, reoccurring flat surfaces in an environment (e.g., walls) can provide information that is useful for determining the direction of travel of a person or vehicle. For example, if the
walls inside a rectangular room are known to face North/South and East/West, this information is useful in determining one’s orientation in the global coordinate frame. Similarly, if a city is laid out in a grid pattern, the orientations of the exterior walls of the buildings can be helpful in determining the heading of a vehicle. Because of the lack of positional information in axis maps, they can usually be kept succinct for physically large environments (e.g., a two-entry axis map is sufficient to describe the axes in a city grid). In less structured environments, this idea of using the axes of flat surfaces to help determine the orientation of a moving body can still be applied if an accurate axis map can be created. This is one application of axis mapping that is explored in this thesis. More specifically, the two-step process of creating an axis map of the environment and then using it to estimate the heading of a moving vehicle is explored in detail. In this capacity, the mapping algorithm is called two-dimensional axis mapping, and the use of a LiDAR sensor to extract axes in the environment and compare them with the axis map for heading estimation is called the LiDAR compass.

In three dimensions, axis mapping has an important application in geology and geotechnical engineering, where axis maps provide information about the properties of rock masses. A rock mass is any volume of rock embedded in the earth whose exposed areas are called rock faces, examples of which are shown in Figure 1.2. A safe, efficient, and accurate method of measuring the properties of rock masses is critical in many engineering and geological applications. These applications include ensuring a stable foundation for civil engineering projects (e.g., constructing highways, buildings, and bridges), building safe and efficient mines (e.g., tunnel excavation or rock wall maintenance), and geological surveys (e.g., mapping). Rock masses have highly anisotropic properties due to the existence of discontinuities (i.e., fractures) caused by tectonic activity, heating and cooling events, or sudden changes in stress. The properties of these discontinuities are important because they largely determine the mechanical behaviour (i.e., stress and displacement) of the rock mass. There is a re-
Figure 1.2: Four examples of rock faces: a rock face that runs alongside a road (top left), rock faces in an underground mine (top right), expansive rock faces in an open-pit mine (bottom left), and a close-up of a rock face showing its discontinuities (bottom right).

The production of shear strength along the discontinuities, and the tensile strength between them is zero. Also, the distribution of discontinuities greatly affects permeability, influencing how fluids flow through a rock mass.

Discontinuities in rock masses manifest themselves as a small number of sets of planar surfaces (typically less than five) that are visible on rock faces, as shown in Figure 1.3. These are called joint sets, which are each statistically distributed in orientation. Joint orientation estimation is the act of determining how each joint set in a rock mass is oriented with respect to the global coordinate frame (e.g., North-East-Down). Because axes can be used to represent the orientations of planes in three dimensions, they are suitable representations of the orientations of joint sets in rock masses. In this context, an axis map is called a stereonet, which is used by geotechnical engineers
Figure 1.3: Discontinuities in a rock face. Planar surfaces belonging to three different joint sets are highlighted (solid outlines, dashed outlines, and dash-dotted outlines).

to analyze the stability of a rock face. A major application of axis mapping explored in this thesis is the automated estimation of stereonets using three-dimensional axis mapping, which transforms data collected by a mobile platform into a stereonet of a rock face. Because axis mapping is specifically designed to take advantage of the lack of positional information in axis maps, this novel form of joint orientation estimation has many advantages compared to the traditional approach of collecting data from static sensors.

Axis mapping is a form of state estimation, which aims to estimate not only the axes of the flat surfaces in the environment, but also the orientation of the mobile platform (e.g., a car or a robot) from which the measurements of the environment originate. Put differently, axis mapping is a variant of simultaneous localization and mapping (SLAM) in the space of orientations and axes. Like SLAM, the necessity of estimating the orientation of the mobile platform stems from the oft-stated dilemma: to create a map, the location of the measurements must be known, and to know the location of the measurements, they must be localized within the map. Because axis maps contain only the orientations of planar surfaces in the environment, only the orientation of the mobile platform must be estimated to place measurements of axes
in the axis map.

State estimation algorithms typically require the state being estimated to exist in a vector space to allow for the application of simple arithmetic operators (e.g., addition and subtraction). Unfortunately, many topological spaces that are often useful representations of the estimated variable (or measurements) do not meet this requirement. In particular, the topological spaces of both axes and orientations are not vector spaces. Although this appears to violate the guidelines of selecting a map representation described above, it is often the case that these topologies are manifolds, where each member of the space has a local neighbourhood that is homeomorphic to a vector space. Fortunately, the comparisons and perturbations of variables in state estimation algorithms are inherently local, which allows the use of homeomorphisms to locally transform the original topological space to a vector space (for use in state estimation operations) and back to the original space (to maintain a singularity-free global parameterization). This is the case for both axes and orientations, which permits their use in axis mapping and retains the advantages of the map representation for the LiDAR compass and for joint orientation estimation.

1.1 Organization of this Thesis

This thesis is divided into two parts: theory (Part I) and applications (Part II). In Part I, parameterizations of axes (Chapter 2) and rotations (Chapter 3) are developed in detail, including sections that specifically address the challenges of using these quantities in state estimation algorithms. The parameterization of axes—in particular in three dimensions—is one of the contributions of this thesis. These parameterizations are used in the formal description of axis mapping (Chapter 4). This contribution is presented generically to cover implementations in both two and three dimensions. Part II begins with a detailed description of the LiDAR compass (Chapter 5), which includes an implementation of two-dimensional axis mapping. The LiDAR compass
is tested in multiple environments as part of a localization algorithm. Next, a comprehensive outline of how three-dimensional axis mapping can be used for joint orientation estimation is provided (Chapter 6), including extensive testing in various environments and comparisons against established methods. The development, implementation, and testing of the two applications in Part II are a major contribution of this thesis.

Rather than providing a review of all related work at the outset of the thesis, each chapter includes an overview of the related literature pertaining to the topic covered in that chapter. These just-in-time literature reviews help frame the material covered in each chapter.

1.2 Publications

A number of peer-reviewed publications that include parts of the work presented in this thesis have been published. This section provides references to these publications and describes how their content relates to the content of this thesis.


This conference paper includes a preliminary description of the LiDAR compass and some experimental results in a warehouse environment. The LiDAR compass algorithm described in Chapter 5 is an updated version of the algorithm.


This journal article provides a detailed description of two-dimensional axis mapping and includes experimental results in the same environments as those presented in Chapter 5. The parameterizations of axes and rotations has since been updated to be consistent with three-dimensional axis mapping, which is outlined in Chapter 4.

This conference paper provides a description of three-dimensional axis mapping, and includes the results of joint orientation estimation experiments. The description of the algorithm is greatly expanded in this thesis in Chapters 4 and 6, as are the experimental results presented in Chapter 6.


This journal article includes a detailed overview of the LiDAR compass and includes experimental results in the same environments as those presented in Chapter 5. As is the case with two-dimensional axis mapping, the parameterizations of axes and rotations has been updated in this thesis.


This journal article provides a high-level overview of the application of three-dimensional axis mapping to joint orientation estimation. It describes the algorithm with particular emphasis on how different aspects of joint orientation estimation influenced the design of the axis mapping algorithm. The paper presents the comprehensive experimental results reported in Chapter 6 in this thesis.
Part I

Theory
Chapter 2

Axes

Directions are found in many everyday phenomena. For example, the cardinal directions measured by a compass, the direction of gravitational acceleration, or simply the direction that a car is driving relative to other cars. However, it is sometimes the case that the opposite direction conveys the same information as the direction itself. The quintessential example of this occurrence is the orientation of a plane, such as the (flat) wall of a building. One way to describe the orientation of the wall is its normal vector, which (despite its appearance) is not a direction. Because the orientation of the wall can equivalently be described by flipping the normal vector to the opposing side of the wall, one must distinguish between the topological spaces of normal vectors and directions.

An axis is an unordered pair of opposing directions. As mentioned in the above example, a phenomenon that is well-represented by axes is the orientation of planes in $\mathbb{R}^3$. In this example, the unordered pair of opposing directions consists of the normal vectors on the two sides of the plane, which can be represented by a three-dimensional axis. Similarly, in $\mathbb{R}^2$ the orientation of a line (which is just a lower-dimensional plane) can be represented by a two-dimensional axis. This chapter develops parameterizations of two-dimensional (Section 2.2) and three-dimensional (Section 2.3) axes, with a focus on using them in state estimation problems. In particular, the encapsulation of the axis parameterizations as manifolds (Section 2.4) and representing axes as ran-
dom variables (Section 2.5) are developed in the subsequent sections.

## 2.1 Related Work

The etymology of the word *axis* in the context of probabilistic data comes from the field of directional statistics [1]. Directional statistics deals with cyclical data, such as directions, axes, and rotations. Axes, in particular, are described as any quantity that can represented by a line through the origin of $\mathbb{R}^n$. Because of their cyclical nature, a random variable representing axial data cannot be exactly described by a Gaussian distribution. Instead, the wrapped normal distribution (see Section 2.5) or the Bingham distribution [2] are often used. However, because axis mapping and its applications typically deal with axes with relatively small variances (i.e., $3\sigma < \pi/2$ in two dimensions, as discussed in Section 2.5), they are approximated with Gaussian distributions.

In two dimensions, axes are geometrically described as two diametrically opposed points on the unit circle. As a result, they can parameterized as a pair of complex numbers [3]. The parameterization of two-dimensional unit axes in Section 2.2 can be summarized as unit complex numbers with a change of notation to be consistent with three-dimensional axes and rotations. For example, the compound operator defined in Proposition 2.1 is equivalent to the representation of a complex number in matrix form, and the inverse operation defined in Proposition 2.4 is equivalent to the complex conjugate of a unit complex number.

The explicit parameterization of three-dimensional axes in the context of mapping has received limited attention in the literature, focusing mostly on estimating the position and orientations of planar surfaces. A minimal parameterization of the orientation of planar surfaces (i.e., two parameters) was used by Lee et al. [4], which is convenient to represent as a random variable but is subject to singularities. Kaess [5] showed that through projective geometry, a full description of an infinite plane
(including its position) can be represented as a point in $S^3$, allowing unit quaternions to be used for parameterization (which have considerable heritage in state estimation). Finally, Hertzberg et al. [6] described a general approach to parameterizing many different geometries for state estimation, including points in $S^2$ (e.g., a normal vector in $\mathbb{R}^3$). The parameterization of axes described in this thesis is inspired by the manifold representation of $S^2$ described by Hertzberg et al.; however, this chapter introduces new properties, a new composition operator, a new notation that is analogous to rotations, and new constraints on the manifold operators to distinguish axes and directions.

The notation and nomenclature used to describe axes in this chapter is inspired by the representation of rotations used by Hughes [7] and later expanded upon by Barfoot [8]. Chapter 3 presents rotations with similar notation.

### 2.2 Two-Dimensional Unit Axes

Two-dimensional axes can be used to represent the orientation of a line in $\mathbb{R}^2$ (Figure 2.1) and have a single degree of freedom. This section introduces two-dimensional unit axes, which are an over-parameterization (two parameters, one constraint) of two-dimensional axes that are well suited for use in state estimation problems. A two-dimensional unit axis $\mathbf{m}$ is the $2 \times 1$ column

$$
\mathbf{m} := \begin{bmatrix} \lambda \\ \kappa \end{bmatrix}, \quad (2.1)
$$

where $\lambda, \kappa \in \mathbb{R}$. Unit axes are constrained to the unit circle $S^1$; that is, $\lambda^2 + \kappa^2 = 1$. Additionally, $\mathbf{m} \equiv -\mathbf{m}$ because they both represent the same line (as shown in Figure 2.1). Furthermore, the angle $\phi \in (-\pi, \pi]$ of a unit axis is defined as

$$
\phi := \text{atan2}(\kappa, \lambda). \quad (2.2)
$$
Figure 2.1: The orientation of a line can be represented by a two-dimensional axis. A two-dimensional unit axis \( \mathbf{m} \) is one parameterization of a two-dimensional axis.

Figure 2.2: A two-dimensional unit axis \( \mathbf{m} \in S^1 \) has parts \( \lambda, \kappa \in \mathbb{R} \). The angle \( \phi \in (-\pi, \pi] \) is related to \( \lambda \) and \( \kappa \) by the relationship in (2.2).

An illustration of the components of a unit axis, as well as its angle, are shown in Figure 2.2.

The composition of \( \mathbf{m}, \mathbf{n} \in S^1 \) is achieved with the two-dimensional axis product \( \mathbf{m} \otimes \mathbf{n} \). The axis product is defined as

\[
\mathbf{m} \otimes \mathbf{n} := \mathbf{m}^+ \mathbf{n},
\]

such that \( \mathbf{m}^+ \mathbf{o} = \mathbf{m} \), where \( \mathbf{m}^+ \in SO(2) \) is the compound operator on \( \mathbf{m} \), and \( \mathbf{o} \in S^1 \) is
Axis Mapping

Chapter 2: Axes

the identity unit axis

\[
o := \begin{bmatrix} 1 \\ 0 \end{bmatrix},
\]

(2.4)

A derivation of the compound operator is shown in Proposition 2.1.

**Proposition 2.1 (Compound Operator):** Let \( m = \begin{bmatrix} \lambda & \kappa \end{bmatrix}^T \in S^1 \). Then the compound operator

\[
m^+ = \begin{bmatrix} \lambda & -\kappa \\ \kappa & \lambda \end{bmatrix}
\]

(2.5)

is a rotation matrix \( m^+ \in SO(3) \) such that \( m^+ o = m \).

**Proof:** In two dimensions, every rotation matrix \( R \in SO(2) \) has the form

\[
R = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix},
\]

(2.6)

where \( R a \) rotates \( a \in \mathbb{R}^2 \) counterclockwise by the angle \( \phi \). From the geometry illustrated in Figure 2.2, note that \( \lambda = \cos(\phi) \) and \( \kappa = \sin(\phi) \), where \( \phi \) is the angle of \( m \). Substituting these into (2.5) generates a rotation matrix of the form (2.6). Finally,

\[
m^+ o = \begin{bmatrix} \lambda & -\kappa \\ \kappa & \lambda \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = m.
\]

(2.7)

\[
\]

**Proposition 2.2 (Axis Product):** Let \( m, n, p \in S^2 \) such that \( p = m^+ n \). Then

\[
p = \begin{bmatrix} \cos(\phi_m + \phi_n) \\ \sin(\phi_m + \phi_n) \end{bmatrix},
\]

(2.8)

where \( \phi_m \) and \( \phi_n \) are the angles of \( m \) and \( n \), respectively.
Proof: Let \( m = \begin{bmatrix} \lambda_m & \kappa_m \end{bmatrix}^T \) and \( n = \begin{bmatrix} \lambda_n & \kappa_n \end{bmatrix}^T \). Then by direct computation,

\[
p = \begin{bmatrix} \lambda_m & -\kappa_m \\ \kappa_m & \lambda_m \end{bmatrix} \begin{bmatrix} \lambda_n \\ \kappa_n \end{bmatrix} = \begin{bmatrix} \lambda_m \lambda_n - \kappa_m \kappa_n \\ \kappa_m \lambda_n + \lambda_m \kappa_n \end{bmatrix}
\]

\[
= \begin{bmatrix} \cos(\phi_m) \cos(\phi_n) - \sin(\phi_m) \sin(\phi_n) \\ \sin(\phi_m) \cos(\phi_n) + \cos(\phi_m) \sin(\phi_n) \end{bmatrix}
\]

\[
= \begin{bmatrix} \cos(\phi_m + \phi_n) \\ \sin(\phi_m + \phi_n) \end{bmatrix}.
\]

(2.9)

Proposition 2.3 (Commutativity): Let \( m, n \in S^1 \). Then \( m^+ n = n^+ m \).

Proof: Let \( m = \begin{bmatrix} \lambda_m & \kappa_m \end{bmatrix}^T \) and \( n = \begin{bmatrix} \lambda_n & \kappa_n \end{bmatrix}^T \). Then by direct computation,

\[
m^+ n = \begin{bmatrix} \lambda_m & -\kappa_m \\ \kappa_m & \lambda_m \end{bmatrix} \begin{bmatrix} \lambda_n \\ \kappa_n \end{bmatrix} = \begin{bmatrix} \lambda_m \lambda_n - \kappa_m \kappa_n \\ \kappa_m \lambda_n + \lambda_m \kappa_n \end{bmatrix}
\]

\[
= \begin{bmatrix} \lambda_m \\ \kappa_m \end{bmatrix} = n^+ m
\]

(2.10)

Proposition 2.4 (Inverse): Let \( m \in S^1 \). Then \( m^+ m^{-1} = o \) if and only if

\[
m^{-1} = \begin{bmatrix} \lambda \\ -\kappa \end{bmatrix}.
\]

(2.11)

Proof: If \( m^+ m^{-1} = o \), then \( m^{-1} = (m^+)^T o \) since by Proposition 2.1, \( m^+ \) is a rotation matrix and

\[
(m^+)^T o = \begin{bmatrix} \lambda \\ -\kappa \end{bmatrix}.
\]

(2.12)

Furthermore, if \( m^{-1} \) is given by (2.11), then by direct computation \( m^+ m^{-1} = o \). □

Given \( m \in S^1 \), one can combine Propositions 2.1 and 2.4 to define \( m^- \in SO(2) \), where

\[
m^- := (m^{-1})^+.
\]

(2.13)
Combining this definition with the results of Propositions 2.1–2.4 reveals the following identities:

\[ m^- = (m^+)^{-1} = (m^+)^\top \]  
\[ m^- m^+ \equiv m^+ m^- \equiv o^+ \equiv o^- \equiv I_2 \]  
\[ m^+ o \equiv m \quad m^+ m^{-1} \equiv o \]  
\[ m^- o \equiv m^{-1} \quad m^- m \equiv o, \]

where \( I_n \) is the \( n \times n \) identity matrix.

**Proposition 2.5 (Difference):** Let \( m, n, p \in S^1 \) such that \( p = m^- n \). Then

\[ p = \begin{bmatrix} \cos(\phi_n - \phi_m) \\ \sin(\phi_n - \phi_m) \end{bmatrix}, \]

where \( \phi_m \) and \( \phi_n \) are the angles of \( m \) and \( n \), respectively.

**Proof:** Let \( m = \begin{bmatrix} \lambda_m \\ \kappa_m \end{bmatrix}^\top \) and \( n = \begin{bmatrix} \lambda_n \\ \kappa_n \end{bmatrix}^\top \). Then by direct computation,

\[ p = \begin{bmatrix} \lambda_m & \kappa_m \\ -\kappa_m & \lambda_m \end{bmatrix} \begin{bmatrix} \lambda_n \\ \kappa_n \end{bmatrix} = \begin{bmatrix} \lambda_m \lambda_n + \kappa_m \kappa_n \\ -\kappa_m \lambda_n + \lambda_m \kappa_n \end{bmatrix} \]

\[ = \begin{bmatrix} \cos(\phi_n - \phi_m) \cos(\phi_n) + \sin(\phi_n - \phi_m) \sin(\phi_n) \\ -\sin(\phi_n - \phi_m) \cos(\phi_n) + \cos(\phi_n - \phi_m) \sin(\phi_n) \end{bmatrix} \]

\[ = \begin{bmatrix} \cos(\phi_n - \phi_m) \\ -\sin(\phi_n - \phi_m) \\ \sin(\phi_n - \phi_m) \end{bmatrix}. \]

\[ \text{Finally:} \]

**2.3 Three-Dimensional Unit Axes**

Three-dimensional axes can be used to represent the orientation of a plane in \( \mathbb{R}^3 \) (Figure 2.3) and have two degrees of freedom. This section introduces three-dimensional
unit axes, which are an over-parameterization (three parameters, one constraint) of three-dimensional axes that are well suited for use in state estimation problems. A unit axis $\mathbf{m}$ is the $3 \times 1$ column

$$
\mathbf{m} := \begin{bmatrix} \lambda \\ \kappa \end{bmatrix},
$$

(2.17)

where $\lambda \in \mathbb{R}$ and $\kappa = (\kappa_1, \kappa_2) \in \mathbb{R}^2$ are the scalar and vector parts of the unit axis, respectively. Unit axes are constrained to the unit sphere $S^2$; that is, $\lambda^2 + \kappa^\top \kappa = 1$. Additionally, $\mathbf{m} \equiv -\mathbf{m}$ because they both represent the same axis (as shown in Figure 2.3). Furthermore, the inclination $\phi \in [0, \pi]$ and direction $\mathbf{r} \in S^1$ of a unit axis are defined as

$$
\phi := \cos^{-1}(\lambda), \quad \mathbf{r} := \frac{\kappa}{||\kappa||}.
$$

(2.18)

An illustration of the scalar and vector components of a unit axis, as well as its inclination and direction, are shown in Figure 2.4.

The composition of unit axes $\mathbf{m}, \mathbf{n} \in S^2$ is achieved using the three-dimensional axis product $\mathbf{m} \otimes \mathbf{n}$. The axis product is defined as

$$
\mathbf{m} \otimes \mathbf{n} := \mathbf{m}^+ \mathbf{n},
$$

(2.19)

such that $\mathbf{m}^+ \mathbf{o} = \mathbf{m}$, where $\mathbf{m}^+ \in SO(3)$ is the compound operator on $\mathbf{m}$, and $\mathbf{o} \in S^2$ is the identity axis

$$
\mathbf{o} := \begin{bmatrix} 1 \\ 0 \end{bmatrix}.
$$

(2.20)

A derivation of the compound operator is shown in Proposition 2.6.
Figure 2.4: A three-dimensional unit axis $m \in S^2$ has a scalar part $\lambda \in \mathbb{R}$ and vector part $\kappa \in \mathbb{R}^2$. The inclination $\phi \in [0, \pi]$ and direction $r \in S^1$ of $m$ are related to $\lambda$ and $\kappa$ by the relationships in (2.18).

**Proposition 2.6 (Compound Operator):** Let $m = \begin{bmatrix} \lambda & \kappa^\top \end{bmatrix}^\top \in S^2$, $\kappa = \begin{bmatrix} \kappa_1 & \kappa_2 \end{bmatrix}^\top$, and $u = \begin{bmatrix} -\kappa_2 & \kappa_1 \end{bmatrix}^\top$. Then the compound operator

$$m^+ = \begin{bmatrix} \lambda & -\kappa^\top \\ \kappa & \lambda I_2 + \frac{1}{\lambda+1}uu^\top \end{bmatrix}$$

(2.21)

is a rotation matrix such that $m^+ o = m$.

**Proof:** Let $\phi$ be the inclination of $m$. If $\phi = 0$, then $m = o$, $m^+ = I_3$ (which is a rotation matrix), and $m^+ o = m$. If $\phi \in (0, \pi]$, one can construct $m^+$ from a rotation of $\phi$ about $a \in S^2$, where $a$ is perpendicular to both $m$ and $o$. For $b = (b_1, b_2, b_3) \in \mathbb{R}^3$, let $b^\times$ be defined as the skew-symmetric matrix

$$b^\times := \begin{bmatrix} 0 & -b_3 & b_2 \\ b_3 & 0 & -b_1 \\ -b_2 & b_1 & 0 \end{bmatrix}.$$
Then by the definition of the cross product,
\[ \mathbf{o} \times \mathbf{m} = \|\mathbf{o}\| \|\mathbf{m}\| \sin(\phi) \mathbf{a} = \sin(\phi) \mathbf{a}. \] (2.23)

Therefore,
\[ \mathbf{a} = \frac{1}{\sin(\phi)} \mathbf{o} \times \mathbf{m} = \frac{1}{\sin(\phi)} \begin{bmatrix} 0 \\ \mathbf{u} \end{bmatrix}. \] (2.24)

Rodrigues’ rotation formula states that
\[ \mathbf{m}^+ = \cos(\phi) \mathbf{I}_3 + (1 - \cos(\phi)) \mathbf{a} \mathbf{a}^T + \sin(\phi) \mathbf{a}^x, \] (2.25)

where \( \mathbf{m}^+ \in SO(3) \) is a rotation of angle \( \phi \) about \( \mathbf{a} \). Substituting (2.24) into (2.25) yields
\[ \mathbf{m}^+ = \cos(\phi) \mathbf{I}_3 + \frac{1 - \cos(\phi)}{\sin(\phi)^2} \begin{bmatrix} 0 \\ \mathbf{u} \end{bmatrix} \begin{bmatrix} 0 \\ \mathbf{u} \end{bmatrix}^T + \begin{bmatrix} 0 \\ \mathbf{u} \end{bmatrix}^x \] (2.26)

and the following substitutions transform (2.26) into (2.21):
\[ \frac{1 - \cos(\phi)}{\sin(\phi)^2} = \frac{1}{\cos(\phi) + 1} \] (trig. identity),
\[ \cos(\phi) = \lambda \] (definition of inclination).

Finally,
\[ \mathbf{m}^+ \mathbf{o} = \begin{bmatrix} \lambda & -\kappa^T \\ \kappa & \lambda \mathbf{I}_2 + \frac{1}{\lambda+1} \mathbf{u} \mathbf{u}^T \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \mathbf{m}. \] (2.28)

**Proposition 2.7 (Inverse):** Let \( \mathbf{m} \in S^2 \). Then \( \mathbf{m}^+ \mathbf{m}^{-1} = \mathbf{o} \) if and only if
\[ \mathbf{m}^{-1} = \begin{bmatrix} \lambda \\ -\kappa \end{bmatrix}. \] (2.29)
Proof: If $m^+ m^{-1} = o$, then $m^{-1} = (m^+)^T o$ since by Proposition 2.6, $m^+$ is a rotation matrix and

$$
(m^+)^T o = \begin{bmatrix} \lambda \\ -\kappa \end{bmatrix}.
$$

(2.30)

Furthermore, if $m^{-1}$ is given by (2.29), then by direct computation $m^+ m^{-1} = o$. ■

Given $m \in S^2$, one can combine Propositions 2.6 and 2.7 to define the inverse compound operator $m^- \in SO(3)$, where

$$
m^- := (m^{-1})^+.
$$

(2.31)

Using this definition along with the results of Proposition 2.6 and 2.7 reveals the following identities:

$$
m^- \equiv (m^+)^{-1} \equiv (m^+)^T
$$

(2.32a)

$$
m^- m^+ \equiv m^+ m^- \equiv o^+ \equiv o^- \equiv I_3
$$

(2.32b)

$$
m^+ o \equiv m \quad m^+ m^{-1} \equiv o
$$

(2.32c)

$$
m^- o \equiv m^{-1} \quad m^- m \equiv o.
$$

(2.32d)

Proposition 2.8 (Axis Product): Let $m, n \in S^2$. Then $m^+ n$ is a rotation of $m$ by $C \in SO(3)$, where $C = m^+ n^+ m^-$.

Proof: By making use of identities in (2.32),

$$
m^+ n \overset{(2.32c)}{=} m^+ n^+ o \overset{(2.32d)}{=} m^+ m^- m^+ m^- m = m^+ n m^- m
$$

and thus $C = m^+ n^+ m^-$. ■

Corollary 2.1 (Angle of Rotation): The angle of rotation of $C$ is the inclination of $n$. 

20
Proof: Rotation matrices have the property \( \text{Tr}(A) = 2 \cos(\theta) + 1 \), where \( A \in SO(3) \) and \( \theta \) is the angle of rotation of \( A \). Because the trace of a matrix is invariant under cyclic permutations,

\[
\text{Tr}(C) = \text{Tr}(m^+ n^+ m^-) = \text{Tr}(n^+ m^- m^+) = \text{Tr}(n^+).
\]

Therefore, \( C \) and \( n^+ \) have the same angle of rotation, which is the inclination of \( n \) by Proposition 2.6.

Corollary 2.2 (Axis of Rotation): Let \( a_n \in S^2 \) be the axis of rotation of \( n^+ \). Then for \( n \neq o \), the axis of rotation of \( C \) is \( m^+ a_n \).

Proof: Let \( a \) be the axis of rotation of \( C \). Then

\[
a = Ca = m^+ n^+ m^- a
\]

\[
m^- a = n^+ m^- a.
\]

By (2.35), \( m^- a \) is invariant to a rotation by \( n^+ \); therefore, \( m^- a = a_n \), or \( a = m^+ a_n \).

An illustration of the rotation of \( m \) by \( C \) is shown in Figure 2.5. This figure also illustrates the angle and axis of rotation.

Proposition 2.9 (Axis Difference): Let \( m, n, p \in S^2 \) such that \( p = m^- n \). Then the inclination of \( p \) is the angle between \( m \) and \( n \).

Proof: Let \( \phi_p \) be the inclination of \( p \). By definition, \( \phi_p = \cos^{-1}(\lambda_p) \), where \( \lambda_p \) is the scalar part of \( p \). By directly computing \( m^- n \), \( \lambda_p = m^+ n \), and \( \cos^{-1}(m^+ n) \) is the angle between \( m \) and \( n \).
Figure 2.5: The axis product \( \mathbf{m}^+ \mathbf{n} = \mathbf{Cm} \) is a rotation of \( \mathbf{m} \) by \( \phi_n \) (the inclination of \( \mathbf{n} \)) about \( \mathbf{m}^+ \mathbf{a}_n \) (where \( \mathbf{a}_n \) is the axis of rotation of \( \mathbf{n}^+ \)).

### 2.4 Manifold Encapsulation of Axes

The topological spaces of axes prevent the direct application of traditional state estimation algorithms, which require the state and measurements to be vector spaces. This problem is addressed by Hertzberg et al. [6] by taking advantage of the inherently local operations applied by state estimation algorithms. More specifically, if a homeomorphism can be defined that locally transforms the original topological space of a variable to a subspace of \( \mathbb{R}^n \) (i.e., the variable exists on a manifold), one can apply the required state estimation operations in the local subspace, while maintaining a singularity-free global parameterization in the original topological space.

The manifold encapsulation of unit axes described in this section is based on the similar approach for unit directions in \( S^1 \) and \( S^2 \) presented by Hertzberg et al. [6]. However, if \( \mathbf{d} \) is a unit direction, then unlike unit axes, \( \mathbf{d} \neq -\mathbf{d} \). As a result, because of the equivalency \( \mathbf{m} \equiv -\mathbf{m} \) for unit axes, as well as the use of a different composition op-
Figure 2.6: A geometric representation of $h^1$ is the real line of length $\pi$ centred at the origin, with one end removed. This line is the domain of the axis vectors $\phi$ in (2.39) and (2.40).

operator in the three-dimensional case, the manifold encapsulation of unit axes differs from unit directions and is presented in this section.

### 2.4.1 Two-Dimensional Axes

The unit circle $S^1$ is a differentiable manifold; that is, it is continuously differentiable at every point and is locally homeomorphic to $\mathbb{R}$. The homeomorphism between $m \in S^1$ and $\phi \in \mathbb{R}$ is described by the two-dimensional axis exponential and axis logarithm; i.e.,

$$m = \exp(\phi) := \begin{bmatrix} \cos(\phi) \\ \sin(\phi) \end{bmatrix}$$

and

$$\phi = \log(m) := \arctan2(\kappa, \lambda).$$

These relationships can be derived geometrically from Figure 2.2. Note that $\phi$ in (2.37) is the angle of $m$. It is denoted the axis vector parameterization (i.e., it is a one-dimensional vector) to be consistent with the analogous parameterizations in higher dimensions (e.g., axis vectors in $S^2$, as described in Section 2.4.2). Note that $\phi = \log(\exp(\phi))$ and $m = \exp(\log(m))$ for all $m$ and $\phi$.

Following the notation used in [6], the $\boxplus$ ("box plus") and $\boxminus$ ("box minus") operators are defined to be used for state estimation on axes. These are

$$\boxplus: S^1 \times h^1 \rightarrow S^1$$

and

$$\boxminus: S^1 \times S^1 \rightarrow h^1,$$

where $h^1 := (-\pi/2, \pi/2]$ is the real line of length $\pi$ centred at the origin, with one end removed (i.e., $h^1 \subset \mathbb{R}$), which is illustrated in Figure 2.6. The reason why $h^1$ is required in (2.38) is clarified in Section 2.4.3.
Let \( m, n \in S^1 \) and \( \phi \in h^1 \), where \( \phi \) is a perturbation of \( m \) and results in \( n \). This perturbation of \( m \) by \( \phi \) is a typical operation used in state estimation algorithms (e.g., suppose \( \phi \) is a correction to be applied to \( m \)). It is achieved by mapping \( \phi \) to \( S^1 \) using the axis exponential, and then applying it to \( m \) using the axis product; i.e.,

\[
    m \circ \phi := m^+ \exp(\phi) = n. \tag{2.39}
\]

Recall that the axis product \( m^+ \exp(\phi) \) perturbs the angle of \( m \) by \( \phi \) (Proposition 2.2). Similarly, suppose that the difference between \( m \) and \( n \) is required by the state estimation algorithm (e.g., to determine a measurement error). This is achieved by calculating a unit axis that represents their difference and then mapping the result to \( h^1 \) using the axis logarithm; i.e.,

\[
    \phi = n \circ m := \begin{cases} 
        \log(m^- n) & \text{if } m^\top n \geq 0 \\
        \log(-(m^- n)) & \text{otherwise} 
    \end{cases} \tag{2.40}
\]

The cases in (2.40) ensure that \( \phi \in h^1 \) for all \( m \) and \( n \). Recall that the axis product \( m^- n \) yields a unit axis whose angle is the angle from \( m \) to \( n \) (Proposition 2.5).

### 2.4.2 Three-Dimensional Axes

The unit sphere \( S^2 \) is not a differentiable manifold because it must have at least one point whose tangent is the null vector, as stated by the hairy ball theorem [9]. However, one can still define a homeomorphism between \( S^2 \) and \( \mathbb{R}^2 \) that explicitly addresses this singularity. Let \( m \in S^2 \) and \( \phi \in \mathbb{R}^2 \). Then the three-dimensional axis exponential and axis logarithm are defined as

\[
    m = \exp(\phi) := \begin{bmatrix} 
        \cos(|\phi|) \\
        \text{sinc}(|\phi|) \phi 
    \end{bmatrix}, \tag{2.41}
\]
where the unnormalized sinc function has the property \( \text{sinc}(0) := 1 \). These relationships can be derived geometrically from Figure 2.4. Note that \( \phi \) in (2.42) is the product of the direction and inclination of \( m \) (i.e., \( \phi = \phi_r \)). This is clear when one considers that \( \text{atan2}(\|\kappa\|, \lambda) \) is the inclination of \( m \) (see Figure 2.4). To highlight its similarities to rotation vectors (which are the product of the angle and axis of rotation), \( \phi \) is denoted the axis vector parameterization. The terms exponential and logarithm are used loosely here because \( S^2 \) is not continuously differentiable. Nonetheless, \( \phi = \log(\exp(\phi)) \) and \( m = \exp(\log(m)) \) for all \( m \) and \( \phi \).

The manifold state estimation operators analogous to those defined in (2.38), but for three-dimensional unit axes are

\[
\begin{align*}
\boxplus & : S^2 \times h^2 \to S^2, \\
\boxminus & : S^2 \times S^2 \to h^2,
\end{align*}
\]

(2.43)

where for \( \phi = (\phi_1, \phi_2) \in \mathbb{R}^2 \),

\[
h^2 := \{ \phi : \|\phi\| \leq \pi/2, \text{atan2}(\phi_1, \phi_2) \in [0, \pi) \}
\]

(2.44)

is a disk of radius \( \pi/2 \) with half of its edge removed (i.e., \( h^2 \subset \mathbb{R}^2 \)), which is illustrated in Figure 2.7. The reason why \( h^2 \) is required in (2.43) is clarified in Section 2.4.3.

Let \( m, n \in S^2 \) and \( \phi \in h^2 \), where \( \phi \) is a perturbation of \( m \) that results in \( n \). It is achieved by mapping \( \phi \) to \( S^2 \) by using the axis exponential, and then applying it to \( m \) by using the axis product; i.e.,

\[
n = m \boxplus \phi := m^+ \exp(\phi).
\]

(2.45)

Recall that the axis product \( m^+ \exp(\phi) \) perturbs \( m \) by rotating it (Proposition 2.8) by \( \phi \) (Corollary 2.1), where \( \phi \) is the magnitude of \( \phi \) and the inclination of \( \exp(\phi) \).
Figure 2.7: A geometric representation of $h^2$ is a disk of radius $\pi/2$ centred at the origin with half of its edge removed. This disk is the domain of the axis vectors $\phi = (\phi_1, \phi_2)$ in (2.45) and (2.46).

Similarly, suppose that the difference between $m$ and $n$ is required by the state estimation algorithm. This is achieved by calculating a unit axis that represents their difference and then mapping the result to $h^2$. Let $\lambda_m$ be the scalar part of $m$ and let $d = m^\top n$. Then

$$\phi = n \boxminus m := \begin{cases} 
\log(m^{-} n) & \text{if } d \geq 0, \lambda_m \geq 0 \\
\log((-m)^{-} (-n)) & \text{if } d \geq 0, \lambda_m < 0 \\
\log((-((-m)^{-} (-n)))) & \text{if } d < 0, \lambda_m \geq 0 \\
\log(-(m^{-} n)) & \text{if } d < 0, \lambda_m < 0.
\end{cases} \tag{2.46}$$

The cases in (2.46) ensure that $\phi \in h^2$ and is unique for all $m$ and $n$. Note that the inclination of $m^{-} n$ is $\phi$, the angle between $m$ and $n$ (Proposition 2.9), and the magnitude of $\phi$. 

26
2.4.3 Axioms of Manifold Operators

Let \( m, n \in S^n \) and \( \phi_1, \phi_2 \in h^n \) for \( n = 1 \) (two-dimensional axes) and \( n = 2 \) (three-dimensional axes). To be used for state estimation, Hertzberg et al. [6] specified that the \( \boxplus \) and \( \boxminus \) operators are required to be smooth in \( \phi_1, \phi_2, \) and \( n \) (but not \( m \)), and that the following four axioms to hold for all \( m, n, \phi_1 \) and \( \phi_2 \):

\[
\begin{align*}
\text{(2.47a)} & \quad m \boxplus 0 = m \\
\text{(2.47b)} & \quad m \boxplus (n \boxminus m) = n \\
\text{(2.47c)} & \quad (m \boxplus \phi_1) \boxminus m = \phi_1 \\
\text{(2.47d)} & \quad \| (m \boxplus \phi_1) \boxminus (m \boxplus \phi_2) \| \leq \| \phi_1 - \phi_2 \|. 
\end{align*}
\]

Here, (2.47c) requires \( \boxplus \) to be injective to guarantee that its right hand side is unique. For this axiom to hold, it is required that \( \phi_1 \in h^n \). Consequently, (2.47b) requires the range of \( \boxminus \) to be \( h^n \) to properly satisfy the injectivity of \( \boxplus \). For these reasons, \( h^n \) is needed in place of \( \mathbb{R}^n \) in (2.38) and (2.43). Proofs demonstrating that the definitions of \( \boxplus \) and \( \boxminus \) in (2.39), (2.40), (2.45), and (2.46) abide by these axioms are provided in Appendix A.

2.5 Random Axes

Parameterizing an axis as a random variable (i.e., a random axis) with a unit axis is troublesome because it represents one degree (\( S^1 \)) or two degrees (\( S^2 \)) of freedom with an additional parameter and the unit constraint. As a result, the covariance matrices representing the uncertainty in the parameters would be singular. This issue is addressed by parameterizing the error of a unit axis by its axis vector parameterization, which has the same number of parameters as the number of degrees of freedom. For example, assuming that a Gaussian distribution is used to represent a
two-dimensional random axis by its mean $\hat{\mathbf{m}} \in S^1$ and variance $\sigma^2_{\delta\phi}$,

$$\hat{\mathbf{m}} = \mathbf{m} \mp \delta\phi, \quad \delta\phi \sim \mathcal{N}(0, \sigma^2_{\delta\phi}),$$  \hspace{1cm} (2.48)

where the error $\delta\phi \in h^1$ is a perturbation of the true axis $\mathbf{m} \in S^1$. Similarly, a three-dimensional random axis is represented by its mean $\hat{\mathbf{m}} \in S^2$ and covariance matrix $\mathbf{P}_{\delta\phi}$ such that

$$\hat{\mathbf{m}} = \mathbf{m} \mp \delta\phi, \quad \delta\phi \sim \mathcal{N}(0, \mathbf{P}_{\delta\phi}),$$  \hspace{1cm} (2.49)

where the error $\delta\phi \in h^2$ is a perturbation of the true axis $\mathbf{m} \in S^2$.

Random axes cannot be exactly represented by Gaussian distributions because the topological spaces of the errors ($h^1$ and $h^2$) are bounded. One possible distribution for random axes is the wrapped normal (WN) distribution $[1]$, denoted $WN(\theta, \sigma^2)$, whose (univariate) probability density function (PDF) is

$$f_{WN}(x|\theta, \sigma^2) := \frac{1}{\sigma \sqrt{2\pi}} \sum_{k=-\infty}^{\infty} \exp \left[ \frac{-(x - \theta + n\pi k)^2}{2\sigma^2} \right],$$  \hspace{1cm} (2.50)

where $n = 1$ for axis vectors $x, \theta \in h^1$. Note how the WN PDF is simply an infinite series of Gaussian PDFs $f_N(x|\theta - n\pi k, \sigma^2)$. As a result, for the same range, the probability enclosed by a WN distribution within some distance from the mean (e.g., $3\sigma$) is larger than that of a Gaussian distribution in the same interval. However, for a two-dimensional random axis with $3\sigma_{\delta\phi} < \pi/2$ (i.e., it is highly probable that the error is in $h^1$), the vast majority of the probability is enclosed by a single term ($k = 0$) of a WN distribution, making a Gaussian distribution a reasonable approximation. This example is illustrated in Figure 2.8.
Figure 2.8: The error $\delta \phi \in h^1$ of a two-dimensional axis is described by a zero-mean Gaussian random variable with standard deviation $\sigma_{\delta \phi}$, which requires $3\sigma_{\delta \phi} < \pi/2$ for this to be a valid representation. The probability that $-3\sigma_{\delta \phi} \leq \delta \phi \leq 3\sigma_{\delta \phi}$ is approximately 0.9973.
Chapter 3

Rotations

A rotation of a rigid body is a circular movement about a fixed point or axis. Rotations are used to describe the relative orientations of rigid bodies, or of a single rigid body in motion. In two dimensions, the rotation between an inertial (non-moving) coordinate frame and the coordinate frame of a moving object is commonly called the heading of the object. In three dimensions, this rotation is commonly called the orientation or attitude of the object. In the context of state estimation, the heading (or orientation) of a moving object (and, less commonly, of static objects in the environment) is often estimated as part of a localization or mapping algorithm.

The parameterization of rotations has received considerable attention in many fields of research. In three dimensions, rotations have three degrees of freedom, but all three-parameter representations are singular for certain orientations (i.e., there is a loss of a degree of freedom). As a result, over-parameterizations of rotations (i.e., greater than three parameters and subject to constraints) are commonly used to avoid singularities. This chapter reviews over-parameterizations of two-dimensional (Section 3.2) and three-dimensional (Section 3.3) rotations, with a focus on their use in state estimation problems. In particular, the encapsulation of the reviewed parameterizations as manifolds (Section 3.4) and the representation of rotations as random variables (Section 3.5) are presented in the subsequent sections.
3.1 Related Work

Representing a rotation with a minimal parameterization (i.e., the same number of parameters as degrees of freedom) inevitably requires special attention to globally represent all possible rotations. For example, in two dimensions, representing the heading by a scalar angle requires that the difference between two headings is properly wrapped to the unit circle. In state estimation, angle wrapping is typically viewed as an implementation detail, and Gaussian distributions are used to approximate angles as random variables. However, there have recently been efforts to remove this approximation. Traa and Smaragdis [10] introduced the wrapped Kalman filter (WKF) that uses the wrapped Gaussian distribution to more accurately represent headings as random variables. Kurz et al. [11] extended this approach for nonlinear measurement models, and Gilitschenski et al. [12] developed a similar filter using the von Mises distribution. The two-dimensional unit rotations described in this chapter assume that a Gaussian distribution is sufficient to represent them as random variables (assuming sufficiently small variance; see Section 3.5); however, the wrapping problem is avoided by using a constrained two-parameter representation.

There has been substantial research on the parameterization of three-dimensional rotations for use in state estimation algorithms. Markley [13] presents a good overview of these parameterizations in the context of Kalman filtering. To avoid singularities, early implementations simply avoided some orientations or switched between parameter orderings [14]. More recently, singularity-free over-parameterizations such as unit quaternions or rotation matrices have been used, but with periodic renormalization to ensure that the constraints of the parameterizations are met [15]. Another approach is the introduction of fictitious low variance measurements of the constraints themselves [16]. A major disadvantage of these approaches is the recurrent optimization of non-existent degrees of freedom of the rotation. Modern approaches take advantage of the fact that the group of rotations forms a matrix Lie
group \([17, 18, 6]\), which permits the inherently local operations of state estimation algorithms to be applied in the tangent space of an over-parameterization. This chapter follows the modern approach, describing unit quaternions, their manifold encapsulation, and how they are used to represent rotations as random variables.

### 3.2 Two-Dimensional Unit Rotations

Two-dimensional rotations can be used to represent the heading of a vehicle in \(\mathbb{R}^2\) (Figure 3.1) and have a single degree of freedom. This section introduces two-dimensional unit rotations, which are an over-parameterization (two parameters, one constraint) of two-dimensional rotations that are well suited for use in state estimation problems. A two-dimensional unit rotation \(q\) is the \(2 \times 1\) column

\[
q := \begin{bmatrix} \eta \\ \epsilon \end{bmatrix}, \tag{3.1}
\]

where \(\eta, \epsilon \in \mathbb{R}\). Unit rotations are constrained to the unit circle \(S^1\); that is, \(\eta^2 + \epsilon^2 = 1\).

In two dimensions, unit rotations are very similar to unit axes. The only difference between the two parameterizations is \(q \neq -q\) for a unit rotation \(q\), whereas \(m \equiv -m\) for a unit axes \(m\). As result, because \(q\) and \(m\) are both in \(S^1\), all of the properties of two-dimensional unit axes described in Section 2.2 also apply to two-dimensional unit rotations. In summary, for unit rotations \(q, p, \epsilon \in S^1\),

\[
\iota := \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \tag{3.2}
\]

\[
q \otimes p = q^+ p \tag{3.3}
\]

\[
q^+ = \begin{bmatrix} \eta & -\epsilon \\ \epsilon & \eta \end{bmatrix}, \quad q^+ p = \begin{bmatrix} \cos(\theta_q + \theta_p) \\ \sin(\theta_q + \theta_p) \end{bmatrix} \tag{3.5}
\]

\[
q^+ p = p^+ q \tag{3.6}
\]

\[
q^{-1} = \begin{bmatrix} \eta \\ -\epsilon \end{bmatrix} \tag{3.7}
\]
The heading of a vehicle can be represented by a two-dimensional rotation. For example, the heading can be defined as a counterclockwise rotation from an inertial coordinate frame to the vehicle coordinate frame. A two-dimensional unit rotation is one parameterization of a two-dimensional rotation.

\[ q^- := (q^{-1})^+ \quad (3.8) \]

\[ q^- p = \begin{bmatrix} \cos(\theta_p - \theta_q) \\ \sin(\theta_p - \theta_q) \end{bmatrix} \quad (3.9) \]

where \( q \) is the identity unit rotation, and \( \theta_q, \theta_p \) are the angles of \( q \) and \( p \), respectively. Note that because \( q \neq -q \), the manifold encapsulations of two-dimensional rotations and axes are different. These differences are highlighted in Section 3.4.

### 3.3 Unit Quaternions

Three-dimensional rotations can be used to represent the orientation of a vehicle in \( \mathbb{R}^3 \) (Figure 3.2) and have three degrees of freedom. This section introduces unit quaternions, which are an over-parameterization (four parameters, one constraint) of three-dimensional rotations that are well suited for use in state estimation problems.

A unit quaternion \( q \) is the \( 4 \times 1 \) column

\[ q := \begin{bmatrix} \eta \\ \epsilon \end{bmatrix} \quad (3.10) \]

where \( \eta \in \mathbb{R} \) and \( \epsilon = (\epsilon_1, \epsilon_2, \epsilon_3) \in \mathbb{R}^3 \) are the scalar and vector parts of the unit quaternion, respectively. Unit quaternions are constrained to the unit 3-sphere \( S^3 \); that is, \( \eta^2 + \epsilon^\top \epsilon = 1 \). Note that, unlike the relationship between two-dimensional axes and rotations, three-dimensional axes and rotations do not belong to the same topological space. As a result, the necessary properties of unit quaternions for state
Figure 3.2: The orientation of a vehicle can be represented by a three-dimensional rotation. The orientation is defined as a rotation from an inertial coordinate frame to the vehicle coordinate frame. A unit quaternion is one parameterization of a three-dimensional rotation.

estimation are explicitly outlined in this section. The properties described in this section are presented without proof because they are well documented elsewhere [19, 8].

The angle $\theta \in [0, \pi]$ and axis $a \in S^2$ of a rotation parameterized by a unit quaternion are

$$\theta = 2 \cos^{-1}(\eta), \quad a = \frac{\epsilon}{||\epsilon||}.$$ \hspace{1cm} (3.11)

Here it is clear that $q \equiv -q$ because they represent the same rotation (i.e., the signs of both the axis and angle of rotation are negated). The null rotation ($\theta = 0$) results in the identity unit quaternion

$$\iota := \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$ \hspace{1cm} (3.12)

and the opposite rotation ($-\theta$ about $a$) results in the inverse unit quaternion

$$q^{-1} := \begin{bmatrix} \eta \\ -\epsilon \end{bmatrix} = -\begin{bmatrix} -\eta \\ \epsilon \end{bmatrix}.$$ \hspace{1cm} (3.13)

The composition of unit quaternions $q, p \in S^3$ is achieved using the quaternion product $q \otimes p$, where

$$q \otimes p := \begin{bmatrix} \eta_q \eta_p - \epsilon_q^T \epsilon_p \\ \eta_q \epsilon_p + \eta_p \epsilon_q - \epsilon_q \times \epsilon_p \end{bmatrix},$$ \hspace{1cm} (3.14)

and $\epsilon^\times$ is the skew-symmetric matrix given by (2.22). Factoring $q$ or $p$ out of (3.14) transforms $q \otimes p$ into $q^+ p$ or $p^\oplus q$, where

$$q^+ = \begin{bmatrix} \eta_q & -\epsilon_q^T \\ \epsilon_q & \eta_q I_{3 \times 3} - \epsilon_q \times \end{bmatrix}, \quad p^\oplus = \begin{bmatrix} \eta_p & -\epsilon_p^T \\ \epsilon_p & \eta_p I_{3 \times 3} + \epsilon_p \times \end{bmatrix}.$$ \hspace{1cm} (3.15)
are the left-hand (+) and right-hand (⊕) compound operators on a unit quaternion. Several identities involving these operators are noted by Barfoot et al. [8].

Given \( q \in S^3 \), one can combine (3.13) and (3.15) to define the inverse left hand (−) and right-hand (⊖) compound operators

\[
q^- \equiv (q^{-1})^+ \quad q^\ominus \equiv (q^{-1})^\oplus.
\]

Combining these definitions with the compound operators in (3.15) reveals the following identities:

\[
\begin{align*}
q^- q^+ &\equiv q^- q^- \equiv q^\ominus q^\ominus \equiv q^\ominus q^- \equiv \iota^+ \equiv \iota^- \equiv \iota^\ominus \equiv \iota^\oplus \equiv I_4 \\
q^+ \iota \equiv q \quad q^\ominus q^- \equiv \iota \quad q^\ominus \iota \equiv q \quad q^\ominus q^{-1} \equiv \iota \\
q^- \iota \equiv q^{-1} \quad q^- q \equiv \iota \quad q^\ominus \iota \equiv q^{-1} \quad q^\ominus q \equiv \iota.
\end{align*}
\]

Let \( q_{ji} \) be a unit quaternion parameterizing a rotation from coordinate frame \( i \) to coordinate frame \( j \), and let \( x_i, x_j \in \mathbb{R}^3 \) be the same vector expressed in coordinate frames \( i \) and \( j \), respectively. Then,

\[
\begin{bmatrix} 0 \\ x_j \end{bmatrix} = q_{ji}^+ \begin{bmatrix} 0 \\ x_i \end{bmatrix} q_{ji}^{-1},
\]

or more compactly, \( x_j = C(q_{ji})x_i \), where

\[
q_{ji}^+ q_{ji}^\ominus = \begin{bmatrix} 1 & 0^T \\ 0 & C(q_{ji}) \end{bmatrix}.
\]

and \( C(q_{ji}) \in SO(3) \).
3.4 Manifold Encapsulation of Rotations

Like the manifold encapsulation of axes outlined in Section 2.4, the topologies of both two-dimensional and three-dimensional rotations are manifolds, permitting the use of homeomorphisms to locally transform the parameterizations to and from $\mathbb{R}^n$ for use in state estimation algorithms. The homeomorphisms given in this section are based on those described by Hertzberg et al. [6].

3.4.1 Two-Dimensional Rotations

Because two-dimensional unit rotations share their topology with two-dimensional unit axes, the homeomorphism between $q \in S^1$ and $\theta \in \mathbb{R}$ (i.e., the two-dimensional rotation exponential and rotation logarithm) are identical to the axis exponential and axis logarithm given in (2.39) and (2.40), respectively; i.e.,

$$q = \exp(\theta) := \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix}$$

$$\theta = \log(q) := \text{atan2}(\epsilon, \eta).$$

Like two-dimensional axes, $\theta$ in (3.21) is the angle of $q$, and is denoted the rotation vector parameterization of the rotation. As expected, $\theta = \log(\exp(\theta))$ and $q = \exp(\log(q))$ for all $q$ and $\theta$.

Two-dimensional rotations and axes differ in their definition of the state estimation manifold operators. That is, for two-dimensional rotations,

$$\boxplus : S^1 \times S^1 \rightarrow S^1$$

$$\boxminus : S^1 \times S^1 \rightarrow s^1,$$

where $s^1 := (-\pi, \pi]$ is the real line of length $2\pi$ centred at the origin, with one end removed (i.e., $s^1 \subset \mathbb{R}$), which is illustrated in Figure 3.3. Note that, compared to the analogous operators on two-dimensional axes in (2.38), $s^1$ takes the place of $h^1 := (-\pi/2, \pi/2]$. Like $h^1$, $s^1$ is required to ensure that $\boxplus$ is injective.
Figure 3.3: A geometric representation of \( \theta \in s^1 \) is the real line of length \( 2\pi \) centred at the origin, with one end removed. The domains of the rotations vectors in (3.23) and (3.24) must be in \( s^1 \).

Let \( q, p \in S^1 \) and \( \theta \in s^1 \), where \( \theta \) is a perturbation of \( q \) that results in \( p \). The operation that applies this perturbation is identical to the analogous operation on two-dimensional axes in (2.39), except \( \theta \in s^1 \) (whereas in (2.39), \( \phi \in h^1 \)); i.e.,

\[
q \triangleq \theta := q^+ \exp(\theta) = p. \tag{3.23}
\]

Conversely, the operation that calculates the difference between \( q \) and \( p \) differs from the analogous operation on two-dimensional axes in (2.40); i.e.,

\[
p \triangleq q := \log(q^- p) = \theta. \tag{3.24}
\]

No special cases are required in (3.24) because the result is already guaranteed to be in \( s^1 \) (because the range of atan2 is \( s^1 \)).

### 3.4.2 Three-Dimensional Rotations

Unlike the close relationship between two-dimensional axes and rotations, the topologies of three-dimensional axes and rotations are different (\( S^2 \) and \( S^3 \), respectively). Let \( q \in S^3 \) and \( \theta \in \mathbb{R}^3 \). The homeomorphism between \( S^3 \) and \( \mathbb{R}^3 \) is

\[
q = \exp \left( \frac{\theta}{2} \right) := \begin{bmatrix} \cos \left( \frac{||\theta||}{2} \right) \\ \sin \left( \frac{||\theta||}{2} \right) \theta / ||\theta|| \end{bmatrix}, \tag{3.25}
\]

\[
\frac{\theta}{2} = \log(q) := \begin{cases} 
0 & \text{for } \epsilon = 0 \\
\text{atan2}(||\epsilon||, \eta) \frac{\epsilon}{||\epsilon||} & \text{otherwise.} \tag{3.26}
\end{cases}
\]

Note that \( \theta \) in (3.25) and (3.26) is the product of the axis and angle of rotation of \( q \) (i.e., \( \theta = \theta a \)). In other words, it is the rotation vector parameterization of a rotation. As expected, \( \theta = \log(\exp(\theta)) \) and \( q = \exp(\log(q)) \).
Figure 3.4: A geometric representation of $h^3$ is a ball of radius $\pi$ centred at the origin with half of its surface removed. This ball is the domain of the rotation vectors $\theta = (\theta_1, \theta_2, \theta_3)$ in (3.28) and (3.29).

The $\oplus$ and $\ominus$ operators that facilitate state estimation for unit quaternions are

\begin{align}
\oplus &: S^3 \times h^3 \to S^3 \\
\ominus &: S^3 \times S^3 \to h^3,
\end{align}

where $h^3$ is a ball of radius $\pi$ centred at the origin in $\mathbb{R}^3$, with half of its surface removed, which is illustrated in Figure 3.4. This topological space is denoted $h^3$ to signify that $h^3 \subset \mathbb{R}^3$ is analogous to $h^2 \subset \mathbb{R}^2$ in a higher dimension. Like $h^2$, $h^3$ is required to ensure the injectivity of the $\oplus$ operator.

Let $q, p \in S^3$ and $\theta \in h^3$, where $\theta$ is a perturbation of $q$ that results in $p$. Then

$$q \oplus \theta := q^+ \exp \left( \frac{\theta}{2} \right) = p. \quad (3.28)$$

The rotation vector $\theta$ parameterizing the difference between $q$ and $p$ represents ro-
Axis Mapping

Chapter 3: Rotations

tation from \( q \) to \( p \). Let \( d = q^\top p \). Then

\[
\theta = p \bowtie q := \begin{cases} 
2 \log (q^{-}p) & \text{if } d \geq 0 \\
2 \log (- (q^\top p)) & \text{otherwise.}
\end{cases}
\] (3.29)

The cases in (3.29) ensure that \( \theta \in h^3 \) and is unique for all \( q \) and \( p \).

3.5 Random Rotations

For the same reasons described in Section 2.5, parameterizing a rotation as a random variable (i.e., a random rotation) with a unit rotation (in two dimensions) or a unit quaternion (three dimensions) is troublesome because one degree (\( S^1 \)) and three degrees (\( S^3 \)) of freedom are each represented with an additional parameter and the unit constraint. As a result, a two-dimensional random rotation is represented by its mean \( \bar{q} \in S^1 \) and variance \( \sigma^2_{\delta\theta} \) such that

\[
\bar{q} = \bar{q} \bowtie \delta\theta, \quad \delta\theta \sim \mathcal{N}(0, \sigma^2_{\delta\theta}),
\] (3.30)

where the error \( \delta\theta \in s^1 \) is a perturbation of the true rotation \( \bar{q} \in S^1 \). Similarly, a three-dimensional random rotation is represented by its mean \( \bar{q} \in S^3 \) and covariance matrix \( P_{\delta\theta} \) such that

\[
\bar{q} = \bar{q} \bowtie \delta\theta, \quad \delta\theta \sim \mathcal{N}(0, P_{\delta\theta}),
\] (3.31)

where the error \( \delta\theta \in h^3 \) is a perturbation of the true rotation \( \bar{q} \in S^3 \). Like random axes, representing random rotations using Gaussian distributions is an approximation because of the bounded topological spaces of the errors. However, this approximation is usually sufficient for most practical applications of state estimation. For example, a Gaussian distribution representing a two-dimensional random rotation with \( 3\sigma^2_{\delta\theta} < \pi \) will enclose that vast majority of the probability within \( s^1 \).
“Even before you understand them, your brain is drawn to maps.”
— Ken Jennings

Chapter 4

Axis Mapping

This chapter formally introduces axis mapping in two (2DAM) and three (3DAM) dimensions. In addition to the axis map of the environment, axis mapping estimates the orientation path of a mobile platform (e.g., a wheeled robot, a UAV, or a handheld unit). The orientation path is a discrete sequence of rotations of the mobile platform as it traverses the environment. The orientation path and axis map are estimated using different types of measurements that correlate their entries with the global coordinate frame. A large optimization problem in the space of orientations and axes is constructed, whose solution maximizes the likelihood of all the measurements. The different components estimated by axis mapping are illustrated in Figure 4.1.

An interesting consequence of the non-injectivity of axis maps is the possibility of establishing correlations between non-sequential orientations without revisiting previously mapped areas (i.e., establishing loop closures), which relaxes a common requirement of creating accurate maps. This is accomplished by re-observing the axes of physically distinct but similarly orientated flat surfaces. Additionally, because axis mapping estimates only orientations and axes, the dimensionality of the optimization problem is significantly reduced compared to using a traditional mapping algorithm to create a large metric map of the environment (e.g., by using graph-based SLAM [17]) and then extracting axes from the map. Although this may reduce the computation required when compared to metric mapping, the non-injectivity of the axis map
Figure 4.1: Illustrations of axis mapping in (a) two-dimensions (2DAM) and (b) three-dimensions (3DAM). The orientation path of a mobile platform (i.e., its trajectory of orientations) and the axis map (axes of flat surfaces in the environment) are estimated in the global coordinate frame (North-East or North-East-Down). Note that physically distinct flat surfaces may share the same entry in the axis map (e.g., the solid and dashed normal vectors shown here result in only two entries in the axis map). As a result, loop closures can occur without revisiting previously mapped areas.
and the frequency of the resulting loop closures was the main driver in selecting the map representation.

This chapter begins by reviewing previous work related to axis mapping in Section 4.1. Next, Section 4.2 formally states the problem solved by axis mapping and explicitly defines some necessary variables. The optimization performed by axis mapping is the maximization of an objective function, whose construction is presented in Section 4.3 and is maximized using the method described in Section 4.4. In all sections, an attempt is made generalize variables and equations to both 2DAM and 3DAM, highlighting differences only when necessary.

4.1 Related Work

Considerable previous work has investigated the use of flat surfaces as features in the context of simultaneous localization and mapping (SLAM). These efforts can be divided into line-based methods for two-dimensional maps (Section 4.1.1), and plane-based methods for three-dimensional maps (Section 4.1.2). Furthermore, because axis maps are devoid of positional information, they are a form of directional data. As a result, the optimization performed by axis mapping (in particular for 2DAM) is a form of regression on directional data, which has some heritage in directional statistics (Section 4.1.3).

4.1.1 Line Mapping

In an early implementation, sonar was used to observe a priori line-based maps of the environment before relatively low-cost LiDAR became available [20]. Here, the idea of improving localization by incorporating geometric constraints in the environment was exploited. One of the first implementations using LiDAR provided online localization given an a priori line-based map [21], where it was noted that many indoor environments are suitable for this type of map representation. More recent
Axis Mapping

Chapter 4: Axis Mapping

efforts have used line-based maps in SLAM implementations by employing the position and axis of line segments to update the pose estimate of the robot. One EKF SLAM implementation [22] demonstrated the accuracy of using line segments in the SLAM state, whose compactness also reduces the burden of the computational complexity of SLAM. The popular line-based Orthogonal SLAM [23] takes advantage of an orthogonality assumption of the surfaces in the environment (e.g., the perpendicular walls common in most indoor areas). This approach was later extended to create a lightweight Rao-Blackwellized particle filter Orthogonal SLAM [24]. Here, only orthogonal lines extracted from the environment are used to update the SLAM state. By fixing the possible axes of lines to an absolute reference (and discarding lines that do not meet this criterion), remarkable mapping and localization accuracy is achieved by limiting the growth of heading errors. Recent work [25] has explored automatically identifying additional types of structure (e.g., point-to-point distances and circles) and incorporating these constraints into the graph-based optimization of the map.

Unlike 2DAM, these implementations generally attempt to simplify metric maps by fitting line segments to flat surfaces and using them as features. In contrast, because 2DAM uses only the axes of flat surfaces, the resulting axis map is not a physical representation of the layout of the environment, but rather it is a description of its dominant orientations. It is shown in Chapter 5 that 2DAM provides a lightweight alternative to traditional line mapping algorithms, while still producing maps that can substantially improve localization.

4.1.2 Plane Mapping

An early effort by Weingarten and Siegwart [26] significantly reduced the dimensionality of the map in EKF-SLAM by using a representation consisting of polygons comprised of planar segments. Nguyen et al. [27] observed that planar surfaces tend to be orthogonal in indoor environments, and were able to build extremely accurate in-
door maps by exploiting this environmental structure. Pathak et al. [28] emphasized that plane registration is a very accurate and robust at determining rotations between poses, justifying optimization of only the translation components of the constructed pose-graph and greatly simplifying the computation of the final map. Recent work by Taguchi et al. [29] demonstrated that real-time SLAM in an indoor environment with a handheld sensor is possible by using a combination of point and plane registration. These methods all compute correspondences between planar surfaces based on their positions and orientations by using plane-to-plane or point-to-plane matching. In contrast, 3DAM uses axis-to-axis matching (i.e., only the orientations of the planes), which allows for correspondences among axes even if they represent physically distinct surfaces.

In the special case when the environment consists of sets of orthogonal planar surfaces (e.g., most indoor spaces and urban areas), Straub et al. [30] developed a model called a Mixture-of-Manhattan-Frames (MMF). Similarly, Schindler et al. [31] developed the similar Atlanta World (AW) model that assumes all frames in the set share an axis. These frameworks parameterize the environment as a mixture of orthogonally-coupled clusters in $S^2$. Although similar to 3DAM in that the underlying structure of a planar environment is extracted, the MMF and AW models are a tool for scene representation and are themselves not a mapping algorithm (although they could be incorporated into one). Furthermore, 3DAM is designed for environments that do not necessarily contain orthogonal planar surfaces (e.g., naturally occurring rock faces).

### 4.1.3 Regression Analysis on Directional Data

One of the first to develop a regression analysis that specifically accounts for directional data was Gould [32], who developed a technique to model the spatial orientation of vectorcardiograms (i.e., the electric activity of the heart). Gould modelled a vector of linear covariates $\mathbf{x}$ (i.e., independent variables) as the mean $\mu + \beta^\top \mathbf{x}$ of a von
Mises distribution [1], and solved for the maximum-likelihood estimate of the vector of parameters $\beta$. This approach yields an infinite number of solutions (which Gould addressed with heuristics), and was later improved [33, 34] by wrapping $\beta^T x$ to the unit circle. A regression technique where the independent variables $x$ are also directional random variables was developed by Sarma and Jammalamadaka [35]. Here, the real and imaginary components of a directional observation $e^{i\theta} = \cos(\theta) + i \sin(\theta)$ are each modelled as a finite Fourier series (given an angular covariate $\alpha$). The Fourier coefficients are then determined to maximize the likelihood of all the observations. Lund [36] later developed a similar approach that considers the case of a combination of both linear and angular covariates. Although developed for directional data, these regression techniques do not address the case where the parameters being determined (e.g., $\beta$) are themselves directional, which is the case for axis mapping.

4.2 Problem Description

Axis mapping finds an estimate of $x = (Q, A)$, which consists of the orientation path of the mobile platform $Q = (q_1, \ldots, q_n)$ and an axis map of the environment $A = (m_1, \ldots, m_m)$. The orientation path is a sequence of discrete orientations of the mobile platform as it passes through the environment, and an axis map is a discrete set of axes of flat surfaces in the environment. The topologies of these variables depend on the dimensionality of the axis mapping problem, as illustrated in Figure 4.2. Table 4.1 provides descriptions of the axis mapping variables and their topologies for two-dimensional (2DAM) and three-dimensional (3DAM) axis mapping.

As presented in Table 4.1, axis mapping simultaneously estimates $Q$ and $A$ by using some combination of three types of observations: (i) observations of orientation path entries $z_{qi}$, (ii) observations of rotations between orientation path entries $z_{qi,qj}$, and (iii) observations of axes of flat surfaces in the environment $z_{qi,mj}$. The methods used to obtain these observations are dependent on the sensors used and the dimension-
Figure 4.2: Examples of (a) two-dimensional axis mapping (2DAM) and (b) three-dimensional axis mapping (3DAM). The topologies of the mobile platform orientation $q_i$ and an axis map entry $m_j$ depend on the dimensionality of the problem. In 2DAM, $q_i \in S^1$ and $m_j \in S^1$, and in 3DAM, $q_i \in S^3$ and $m_j \in S^2$.

Table 4.1: Descriptions and topological spaces of the axis mapping variables.

<table>
<thead>
<tr>
<th>Variable</th>
<th>2DAM</th>
<th>3DAM</th>
<th>Description</th>
<th>Topology 2DAM</th>
<th>Topology 3DAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q$</td>
<td>$Q$</td>
<td></td>
<td>Orientation path $Q = (q_1, \ldots, q_n)$</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$q_i$</td>
<td>$q_i$</td>
<td></td>
<td>Rotation from global to mobile platform coordinate frame ($i$-th entry of $Q$)</td>
<td>$S^1$</td>
<td>$S^3$</td>
</tr>
<tr>
<td>$A$</td>
<td>$A$</td>
<td></td>
<td>Axis map $A = (m_1, \ldots, m_m)$</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$m_j$</td>
<td>$m_j$</td>
<td></td>
<td>Axis expressed in the mobile platform coordinate frame ($j$-th entry of $A$)</td>
<td>$S^1$</td>
<td>$S^2$</td>
</tr>
<tr>
<td>$x$</td>
<td>$x$</td>
<td></td>
<td>Orientation path and axis map $x = (Q, A)$</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$z_{q_i}$</td>
<td>$z_{q_i}$</td>
<td></td>
<td>Observation of $q_i$</td>
<td>$S^1$</td>
<td>$S^3$</td>
</tr>
<tr>
<td>$\sigma^2_{q_i}$</td>
<td>$R_{q_i}$</td>
<td>(Co)variance of $z_{q_i}$</td>
<td>$\mathbb{R}$</td>
<td>$\mathbb{R}^{3 \times 3}$</td>
<td></td>
</tr>
<tr>
<td>$z_{q_i,q_j}$</td>
<td>$z_{q_i,q_j}$</td>
<td></td>
<td>Observation of rotation from $q_i$ to $q_j$</td>
<td>$S^1$</td>
<td>$S^3$</td>
</tr>
<tr>
<td>$\sigma^2_{q_i,q_j}$</td>
<td>$R_{q_i,q_j}$</td>
<td>(Co)variance of $z_{q_i,q_j}$</td>
<td>$\mathbb{R}$</td>
<td>$\mathbb{R}^{3 \times 3}$</td>
<td></td>
</tr>
<tr>
<td>$z_{q_i,m_j}$</td>
<td>$z_{q_i,m_j}$</td>
<td></td>
<td>Observation of $m_j$ made from $q_i$ expressed in the mobile platform coordinate frame</td>
<td>$S^1$</td>
<td>$S^2$</td>
</tr>
<tr>
<td>$\sigma^2_{q_i,m_j}$</td>
<td>$R_{q_i,m_j}$</td>
<td>(Co)variance of $z_{q_i,m_j}$</td>
<td>$\mathbb{R}$</td>
<td>$\mathbb{R}^{2 \times 2}$</td>
<td></td>
</tr>
</tbody>
</table>
The entries of $x$ are modelled as random variables; that is, $q_i$ is a random rotation and $m_j$ is a random axis. An estimate of $x$ is represented by its mean $\hat{x} = (\hat{Q}, \hat{A})$, where $\hat{Q} = (\hat{q}_1, \ldots, \hat{q}_n)$ and $\hat{A} = (\hat{m}_1, \ldots, \hat{m}_m)$ are the means of the respective random rotations and random axes. Similarly, the observations are also modelled as random variables; that is, $z_{q_i}$ and $z_{q,q_j}$ are random rotations and $z_{q,m_j}$ is a random axis. Because the (co)variances of the observations are used in the process of estimating $x$, their topological spaces are explicitly stated in Table 4.1.

Axis mapping determines the estimate $\hat{x}$ that maximizes the likelihood of all the observations simultaneously. It requires an initial estimate of $\hat{x}$, which is obtained by using different methods that are implementation and sensor dependent. In general, axis mapping answers the following question:

*Given an initial estimate $\hat{x}$ and a set of observations of type $z_{q_i}$, $z_{q,q_j}$, and $z_{q,m_j}$,*
Figure 4.3: Four possible axis mapping factor graph configurations. In these examples, the orientation path has four entries $q_1, \ldots, q_4$ and the axis map has two entries $m_1, m_2$. (a) All observation types are frequently available. (b) The orientation of the mobile platform is not directly observable with the sensor data. Only one observation of the type $z_{q_i}$ is available, measuring the initial orientation. (c) The orientation of the mobile platform is directly observable with the sensor data and no observations of the type $z_{q_i}$ are used. (d) Only one observation of the type $z_{q_i}$ is available, measuring the initial orientation of the mobile platform. No other observations of types $z_{q_i}$ or $z_{q_i,q_j}$ are available.
what is the optimal estimate $\hat{x}^*$ that maximizes the likelihood of the observations?

Axis mapping solves this problem by building an objective function whose input is an estimate $\hat{x}$ and whose output is proportional to the likelihood of all the observations. The optimal estimate $\hat{x}^*$ is then determined by maximizing the objective function by using an optimization algorithm that carefully addresses the topological spaces of the entries of $x$ and the observations.

### 4.3 Building the Objective Function

The objective function $J(\hat{x})$ maps $\hat{x}$ to the log-likelihood of all the observations. As a result, maximizing $J(\hat{x})$ is equivalent to determining the maximum likelihood estimate of $\hat{x}$. This section details how the objective function is constructed. In Section 4.3.1, the log-likelihood of a generic observation in $\mathbb{R}^n$ is derived to demonstrate how it is related to the probability density function of the observation. The log-likelihoods of the axis mapping observations are then derived in Section 4.3.2.

#### 4.3.1 Log-Likelihood of a Gaussian Random Variable

The probability density function (PDF) of a Gaussian random variable with mean $\hat{z} \in \mathbb{R}^n$ and covariance $R \in \mathbb{R}^{n \times n}$ is

$$f(z|\hat{z}, R) = \frac{1}{\sqrt{(2\pi)^n \det R}} \exp \left( -\frac{1}{2} (z - \hat{z})^\top R^{-1} (z - \hat{z}) \right),$$

which evaluates the relative likelihood of an observation $z \in \mathbb{R}^n$ given its covariance and the expected observation (i.e., the mean). In other words, $f(z|\hat{z}, R)$ is larger for values of $z$ that are closer to what is expected. The difference between $z$ and the expected observation is called the error $e$, where $e = z - \hat{z}$ in (4.1).

Suppose that the error is a function of some variable $x$, such that the error function $e(x)$ maps $x$ to the error given an observation $z$. One can then maximize the likelihood
of \( z \) by determining the value of \( x \) that maximizes \( f(z|x, R) \), where

\[
f(z|x, R) = \frac{1}{\sqrt{(2\pi)^n \det R}} \exp \left( -\frac{1}{2} \mathbf{e}(x) \mathbf{R}^{-1} \mathbf{e}(x) \right).
\]  

(4.2)

Maximizing \( f(z|x, R) \) involves setting its partial derivative with respect to \( x \) equal to zero. This process can often be made simpler by first calculating its natural logarithm. This does not affect the maximization problem because the natural logarithm is a monotonic function (i.e., maximizing the natural logarithm of a function gives the same result as maximizing the function itself). The resulting log-likelihood of \( z \) is

\[
\ln (f(z|x, R)) = -\alpha \frac{1}{2} \mathbf{e}(x) \mathbf{R}^{-1} \mathbf{e}(x),
\]

(4.3)

where \( \alpha \) is a constant independent of \( x \) and therefore does not affect the maximization. As a result, one can define an objective function

\[
f(x) = -\frac{1}{2} \mathbf{e}(x) \mathbf{R}^{-1} \mathbf{e}(x)
\]

(4.4)

to be maximized, whose range is proportional to the likelihood of \( z \). The argument of the objective function whose value is to be optimized (\( x \) in this case) is called the design parameter of the objective function.

### 4.3.2 Log-Likelihood of Axis Mapping Observations

When performing axis mapping, the design parameter of the objective function is \( x \); that is, optimal estimates of the orientation path and axis map are determined that maximize the objective function (i.e., the likelihood of the observations). The error functions of the axis mapping observations \( z_{q_i}, z_{q_i q_j}, \) and \( z_{q_i m_j} \) are

\[
\mathbf{e}_{q_i}(x) = z_{q_i} \equiv q_i,
\]

(4.5a)

\[
\mathbf{e}_{q_i q_j}(x) = z_{q_i q_j} \equiv (q_i^{-1} q_j),
\]

(4.5b)

\[
\mathbf{e}_{q_i m_j}(x) = z_{q_i m_j} \equiv (C(q_i)m_j),
\]

(4.5c)
where $C(q_i) := q_i^+$ when $q_i \in S^1$, and is defined in (3.19) when $q_i \in S^3$. Each error function in (4.5) is the difference between the observation and the expected observation constructed from entries of $x$. Note that there is a maximum error in each of the cases in (4.5). For example, if $z_{q_i}, q_i \in S^1$ in (4.5a) are two-dimensional unit rotations (Section 3.2), the range of $e_{q_i}(x)$ is $s^1$ (Figure 3.3 on page 37), which results in a maximum error of $\pi$.

The log-likelihoods of the observations (with the constants omitted) are

\[
\ln \left( f \left( z_{q_i} | \hat{x}, R_{q_i} \right) \right) \propto -\frac{1}{2} e_{q_i}(x)^\top R_{q_i}^{-1} e_{q_i}(x), \quad (4.6a)
\]

\[
\ln \left( f \left( z_{q_i q_j} | \hat{x}, R_{q_i q_j} \right) \right) \propto -\frac{1}{2} e_{q_i q_j}(x)^\top R_{q_i q_j}^{-1} e_{q_i q_j}(x), \quad (4.6b)
\]

\[
\ln \left( f \left( z_{q_i m_j} | \hat{x}, R_{q_i m_j} \right) \right) \propto -\frac{1}{2} e_{q_i m_j}(x)^\top R_{q_i m_j}^{-1} e_{q_i m_j}(x). \quad (4.6c)
\]

Each observation is considered conditionally independent, which makes their joint likelihood simply the product of their individual likelihoods. This results in a joint log-likelihood that is the sum of their individual log-likelihoods. As a result, the objective function $J(x)$ over all the axis mapping observations is

\[
J(x) = -\frac{1}{2} \sum_{q_i} \left( e_{q_i}(x)^\top R_{q_i}^{-1} e_{q_i}(x) + \sum_{q_j} e_{q_i q_j}(x)^\top R_{q_i q_j}^{-1} e_{q_i q_j}(x) + \sum_{m_j} e_{q_i m_j}(x)^\top R_{q_i m_j}^{-1} e_{q_i m_j}(x) \right), \quad (4.7)
\]

where there may not be observations $z_{q_i}, z_{q_i q_j},$ and $z_{q_i m_j}$ for all combinations of $q_i, q_j,$ and $m_j$.

### 4.4 Maximizing the Objective Function

Maximizing the objective function is relatively straightforward when the error functions are linear with respect to the design parameter. For example, given observations $z_i \in \mathbb{R}^m$, design parameter $x \in \mathbb{R}^n$, observation matrices $C_i \in \mathbb{R}^{m \times n}$, and observation
Axis Mapping

covariance matrices $R_i \in \mathbb{R}^{m \times m}$, the objective function $J(x)$ is quadratic in $x$; i.e.,

$$J(x) = -\frac{1}{2} \sum_i (z_i - C_i x)^\top R_i^{-1} (z_i - C_i x)$$

$$= -\frac{1}{2} \sum_i \left( z_i^\top R_i^{-1} z_i - 2x^\top C_i^\top R_i^{-1} z_i + x^\top C_i^\top R_i^{-1} C_i x \right).$$  \hspace{1cm} (4.8)

By defining the terms

$$a = \sum_i z_i^\top R_i^{-1} z_i, \quad b = \sum_i C_i^\top R_i^{-1} z_i, \quad H = \sum_i C_i^\top R_i^{-1} C_i,$$  \hspace{1cm} (4.9)

where $a \in \mathbb{R}$, $b \in \mathbb{R}^n$, and $H \in \mathbb{R}^{n \times n}$, we have

$$J(x) = -\frac{1}{2} a + x^\top b - \frac{1}{2} x^\top H x.$$  \hspace{1cm} (4.10)

The design parameter $x^*$ that maximizes the objective function is

$$x^* = \arg \max_x J(x),$$  \hspace{1cm} (4.11)

which is calculated by setting the partial derivative of $J(x)$ with respect to $x$; i.e.,

$$\frac{\partial J(x)}{\partial x} = b - Hx^* = 0$$

$$\Rightarrow Hx^* = b.$$  \hspace{1cm} (4.12)

Note that one does not normally take the direct inverse of $H$ to solve (4.12). The construction in (4.9) often leads to a structure of $H$ that allows for a more efficient means of solving the linear system of equations for $x^*$.

It is possible to directly solve for the optimal design parameter in (4.12) because the error functions are linear (which makes the objective function quadratic) with respect to the design parameter. This leads to a linear system of equations after setting its partial derivative to zero. Unfortunately, the axis mapping observation error functions in (4.5) are not necessarily linear; therefore, they must be linearized to maximize the objective function in this way.
4.4.1 Linearizing the Axis Mapping Observation Errors

Linearizing the observation error functions requires differentiation with respect to the design parameter $x$. However, because $x$ is a constrained parameterization (i.e., it has more parameters than degrees of freedom), differentiation must be performed with respect to changes in its degrees of freedom rather than its parameters. As a result, it is useful to explicitly define the topological space of $x$ and to define a new topological space that represents an unconstrained parameterization of $x$.

Recall that $x$ is the Cartesian product of the entries of $Q$ and $A$. The topological space of $x$ is defined as

$$
\mathcal{S} := \begin{cases} 
S^1 \times \ldots \times S^1 \times S^1 \times \ldots \times S^1 & \text{for 2DAM} \\
\text{orientation path} & \text{axis map} \\
S^3 \times \ldots \times S^3 \times S^2 \times \ldots \times S^2 & \text{for 3DAM.} \\
\text{orientation path} & \text{axis map}
\end{cases}
$$

(4.13)

Now let $\bar{x} = \log(x)$ be the unconstrained parameterization of $x$, where

$$
\log(x) := (\log(q_1), \ldots, \log(q_n), \log(m_j), \ldots, \log(m_m)).
$$

(4.14)

Then the topological space of $\bar{x}$ is defined as $\mathcal{R} \subset \mathbb{R}^n$, where

$$
\mathcal{R} := \begin{cases} 
s^1 \times \ldots \times s^1 \times h^1 \times \ldots \times h^1 & \text{for 2DAM} \\
\text{orientation path} & \text{axis map} \\
h^3 \times \ldots \times h^3 \times h^2 \times \ldots \times h^2 & \text{for 3DAM.} \\
\text{orientation path} & \text{axis map}
\end{cases}
$$

(4.15)

The axis mapping observation error functions have the form $e(x) = z \circ h(x)$, which is the difference between an observation $z$ and a function $h$ that maps the entries of $x$ to the expected observation. Given an initial estimate $\hat{x}$ of $x$, the Taylor series expansion of $e(x)$ is

$$
e(\hat{x} \oplus \Delta x) = e(\hat{x}) + \left. \frac{\partial e(x)}{\partial x} \right|_{\hat{x}} \Delta x + \frac{1}{2} \left. \Delta x^\top \frac{\partial^2 e(x)}{\partial x \partial x^\top} \right|_{\hat{x}} \Delta x + \ldots
$$

(4.16)
where $\Delta x \in \mathbb{R}$ is a perturbation of the estimate $\hat{x}$. As mentioned above, the partial derivatives in (4.16) must be taken with respect to the unconstrained parameterization. The domain and range of each error function depends on the specific type of observation, which determines the mapping performed by the $\Box$ operator. For 2DAM and 3DAM, the following mappings are possible:

$$
\Box : S^1 \times S^1 \to h^1 \text{ (or } s^1), \quad \Box : S^2 \times S^2 \to h^2, \quad \Box : S^3 \times S^3 \to h^3.
$$

(4.17)

Recall that $h^1, s^1 \subset \mathbb{R}$, $h^2 \subset \mathbb{R}^2$, and $h^3 \subset \mathbb{R}^3$. The evaluation of (4.16) for each of these scenarios follows.

**Error Functions Mapping** $S^1 \times S^1 \to h^1 \text{ or } S^1 \times S^1 \to s^1$

These scenarios occur when the error is the difference between two-dimensional unit axes or unit rotations. In 2DAM, all three error functions in (4.5) are this type of error. Using $e_{q_i}(x)$ in (4.5a) as an example, where $z_{q_i} \in S^1$ is an observation of the $i$-th entry of the orientation path $q_i \in S^1$, the error function is

$$
e_{q_i}(x) = z_{q_i} \Box q_i
$$

$$
= \log (q_i^{-1}z_{q_i})
$$

$$
= \log \begin{bmatrix} \cos (\theta_z - \theta_i) \\ \sin (\theta_z - \theta_i) \end{bmatrix}
$$

$$
= \text{atan2} (\sin (\theta_z - \theta_i), \cos (\theta_z - \theta_i)),
$$

(4.18)

which makes use of Proposition 2.5 on page 16, where $\theta_z = \log(z_{q_i})$ and $\theta_i = \log(q_i)$. The first partial derivative in the Taylor series expansion evaluated at an estimate $\hat{x}$ is then

$$
\frac{\partial e(x)}{\partial x} \bigg|_{\hat{x}} = \frac{\partial \text{atan2} (\sin (\theta_z - \theta_i), \cos (\theta_z - \theta_i))}{\partial x} \bigg|_{\hat{x}}
$$

$$
= \begin{bmatrix} 0_1 & \ldots & -1_i & \ldots & 0_n & 0_1 & \ldots & 0_m \end{bmatrix},
$$

(4.19)
where \( \mathbf{x} = (\theta_1, \ldots, \theta_n, \phi_1, \ldots, \phi_m) \) for \( n \) orientations in the orientation path and \( m \) axes in the axis map. As a result, \( \frac{\partial^k e(\mathbf{x})}{\partial \mathbf{x}^k} \bigg|_{\hat{x}} = 0^T \) (4.20) for all \( k \geq 2 \). Therefore, by substituting (4.19) and (4.20) into (4.16), the full Taylor series expansion of \( e(\mathbf{x}) \) evaluated at \( \hat{x} \) is
\[
e_i(\hat{x} \oplus \Delta \mathbf{x}) = z_{q_i} \ominus \hat{q}_i + \begin{bmatrix} 0 \ldots -1 \ldots 0_n \ 0_1 \ldots 0_m \end{bmatrix} \Delta \mathbf{x} \quad (4.21)
\]
which is linear with respect to \( \Delta \mathbf{x} \) and is not an approximation.

**Error Functions Mapping** \( S^2 \times S^2 \rightarrow h^2 \)

This scenario occurs when the error is the difference between three-dimensional unit axes. In 3DAM, \( e_{q_i,m_j}(\mathbf{x}) \) defined in (4.5c) is this type of error. Using this error function as an example, where \( z_{q_i,m_j} \) is an observation of \( j \)-th entry of the axis map \( m_j \in S^2 \) taken from the \( i \)-th entry of the orientation path \( q_i \in S^3 \), the error function is
\[
e_{q_i,m_j}(\mathbf{x}) = z_{q_i,m_j} \ominus (C(q_i)m_j)
\]
\[
= \log \left( (C(q_i)m_j)^{-1} z_{q_i,m_j} \right) \quad (4.22)
\]

Analytically calculating the partial derivative of (4.22) with respect to \( \mathbf{x} \) is exceedingly difficult (as are the higher order derivatives). As a result, the partial derivative is calculated numerically; i.e.,
\[
\frac{\partial e_{q_i,m_j}(\mathbf{x})}{\partial \mathbf{x}} \bigg|_{\hat{x}} \approx \begin{bmatrix} 0_1 \ldots a_1 \ a_2 \ a_3 \ldots 0_n \ 0_1 \ldots b_1 \ b_2 \ldots 0_m \end{bmatrix} \quad (4.23)
\]
where given a small perturbation \( \epsilon \in \mathbb{R} \), and letting \( \log(q_i) = (\theta_1, \theta_2, \theta_3) \) and \( \log(m_j) = (\phi_1, \phi_2) \),
\[
a_1 = \frac{\partial e_{q_i,m_j}(\mathbf{x})}{\partial \theta_1} \bigg|_{\hat{x}} = \frac{z_{q_i,m_j} \ominus C \left( \hat{q}_i \ominus \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right) \hat{m}_j - z_{q_i,m_j} \ominus C \left( \hat{q}_i \ominus \begin{bmatrix} \epsilon \\ 0 \end{bmatrix} \right) \hat{m}_j}{2\epsilon}
\]
The perturbations in (4.24) are applied along the three degrees of freedom of \( \hat{q}_i \) and the two degrees of freedom of \( \hat{m}_j \). A first-order Taylor series expansion is then used to approximate \( e_{q,m_j}(x) \) as a linear function with respect to \( \Delta x \); i.e.,

\[
\begin{align*}
e_{q,m_j}(\hat{x} \oplus \Delta x) & \approx z_{q,m_j} \oplus (C(\hat{q}_i)\hat{m}_j) + \frac{\partial e_{q,m_j}(x)}{\partial x} \bigg|_{\hat{x}} \Delta x. \\
& = \log \left( q_i^{-1} z_{q,i} \right) \\
& = \log \left( \left[ \begin{array}{c} \eta_q \eta_z + \epsilon_q^T \epsilon_z \\
\eta_q I_{3 \times 3} + \epsilon_q^X \epsilon_z - \eta_z \epsilon_q \end{array} \right] \right),
\end{align*}
\]

(4.25)
where \( \eta_q, \kappa_q, \eta_z, \kappa_z \) are the scalar and vector parts of \( q_i \) and \( z_q \), respectively. Once again, analytically calculating the partial derivatives of (4.26) is difficult. As a result, the partial derivative is calculated numerically; i.e.,

\[
\frac{\partial e_{q_i}(x)}{\partial x} \bigg|_{\hat{x}} \approx [0_1 \ldots a_1 a_2 a_3 \ldots 0_n \ 0_1 \ldots 0_m]
\]  

(4.27)

where given a small perturbation \( \epsilon \in \mathbb{R} \), and letting \( \log(q_i) = (\theta_1, \theta_2, \theta_3) \),

\[
a_1 = \frac{\partial e_{q_i}(x)}{\partial \theta_1} \bigg|_{\hat{x}} = \frac{z_{q_i} \Box (\hat{q}_i \Box \left[ \begin{array}{c} \epsilon \\ 0 \\ 0 \end{array} \right]) - z_{q_i} \Box (\hat{q}_i \Box \left[ \begin{array}{c} \epsilon \\ 0 \\ 0 \end{array} \right])}{2\epsilon}
\]

\[
a_2 = \frac{\partial e_{q_i}(x)}{\partial \theta_2} \bigg|_{\hat{x}} = \frac{z_{q_i} \Box (\hat{q}_i \Box \left[ \begin{array}{c} 0 \\ \epsilon \\ 0 \end{array} \right]) - z_{q_i} \Box (\hat{q}_i \Box \left[ \begin{array}{c} 0 \\ \epsilon \\ 0 \end{array} \right])}{2\epsilon}
\]

\[
a_3 = \frac{\partial e_{q_i}(x)}{\partial \theta_3} \bigg|_{\hat{x}} = \frac{z_{q_i} \Box (\hat{q}_i \Box \left[ \begin{array}{c} 0 \\ 0 \\ \epsilon \end{array} \right]) - z_{q_i} \Box (\hat{q}_i \Box \left[ \begin{array}{c} 0 \\ 0 \\ \epsilon \end{array} \right])}{2\epsilon}
\]

(4.28)

The perturbations in (4.28) are applied along the three degrees of freedom of \( \hat{q}_i \). The first-order Taylor series expansion of \( e_{q_i}(x) \) is then

\[
e_{q_i}(\hat{x} \Box \Delta x) \approx z_{q_i} \Box \hat{q}_i + \frac{\partial e_{q_i}(x)}{\partial x} \bigg|_{\hat{x}} \Delta x.
\]

(4.29)

which is linear with respect to \( \Delta x \).

### 4.4.2 Maximizing the Axis Mapping Objective Function

In Section 4.4.1, Taylor series expansions are used to linearize the different types of observation error functions with respect to \( \Delta x \in \mathcal{R} \), which is a change in \( x \) expressed in its unconstrained parameterization. Given a set of observations, let \( e_i(x) \) be the \( i \)-th
observation error function, whose linearization is of the form

$$e_i(\hat{x} \oplus \Delta x) \approx e_i(x) + \frac{\partial e_i(x)}{\partial x} \bigg|_{\hat{x}} \Delta x,$$  \hspace{1cm} (4.30)

which is one of (4.21) (where it is not an approximation), (4.25), or (4.29). Recall that the objective function to be maximized is

$$J(x) = -\frac{1}{2} \sum_i e_i(x) \top R^{-1}_i e_i(x)$$  \hspace{1cm} (4.31)

where $R_i$ is the covariance matrix of the $i$-th observation. Linearizing $e_i(x)$ and making the substitutions

$$e_i(\hat{x}) = e_i(x) \bigg|_{\hat{x}}, \quad C_i = \frac{\partial e_i(x)}{\partial x} \bigg|_{\hat{x}}$$  \hspace{1cm} (4.32)

results in the linearized objective function

$$J(\hat{x} \oplus \Delta x) = -\frac{1}{2} \sum_i (e_i(\hat{x}) + C_i \Delta x) \top R^{-1}_i (e_i(\hat{x}) + C_i \Delta x),$$  \hspace{1cm} (4.33)

which is quadratic with respect to $\Delta x$. Maximizing $J(\hat{x} \oplus \Delta x)$ proceeds similarly to (4.8)–(4.12), leading to the linear system of equations

$$H \Delta x^* = b,$$  \hspace{1cm} (4.34)

where $\Delta x^*$ is the optimal perturbation of the linearized objective function and

$$b = \sum_i C_i \top R^{-1}_i e_i(\hat{x}), \quad H = \sum_i C_i \top R^{-1}_i C_i,$$  \hspace{1cm} (4.35)

are the Jacobian and Hessian, respectively, of the objective function evaluated at $\hat{x}$. Put differently,

$$b = \frac{\partial J(x)}{\partial x} \bigg|_{\hat{x}}, \quad H \approx \frac{\partial^2 J(x)}{\partial x \partial x \top} \bigg|_{\hat{x}},$$  \hspace{1cm} (4.36)

where the Hessian approximation is only accurate near $J(x) = 0$ because of the use of a first-order Taylor series approximation of the error functions. Consequently, the Hessian in (4.36) is not an approximation when the first-order Taylor series is exact,
as is the case in (4.21). At this point, it is important to distinguish between scenarios when the error function linearization is exact (e.g., 2DAM) and when the error function linearization is an approximation (e.g., 3DAM).

**Exact Linearization**

If the error functions are exactly linear with respect to the perturbation $\Delta x$, then the linearized objective function $J(\hat{x} \oplus \Delta x)$ in (4.33) is not an approximation and

$$\Delta x^* = H^{-1}b$$

(4.37)

is the difference between the initial estimate $\hat{x}$ and the optimal estimate $\hat{x}^*$; i.e.,

$$\hat{x}^* = \hat{x} \oplus \Delta x^* = \begin{bmatrix} \hat{q}_1 \oplus \Delta \theta_1 \\ \vdots \\ \hat{q}_n \oplus \Delta \theta_n \\ \hat{m}_1 \oplus \Delta \phi_1 \\ \vdots \\ \hat{m}_m \oplus \Delta \phi_m \end{bmatrix}.$$  

(4.38)

In other words, determining the orientation path and axis map that maximize the likelihood of the observations requires a single update of an initial estimate.

It is worth highlighting why it is necessary to maximize the objective function by a single update of an initial estimate, rather than directly solving for the optimal estimate in the linear case outlined in (4.8)–(4.12). In both of these cases, the error functions are linear; however, in the axis mapping case, the fact that the variables are in $S^1$ and not $\mathbb{R}$ make directly solving for the optimal estimate problematic. For example, let $z_1, z_2 \in S^1$ be two observations of a rotation $q_1 \in S^1$, where

$$\theta_{z_1} = \log(z_1) = -177^\circ, \quad \theta_{z_2} = \log(z_2) = 179^\circ, \quad \theta_{q_1} = \log(q_1) = -179^\circ.$$  

(4.39)
Now suppose one attempted to directly solve for $\theta_{q_1}$ using the linear error functions

$$e_1(x) = \theta_z - \theta_{q_1}, \quad e_2(x) = \theta_z - \theta_{q_1}.$$  \hspace{1cm} (4.40)

Here, there is no value of $\theta_{q_1}$ that properly represents the actual errors correctly in $\mathbb{R}$ (e.g., $\theta_{q_1} = -179^\circ$ results in $e_1(x) = 2^\circ$ and $e_2(x) = 358^\circ$). On the other hand, suppose we had an initial estimate $\hat{\theta}_{q_1} = -160^\circ$. Then, using the linearized error function in (4.21),

$$e_1(x) = z_1 \boxtimes \hat{\theta}_1 - \Delta\theta_{q_1} = -17^\circ - \Delta\theta_{q_1}, \quad e_2(x) = z_2 \boxtimes \hat{\theta}_1 - \Delta\theta_{q_1} = -21^\circ - \Delta\theta_{q_1}.$$  \hspace{1cm} (4.41)

Here, assuming the observations have the same variance, maximizing the objective function will produce the solution $\Delta\theta_{q_1} = -19^\circ$, and

$$\hat{\theta}^* = \hat{\theta} \boxtimes \Delta\theta_{q_1} = \exp(-160^\circ) \boxtimes -19^\circ = \exp(-179^\circ).$$  \hspace{1cm} (4.42)

### Approximate Linearization

If the linearization of the error functions is an approximation, the optimal perturbation $\Delta x^*$ is only optimal with respect to the linearized objective function in (4.33) and not the actual objective function in (4.31). As a result, it is possible that $\hat{x} \boxtimes \Delta x^*$ will actually decrease the value of the objective function. There are many algorithms that attempt to solve this nonlinear optimization problem, differing mostly in how they calculate $\Delta x^*$ (Madsen et al. [37] provide an overview of several algorithms). In short, these algorithms begin with an initial estimate $\hat{x}$, calculate $\Delta x^*$, update the initial estimate via $\hat{x} \boxtimes \Delta x^*$ if it increases the objective function, and iterate these steps until some convergence criteria are met. Axis mapping uses a variant of the popular Levenberg-Marquardt (LM) algorithm [38] to maximize the objective function.

Instead of iteratively solving (4.34), LM iteratively solves

$$(H + \alpha I) \Delta x^* = b,$$  \hspace{1cm} (4.43)
where the $\alpha \in \mathbb{R}$ is a nonnegative damping factor and $I$ is the identity matrix with the same dimensions as $H$. For sufficiently large $\alpha$ (with respect to the entries of $H$),

$$\Delta x^* \approx \frac{1}{\alpha} b$$  \hspace{0.5cm} (4.44)

which is a small perturbation in the direction of $b$, which by (4.35) is the Jacobian of the objective function at the current estimate (i.e., the direction of steepest ascent). Conversely, for sufficiently small $\alpha$,

$$\Delta x^* \approx H^{-1} b,$$  \hspace{0.5cm} (4.45)

which is again a perturbation in the direction of steepest ascent, but inversely scaled by the curvature of the objective function at the current estimate. Simply put, (4.45) scales the perturbation such that entries with larger curvatures have smaller magnitudes. Iteratively calculating the perturbations with (4.45) is an approximation of Newton’s method\footnote{Given an objective function $f(x)$, the perturbation of an iteration of Newton’s method is $\Delta x = \frac{f'(x)}{f''(x)}$.} (because the Hessian matrix is only an approximation) called the Gauss-Newton algorithm.

LM adjusts $\alpha$ at each iteration to calculate perturbations using a combination of the steepest ascent and Gauss-Newton algorithms. If the perturbation increases the value of the objective function, the estimate $\hat{x}$ is updated via (4.38). Axis mapping uses heuristics introduced by Neilsen [38] to adjust $\alpha$ and to determine whether convergence criteria are met.

### 4.5 Pseudocode

Pseudocode for the 2DAM and 3DAM algorithms are provided in Algorithm 1 and Algorithm 2, respectively. Note that calculating the variables in the objective functions follows the same process for the two algorithms; it is only when the objective function is maximized do the two algorithms differ.
Algorithm 1: A summary of the two-dimensional axis mapping (2DAM) algorithm.

Require: Initial estimate $\hat{x}$, observations $\{z_{q_i}, \sigma_{q_i}^2\}$, $\{z_{q_iq_j}, \sigma_{q_iq_j}^2\}$, $\{z_{q_im_j}, \sigma_{q_im_j}^2\}$

1: for all observations $\{z_i, \sigma_i^2\}$ do
2:   Calculate observation error $e_i(\hat{x}) = e_i(x)\bigg|_{\hat{x}}$
3:   Calculate observation matrix $C_i = \frac{\partial e_i(x)}{\partial x} \bigg|_{\hat{x}}$
4: end for
5: Calculate Jacobian $b = \sum_i C_i^T R_i^{-1} e_i(\hat{x})$
6: Calculate Hessian $H = \sum_i C_i^T R_i^{-1} C_i$
7: while not converged according to [38] do
8:   Solve $H\Delta x^* = b$ for optimal perturbation $\Delta x^*$
9:   Calculate optimal estimate $\hat{x}^* = \hat{x} \oplus \Delta x^*$
10: return $\hat{x}^*$

Algorithm 2: A summary of the three-dimensional axis mapping (3DAM) algorithm.

Require: Initial estimate $\hat{x}$, observations $\{z_{q_i}, R_{q_i}\}$, $\{z_{q_iq_j}, R_{q_iq_j}\}$, $\{z_{q_im_j}, R_{q_im_j}\}$

1: for all observations $\{z_i, R_i\}$ do
2:   Calculate observation error $e_i(\hat{x}) = e_i(x)\bigg|_{\hat{x}}$
3:   Calculate observation matrix $C_i = \frac{\partial e_i(x)}{\partial x} \bigg|_{\hat{x}}$
4: end for
5: Calculate Jacobian $b = \sum_i C_i^T R_i^{-1} e_i(\hat{x})$
6: Calculate Hessian $H = \sum_i C_i^T R_i^{-1} C_i$
7: while not converged according to [38] do
8:   Solve $(H + \alpha I)\Delta x^* = b$ for optimal perturbation $\Delta x^*$
9:   if $J(\hat{x} \oplus \Delta x^*) > J(\hat{x})$ then
10:      $\hat{x} \leftarrow \hat{x} \oplus \Delta x^*$
11:      Recalculate $e_i(\hat{x})$ and $C_i$ for all observations
12:      Recalculate $b$ and $H$
13: end if
14: Adjust $\alpha$ according to [38]
15: end while
16: Optimal estimate is latest estimate $\hat{x}^* = \hat{x}$
17: return $\hat{x}^*$
Part II

Applications
Chapter 5

LiDAR Compass

Localization of mobile robots or other ground vehicles is an active area of research that has important applications in mapping, planning, and control. In the absence of an absolute positioning system (e.g., GPS), localization is traditionally performed by measuring the internal state of the vehicle with interoceptive sensors (e.g., accelerometers, gyroscopes, and wheel encoders), and/or measuring the local environment surrounding the vehicle with exteroceptive sensors (e.g., cameras and LiDAR). In two dimensions, accurately estimating the robot’s heading is particularly important when the localization algorithm involves dead reckoning. In this common scenario, the estimation of the position and heading components are often tightly coupled, causing heading inaccuracies to be quickly propagated into substantial position errors. As a result, accurate heading estimation can be essential for localization. Although sensors exist that can directly or indirectly be used for heading estimation (e.g., a compass or gyroscope), environmental limitations and/or dead reckoning errors can render these estimates unreliable for many applications. To this end, this chapter describes the LiDAR compass (LC), which transforms data from a horizontally-oriented 2D scanning LiDAR into an absolute heading estimate that can be used in a variety of indoor and outdoor environments.

The key idea behind the LC is its use of two-dimensional axis maps to store information about the dominant orientations of the flat surfaces in the environment. As a
vehicle moves through the environment, it observes the axes of the flat surfaces and compares the observations to the entries the axis map. This information is then used to provide corrections to its heading estimate. As a result, it is assumed that the operating environment contains some approximately flat surfaces. This assumption is satisfied in many common scenarios, including both indoor (e.g., walls and furniture) and outdoor (e.g., buildings and cars) environments. In these environments, the LC is essentially a virtual heading sensor that—when combined with a means to measure translation—can be used to aid localization. Using the LC in this way is analogous to augmenting wheel odometry with a compass and gyroscope. Much like how a compass provides an absolute heading reference by measuring the direction of the Earth’s magnetic field, the LC uses the axes of the dominant flat surfaces in the environment as the absolute reference. Similarly, much like how a gyroscope provides a dead reckoning heading estimate by integrating angular velocity, the LC also provides relative heading estimates by tracking the axes of local surfaces not in the *a priori* axis map. The result is an easy-to-implement, very lightweight, virtual heading sensor that (unlike a gyroscope) provides an absolute heading reference, and (unlike a compass) can be used in any environment from which an axis map can be derived.

### 5.1 Related Work

The LC requires reoccurring flat surfaces in the environment. Although only the axes of these surfaces are used by the LC, map representations containing the axes and positions of flat surfaces in the environment have considerable heritage in mobile robotics. Before relatively low-cost LiDAR became available, sonar was used to observe flat surfaces in the environment given an *a priori* map [20]. Here, the idea of improving localization by incorporating geometric constraints in the environment was exploited. One of the first implementations using LiDAR provided online localization given an *a priori* map of flat surfaces [21], where it was noted that many indoor
environments are suitable for this type of map representation. An early implementation that actually constructed maps of flat surfaces [39] predates modern SLAM and actually decouples mapping and localization. However, even with a rudimentary 2D LiDAR, less structured environments such as underground mines were shown to contain sufficiently flat surfaces for the map representation to be effective.

More recent efforts have used maps of flat surfaces in SLAM implementations by using the positions and axes of extracted line segments to update the estimated pose of the robot. One EKF SLAM implementation [22] demonstrated the accuracy of using line segments in the SLAM state, whose compactness also reduces the burden of the computational complexity of SLAM. However, as line segments have four degrees of freedom, effective data association among line segments requires several correspondence tests. This issue is non-existent in an axis map because its entries have only a single degree of freedom.

The popular line-based Orthogonal SLAM [23] takes advantage of an orthogonality assumption of the flat surfaces in the environment (e.g., the perpendicular walls common in most indoor areas). This approach was later extended to create a lightweight Rao-Blackwellized particle filter Orthogonal SLAM [24]. Here, only orthogonal lines extracted from the environment are used to update the SLAM state. By fixing the possible axes of lines to an absolute reference (and discarding lines that do not meet this criterion), remarkable mapping and localization accuracy is achieved by limiting the growth of heading errors. Recent work [25] has explored automatically identifying additional types of structure (e.g., point-to-point distances and circles) and incorporating these constraints into the graph-based optimization of the map. The LC presented in this chapter shares some of the benefits of these structure-sensitive SLAM algorithms with its \textit{a priori} axis map. Unlike the aforementioned approaches, the LC is intended to use the structure of the environment for online, lightweight, and easily implemented heading estimation. It is effective in environments whose surfaces are
A different approach of heading estimation called a **visual compass** (VC) is currently an active research topic in mobile robot localization. Although many implementations of VCs exist, most implementations use a camera to track changes or features in the image frame to infer rotational information. VCs differ from visual odometry because they usually discard all positional information in the sensor data, not unlike the LC. A common approach is to unwrap sequential omnidirectional images and observe how simple extracted features appear to be displaced as the robot rotates [40, 41]. An overview of this method concluded that its dead reckoning heading estimation performed similarly or better than inertial sensors in appropriate environments [40]. Other forms of VCs do not necessarily require omnidirectional cameras and instead track the motion of specific vanishing points at far distances [42, 43], which work well in large, open environments. Finally, a recent VC implementation addresses the restrictions of many other VCs (e.g., computation, prior environmental knowledge, or calibration requirements) with an algorithm that tracks circles and lines detected using edge detection, RANSAC, and Hough transforms [44]. However, this algorithm was only tested on short traverses (< 30 m) and suffered from problems caused by changes in illumination or sharp turns. Although the LC shares some of the same principles as VCs, it is far simpler to implement, has environmental requirements that are less constrictive, and may be more easily used as an absolute heading sensor (due to the ease of producing axis maps).

A potential application of the LC is to provide an initial guess of the design parameter in graph-based SLAM algorithms. When equipped with a two-dimensional laser scanner and (optionally) wheel encoders and/or a gyroscope, the initial guess of the path travelled by the robot is commonly provided by using scan matching [45]. Simply described, scan matching compares overlapped scans (or local maps generated by the scans) between two poses and estimates their relative rotation and transla-
tion. Incrementally performing scan matching (where motion sensors are often used to provide a first guess to the scan matching algorithm) is a viable form of localization. However, because scan matching is a form of dead reckoning, the correctness of the map degrades severely as the robot’s trajectory lengthens. As is demonstrated in Section 5.7, incorporating the LC as part of a localization algorithm and using it in place of scan matching to provide an initial guess to graph-based SLAM becomes an attractive alternative.

5.2 Problem Description

This chapter describes how an LC is used as part of a two-dimensional localization algorithm. An LC is the process of transforming two-dimensional LiDAR data into estimates of the heading of a moving vehicle. Although the LC can be used solely in this capacity (as described in [46]), it is most useful when the heading estimates it provides are incorporated into a localization algorithm. LiDAR compass localization (LCL) dynamically determines an estimate of the state \( x = (p, q, A) \), which consists of the two-dimensional position of the vehicle \( p \in \mathbb{R}^2 \) (expressed in the global coordinate frame \( \mathcal{F}_{g} \)), the heading of the vehicle as a two-dimensional unit rotation \( q \in S^1 \) (where \( q^+ \in SO(2) \) rotates the vehicle coordinate frame \( \mathcal{F}_v \) to the global coordinate frame \( \mathcal{F}_g \)), and a local axis map \( A = (m_1, \ldots, m_m) \) of the flat surfaces in the immediate vicinity of the vehicle (where \( m_j \in S^1 \) is a two-dimensional unit axis expressed in \( \mathcal{F}_g \)). The components of \( x \) are illustrated in Figure 5.1.

To perform online estimation of \( x \), LCL uses the following measurements and \textit{a priori} information about the environment:

(i) motion measurements \( d = (\Delta d, \Delta q) \) of the translation \( \Delta d \in \mathbb{R} \) and rotation \( \Delta q \in S^1 \) of the vehicle in a short time interval, measured in \( \mathcal{F}_v \);

(ii) axis measurements \( z \in S^1 \) of the axes of the flat surfaces in the local environ-
Figure 5.1: The components of $\mathbf{x} = (\mathbf{p}, q, \mathbf{A})$ estimated by LiDAR compass localization. The position $\mathbf{p} \in \mathbb{R}^2$ and heading $q = \exp(\theta) \in S^1$ of the vehicle, and entries $\mathbf{m}_j = \exp(\phi_j) \in S^1$ of the axis map $\mathbf{A}$ are defined with respect to the global coordinate frame $\mathcal{F}_g$.

(iii) an axis map $\mathbf{\bar{A}} = (\mathbf{\bar{m}}_1, \ldots, \mathbf{\bar{m}}_n)$ of the unit axes describing the dominant, reoccurring flat surfaces in the environment, expressed in $\mathcal{F}_g$.

Obtaining the motion measurements depends on the type of vehicle and the available sensors. A detailed description of deriving motion measurements from rotary encoder and gyroscope measurements for two different types of vehicles is provided in Appendix B. Appendix B also describes how axis measurements are derived from a two-dimensional scanning LiDAR.

It is worth explicitly stating the differences between the axis map in the state $\mathbf{A}$, and the global axis map $\mathbf{\bar{A}}$. The entries of $\mathbf{A}$ are estimated online and describe the flat surfaces in the local environment that are not present in $\mathbf{\bar{A}}$. These axes are dynamically added to and removed from $\mathbf{A}$ as the vehicle moves through the environment by using the method described in Section 5.4.4. The primary purpose of $\mathbf{A}$ is to use locally consistent axes for heading estimation in parts of the environment that are devoid of entries of $\mathbf{\bar{A}}$. Conversely, the entries of the global axis map $\mathbf{\bar{A}}$ represent the dominant, reoccurring flat surfaces that can be found throughout the environment, and are determined \textit{a priori} using one of the methods described in Section 5.3.
The entries of $x$ are modelled as random variables; that is, $p$ is a two-dimensional Gaussian random variable, $q$ is a two-dimensional random rotation (Section 3.5), and the entries of $A$ are two-dimensional random axes (Section 2.5). An estimate of the state is represented by its mean $\hat{x} = (\hat{p}, \hat{q}, \hat{A})$, which are the means of the respective random variables, and its $(3 + m) \times (3 + m)$ covariance matrix $P$. Similarly, the noises of the measurements $d$ and $z$ are modelled as zero-mean Gaussian random variables with covariance $Q \in \mathbb{R}^{2 \times 2}$ and variance $\sigma^2_\delta \phi$, respectively. Finally, each entry $\tilde{m}_j$ of the a priori axis map $\tilde{A}$ is modelled as a random axis with variance $\sigma^2_\delta \tilde{\phi}_j$.

LCL recursively determines an online estimate of $\{\hat{x}_k, P_k\}$ at time step $k$ given the a priori axis map $\tilde{A}$, the previous estimate $\{\hat{x}_{k-1}, P_{k-1}\}$, a motion measurement $\{d_k, Q_k\}$, and $p \geq 0$ axis measurements $\{z_1, \sigma^2_\delta \phi_1\}, \ldots, \{z_p, \sigma^2_\delta \phi_p\}$. This is accomplished by using a Kalman filter [47, p. 40] that predicts the motion of the vehicle with the motion measurements and corrects the prediction with the axis measurements.

### 5.3 Building the A Priori Axis Map

A key component of LCL is $\tilde{A}$, its a priori global axis map of the environment, which contains the axes of the dominant, reoccurring flat surfaces in the environment. $\tilde{A}$ provides a global reference with which the heading of the vehicle can be compared, much like how a compass uses the (relatively) constant direction of the Earth’s magnetic field. There are two primary methods for determining $\tilde{A}$: by using a blueprint or overhead image of the environment to manually estimate the axes, or by performing two-dimensional axis mapping (2DAM). It may be surprising that the former method is actually quite straightforward for a large number of environments. For example, for the environments pictured in Figure 5.2, a reasonable a priori axis map is $\tilde{A} = \begin{pmatrix} [1 & 0]^T, [0 & 1]^T \end{pmatrix}$. The latter method requires using the axis mapping algorithm described in Chapter 4 and summarized for two dimensions (2DAM) in Algorithm 1 on page 62. This section specifies how 2DAM is performed with the motion and axis
Figure 5.2: Examples of two different environments from which $\bar{\bar{A}}$ can be easily determined. On the left is a blueprint of the ground floor of Beamish-Munro Hall at Queen’s University (approximately 35 m × 55 m). Despite some non-orthogonal and curved areas, the axis map $\bar{\bar{A}} = ([1 \ 0]^T, [0 \ 1]^T)$ describes the dominant, reoccurring walls. The solid and dashed paths are the first and second routes used in the experiments described in Section 5.7, respectively, and the positions marked by letters are known ground truth poses. On the right is satellite imagery (©2014 DigitalGlobe) of a section of the Queen’s University campus (approximately 200 m × 400 m). The outer walls of most of the buildings share the same axes. The axis map $\bar{\bar{A}} = ([1 \ 0]^T, [0 \ 1]^T)$ would also be appropriate for this environment. The solid path is the route used in the experiments described in Section 5.7.

measurements described in Section 5.2. It is worth emphasizing that the variable $\mathbf{x}$ used in this section refers to the axis mapping design parameter defined in Table 4.1 on page 46 and not the LCL state defined in Section 5.2.

5.3.1 Obtaining the Axis Mapping Observations

As described in Chapter 4, up to three types of observations are used to perform axis mapping. For 2DAM, these are the random rotations $\mathbf{z}_{q_i}, \mathbf{z}_{q_j} \in S^1$ and the random axes $\mathbf{z}_{q_m} \in S^1$. Table 4.1 on page 46 describes these observations. The implementa-
tion of 2DAM described in this section uses the factor graph illustrated in Figure 4.3b on page 48. This factor graph is used because no sensor is available that directly measures the orientation of the vehicle. As a result, only a single observation of the type \( z_{q_i} \) is available (the initial heading of the vehicle). This section describes how the other two types of observations are obtained from the LCL sensor measurements.

### Obtaining Observations of the Type \( z_{q_i,q_j} \)

Observations of the type \( z_{q_i,q_j} \) represent rotations between \( q_i \) and \( q_j \) (see Table 4.1 on page 46). Suppose there are \( n \) motion measurements \( d_1, \ldots, d_n \) between \( q_i \) and \( q_j \). Then the rotation from \( q_i \) to \( q_j \) can be estimated by accumulating the measured rotations; i.e.,

\[
z_{q_i,q_j} = \Delta q_i^+ \Delta q_{i-1}^+ \cdots \Delta q_n^+ \Delta q_1^+.
\]

The variance \( \sigma^2_{q_i,q_j} \) of this observation is similarly calculated by accumulating the variance of the measured rotations; i.e.,

\[
\sigma^2_{q_i,q_j} = \sigma^2_{\delta\theta_1} + \cdots + \sigma^2_{\delta\theta_n},
\]

where \( \sigma^2_{\delta\theta_k} \) is the variance of \( \Delta q_k \).

### Obtaining Observations of the Type \( z_{q_i,m_j} \)

Observations of the type \( z_{q_i,m_j} \) represent observations of \( m_j \) in \( F_v \) taken at \( q_i \) (see Table 4.1 on page 46). These observations are simply the axis measurements used for LCL; i.e.,

\[
\left\{ z_{q_i,m_1}, \sigma^2_{q_i,m_1}\right\}_1, \ldots, \left\{ z_{q_i,m_p}, \sigma^2_{q_i,m_p}\right\}_p \leftarrow \left\{ z_1, \sigma^2_{\delta\phi_1}\right\}, \ldots, \left\{ z_p, \sigma^2_{\delta\phi_p}\right\},
\]

where the method used to derive this type of observation from a two-dimensional LiDAR measurement is outlined in Section B.3 of Appendix B. Note that each observation is either associated with \( \hat{m}_j \) (i.e., an entry of \( \hat{A} \)) or is marked as an outlier. This
process of associating the observations with the local axis map is called *data association* and is detailed in Section 5.3.2.

### 5.3.2 Incremental Optimal Estimation of the Design Parameter

The 2DAM algorithm summarized in Algorithm 1 requires an initial estimate $\hat{x} = (\hat{Q}, \hat{A})$, which is the design parameter of the objective function. Additionally, associating the observations of type $z_{q,m,j}$ with entries of the axis map implies that an estimate of the axis map $\hat{A}$ is available, as well as an estimate of the orientation path $\hat{Q}$ to allow comparisons of the observations (which are expressed in $\mathcal{F}_v$) with entries of the axis map (which are expressed in $\mathcal{F}_y$).

Recall that the initial heading is the only observation of the type $z_{q_i}$ available. In traditional mapping algorithms (where the positions of the vehicle and map entries are also estimated), the initial estimate of the design parameter is usually obtained by starting with a relatively well known initial position and heading, and then adding entries to the design parameter each time the change in position or heading of the vehicle exceeds a threshold (e.g., 0.5 m or 0.5 rad, respectively) as measured by the sensors (i.e., by using *dead reckoning*) [17]. However, this approach is not appropriate when performing axis mapping because of the lack of positional information in the axis map. More specifically, because each entry of the axis map can potentially be observed from any part of the environment, the heading of the vehicle must always be accurate enough to correctly associate new observations with axes that were originally observed much earlier. This phenomenon is illustrated in Figure 5.3. As a result, the design parameter $\hat{x}$ is *incrementally* optimized to provide the required accuracy for data association.

Obtaining an accurate initial estimate $\hat{x}$ is achieved by calculating the optimal estimate $\hat{x}^*$ each time a new entry is added to $\hat{Q}$. As a result, the best possible estimate of $\hat{x}$ is used each time data association is performed. This strategy is computation-
Figure 5.3: The importance of having an accurate initial estimate $\hat{x} = (\hat{Q}, \hat{A})$. (a) With a good initial estimate, observations of the same (or similar) axes will appear as clusters when expressed in $F_g$. Clustering these observations provides a good initial estimate of the axis map $\hat{A}$. (b) With a poor initial estimate, the same observations will not necessarily appear in clusters in $F_g$ due to the poor heading estimates in $\hat{Q}$. As a result, it would be difficult to cluster the observations to form an estimate of the axis map $\hat{A}$. 
Figure 5.4: Outline of the 2DAM algorithm for obtaining $\hat{A}$ for LCL. After obtaining an initial heading, measurements are processed until a new entry to $\hat{Q}$ is needed and added. At this heading, the observation $z_{qi,j}$ is obtained by accumulating the motion measurements since the previous entry of $\hat{Q}$ was added, and observations of the type $z_{qi,mi}$ are obtained from the axis measurements. All observations of the type $z_{qi,mi}$ are then rotated to $F_{qi}$ and clustered to estimate $\hat{A}$ and to perform data association. The observations and the current estimate of $\hat{x}$ are then used as the inputs to Algorithm 1, which produces the optimal estimate $\hat{x}^\ast$. This process continues until all measurements have been incorporated.

ally tractable because the calculation of $\hat{x}^\ast$ is simply a one-step perturbation of the previous estimate of $\hat{x}$, as described in Section 4.4. The final optimal estimate $\hat{x}^\ast$ is simply the latest incremental estimate once all observations have been incorporated.

This method of iteratively maximizing the objective function is outlined in Figure 5.4. Some blocks in this outline require clarification; in particular, determining when and how new entries are added to $\hat{Q}$ (the “Process measurements until new entry of $\hat{Q}$ is needed” and “Add new entry to $\hat{Q}$” blocks), and how data association and the estimation of $\hat{A}$ is performed (the “Perform data association” block). This section concludes with detailed explanations of these blocks.

Adding New Entries to $\hat{Q}$

Only observations taken at entries of $\hat{Q}$ are used in the axis mapping algorithm. As a result, criteria for determining when to add a new entry of $\hat{Q}$ should ensure that
axis-rich areas in the environment (e.g., many different flat surfaces) result in many entries, and axis-dull areas (e.g., long, straight hallways) result in fewer entries. Furthermore, to prevent poor data association, large rotations of the vehicle should trigger new entries (e.g., to correct undetected wheel slip), as should the accumulation of large heading uncertainty. To meet these criteria, an entry to $\hat{Q}$ is added if any of the following conditions are met:

(i) the measured rotation since the previous entry exceeds a threshold (typically $10^\circ$–$25^\circ$);

(ii) the accumulated uncertainty of the measured rotation is greater than a threshold (to prevent poor data association);

(iii) a timeout occurs (typically 5–25 s) and the mean axis of an axis measurement has changed by a threshold (typically $8^\circ$–$15^\circ$).

The estimated heading $\hat{q}_{n+1}$ of the new entry is initially estimated using the motion measurements. It is simply calculated by rotating the previous entry of $\hat{Q}$ by the accumulated measured rotation; i.e.,

$$\hat{q}_{n+1} = \hat{z}_{q_n q_{n+1}}^+ \hat{q}_n,$$  \hspace{1cm} (5.4)

where $\hat{z}_{q_n q_{n+1}}$ is the measured rotation from the previous entry $\hat{q}_n$ to the new entry $\hat{q}_{n+1}$, as defined in (5.1).

**Data Association and Estimation of $\hat{A}$**

At each estimated heading $\hat{q}_i$, there is a set of $p \geq 0$ observations $(z_{q_i m_j})_1, \ldots, (z_{q_i m_j})_p$, which are expressed in $\mathcal{F}_g$. Using $\hat{q}_i$, each observation is expressed in $\mathcal{F}_g$ by

$$z_{q_i m_j}^{(g)} = \hat{q}_i^+ z_{q_i m_j},$$  \hspace{1cm} (5.5)
where $z_{q,m}^{(g)}$ is the observation expressed in $\mathcal{F}_g$ and $q_i^+ \in SO(2)$ is, by definition, the estimated rotation from $\mathcal{F}_v$ to $\mathcal{F}_g$. Now suppose (5.5) is applied to every observation of the type $z_{q,m}$ at every heading, resulting in every observed axis being expressed in $\mathcal{F}_g$. The observations derived from the reoccurring flat surfaces in the environment should appear as clusters (given a good estimate of $\hat{Q}$, see Figure 5.3). As a result, $\hat{A}$ is estimated by clustering similar axes using DBSCAN [48] and calculating the axial mean [1] of the resulting clusters. The observations making up each cluster are associated with that entry of $\hat{A}$ and the unclustered observations are marked as outliers. This process is illustrated in Figure 5.5.
5.4 LiDAR Compass Localization

The data required to perform LCL (as stated in Section 5.2) are motion measurements of the vehicle, axis measurements of the local flat surfaces in the environment, and an a priori axis map describing the dominant, reoccurring flat surfaces in the environment. LCL uses this data to localize a vehicle by using the method illustrated in Figure 5.6, which shows how the state estimate $\hat{x} = (\hat{p}, \hat{q}, \hat{A})$ evolves as new measurements are taken. This section describes this process in detail.

5.4.1 Motion Prediction

As stated in Section 5.2, LCL employs a Kalman filter to estimate the state. The prediction step of the Kalman filter propagates the previous state estimate forward in time by using a motion measurement. At time step $k$, let $\hat{p}_k = (\hat{x}_k, \hat{y}_k) \in \mathbb{R}^2$ and $\hat{\theta}_k = \log(\hat{q}_k) \in s^1$ be the estimated position and heading of the vehicle, respectively, and let $\Delta d_k$ and $\Delta \theta_k = \log(\Delta q_k) \in s^1$ be the components of the motion measurements. Then the predicted position and heading of the vehicle are represented as

$$
\begin{bmatrix}
\hat{x}_k \\
\hat{y}_k \\
\hat{\theta}_k
\end{bmatrix} =
\begin{bmatrix}
\hat{x}_{k-1} + \Delta d_k \cos(\hat{\theta}_{k-1}) \\
\hat{y}_{k-1} + \Delta d_k \sin(\hat{\theta}_{k-1}) \\
\hat{\theta}_{k-1} + \Delta \theta_k
\end{bmatrix}.
$$

(5.6)

Returning to the original parameterizations of the position, heading, and measurements and making the substitutions

$$
\begin{bmatrix}
\cos(\hat{\theta}_{k-1}) \\
\sin(\hat{\theta}_{k-1})
\end{bmatrix} = \hat{q}_{k-1}, \quad \hat{\theta}_{k-1} + \Delta \theta_k = \log(\Delta q_k^+ \hat{q}_{k-1})
$$

(5.7)

results in the predicted position and heading of the vehicle

$$
\begin{bmatrix}
\hat{p}_k \\
\hat{q}_k
\end{bmatrix} =
\begin{bmatrix}
I_2 & \Delta d_k I_2 \\
0_{2 \times 2} & \Delta q_k^+ \hat{q}_{k-1}
\end{bmatrix}
\begin{bmatrix}
\hat{p}_{k-1} \\
\hat{q}_{k-1}
\end{bmatrix}.
$$

(5.8)
Figure 5.6: LiDAR compass localization (LCL). At time step $k$, the estimated position $\hat{p}_k$ and heading $\hat{q}_k$ of the vehicle is predicted with the motion measurements $d_k$ and corrected with $p \geq 0$ axis measurements $z_1, \ldots, z_p$. In this example, the a priori axis map $\bar{A}$ has only one entry, and measured axes not in $\bar{A}$ are added to $\hat{A}$ as they are observed. For example, at $(\hat{p}_1, \hat{q}_1)$ two axes $z_1$ and $z_2$ are measured, resulting in $z_1$ being associated with the a priori axis map entry $\bar{m}_1$ and $z_2$ being transformed into the new local axis map entry $\hat{m}_1$. 
Because the motion measurements do not contain information about the local axis map, \( \hat{\mathbf{A}}_k = \hat{\mathbf{A}}_{k-1} \). Combining this with (5.8) yields the full state prediction

\[
\hat{x}_k = \begin{bmatrix}
\mathbf{I}_2 & \Delta d_k \mathbf{I}_2 & \mathbf{0}_{2 \times 2m} \\
\mathbf{0}_{2 \times 2} & \mathbf{\Delta q}_k & \mathbf{0}_{2 \times 2m} \\
\mathbf{0}_{2m \times 2} & \mathbf{0}_{2m \times 2} & \mathbf{I}_{2m}
\end{bmatrix}
\hat{x}_{k-1} = F_k \hat{x}_{k-1},
\]

where \( m \) is the number of entries in \( \hat{\mathbf{A}}_{k-1} \).

Given the prediction in (5.9), propagating the covariance matrix \( P_{k-1} \) to time step \( k \) with a Kalman filter would normally take the form

\[
P_k = F_k P_{k-1} F_k + G_k Q_k G_k,
\]

where \( Q_k \) is the covariance matrix of the measurement, and \( G_k \) is the Jacobian of the prediction with respect to the measurement. However, recall that some of the entries of the state are constrained parameterizations: the heading is a two-dimensional unit rotation (Section 3.2) and the axis map entries are two-dimensional unit axes (Section 2.2). As a result, because the covariance matrix represents uncertainty only along the degrees of freedom of the state (for the reasons stated in Section 3.5 and Section 2.5 for the heading and axis map entries, respectively), it is propagated forward by examining how the prediction changes with respect to changes of the unconstrained parameterizations of the state and measurements.

Let \( \mathbf{x} = \log(\mathbf{x}) \in \mathbb{R}^{3+m} \) be the unconstrained parameterization of the state, where

\[
\log(\mathbf{x}) := \begin{bmatrix}
\mathbf{p} \\
\log(\mathbf{q}) \\
\log(\mathbf{m}_1) \\
\vdots \\
\log(\mathbf{m}_m)
\end{bmatrix} = \begin{bmatrix}
\mathbf{p} \\
\theta \\
\phi_1 \\
\vdots \\
\phi_m
\end{bmatrix},
\]

(5.11)
and let the unconstrained parameterization of the prediction be

$$\log(x_k) = \log(F_k x_{k-1})$$

$$\hat{x} = \begin{bmatrix} x_{k-1} + \Delta d_k \cos(\theta_{k-1}) \\ y_{k-1} + \Delta d_k \sin(\theta_{k-1}) \\ \theta_{k-1} + \Delta \theta_k \\ \phi_1 \\ \vdots \\ \phi_m \end{bmatrix}. \quad (5.12)$$

Note that the use of $\log$ here is purely a notational convenience. The unconstrained Jacobian $\mathcal{F}_k$ of (5.12) with respect to the state is then

$$\mathcal{F}_k = \frac{\partial \log(F_k x_{k-1})}{\partial x_{k-1}} \Big|_{x_{k-1}} = \begin{bmatrix} I_2 & \Delta d_k \hat{u}_{k-1} & 0_{2 \times m} \\ 0_{1 \times 2} & 1 & 0_{1 \times m} \\ 0_{m \times 2} & 0_{m \times 1} & I_m \end{bmatrix} \quad (5.13)$$

where $\hat{u}_{k-1} = [-\epsilon \eta]^T$ for $\hat{q}_{k-1} = [\eta \ \epsilon]^T$. Similarly, let $\mathcal{G}_k = \log(d)$ be the unconstrained parameterization of the motion measurement, where

$$\log(d) := \begin{bmatrix} \Delta d \\ \log(\Delta q) \end{bmatrix} = \begin{bmatrix} \Delta d \\ \Delta \theta \end{bmatrix}. \quad (5.14)$$

The unconstrained Jacobian $\mathcal{G}_k$ of (5.12) with respect to the measurement is then

$$\mathcal{G}_k = \frac{\partial \log(F_k x_{k-1})}{\partial d_k} \Big|_{d_k} = \begin{bmatrix} \hat{q}_{k-1} & 0_{2 \times 1} \\ 0 & 1 \\ 0_{m \times 1} & 0_{m \times 1} \end{bmatrix}. \quad (5.15)$$

The covariance matrix $P_{k-1}$ can now be propagated to time step $k$ by using the unconstrained Jacobians; i.e.,

$$P_k = \mathcal{F}_k P_{k-1} \mathcal{F}_k^T + \mathcal{G}_k Q_k \mathcal{G}_k^T. \quad (5.16)$$

### 5.4.2 Motion Correction with the A Priori Axis Map

The correction step of the Kalman filter corrects the predicted state estimate by incorporating an axis measurement of a flat surface in the environment. The a priori
axis map $\bar{A}$ contains the axes of the dominant, reoccurring flat surfaces in the environment, and this section describes how the state estimate is corrected when one of the entries of $\bar{A}$ is measured. However, recall that there are $p \geq 0$ axis measurements at each time step, and $n$ entries of $\bar{A}$. Determining which (if any) entry of $\bar{A}$ is being measured by each of the $p$ axis measurements (i.e., data association) is described in detail in Appendix C. To simplify the notation, the subscript $k$ indicating the time step is omitted in this section.

Suppose data association has determined that the $i$-th measurement $z_i$ is measuring the $j$-th a priori axis map entry $\bar{m}_j$. The expected measurement $\hat{z}_j \in S^1$ is simply the expression of that axis in $\mathcal{F}_v$; i.e.,

$$\hat{z}_j = \hat{q}^- \bar{m}_j,$$

(5.17)

where $\hat{q}$ is the estimated heading from which the measurement was made, and $\hat{q}^- \equiv \hat{q}^{+\top} \in SO(2)$ is, by definition, the estimated rotation from $\mathcal{F}_g$ to $\mathcal{F}_v$ (see Section 3.2 for a definition of the compound operator $q^+$ on a two-dimensional unit rotation $q$). The measurement error $e_{ij} \in h^1$ is then

$$e_{ij} = z_i \boxplus \hat{z}_j,$$

(5.18)

where the $\boxplus$ operator on two-dimensional unit rotations is defined in Section 3.4.1. The variance $\sigma_{e_{ij}}^2$ of the measurement error is

$$\sigma_{e_{ij}}^2 = \sigma_{\delta \phi_i}^2 + \sigma_{\delta \bar{\phi}_j}^2 + HPH^\top,$$

(5.19)

where $\sigma_{\delta \phi_i}^2$ is the variance of $z_i$, $\sigma_{\delta \bar{\phi}_j}^2$ is the variance of $\bar{m}_j$, and $H$ is the Jacobian of the error with respect to $x$; i.e.,

$$H = \left. \frac{\partial (z_i \boxplus q^- \bar{m}_j)}{\partial x} \right|_{\hat{x}} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}.$$
 simplemente extrae la varianza $\sigma_q^2$ de $\hat{q}$ de $P$, lo cual simplifica (5.19) a

$$\sigma_{e_{ij}}^2 = \sigma_{\delta\phi_i}^2 + \sigma_{\delta\bar{\phi}_j}^2 + \sigma_q^2. \quad (5.21)$$

La corrección $\Delta \mathbf{x}_{ij} \in \mathbb{R}^{3+m}$ al estimado de estado es el parámetroización no restringida de una perturbación al estado; i.e.,

$$\Delta \mathbf{x}_{ij} = K_{ij} e_{ij}, \quad (5.22)$$

la cual se escala por la ganancia de Kalman $K_{ij} \in \mathbb{R}^{3+m}$, donde

$$K_{ij} = \frac{\mathbf{P} \mathbf{H}^T}{\sigma_{e_{ij}}^2}. \quad (5.23)$$

La corrección se aplica al estimado de estado usando el operador de la sección 3.4.1 (para aplicar la corrección a la orientación) y la sección 2.4.1 (para aplicar las correcciones a los ejes), donde

$$\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} \boxplus \Delta \mathbf{x}_{ij} = \begin{bmatrix} \hat{\mathbf{p}} + \Delta \mathbf{p}_{ij} \\ \hat{\mathbf{q}} \boxplus \Delta \theta_{ij} \\ \hat{\mathbf{m}}_1 \boxplus (\Delta \phi_1)_{ij} \\ \vdots \\ \hat{\mathbf{m}}_m \boxplus (\Delta \phi_m)_{ij} \end{bmatrix}. \quad (5.24)$$

Porque los términos de la matriz de covarianza ya están representados por parámetros no restringidos, no se necesitan modificaciones especiales a la ecuación de corrección estándar, y por lo tanto

$$\mathbf{P} \leftarrow (I + K_{ij} \mathbf{H}) \mathbf{P}, \quad (5.25)$$

donde la matriz identidad $I$ tiene dimensión $3 + m$. La corrección del estimado de estado y su matriz de covarianza se repite entonces para cada medición asociada.

### 5.4.3 Corrección de Movimiento con el Mapa de Ejes Locales

La etapa de corrección del filtro de Kalman se repite usando el mapa de ejes globales estimado $\hat{A}$ en lugar del mapa de ejes globales $\mathbf{A}$. Solo los ejes que fueron no asociados...
with entries of $\hat{A}$ are used to perform these corrections. Once again, data association is required to determine which (if any) entry of $\hat{A}$ is being measured by each of the remaining measurements.

As before, suppose data association has determined that the $i$-th measurement $z_i$ is a measurement of the $j$-th local axis map entry $\hat{m}_j$. The expected measurement and measurement error are calculated with (5.17) and (5.18), respectively, but with $\hat{m}_j$ used in place of $\tilde{m}_j$. However, the calculation of the variance of the error $\sigma_{e_{ij}}^2$ differs because $\hat{m}_j$ is part of the state estimate; i.e.,

$$\sigma_{e_{ij}}^2 = \sigma_{\delta\phi_{ij}}^2 + H_j P H_j^T,$$

where $H_j$ is once again the Jacobian of the error with respect to $x$; i.e.,

$$H_j = \left. \frac{\partial (z_i \oplus q^+ \hat{m}_j)}{\partial x} \right|_{\hat{x}} = \begin{bmatrix} 0 & 0 & -1 & 0 & \cdots & 1 & \cdots & 0_m \end{bmatrix},$$

$H_j P H_j^T$ transforms $P$ into the variance of the expected measurement $\hat{z}_j$.

The correction $\Delta x_{ij}$ and Kalman gain $K_{ij}$ are calculated with (5.22) and (5.23), respectively, but with $H_j$ used in place of $H$. The correction is then applied to the state estimate with (5.24). Similarly, the covariance matrix $P$ is corrected with (5.25), but with $H_j$ used in place of $H$. The correction step is then repeated for all other previously unassociated measurements that are associated with entries of $\hat{A}$.

### 5.4.4 Adding and Removing Entries of the Local Axis Map

Axis measurements that were not associated with entries of $\tilde{A}$ or $\hat{A}$ represent previously unobserved axes in the environment and are used to add new entries to $\hat{A}$. Suppose that the $i$-th measurement $z_i$ is unassociated with an axis map entry. Then the new entry of $\hat{A}$ is calculated by expressing the measurement in $\mathcal{F}_{gi}$; i.e.,

$$\hat{\mathbf{x}} \leftarrow \begin{bmatrix} \hat{\mathbf{x}} \\ q^+ z_i \end{bmatrix}.$$
where $\hat{q}$ is the estimated heading from which the measurement was made, and $\hat{q}^+ \in SO(2)$ is, by definition, the estimated rotation from $\mathcal{F}_\theta$ to $\mathcal{F}_y$. The updated covariance matrix is

$$P \leftarrow \begin{bmatrix} P & PH^T \\ HP & HPH^T + \sigma_{\delta \phi_i}^2 \end{bmatrix},$$

(5.29)

where $\sigma_{\delta \phi_i}^2$ is the variance of the measurement and $H$ is defined in (5.20).

It is possible that erroneous or isolated measurements result in entries being added to $\hat{A}$. For example, if a door is measured while it is being opened (i.e., while its axis is changing), it is possible that several entries will be added to $\hat{A}$. These entries are unlikely to be re-observed and could be detrimental to data association, hence their presence is detected and they are removed.

To remove extraneous entries of $\hat{A}$, each entry is assigned a brightness $b \in [0,1]$. Measuring an entry causes it to brighten, while unmeasured entries fade. When a new entry is added to $\hat{A}$, it is assigned an initial brightness $b_0 \in (0,1)$. Each time a measurement at time step $k$ is associated with $\hat{m}_j$, its brightness $b_j$ increases by $\Delta b \in (0,1)$ (typically $\Delta b < 0.1$), up to a maximum brightness of 1. If $\hat{m}_j$ is not observed at time step $k$, $b_j$ decreases by $\Delta b$. Entries fading to zero are removed from $\hat{A}$, resulting in a succinct axis map that is comprised solely of map entries actively being observed.

By selecting $b_0$ and $\Delta b$ in conjunction with the frequency of the measurements, one can specify the typical amount of time it takes for an entry to reach maximum brightness or fade to zero. Extraneous entries are quickly removed because they tend not to be re-observed. Additionally, brightnesses can be used to scale motion corrections (Section 5.4.3) by scaling the Kalman gain (e.g., $b_j K_{ij}$). This results in corrections resulting from bright entries (e.g., the axes of the walls of a room) having a greater effect on the state estimate than faded entries.
5.4.5 Merging Entries of the Local Axis Map

As $\hat{x}$ evolves over time, it is possible that two entries of $\hat{A}$ converge to the same axis. These occurrences are detected by calculating the Mahalanobis distance (defined in Appendix C) between the expected difference between each pair of entries (which is the identity unit axis $o \in S^1$ if they are the same) and the estimated axis difference between them (see Proposition 2.5 on page 16 for the definition of the difference between two-dimensional axes). For example, to check if the $i$-th and $j$-th entries of $\hat{A}$ should be merged, the Mahalanobis distance $d_{ij} \in \mathbb{R}$ is

$$d_{ij} = \sqrt{\frac{e_{ij}}{H_{ij}P_{ij}^T}}, \quad e_{ij} = o \boxplus (\hat{m}_i, \hat{m}_j),$$

(5.30)

where the error $e_{ij} \in h^1$ and

$$H_{ij} = \frac{\partial (o \boxplus (m_i, m_j))}{\partial \hat{x}}|_{\hat{x}}^\dagger = \begin{bmatrix} 0 & 0 & 0 & 1 & \cdots & 1_i & \cdots & -1_j & \cdots & 0_m \end{bmatrix}.$$

(5.31)

$H_{ij}P_{ij}^T$ transforms $P$ into the variance of the error $e_{ij}$. If $d_{ij}$ is less than a threshold, the entries $\hat{m}_i$ and $\hat{m}_j$ are merged. This is achieved by a pseudo-correction step where the measurement is $o$ and the expected measurement is $m_i, m_j$. The state estimate and its covariance matrix are then updated in a fashion similar to the correction steps outlined in Section 5.4.2; i.e.,

$$K_{ij} = \frac{PH_{ij}^T}{H_{ij}P_{ij}^T}, \quad \Delta x_{ij} = K_{ij} e_{ij}, \quad \hat{x} \leftarrow \hat{x} \boxplus \Delta x_{ij}, \quad P \leftarrow (I + K_{ij}H)P.$$

(5.32)

After the correction, the $i$-th and $j$-th entries of $\hat{A}$ are identical and represent the merged entry. Because only one of the entries is required, the $j$-th entry of $\hat{A}$ is then removed from $\hat{x}$ and $P$. 
5.5 Pseudocode

Pseudocode for the LCL algorithm is provided in Algorithm 3. Note that the \textit{a priori} axis map \( \hat{\mathbf{A}} \) that is required by the algorithm may be determined by using 2DAM, as described in detail in Section 5.3.

\begin{algorithm}
\caption{A summary of the LiDAR compass localization (LCL) algorithm.}
\begin{algorithmic}[1]
\Require Initial estimates of position \( \hat{\mathbf{p}}_0 \) and heading \( \hat{\mathbf{q}}_0 \), \textit{a priori} axis map \( \hat{\mathbf{A}} \)
\ForAll{time steps \( k \)}
\State Determine motion measurement \( \{ \mathbf{d}_k, \mathbf{Q}_k \} \)
\State Predict motion \( \hat{\mathbf{x}}_k = \mathbf{F}_k \hat{\mathbf{x}}_{k-1} \)
\State Calculate unconstrained Jacobians \( \mathbf{F}_k \) and \( \mathbf{G}_k \)
\State Propagate covariance \( \mathbf{P}_k = \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^\top + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^\top \)
\State Determine \( p \) axis measurements \( \{ \mathbf{z}_i, \sigma_{\delta \phi}^2 \} \)
\State Associate measurements with entries of \( \hat{\mathbf{A}} \)
\ForAll{associations between \( \mathbf{z}_i \) and \( \hat{\mathbf{m}}_j \)}
\State Calculate expected measurement \( \hat{\mathbf{z}}_j \) and error \( \{ e_{ij}, \sigma_{e_{ij}}^2 \} \)
\State Calculate Kalman gain \( \mathbf{K}_{ij} \) and correction \( \Delta \mathbf{x}_{ij} \)
\State Correct state estimate \( \hat{\mathbf{x}}_k \leftarrow \hat{\mathbf{x}}_k \oplus \Delta \mathbf{x}_{ij} \)
\State Correct covariance \( \mathbf{P}_k \leftarrow (\mathbf{I} + \mathbf{K}_{ij} \mathbf{H}) \mathbf{P}_k \)
\EndFor
\State Associate remaining measurements with entries of \( \hat{\mathbf{A}} \)
\ForAll{associations between \( \mathbf{z}_i \) and \( \hat{\mathbf{m}}_j \)}
\State Calculate expected measurement \( \hat{\mathbf{z}}_j \) and error \( \{ e_{ij}, \sigma_{e_{ij}}^2 \} \)
\State Calculate scaled Kalman gain \( b_j \mathbf{K}_{ij} \) and correction \( \Delta \mathbf{x}_{ij} \)
\State Correct state estimate \( \hat{\mathbf{x}}_k \leftarrow \hat{\mathbf{x}}_k \oplus \Delta \mathbf{x}_{ij} \)
\State Correct covariance \( \mathbf{P}_k \leftarrow (\mathbf{I} + b_j \mathbf{K}_{ij} \mathbf{H}) \mathbf{P}_k \)
\State Increase brightness \( b_j \leftarrow b_j + \Delta b \)
\EndFor
\ForAll{unmeasured \( \hat{\mathbf{m}}_j \)}
\State Decrease brightness \( b_j \leftarrow b_j - \Delta b \)
\State Remove \( \hat{\mathbf{m}}_j \) from \( \hat{\mathbf{A}} \) if \( b_j \leq 0 \)
\EndFor
\ForAll{unassociated \( \mathbf{z}_i \)}
\State Add entry to \( \hat{\mathbf{A}} \) with \( \mathbf{z}_i \)
\EndFor
\State Detect and merge similar entries of \( \hat{\mathbf{A}} \)
\EndFor
\end{algorithmic}
\end{algorithm}

87
5.6 Experiments

Experiments were performed in three distinct environments to evaluate the quality of axis maps generated by 2DAM and the accuracy of LCL. The experiments were performed in (i) a simulated environment containing a large number of differently oriented flat surfaces; (ii) an indoor environment containing some walls oriented differently from its main layout; (iii) and an outdoor environment containing a large variety of flat and curved surfaces. This section contains details about each of the environments and descriptions of the vehicles and sensors used in the experiments.

5.6.1 Simulation

The purpose of the simulation experiments was to test 2DAM in challenging environments, such as the jagged wall shown in Figure 5.7. This wall consists of 200 segments, of which the axes of 169 of the segments were randomly drawn from an eight-entry axis map. The axes of the other 31 segments were randomly drawn from a uniform distribution over $h^1$. Algorithm 4 details how the wall was constructed. A simulated LiDAR was modelled after a Hokuyo URG-04LX [49] (same range, resolution, noise, and frequency), and a simulated gyroscope was modelled with noise characteristics similar to a LORD Microstrain 3DM-GX3-25 [50] IMU. A simulated robot equipped with these sensors was driven at 0.5 m/s while following the shape of the wall.
Algorithm 4: Generating a random simulated wall containing axis map entries. For the wall shown in Figure 5.7, \( n = 200, l_{\text{min}} = 0.4 \text{ m}, l_{\text{max}} = 1.0 \text{ m}, m = 8, \) and \( p_{\text{out}} = 20. \)

**Require:** Num. wall segments \( n \), min. segment length \( l_{\text{min}} \), max. segment length \( l_{\text{max}} \), axis map \( A \) containing \( m \) random axes, % outliers \( p_{\text{out}} \)

1: **while** num. wall segments < \( n \) **do**
2: Choose axis or outlier based on \( p_{\text{out}} \)
3:  **if** axis **then**
   4: Randomly select entry \( \{m, \sigma_\delta^2 \} \) from \( A \)
   5: \( \phi = \log(\text{sample}_{-\text{gaussian}}(m, \sigma_\delta^2)) \)
3:  **else**
6: \( \phi = \text{sample}_{-\text{uniform}}(h^1) \)
8: **end if**
9: \( l = \text{sample}_{-\text{uniform}}(l_{\text{min}}, l_{\text{max}}) \)
10: Append line \( (l, \phi) \) to end of previous line
11: **end while**

5.6.2 Beamish-Munro Hall

Experiments were performed on the ground floor of Beamish-Munro Hall (BMH), a building at Queen’s University (Kingston, ON, Canada). A blueprint of the test area is shown in Figure 5.2 on page 71. A Clearpath Robotics Husky A200 [51] mobile robot was equipped with a SICK LMS111 [52] 2D laser scanner (field of view of \( 270^\circ \), resolution \( 0.5^\circ \), range \( 20 \text{ m} \)) oriented to scan in the horizontal plane, and a LORD Microstrain 3DM-GX3-25 IMU [50], used solely for one of its gyroscopes. At the time of the experiments, the environment had a considerable amount of pedestrian traffic.

Hand measurements were taken to determine ground truth poses \( q_A, q_B, \ldots, q_J \) at the ten locations (A, B, \ldots, J) illustrated in Figure 5.2, where \( q_i = (x_i, y_i, \theta_i) \in \mathbb{R}^2 \times s^1 \). The positions of these poses were estimated by taking a series of range and bearing measurements between all poses within line-of-sight of each other, as well as between poses and known positions in the building (e.g., a corner), taken from blueprints. An overdetermined nonlinear system was formed from this set of noisy range and bearing measurements. This system was solved using the popular Levenberg-Marquardt algorithm [38], yielding an estimate of \( (x_A, y_A, x_B, y_B, \ldots, x_J, y_J) \) and its associated co-
variance matrix. The orientations \((\theta_A, \theta_B, \ldots, \theta_J)\) were estimated via hand measurements with a digital protractor relative to a nearby known orientation (e.g., a wall).

Although a reasonable \textit{a priori} axis map of \(\bar{A} = \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^T \right) \) can be determined from the blueprint of BMH in Figure 5.2, \(\bar{A}\) was estimated by performing 2DAM. This served to test 2DAM in an indoor environment and evaluate its effectiveness in generating the \textit{a priori} axis map required for LCL. As a result, an initial mapping phase was performed in which the robot was driven approximately along the solid path illustrated in Figure 5.2. The collected data was then used to estimate \(\bar{A}\) by applying the procedure outlined in Section 5.3 and illustrated in Figure 5.4.

Two routes were selected of varying difficulty to test LCL. The first route (ABCDE-FGA) consistently contained surfaces with axes present in \(\bar{A}\). The second route (AHI-JEFGA) included areas devoid of entries of \(\bar{A}\), surfaces that were oriented slightly differently than those in \(\bar{A}\) (i.e., differing by \(< 8^\circ\) ), and a large, curved surface. Five trials were performed on each route to compare four different localization algorithms:

(i) odometry (motion measurements using only encoders);
(ii) gyroscope (motion measurements using encoders and a gyroscope);
(iii) scan matching (motion measurements using encoders and LiDAR);
(iv) LCL (motion measurements using encoders, axis measurements using LiDAR).

Localization via (i) and (ii) is outlined in Appendix B. Scan matching was performed by using an open source implementation [53] of the popular point-to-line algorithm by Censi [54]. During each trial, the robot was briefly stopped at each ground truth pose to record the pose estimated by each localization algorithm. It was then driven (at varying speeds of 0.1–1.0 m/s and not necessarily in a straight line) to the next ground truth pose.
Figure 5.8: The electric vehicle (Taylor-Dunn SS-534) used for the Queen’s University campus experiments. It has a maximum speed of approximately 4.4 m/s (16 km/h). Rotary encoders were installed on the drive shaft and steering column. The (rear-facing) LiDAR (SICK LMS111) was mounted at the back.

5.6.3 Queen’s University Campus

Data was collected by driving a Taylor-Dunn SS-534 [55] electric vehicle outdoors around the campus of Queen’s University, along the 1.3 km route illustrated in Fig. 5.2. The environment contained many trees, buildings, cars, pedestrians, and gardens. The vehicle (pictured in Figure 5.8) was equipped with the same LiDAR and IMU used in the BMH experiments, along with US Digital A2 [56] encoders on the drive shaft and steering column. To provide ground truth, a Novatel SPAN-SE-2 GPS receiver and CPT IMU [57] was used in conjunction with commercial Novatel GPS/INS software.

Like BMH, although a reasonable a priori axis map of $\tilde{A} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}^T$ can be determined from the satellite image in Figure 5.2 on page 71, $\tilde{A}$ was once again estimated by performing 2DAM. This served to test 2DAM in an outdoor environment to complement the indoor experiment in BMH. As before, an initial mapping phase was performed and the collected data was used to estimate $\tilde{A}$ by applying the procedure outlined in Section 5.3 and illustrated in Figure 5.4.

The section of the campus at which the experiments were performed contained a variety of buildings, trees, pathways, roads, and parking lots, and is typically popu-
lated by many cars, cyclists, and pedestrians. Five trials were performed that covered several types of environments (e.g., narrow paths between buildings, parking lots, open roads, areas with many trees, etc.). The vehicle was driven at speeds between 1.4–4.4 m/s (5–16 km/h). The same four different localization algorithms compared in the BMH experiments (odometry, gyroscope, scan matching, and LCL) are compared in the outdoor experiments.

5.7 Results and Discussion

The results of the experiments in the three different environments detailed in Section 5.6 are presented in this section. More specifically, the 2DAM results from the simulated environment are discussed in Section 5.7.1, and the 2DAM and localization results from BMH and outdoors at the Queen’s University Campus are discussed in Sections 5.7.2 and 5.7.3, respectively.

The axis maps presented in this section are plotted as one-dimensional rose plots (e.g., Figure 5.9). These plots illustrate an axis map by plotting the angles of its entries. Recall that because \( m = -m \) for a unit axis \( m \in S^1 \), each axis is represented by a diameter of the rose plot. Furthermore, two versions of each axis map are plotted: the unclustered and clustered axis maps. In the unclustered axis maps, each axis measurement at every entry of the orientation path is rotated to the global coordinate frame and is plotted (e.g., Figure 5.9c). In the clustered axis map, DBSCAN [48] is applied to the unclustered axis map to automatically group similar clusters and ignore outliers. The axial mean and variance [1] of each cluster is then plotted on the rose plot, where the width of the wedges represent the 95% confidence interval of the axis (e.g., Figure 5.9d). To be consistent with the rose plots, the entries of the axis maps discussed in this section are given as their angles (i.e., in \( h^1 \)), reported in degrees.
5.7.1 Simulation

Four axis maps of the simulated wall are illustrated in Figure 5.9. The true unclustered axis map of the simulated wall (one entry per 200 wall segments) is shown in Figure 5.9a, whose entries were sampled from the axis map shown in Figure 5.9b. The unclustered and clustered axis maps estimated by 2DAM are shown in Figures 5.9c and 5.9d, respectively.

It is clear by comparing Figures 5.9b and 5.9d that the estimated axis map is an accurate representation of axes in the simulated wall. In general, the estimated axes tended to have a similar means with slightly inflated uncertainties. This inflation comes from two sources: the inherent variability introduced by the noise of the simulated gyroscope and LiDAR, and the occurrence of outliers close enough to the true axes to be included by the clustering algorithm. Some of these outliers are visible in Figure 5.9c, especially around the cluster near 60°. Overall, all eight entries of the true axis map are readily identifiable by 2DAM.

5.7.2 Beamish-Munro Hall

The results of the BMH experiments are presented in two parts. First, the results of estimating an axis map of the environment by using 2DAM are presented. Next, the results of using the estimated axis map as the a priori axis map $\bar{A}$ for LCL are discussed.

Axis Mapping Results

The unclustered and clustered axis maps of BMH estimated by 2DAM are illustrated in Figures 5.10a and 5.10b, respectively. The clustered axis map has entries $\bar{A} = (0.7 \pm 3.4^\circ, 90.1 \pm 2.8^\circ)$, where the uncertainty is the standard deviation of the estimate. This is quite similar to the expected axis map of $\bar{A} = (0^\circ, 90^\circ)$ indicated by the blueprint in Figure 5.2.

Comparing Figures 5.10a and 5.10b reveals that two smaller clusters appear to be
**Figure 5.9:** Two-dimensional axis maps of the simulated environment. Axes were sampled from the true axis map in (b) to generate the axes of the wall segments in Figure 5.7.
merged together, increasing the uncertainty of the resulting axis. It was determined that part of the environment with gradually curved surfaces momentarily caused false associations among the axis measurements and axis map. This demonstrates a weakness of using DBSCAN for data association: small changes of the axes of individual points along a curved surface tend to be clustered together, even if the maximum difference between two axes in the cluster is quite large.

Although the path of the robot was approximately 130 m in length, the orientation path contained only 46 entries (a mean distance of 2.8 m between entries). As was expected from the criteria for adding new entries to the orientation path outlined in Section 5.3.2, relatively few entries were added in long corridors compared to areas where the geometry of the environment required substantial heading changes or the axes of surfaces rapidly changed.

**Localization Results**

The axis map $\bar{A} = (0.7 \pm 3.4^\circ, 90.1 \pm 2.8^\circ)$ that was generated by 2DAM was taken as the *a priori* axis map for LCL. With this axis map, the paths estimated by LCL and the other three localization algorithms for both routes are shown in Figure 5.11. Further-
Figure 5.11: The paths estimated in BMH with the four different localization algorithms (odometry, gyroscope, scan matching, and LCL). The results from routes ABCDEFGA and AHIJEFGA of the BMH experiments are plotted in (a) and (b), respectively. In both cases, the trial in which scan matching produced the best result is shown. The robot was briefly parked at each of the ground truth poses, whose positions (A, B, ...) are labelled in the plots.

more, the estimated headings at the ground truth poses of three of the localization algorithms (odometry is not included because its error was far off the scale) for both routes are plotted in Figure 5.12. Finally, the errors of the estimated headings of the localization algorithms over the course of the five trials are summarized in Table 5.1.

As is apparent in Figure 5.11, when odometry, a gyroscope, or scan matching was used for heading estimation, the localization estimates suffered from the accumulation of dead reckoning error. In particular, the error accumulated using scan matching varied significantly among the trials. It is possible that the somewhat dynamic
Figure 5.12: A comparison of the heading errors of the gyroscope, scan matching, and LCL estimates in the BMH experiments. (a) The results of the first trial on the first route (ABCDEFGA). (b) The results of the fifth trial on the second route (AHIJEFGA). These trials were the best results (i.e., smallest root-mean-square error) achieved by scan matching. The shaded areas illustrate the 3σ uncertainty of the errors, and include the (relatively small) uncertainty of the ground truth.
environment (pedestrians) contributed to this inconsistency. Although LCL does not prevent the accumulation of translation errors, it bounds heading errors if entries of $\overline{A}$ are consistently observed. Furthermore, adding and observing entries of $\hat{A}$ limits the growth rate of heading error in areas devoid of surfaces corresponding to entries of $\overline{A}$. Because the position estimate is tightly coupled to the quality of the heading estimate, the full pose estimation was drastically improved when using LCL as compared to the other methods. Despite the changing environment due to considerable pedestrian traffic, Table 5.1 demonstrates the consistency of LCL over multiple trials.

All trials for each route showed very similar behaviour. One exception occurred at ground truth pose H in the more difficult second route (AHIJFEGA), where LCL consistently produced an incorrect heading estimate (see Figure 5.12b). The cause of this discrepancy revealed a short-coming of LCL when there are surfaces with axes that differ only slightly from the entries of $\overline{A}$. In this case, a wall was frequently observed with an axis only a few degrees different from an entry of $\overline{A}$. There was sufficient combined uncertainty from the axis measurement and the current heading estimate
that incorrect data association occurred. This mild failure highlights the sensitivity of LCL on selecting an appropriate Mahalanobis distance threshold for data association in this type of scenario (see Appendix C for details). The threshold must be large enough to capture the expected uncertainty of the measurements, which may result in slight misalignments (e.g., a shelf not quite aligned with a wall) being incorrectly associated. However, the LC easily recovered once sufficient subsequent measurements of true entries of $\bar{A}$ were made. All the trials concluded with nearly zero heading error when LCL is used.

### 5.7.3 Queen’s University Campus

Like the BMH results, the results of the Queen’s University campus experiments begin with a presentation of the 2DAM results, followed by a discussion of the LCL results that use the axis map estimated by 2DAM as its a priori axis map.

**Axis Mapping Results**

The unclustered and clustered axis maps of the Queen’s University campus estimated by 2DAM are illustrated in Figures 5.13a and 5.13b, respectively. The clustered axis map has entries $\bar{A} = (1.3 \pm 2.5^\circ, 91.9 \pm 2.6^\circ)$. Like the BMH experiments, this is quite similar to the expected axis map of $\bar{A} = (0^\circ, 90^\circ)$ indicated by orientations of the buildings in the satellite image in Figure 5.2.

The unclustered axis map in Figure 5.13a shows a greater variety of axes compared to the indoor experiment at BMH. The axes in the less dense parts of the axis map are primarily attributed to two sources: environmental anomalies (i.e., flat surfaces with axes not in the a priori axis map, which includes the exterior walls of some buildings, the sides of parked cars, and the infrastructure around pathways) and false positive axis extractions (i.e., a sufficiently large number of points from different parts of the LiDAR measurement having similar axes and being clustered together). Although the environmental anomalies are expected to (and should) appear in the axis map, one
way to reduce false positive axis extractions could be to cluster points based on both their axes and relative positions.

As was the case in the BMH experiments, straight pathways and areas whose surfaces were consistently oriented resulted in few entries being added to the orientation path, this time averaging nearly 12 m between entries. Unlike traditional mapping algorithms (e.g., pose-graph SLAM), this distance is permitted because overlap between LiDAR measurements among the entries of the orientation path is unnecessary. Note that reducing the minimum number of axes needed to form a cluster in Figure 5.13a introduces another “dominant” surface (and a potential new entry to $\bar{\mathbf{A}}$) at approximately $45^\circ$. It is speculated that chamfered corners on some buildings contributed to this cluster.

**Localization Results**

The axis map $\bar{\mathbf{A}} = (1.3 \pm 2.5^\circ, 91.9 \pm 2.6^\circ)$ that was generated by 2DAM was taken as the *a priori* axis map for LCL. With this axis map, the paths estimated by LCL and the other three localization algorithms are shown in Figure 5.14. Furthermore, the heading estimates of each localization algorithm (except odometry) are plotted in Fig-

![Figure 5.13: Two-dimensional axis maps of the Queen’s University campus estimated with 2DAM.](image-url)
Figure 5.14: The paths estimated by three different localization algorithms (gyroscope, scan matching, and LCL) for the first trial of the Queen’s University campus data set. The odometry localization path is not shown; it is the least accurate and cannot be contained in the scale of this plot. This trial is illustrated because it represents the best result (i.e., smallest root-mean-square error) achieved by scan matching. The vehicle both started and finished at the origin of the plot (driving clockwise).

Figure 5.15. Finally, comparisons of the estimated and true headings of each localization algorithm over the course of the five trials is summarized in Table 5.2.

The heading estimated by scan matching sometimes suffered singular points of failure due to the experiments taking place in a three-dimensional environment. When the vehicle transitioned from a downward slope to a flat surface, its rear-facing LiDAR would sometimes suddenly measure the ground behind the vehicle. This situation is illustrated in Figure 5.16. When this occurred near a turn, the heading estimate provided by scan matching would sometimes incorrectly shift by up to tens of degrees. LCL was unaffected by these abrupt changes in the scans. As a result, to fairly com-
Figure 5.15: A comparison of the heading errors of the gyroscope, scan matching, and LCL estimates for the first trial on the Queen’s University campus. This trial is illustrated because it represents the best result (i.e., smallest root-mean-square error) achieved by scan matching. To improve the readability of the plot, only every tenth point is plotted. Similarly to Figure 5.12, the shaded areas illustrate the 3σ uncertainty of the errors, and include the (relatively small) uncertainty of the ground truth.

Table 5.2: A summary of the results from all trials on the Queen’s University campus. The number in parentheses next to each scan matching result is the number of singular points of failure that were manually corrected.

<table>
<thead>
<tr>
<th></th>
<th>Trial</th>
<th>Time [s]</th>
<th>Root Mean Square Error [deg]</th>
<th>Gyroscope</th>
<th>Scan Matching</th>
<th>LC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>365</td>
<td></td>
<td>16.64</td>
<td>11.58 (2)</td>
<td>2.79</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>358</td>
<td></td>
<td>15.57</td>
<td>36.49 (3)</td>
<td>4.06</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>365</td>
<td></td>
<td>15.76</td>
<td>31.66 (3)</td>
<td>3.03</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>338</td>
<td></td>
<td>16.47</td>
<td>14.89 (2)</td>
<td>2.81</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>346</td>
<td></td>
<td>19.76</td>
<td>30.58 (3)</td>
<td>3.61</td>
</tr>
</tbody>
</table>
Figure 5.16: A LiDAR measurement moments (a) before, and (b) shortly after the vehicle transitioned from a downward slope to a flat surface. The arrow indicates the approximate pose of the LiDAR. The time between these two measurements is approximately 0.25 s. This sharp transition often caused singular points of failure in the heading estimated by scan matching.

pare the performance of scan matching, these erroneous occurrences were manually removed from the resulting estimates. The number of manual corrections required for each trial is shown in parentheses in Table 5.2.

Comparing the estimated paths in Figure 5.14 illustrates how very simple a priori information can vastly improve heading estimation, and by extension, localization. Although the vehicle was often driven in areas where surfaces corresponding to entries of $\bar{A}$ were not present, tracking local surfaces in $\hat{A}$ sufficiently limited the growth of heading errors until entries of $\bar{A}$ could be re-observed. The result is an estimated path that differs from ground truth mostly in scale, and not in rotation. This scale difference can mostly be attributed to two sources: (i) the translation error of the encoder measurements; and (ii) changes in elevation, which are reflected in the ground truth but not the two-dimensional pose estimate provided by odometry/LC. One application previously mentioned for LCL is the generation of a good initial guess for pose-graph SLAM algorithms. The (mostly) scale-only difference between the LCL estimate and the ground truth presents the interesting possibility of developing a
lightweight pose-graph implementation that uses this constraint. In other words, an acceptable map could be produced by assuming the orientation of poses in the pose-graph are correct, and optimizing only over positions, which makes the optimization completely linear. Although not as accurate as a full pose-graph SLAM implementation, this simplification may be sufficient for some practical applications.

The heading error plots illustrated in Figure 5.15 show that the LCL estimate typically stayed near zero for the duration of the trial, with occasional spikes in its uncertainty. These spikes (e.g., at 55 s, 205 s, and 315 s) were consistent in all of the trials. They tended to occur while the vehicle underwent a rapid change in heading while few local surfaces were available for axis extraction. As a result, heading estimation briefly relied on the highly uncertain prediction provided by the steering encoder on the vehicle; i.e., the assumption that flat surfaces are frequently observable was violated. A temporary lack of surfaces when the vehicle is moving relatively straight is not a problem if the sensor providing the prior estimate has minimal bias (i.e., it reports near-zero heading change when driving straight). In all cases, recovery was rapid once flat surfaces were re-observed.

The importance of accurately modelling the covariance matrices of the motion measurements and the variance of the axis measurements (particularly while no entries of $\bar{A}$ are being observed) cannot be overstated; overconfidence in these measurements may prevent the association of future axis measurements with entries of $\bar{A}$ (false negative), while underconfidence may incorrectly associate an axis measurement with an entry of $\bar{A}$ (false positive). The latter case was responsible for the spike at approximately 205 s in the plots in Figure 5.15, where a wall whose axis differed only slightly from an entry of $\bar{A}$ was associated incorrectly.
Chapter 6

Joint Orientation Estimation

Joint orientation estimation is the process of determining the joint sets present in a rock mass from a set of measurements. It is usually performed by using hand tools (e.g., a compass and inclinometer), or more recently, by using LiDAR. These methods are pictured in Figure 6.1. The use of LiDAR for joint orientation estimation is an active field of research [58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68]. In general, these methods use a stationary tripod-mounted LiDAR to capture one or more point clouds of the exposed rock face. The orientation of the sensor is measured beforehand such that the collected data can be transformed to the global coordinate frame. Using the geometry of the captured point cloud, the orientations of planar surfaces are extracted using various methods.

The use of stationary LiDAR (e.g., Leica ScanStation C10 [69] or Renishaw Quarryman Pro [70]) has several advantages compared to hand measurements. Safety is improved by limiting human exposure to potentially unstable rock faces, it allows greater access to hard-to-reach locations, it is faster and less labour intensive, and the opportunity for introducing erroneous measurements due to procedural difficulties, human bias, and human error is significantly reduced. Despite these advantages, there are barriers preventing widespread adaption of stationary LiDAR. Its high cost can be prohibitive, it is often physically too large and heavy for some remote deployments, processing the resulting point clouds often requires manual intervention.

“What are men to rocks and mountains?”
— Jane Austen
to remove outliers, and several measurements of the same rock face from different points of view are usually required to avoid occlusions. This last drawback can make the data collection process very time-consuming (i.e., moving, reorienting, and taking new measurements with the sensor can take tens of minutes) and is particularly challenging in enclosed spaces (e.g., underground mines).

Because axes can be used to represent the orientations of planes in three dimensions, they are suitable representations of the orientations of joint sets in rock masses. Furthermore, joint orientation estimation is a direct application of three dimensional axis mapping (3DAM); that is, they share the same goal of mapping the axes of planar surfaces in an environment. Many of the disadvantages of using stationary LiDAR are avoided with 3DAM. The use of low cost LiDAR (e.g., Microsoft Kinect v2 [71] or Fotonic E70 [72]) is permitted because lower resolution point clouds can be captured more frequently while the mobile platform is in motion. Occlusions are easily and quickly eliminated by simply maneuvering around the rock face. The time and effort required to collect data is significantly reduced compared to both manual techniques and stationary LiDAR. Finally, the mobile platform can take any form capable of car-
ry ing the required sensors, from a handheld configuration to remote operations on a mobile robot.

## 6.1 Related Work

Research concerning joint orientation estimation using LiDAR has focused on different strategies for extracting the orientation of planar surfaces from point clouds. In general, these strategies can be categorized as either *top-down* or *bottom-up* approaches. The top-down approach (sometimes called *surface reconstruction*) uses a mesh generated from the point cloud (usually by interpolating between points) to approximate the surface of the measured rock face, and calculates the normal vector at each segment in the mesh. Many researchers depend on commercial software (e.g., PolyWorks [73]) to perform surface reconstruction. Slob *et al.* [74] evaluated this approach and compared it to manual techniques from a cost-benefit perspective. Other researchers have studied optimizing the point cloud processing procedure [61], correcting for biases in data collection [63], and have analyzed the sensitivity to point cloud resolution [62, 66]. The top-down approach often requires manual intervention to prevent unwanted points from being included in the surface reconstruction [66] (e.g., vegetation, outliers).

The bottom-up approach attempts to find subsets of points on planar surfaces and uses least-squares techniques (e.g., principal component analysis) to fit planes to the data. This approach requires finding subsets of points in the point cloud that measure planar surfaces. A plane is then fit to the subset of points and its normal vector is calculated. Various methods have been used to find the appropriate planar subsets. For example, random sample consensus (RANSAC) has been shown to be an effective way to find subsets belonging to planar surfaces by iteratively comparing subsets to the model of a plane [58, 60, 67]. Gigli *et al.* [65] divide the point cloud into a cubic grid of various resolutions to evaluate the planarity of points in each grid cell. Olariu
et al. [59] apply a $k$-means clustering algorithm at different resolutions and measure the planarity of the clusters.

A thorough comparison by Slob [75] of the two techniques concluded that the bottom-up approach is preferable because it is easier to automate, retains the original point cloud, and is better at dealing with outliers. The implementation of 3DAM described in this chapter uses a modified form of the bottom-up approach for axis extraction. Instead of fitting planes to discrete planar surfaces, it estimates the planarity at each point in the point cloud and then clusters similar points across the entire point cloud.

There has been limited research on the use of a mobile platform for joint orientation estimation. De Agostino et al. [76] mounted a GPS, IMU, LiDAR, and camera to a truck, but only captured point clouds while the LiDAR was stationary. As a result, this system was very similar to the established techniques, with the exception that the sensors were more easily moved from one scanning location to the next. A mobile LiDAR system constructed by Terrapoint Canada [77] (now Ambercore) and employed by Lato et al. [78] captured point clouds while in motion. This system combines multiple LiDAR sensors, a high-end IMU, and a differential GPS system to track the movement of a truck mounted on railway tracks. The system constructs a massive point cloud in the global coordinate frame, after which a top-down approach was applied to manually selected portions of the point cloud. Put differently, the point cloud was treated exactly as if it were captured with a stationary LiDAR; that is, there were no modifications made to the joint orientation algorithm to account for mobile data collection.

Rather than constructing a large, high-resolution point cloud and applying standard techniques to extract the joint orientations, the implementation of 3DAM described in this chapter is developed with joint orientation estimation in mind. It is designed to be relatively inexpensive, easily carried and operated by a single person,
and have no dependence on GPS. By meeting these constraints, it is deployable in a large variety of environments (i.e., anywhere where a person or remotely operated vehicle can reach, including underground) and is both economically and physically viable as a drop-in replacement for a compass and inclinometer. Although the expensive stationary LiDAR devices (i.e., tens of thousands of dollars [79]) used in the literature discussed above produce large, high-resolution point clouds, they must remain stationary during measurements that can take several minutes. On the other hand, 3DAM can use a low-cost (i.e., hundreds of dollars), lower resolution LiDAR. The need for high resolution is mitigated by using a LiDAR with a much higher frequency, which captures multiple complete point clouds per second.

6.2 Problem Description

This chapter describes how 3DAM is used to perform joint orientation determination. As explained in Chapter 4, 3DAM estimates \( \hat{x} = (\hat{Q}, \hat{A}) \), which consists of the estimated orientation path \( \hat{Q} = (\hat{q}_1, \ldots, \hat{q}_n) \) of the mobile platform, and an estimated axis map \( \hat{A} = (\hat{m}_1, \ldots, \hat{m}_m) \) of the environment. Each entry \( \hat{q}_i \in S^3 \) of the orientation path is a unit quaternion (Section 3.3) representing a rotation from the global coordinate frame \( \mathcal{F}_g \) to the mobile platform coordinate frame \( \mathcal{F}_m \), such that a vector \( \mathbf{a} \in \mathbb{R}^3 \) expressed in \( \mathcal{F}_g \) is rotated to \( \mathcal{F}_m \) by the transformation \( \mathbf{C}(\hat{q}_i)\mathbf{a} \). Each entry \( \hat{m}_j \) of the axis map is a three-dimensional unit axis (Section 2.3) of a reoccurring planar surface in the environment expressed in \( \mathcal{F}_g \). In the context of joint orientation estimation, the entries of the axis map are the joint sets of a rock face.

To collect the data required for 3DAM, a mobile platform carrying an inertial measurement unit (IMU) and a time-of-flight (ToF) camera is used. A ToF camera is a type of scannerless LiDAR that captures entire point clouds simultaneously. The IMU consists of a three-axis accelerometer, three-axis gyroscope, and three-axis magnetometer. The measurements from each of these sensors undergo some processing...
before being used for axis mapping, which is outlined in detail in Appendix D. In particular, the bias of the gyroscope is estimated (Section D.1), the accelerometer measurements are transformed into measurements of the direction of gravitational acceleration (Section D.2), the magnetometer measurements are transformed in measurements of the direction of the Earth’s magnetic field (Section D.3), and the LiDAR measurements are transformed into measurements of the axes of planar surfaces in its field of view (Section D.4). A summary of these transformations is provided in Table 6.1. It is important to note that the processed accelerometer and magnetometer measurements are unit directions in $S^2$, which share the same topological space as unit axes, but differ in their manifold encapsulation. These differences are detailed in Appendix E. All measurements by the IMU and the LiDAR are expressed in $F_m$.

A high-level description of the steps required to perform joint orientation estimation are enumerated below. Detailed descriptions of these steps are provided in the subsequent sections.

(i) Determine an initial estimate of the orientation path $\hat{Q}$ by using the IMU measurements in a Kalman filter (Section 6.3).

(ii) Determine an initial estimate of the axis map $\hat{A}$ by rotating all the axes measured by the LiDAR to $F_g$ by using the initial estimate of the orientation path
(iii) Transform the IMU and LiDAR measurements into the observations required for axis mapping (Section 6.5).

(iv) Execute a slightly modified version of the 3DAM algorithm that is outlined in Algorithm 2 (Section 6.6).

(v) Generate the final axis map of the rock face, which is called a stereonet in the context of joint orientation estimation (Section 6.7).

6.3 Initializing the Orientation Path

A initial estimate of the orientation path is required to maximize the objective function (as described in detail in Chapter 4). The implementation of 3DAM described in this chapter simply adds a new entry to $\mathbf{Q}$ at each LiDAR measurement (whose frequency is assumed to be much smaller than the IMU). The initial estimate of the orientation path is determined by fusing the processed IMU measurements with a Kalman filter. Estimating the orientation of a mobile platform with a Kalman filter has extensive heritage, especially in the context of spacecraft attitude determination [80]. The implementation outlined here combines some of the elements of the Kalman filters described by Trawny [81] and Kraft [82].

At time step $k$, let $\{\hat{\mathbf{q}}_k, \mathbf{P}_k\}$ be a random rotation (Section 3.5) representing the estimated orientation of the mobile platform. The initial estimate $\{\hat{\mathbf{q}}_0, \mathbf{P}_0\}$ is calculated with the factored quaternion algorithm [83], which first estimates the roll and pitch of the mobile platform with an accelerometer measurement, and then estimates the yaw with a magnetometer measurement. Subsequent estimates of $\{\hat{\mathbf{q}}_k, \mathbf{P}_k\}$ are predicted with the processed gyroscope measurements and corrected using the processed accelerometer and magnetometer measurements.
6.3.1 Orientation Prediction

The prediction step of the Kalman filter propagates the previous orientation estimate $\hat{q}_{k-1}$ forward in time by using an unbiased gyroscope measurement $\tilde{\omega}_k$. This prediction is calculated by using a zero-th order quaternion integrator (i.e., the angular velocity is assumed constant over the short interval between measurements); i.e.,

$$\hat{q}_k = \exp \left( \frac{\tilde{\omega}_k \Delta t}{2} \right) \hat{q}_{k-1}, \quad (6.1)$$

where $\Delta t$ is the duration from time step $k - 1$ to $k$. This prediction corresponds to rotating the previous estimate around the axis of rotation $a_k = \tilde{\omega}_k / \|\tilde{\omega}_k\|$ through the angle $\theta_k = \|\tilde{\omega}_k\| \Delta t$. The covariance matrix $P_{k-1}$ is propagated to time step $k$ by

$$P_k = F_k P_{k-1} F_k^T + Q_k \Delta t, \quad (6.2)$$

where $F_k$ is the rotation matrix parameterization of the integrated angular velocity measurement calculated with Rodrigues’ rotation formula; i.e.,

$$F_k = \cos(\theta_k) + I_3 a_k a_k^T (1 - \cos(\theta_k)) - a_k^\times \sin(\theta_k). \quad (6.3)$$

A detailed derivation of a similar prediction step is available in [81].

6.3.2 Orientation Correction

The correction step of the Kalman filter corrects the predicted orientation by incorporating a processed accelerometer and magnetometer measurement. The known directions of gravitational acceleration and the Earth’s magnetic field in $\mathcal{F}_g$ are rotated to $\mathcal{F}_m$ and are compared with the corresponding directions measured by the accelerometer and magnetometer, respectively. Let $h : S^3 \times S^2 \rightarrow S^2$, where

$$h(q, d) = C(q)d \quad (6.4)$$
Axis Mapping  Chapter 6: Joint Orientation Estimation

rotates the unit direction \( \mathbf{d} \in S^2 \) from \( \mathcal{F}_g \) to \( \mathcal{F}_m \) via the unit quaternion \( \mathbf{q} \in S^3 \), where the rotation matrix \( C(\mathbf{q}) \in SO(3) \) is defined in (3.19) on page 35. Because the predicted orientation \( \hat{\mathbf{q}}_k \) represents the estimated rotation from \( \mathcal{F}_g \) to \( \mathcal{F}_m \), the unit direction \( \hat{\mathbf{z}}_k = h(\hat{\mathbf{q}}_k, \mathbf{d}) \) is the expected direction of gravitational acceleration (i.e., \( \mathbf{d} = [0 \ 0 \ -1]^\top \)) or the expected direction of the Earth’s magnetic field (i.e., \( \mathbf{d} \) is the normalized magnetic field vector obtained from a database) expressed in \( \mathcal{F}_m \). Furthermore, let \( \{\hat{\mathbf{z}}_k, \hat{\mathbf{R}}_k\} \) be either the measured direction of gravitational acceleration \( \{\hat{\mathbf{a}}_k, \hat{\mathbf{R}}_k^{(a)}\} \) by the accelerometer, or the measured direction of the Earth’s magnetic field \( \{\hat{\mathbf{m}}_k, \hat{\mathbf{R}}_k^{(m)}\} \) by the magnetometer. The correction step of an unscented Kalman filter (UKF) [84] is used to correct the predicted orientation because \( h \) is nonlinear with respect to \( \hat{\mathbf{q}}_k \).

To begin, a set of sigma points \( q_1, \ldots, q_6 \in S^3 \) are sampled from the predicted orientation; i.e.,

\[
q_i = \begin{cases} 
\hat{\mathbf{q}}_k \boxplus \left( \sqrt{3}P_k \right)_i & \text{for } i = 1, 2, 3 \\
\hat{\mathbf{q}}_k \boxplus - \left( \sqrt{3}P_k \right)_{i-3} & \text{for } i = 4, 5, 6 
\end{cases}
\] (6.5)

where \((A)_i\) is the \( i \)-th column of the matrix \( A \). Each of these sigma points is then passed through \( h \); i.e.,

\[
\mathbf{z}_i = h(q_i, \mathbf{d}), \quad \text{for } i = 1, \ldots, 6,
\] (6.6)

which transforms each sigma point and the known unit direction \( \mathbf{d} \in S^2 \) expressed in \( \mathcal{F}_g \) to a new sigma point \( \mathbf{z}_i \in S^2 \) that represents the known direction expressed in \( \mathcal{F}_m \). The expected measurement \( \hat{\mathbf{z}}_k \) is the normalized mean of the transformed sigma points; i.e.,

\[
\hat{\mathbf{z}}_k = f\left( \frac{1}{6} \sum_{i=1}^{6} \mathbf{z}_i \right),
\] (6.7)

where \( f : \mathbb{R}^3 \to S^2 \) is the normalization function given in (D.4). The covariance matrix
The correction $\Delta \theta \in h^3$ to the predicted orientation is proportional to the difference between the measured and expected direction; i.e.,

$$\Delta \theta = K (\tilde{z}_k \ominus \hat{z}_k)$$

(6.10)

which is scaled by the Kalman gain $K \in R^{3x2}$, where

$$K = P_{qz} P_{zz}^{-1}.$$  

(6.11)

The correction is applied to the predicted orientation using the $\boxplus$ operator (defined for three-dimensional rotations in Section 3.4), where

$$\hat{q}_k \leftarrow \hat{q}_k \boxplus \Delta \theta.$$  

(6.12)

Because the covariance matrices of rotations and directions are represented by unconstrained parameterizations, the covariance matrix $P_k$ of the corrected orientation is simply the standard UKF update

$$P_k \leftarrow P_k - KP_{zz} K^T.$$  

(6.13)

The correction step is performed twice: once each for the processed accelerometer and magnetometer measurements.

The prediction and correction of the orientation are performed at every IMU measurement. However, recall that the orientation path consists of only orientations at
the LiDAR measurements. Because the IMU data is collected at a much higher frequency than the LiDAR data (e.g., 100 Hz compared to 2 Hz), the Kalman filter estimate closest to each LiDAR measurement is used to initialize the orientation path. Note that the orientation of the mobile platform is fully observable given the IMU data (as long as the magnetic field and gravity vectors are not collinear [85]), that is, the uncertainty of the orientation estimate is bounded. This is critical in determining a good initial estimate of the axis map (Section 6.4), for the reason illustrated in Figure 5.3 on page 74.

### 6.4 Initializing the Axis Map

An initial estimate of the axis map is required to maximize the objective function. The method used to initialize the axis map is analogous to the two-dimensional approach described in Section 5.3.2 and illustrated in Figure 5.5 on page 77. At each estimated orientation \( \hat{q}_i \) in the orientation path, the LiDAR measurement produces a set of \( p \geq 0 \) random axes \( \{z_1, R^{(\delta\phi_1)}\}, \ldots, \{z_p, R^{(\delta\phi_p)}\} \) expressed in \( F_m \) by using the method described in Section D.4 in Appendix D. Each measured axis is transformed to \( F_g \) by

\[
    z_j^{(g)} = C \left( \hat{q}_i^{-1} \right) z_j,
\]

where \( z_j^{(g)} \) is the measured axis expressed in \( F_g \) and \( C \left( \hat{q}_i^{-1} \right) \in SO(3) \) is the estimated rotation from \( F_m \) to \( F_g \) by definition. Applying (6.14) to every measurement \( z_j \) at every orientation \( \hat{q}_i \) results in all measured axes being expressed in \( F_g \). Due to the nature of the rock face, this should result in clusters of axes at each of the joint sets. As before, the clusters are extracted (with DBSCAN [48]) and their means are calculated (with the method described by Beran and Fisher [86]), forming an initial estimate of the axis map \( \hat{A} \). Furthermore, each measured axis belonging to a cluster is considered associated with the corresponding entry of \( \hat{A} \) derived from that cluster.
6.5 Obtaining the Axis Mapping Observations

As described in Chapter 4, up to three types of observations are used to perform axis mapping. For 3DAM, these are the random rotations $z_q, z_q, z_q ∈ S^3$, and the random axes $z_q, z_q ∈ S^2$. Table 4.1 on page 46 describes these observations. The implementation of 3DAM described in this chapter uses the factor graph illustrated in Figure 4.3c on page 48. Although all the sensors required to use the factor graph in Figure 4.3a are available, the gyroscope measurements (which would normally be integrated to produce observations of the type $z_q,q$) are already fused into the Kalman filter estimate, which is used to produce observations of the type $z_q$ (as described in Section 6.5.1 below). This section describes how the two types of observations in the factor graph are obtained from the processed LiDAR and IMU measurements.

6.5.1 Obtaining Observations of the Orientation Path Entries

An observation $z_q$ is a direct measurement of the $i$-th orientation path entry $q_i$, which represents the rotation from the global coordinate frame $F_g$ to the mobile platform coordinate frame $F_m$. Recall that the initialization of the orientation path (Section 6.3) produces this type of observation by way of fusing the processed gyroscope, accelerometer, and magnetometer measurements with a Kalman filter. More specifically,

$$z_q ← \hat{q}_k, \quad R_q ← P_k,$$

where $\{\hat{q}_k, P_k\}$ is the random rotation of the mobile platform estimated by the Kalman filter and $k$ is the time step of the $i$-th LiDAR measurement.

6.5.2 Obtaining Observations of the Axis Map Entries

An observation $z_{q|m}$ is a measurement of the $j$-th axis map entry $m_j$ taken from the $i$-th orientation path entry $q_i$, expressed in $F_m$. Recall that this type of observation is produced by processing the LiDAR measurement (Section D.4 in Appendix D) and
associating them with entries of the axis map when it is initialized (Section 6.4). More specifically,

\[ z_{q,m_j} \leftarrow z_j, \quad R_{q,m_j} \leftarrow R^{(\delta \phi_j)}, \quad (6.16) \]

where \( \{z_j, R^{(\delta \phi_j)}\} \) is the random axis extracted from a LiDAR measurement at the \( i \)-th orientation path entry, associated with the \( j \)-th axis map entry.

### 6.6 Execution of the 3DAM Algorithm

At this point, an initial estimate of the orientation path has been calculated by fusing the IMU data with a Kalman filter (Section 6.3) and an initial estimate of the axis map has been calculated by extracting axes from the LiDAR data, rotating them from \( F_m \) to \( F_g \), and grouping them into clusters (Section 6.4). Furthermore, the sensor data has been processed into the types of observations used by 3DAM (Section 6.5). This meets all the requirements necessary to execute the 3DAM algorithm outlined in Algorithm 2 on page 62. As a result, the algorithm is executed with one minor change to allow for possible amendments to data association as the design parameter is updated, as described next.

The initialization of an estimate of the axis map \( \hat{A} \) and the association of the measured axes with entries of \( \hat{A} \) requires an estimate of the orientation path \( \hat{Q} \), as described in detail in Section 6.4. Because each iteration of the 3DAM algorithm perturbs the estimate of the design parameter \( \hat{x} \) (which contains \( \hat{Q} \)), it is possible that a new estimate of \( \hat{A} \) that uses the perturbed \( \hat{Q} \) may result in different associations of the measured axes. As a result, each time the design parameter is updated, the steps described in Section 6.4 are redone. This requires a minor amendment to Algorithm 2. After the design parameter is updated on line 10, the following step is added:

*Update the estimate of \( \hat{A} \) and the associations among observations of the type \( z_{q,m_j} \) and the entries of \( \hat{A} \).*
Figure 6.2: Dip, dip direction, and a stereonet. (a) The dip angle $\alpha$ and dip direction angle $\beta$ of a plane, as it relates to the global coordinate frame and the axis of the plane. (b) An axis plotted on a stereonet parameterized by its dip $\alpha$ and dip direction $\beta$.

6.7 Generating Stereonets

The most common method, particularly in the geosciences community, used to report the orientation of the planar surfaces visible in a rock face (and therefore, the joint sets) is by producing a stereonet. A point plotted on a stereonet is the result of applying a stereographic projection to an axis defined in the global coordinate frame (North-East-Down) and parameterizing the result as the dip and dip direction of the axis. The dip $\alpha \in [0, \pi/2]$ is defined as the angle from the negative down (i.e., up) direction to the axis of the plane, or equivalently, from the North-East plane (i.e., the horizontal plane) to the measured plane. The dip direction $\beta \in [0, 2\pi]$ is the angle from North to the direction of the axis measured clockwise. These quantities are illustrated in Figure 6.2a and are plotted on a stereonet in Figure 6.2b.

Let $\mathbf{m} = [\lambda \ \kappa_1 \ \kappa_2]^\top$ be a unit axis, and let $\mathbf{\phi} = \log(\mathbf{m}) = [\phi_1 \ \phi_2]^\top$ be the axis
vector parameterization of \( \mathbf{m} \). Then for \( \lambda \geq 0 \),

\[
\alpha = \frac{\pi}{2} - \text{atan2} \left( \lambda, \sqrt{\kappa_1^2 + \kappa_2^2} \right) = \sqrt{\phi_1^2 + \phi_2^2}
\]

\[
\beta = \text{atan2}(\kappa_2, \kappa_1) + \pi = \text{atan2}(\phi_1, \phi_2) + \pi.
\]

(6.17)

After calculating the optimal estimate of the design parameter \( \hat{x}^* \) with the procedure outlined in Section 6.6, a stereonet of the rock face is generated by first rotating each measured axis from \( F_m \) to \( F_y \) with the entries of \( \hat{Q}^* \), and then converting the result to a dip and dip direction with (6.17). Although \( \hat{A}^* \) represents an estimate of the joint sets, stereonets typically contain the individual measurements of the planar surfaces. Given the stereonet, it is left to the geologist or geotechnical engineer to interpret the data.

6.8 Pseudocode

Pseudocode outlining how joint orientation estimation algorithm is performed with 3DAM is provided in Algorithm 5. The variables used in this algorithm are defined throughout this chapter.

6.9 Experiments

Experiments were performed at three different field sites, two of which employed a handheld configuration of the mobile platform and one which employed a remotely operated mobile robot. This section provides a detailed description of both the apparatus and the field sites.

6.9.1 Apparatus

A mobile platform was constructed consisting of a Microsoft Kinect v2 time-of-flight camera [71] and a LORD MicroStrain 3DM-GX3-25 IMU [50]. The two sensors were rigidly connected on a short length of aluminum extrusion, which was outfitted with
Algorithm 5: A summary of the 3DAM joint orientation estimation algorithm.

**Require:** Timestamped and processed IMU and LiDAR measurements
1: Estimate initial orientation \( \{ \hat{q}_0, P_0 \} \) from first IMU measurement with factored quaternion algorithm
2: **for all** processed IMU measurements \( \{ \tilde{\omega}_k, Q_k \}, \{ \tilde{a}_k, \tilde{R}^{(a)}_k \}, \{ \tilde{m}_k, \tilde{R}^{(m)}_k \} \) **do**
3: Predict \( \{ \hat{q}_k, P_k \} \) with \( \{ \tilde{\omega}_k, Q_k \} \) and \( \{ \hat{q}_{k-1}, P_{k-1} \} \)
4: Correct \( \{ \hat{q}_k, P_k \} \) with \( \{ \tilde{a}_k, \tilde{R}^{(a)}_k \} \)
5: Correct \( \{ \hat{q}_k, P_k \} \) with \( \{ \tilde{m}_k, \tilde{R}^{(m)}_k \} \)
6: **end for**
7: Initialize \( \hat{Q} \) by sampling \( \hat{q}_k \) at LiDAR measurement time steps \( k \)
8: Initialize \( \hat{A} \) and by rotating all measurements to \( F_g \) and clustering similar axes
9: Obtain observations of types \( \{ z_q, R_q \} \) and \( \{ z_q, R_q, R_{m} \} \) from processed measurements
10: Execute amended version of Algorithm 2
11: Rotate each measured axis \( z_j \) from \( F_m \) to \( F_g \) with entries of \( \hat{Q}^* \)
12: Convert measured axes to dips and dip directions
13: **return** stereonet

handles for handheld use (Figure 6.3). The sensors were wired to a laptop and a small battery carried inside a backpack, and a tablet computer was connected to the mobile platform that provided information to the user (sensor status, distance from rock face, estimated planarity of point cloud, and acceleration). Specifications of the Microsoft Kinect v2 are listed in Table 6.2. Note that the noise depends on both the range and operating conditions. An in-depth analysis of the noise characteristics of the Kinect v2 was performed by Fankhauser et al. [87]. Comprehensive specifications of the LORD MicroStrain 3DM-GX3-25 IMU are available in its manual [50]. Note that this version of the mobile platform is a proof-of-concept prototype, and it is conceivable that it could be made much more compact.

In one experiment, a stereonet was generated by processing the point cloud measured by an MDL (now Renishaw) Quarryman Pro 3D laser scanner [70] (stationary time-of-flight LiDAR). More details about this process are described in Section 6.9.2 and its results are presented in Section 6.10.2.
Figure 6.3: The mobile platform constructed for joint orientation estimation. A Microsoft Kinect v2 time-of-flight camera (scannerless LiDAR) and a LORD MicroStrain 3DM-GX3-25 IMU are mounted to a piece of aluminum extrusion. The attached handles allow for handheld operation.

Table 6.2: Specifications of the Microsoft Kinect v2.

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimensions (mm)</td>
<td>249 × 66 × 67</td>
</tr>
<tr>
<td>Mass (kg)</td>
<td>0.970</td>
</tr>
<tr>
<td>Resolution (pixels)</td>
<td>512 × 424</td>
</tr>
<tr>
<td>Min. range (m)</td>
<td>0.5</td>
</tr>
<tr>
<td>Max. range (m)</td>
<td>4.5</td>
</tr>
<tr>
<td>Horiz. field of view (deg)</td>
<td>70</td>
</tr>
<tr>
<td>Vert. field of view (deg)</td>
<td>60</td>
</tr>
<tr>
<td>Axial and lateral noise (mm)</td>
<td>&lt; 10</td>
</tr>
<tr>
<td>Max. frequency (frames per second)</td>
<td>30</td>
</tr>
<tr>
<td>Power usage (W)</td>
<td>15</td>
</tr>
<tr>
<td>Cost (USD)</td>
<td>200</td>
</tr>
</tbody>
</table>
6.9.2 Experiments

Three experiments were performed to assess the application of 3DAM for joint orientation estimation and to compare its results to other established methods. The locations of these experiments were

(i) a road cut near Brewer Lake (44°24′04″ N, 76°18′46.9″ W), approximately 25 km northeast of Kingston, ON, Canada;

(ii) a road cut near Ivy Lea (44°22′13.2″ N, 75°58′37.5″ W), approximately 45 km east of Kingston, ON, Canada; and

(iii) an old railway tunnel on the Kettle Valley Trail in Myra-Bellevue Provincial Park (49°46′58.5″ N, 119°18′29.2″ W), approximately 20 km southeast of Kelowna, BC, Canada.

The 19.5 m × 2.5 m section of the road cut at Brewer Lake used in the experiments is pictured in Figure 6.4. Brewer Lake was selected as a test site because joint orientation estimation was previously performed on the same rock face by Lato et al. [61], which included results from both hand measurements (compass and inclinometer) and stationary LiDAR. These results are compared with 3DAM in Section 6.10.1.

The 6.1 m × 2.5 m section of the road cut at Ivy Lea used in the experiments is pictured in Figure 6.5. In addition to hand measurements and the data collected for 3DAM, a point cloud of the rock face was collected using the MDL Quarryman Pro. A stereonet was generated from this point cloud extracting axes at each point using
Figure 6.5: The section of rock face at Ivy Lea used in the joint orientation estimation experiments. Note that a small ditch occludes the bottom portion of the rock face in this picture.

Figure 6.6: The Clearpath Husky A200 mobile robot in the old railway tunnel in Myra-Bellevue. The 3DAM mobile platform illustrated in Figure 6.3 is mounted on top of the robot. The rock face used in the joint orientation estimation experiments is shown the background.

The method outlined in Section D.4 in Appendix D (i.e., processing the point cloud in the same way as the time-of-flight camera point clouds). Results of these different methods of joint orientation estimation are presented in Section 6.10.2.

The mobile platform shown in Figure 6.3 was mounted on a Clearpath Husky A200 [51] mobile robot and driven through an old railway tunnel in Myra-Bellevue (pictured in Figure 6.6). The robot was remotely driven along a 6 m section of the tunnel wall at approximately 0.5 m/s using a joystick. Hand measurements were taken at the same section of the tunnel and the results are presented in Section 6.10.3.

At Brewer Lake and Ivy Lea, data was collected with the mobile platform in two different ways: biased and unbiased handheld LiDAR scanning. In the biased experiments, the operator smoothly moved the mobile platform through the environment
while deliberately pointing the LiDAR at planar surfaces. In the unbiased experiments, the operator moved the mobile platform in the predefined pattern illustrated in Figure 6.7. In the Myra-Bellevue experiments, the mobile robot was simply driven in a straight line roughly parallel to the tunnel wall. In all experiments, data was collected with the LiDAR at 2 Hz and with the IMU at 100 Hz.

### 6.10 Results and Discussion

This section presents the results of the individual experiments at Brewer Lake (Section 6.10.1), Ivy Lea (Section 6.10.2), and Myra-Bellevue (Section 6.10.3). The dips and dip directions of the joint sets presented in this section were calculated using the fuzzy cluster analysis tool in the Rocscience Dips software package [88]. This tool automatically determines the axes belonging to a joint set within a 30 degree radius of a highly concentrated area in a stereonet and reports the mean dip and dip direc-
tion of the resulting cluster. By using a common convention [89], the dip and dip direction are reported as 71/054 (for example) for a dip of 71° and a dip direction of 54°. Furthermore, because it is a common practice, contours representing the Fisher concentration of the axes (axes per 1% area) are included on the stereonets.

It is important to remark that the stereonets produced by hand measurements presented in the following sections should not be considered ground truth for the following reasons: (i) the tool used for hand measurements has a resolution of no less than one degree in both dip and dip direction; (ii) measuring the dip direction manually is very difficult for dips near zero; (iii) hand measurements assume the surface being measured is perfectly planar; (iv) hand measurements are subject to selection bias; that is, the measured surfaces may not be representative of the rock face as a whole because it is the responsibility of the (skilled) operator to choose which surfaces are measured.

A comprehensive manual survey of all axes by taking hand measurements is usually impractical due to hard-to-reach areas, safety concerns (e.g., rock falls and elevated areas), and time (i.e., the work is labourious and time-consuming). Furthermore, before comparing the stereonets, one must consider two sources of uncertainty in the mapped joint sets. First, uncertainty is introduced by noisy sensor measurements and approximations in the algorithms determining the joint sets (present in all mapping algorithms). Second, due to the various natural forces that form the rock faces, planar surfaces belonging to the same joint set will not be exactly parallel. As a result, if perfect ground truth were available (with one axis for each planar surface), one would still expect the stereonet to contain moderately loose clusters.

6.10.1 Brewer Lake

The results of four different joint orientation estimation experiments are compared and discussed in this section: (i) hand measurements using a compass and inclinome-
ter (Fig. 6.8a); (ii) the top-down approach using a stationary LiDAR (Fig. 6.8b); (iii) biased 3DAM (Fig. 6.8c); (iv) unbiased 3DAM (Fig. 6.8d). The results of (i) and (ii) are reproduced from [61] with the permission of the authors. Note that the reproduced stereonets differ slightly compared to those in [61] because the originals did not account for the local magnetic declination. A summary of the results is provided in Table 6.3.

The orientations of the three joint sets extracted from both the biased and unbiased 3DAM stereonets are comparable to those extracted with hand measurements. The largest deviation between 3DAM and the hand measurements is the orientation of Set 3. A major contributing factor to this discrepancy is the continuum of poles between Sets 2 and 3 in the 3DAM results, which perturbs the dip direction of Set 3 and is not present in the hand measurements. It should be noted that these poles are present in all trials of both the biased and unbiased 3DAM. The presence of only two poles between Sets 2 and 3 in the hand measurements suggest planar surfaces at this orientation were possibly undersampled by the hand measurements.

Compared to the top-down approach, the 3DAM stereonets have considerably less scatter and a greater concentration of poles at the three joint sets. This suggests that the robust bottom-up approach used by 3DAM is effective at rejecting points measuring non-planar surfaces, which supports the conclusions of Slob [75]. Various causes for the considerable amount of scatter in the top-down approach stereonet are discussed in [61], including difficulties in meshing areas containing low point density, and field setup issues such as position and calibration of the LiDAR. As a result, all joint sets are not easily identifiable.

The similarity of the biased and unbiased 3DAM results implies that deliberately pointing the LiDAR at planar surfaces is not necessarily required. This suggests that 3DAM is effective at ignoring the additional non-planar surfaces that may be measured by an unbiased scanning pattern, which again is likely a consequence of the
Figure 6.8: Stereonets illustrating the results of the four different joint orientation estimation methods used in the Brewer Lake experiments.
<table>
<thead>
<tr>
<th></th>
<th>Hand measurements</th>
<th>Stationary LiDAR</th>
<th>Biased 3DAM</th>
<th>Unbiased 3DAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of poles</td>
<td>101</td>
<td>257</td>
<td>605</td>
<td>688</td>
</tr>
</tbody>
</table>

Joint orientations (dip / dip direction)

<table>
<thead>
<tr>
<th>Set</th>
<th>Hand measurements</th>
<th>Stationary LiDAR</th>
<th>Biased 3DAM</th>
<th>Unbiased 3DAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set 1</td>
<td>77/137</td>
<td>61/134</td>
<td>77/134</td>
<td>75/137</td>
</tr>
<tr>
<td>Set 2</td>
<td>89/060</td>
<td>83/056</td>
<td>89/234</td>
<td>89/057</td>
</tr>
<tr>
<td>Set 3</td>
<td>28/314</td>
<td>39/308</td>
<td>40/303</td>
<td>22/297</td>
</tr>
</tbody>
</table>

Angular difference with respect to hand measurements (deg)

<table>
<thead>
<tr>
<th>Set</th>
<th>Hand measurements</th>
<th>Stationary LiDAR</th>
<th>Biased 3DAM</th>
<th>Unbiased 3DAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set 1</td>
<td>–</td>
<td>16.2</td>
<td>2.9</td>
<td>2.0</td>
</tr>
<tr>
<td>Set 2</td>
<td>–</td>
<td>7.2</td>
<td>6.3</td>
<td>3.0</td>
</tr>
<tr>
<td>Set 3</td>
<td>–</td>
<td>11.5</td>
<td>13.7</td>
<td>9.3</td>
</tr>
</tbody>
</table>

Approximate effort (min)

<table>
<thead>
<tr>
<th></th>
<th>Hand measurements</th>
<th>Stationary LiDAR</th>
<th>Biased 3DAM</th>
<th>Unbiased 3DAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setup / take down</td>
<td>0</td>
<td>~30^a</td>
<td>5^b</td>
<td>5^b</td>
</tr>
<tr>
<td>Data collection</td>
<td>~120^c</td>
<td>~105^a</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Data processing</td>
<td>10</td>
<td>~180^a</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Total</td>
<td>~130</td>
<td>~315</td>
<td>16</td>
<td>18</td>
</tr>
</tbody>
</table>

^a These times are not reported in [61]. Times are estimated based on those reported in a similar experiment using the top down approach [66].

^b Setup only needs to be performed once for multiple trials.

^c This time is not reported in [61]. This number is estimated based on the experience of the author and the number of measurements.
bottom-up extraction of joint orientations.

Perhaps the most significant difference between 3DAM and other methods of joint orientation estimation is the amount of effort required. Stationary LiDAR often requires multiple scans from different viewpoints in order to avoid occlusions, which can introduce bias to the extracted joint sets [63]. Fig. 6.8b is the result of combining seven separate LiDAR measurements from three different locations. Each scan adds effort both in the field and when processing the data. Conversely, 3DAM requires substantially less effort; in fact, the actual data collection took only a few minutes in each experiment. This is the result of using scannerless LiDAR (capturing a point cloud takes a fraction of a second instead of several minutes) and the mobility of 3DAM (occlusions are eliminated by simply and easily maneuvering the device).

6.10.2 Ivy Lea

This section compares and discusses the results of four different joint orientation estimation experiments: (i) hand measurements using a compass and inclinometer (Fig. 6.9a); (ii) the 3DAM bottom-up approach using a stationary LiDAR (Fig. 6.9b); (iii) biased 3DAM (Fig. 6.9c); (iv) unbiased 3DAM (Fig. 6.9d). The joint orientations determined in (ii) are the point axes extracted from a single scan of the stationary LiDAR, using the same bottom-up approach employed in (iii) and (iv). A summary of the results is provided in Table 6.4.

All three LiDAR based methods produced joint set orientations similar to those resulting from the hand measurements. In fact, for Sets 1 and 2, the angular differences are less than five degrees. However, the angular differences relative to Set 3 are more significant. It is speculated that a major contribution to this discrepancy is the difficulty of accurately measuring the dip direction by hand on surfaces with small dips. On these surfaces, the measured dip direction becomes increasingly sensitive to minor variations of the roughness, geometry, and accessibility of the point of con-
Table 6.4: Results of the Ivy Lea experiments.

<table>
<thead>
<tr>
<th></th>
<th>Hand measurements</th>
<th>Stationary LiDAR</th>
<th>Biased 3DAM</th>
<th>Unbiased 3DAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of poles</td>
<td>70</td>
<td>18101</td>
<td>359</td>
<td>1229</td>
</tr>
<tr>
<td>Joint orientations</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(dip / dip direction)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Set 1</td>
<td>90/352</td>
<td>86/354</td>
<td>88/169</td>
<td>89/171</td>
</tr>
<tr>
<td>Set 2</td>
<td>73/069</td>
<td>72/066</td>
<td>75/065</td>
<td>76/066</td>
</tr>
<tr>
<td>Set 3</td>
<td>11/203</td>
<td>23/237</td>
<td>14/253</td>
<td>17/234</td>
</tr>
<tr>
<td>Angular difference with respect to hand measurements (deg)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Set 1</td>
<td>–</td>
<td>4.5</td>
<td>3.6</td>
<td>1.4</td>
</tr>
<tr>
<td>Set 2</td>
<td>–</td>
<td>3.0</td>
<td>4.3</td>
<td>4.2</td>
</tr>
<tr>
<td>Set 3</td>
<td>–</td>
<td>15.1</td>
<td>10.8</td>
<td>9.4</td>
</tr>
<tr>
<td>Approximate effort (min)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Setup / take down</td>
<td>0</td>
<td>10</td>
<td>5(^a)</td>
<td>5(^a)</td>
</tr>
<tr>
<td>Data collection</td>
<td>~85</td>
<td>40(^b)</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Data processing</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Total</td>
<td>95</td>
<td>60</td>
<td>17</td>
<td>19</td>
</tr>
</tbody>
</table>

\(^a\) Setup only needs to be performed once for multiple trials.  
\(^b\) This time depends on the desired resolution of the resulting point cloud. In this case, the point cloud was downsampled so data collection could have been performed faster.

tact. Note that the offset of the dip direction in all three LiDAR based methods is in the same direction.

It appears that the stationary LiDAR was well-oriented to capture all three joint sets. Due to the local terrain, the LiDAR stood slightly above the centre of the scanned area, allowing for a better line of sight of the nearly-horizontal Set 3. Note that no manual pruning of the point cloud was performed and points belonging to planar surfaces are discovered automatically.
Figure 6.9: Stereonets illustrating the results of the four different joint orientation estimation methods used in the Ivy Lea experiments.
Figure 6.10: Stereonets illustrating the results of the two different joint orientation estimation methods used in the Myra-Bellevue experiments.

6.10.3 Myra-Bellevue

The purpose of the Myra-Bellevue experiment is to demonstrate that joint orientation estimation with 3DAM can be performed by remotely operating a vehicle to collect the data. In this section, this approach (Fig. 6.10b) is compared with hand measurements using a compass and inclinometer (Fig. 6.10a). A summary of the results is provided in Table 6.5.

Compared to the results at Brewer Lake and Ivy Lea, the angular differences between the hand measured and 3DAM joint sets are slightly larger at Myra-Bellevue. One possible cause of this discrepancy is the path driven by the robot. Unlike the biased and unbiased 3DAM used in the previous experiments, the orientation of the LiDAR was relatively constant while the robot was driven in a straight line roughly parallel to the rock face. In particular, as pictured in Fig. 6.6, the dip of the LiDAR was fixed. It is speculated that performing 3DAM from a remotely operated robot could be
Table 6.5: Results of the Myra-Bellevue experiments.

<table>
<thead>
<tr>
<th></th>
<th>Hand measurements</th>
<th>3DAM using a mobile robot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of poles</td>
<td>25</td>
<td>234</td>
</tr>
<tr>
<td>Joint orientations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(dip / dip direction)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Set 1</td>
<td>87/237</td>
<td>87/065</td>
</tr>
<tr>
<td>Set 2</td>
<td>89/292</td>
<td>89/286</td>
</tr>
<tr>
<td>Angular difference with respect to hand measurements (deg)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Set 1</td>
<td>–</td>
<td>9.9</td>
</tr>
<tr>
<td>Set 2</td>
<td>–</td>
<td>6.0</td>
</tr>
<tr>
<td>Approximate effort (min)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Setup / take down</td>
<td>0</td>
<td>10&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>Data collection</td>
<td>45</td>
<td>1</td>
</tr>
<tr>
<td>Data processing</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Total</td>
<td>55</td>
<td>21</td>
</tr>
</tbody>
</table>

<sup>a</sup> Setup only needs to be performed once for multiple trials.
improved by ensuring there is a greater variety of viewpoints of the rock face. This could be achieved by actuating the LiDAR, deliberately maneuvering the robot relative to the scanned surfaces, or using a robot with more degrees of freedom in its motion (e.g., a flying robot).
Chapter 7

Conclusion

This thesis explores how a map representation comprised entirely of the orientations of flat surfaces can be accurately estimated and applied in useful real-world applications. Through the careful parameterization of axes (Chapter 2) and rotations (Chapter 3) that specifically addresses how they are used for state estimation, an axis mapping (Chapter 4) algorithm is developed that takes advantage of the non-injectivity of axes to produce an accurate axis map of the environment. Two major applications of axis maps are explored that exploit its representation and succinctness. The LiDAR compass (Chapter 5) uses a compact axis map to provide accurate heading estimation to a two-dimensional vehicle. As long as the entries of the axis map are frequently observed, heading error is bounded and excellent localization is possible when combined with translation measurements. Finally, joint orientation estimation (Chapter 6) is a direct application of three-dimensional axis mapping that offers significant improvements to the traditional methods of measuring joint sets. In this application, the map representation allows physically distinct discontinuities in the rock face with similar axes to be correlated, which can greatly improve the estimation of the joint sets.
7.1 Contributions

Axis mapping is a specialized implementation of simultaneous localization and mapping (SLAM) in the space of axes and orientations. The contributions of this thesis are centred around enabling axis maps to be correctly and efficiently used in SLAM, as well as the applications resulting from the advantages offered by the representation of the map. The primary contributions of this thesis are listed below.

- Unit axis and axis vector parameterizations of axes in two and three dimensions are developed. In particular, the notation and operations acting on axes are established and made consistent with rotations of the same dimensionality. Furthermore, the manifold encapsulation of axes is explicitly described, including geometric representations of the topological spaces involved.

- Novel axis mapping algorithms in two and three dimensions are developed, which explore how a map representation with a non-injective relationship with physical parts of the environment can be incorporated into the mapping process. The algorithms also demonstrate how the manifold encapsulation of axes can be incorporated into a state estimation algorithm.

- The LiDAR compass is introduced as a method of heading estimation. A localization algorithm that incorporates the LiDAR compass is developed, implemented, and tested in multiple environments. This contribution also includes a experimental demonstration of how two-dimensional axis maps can be created with both real and simulated sensor data in various challenging environments.

- A novel form of joint orientation estimation that uses a mobile platform for data collection is fully developed and tested. This major contribution bridges the fields of state estimation and robotics and geotechnical engineering by establishing and testing an algorithm to rapidly, accurately, safely, and inexpensively
generate stereonets. The advantages of axis mapping in this context are definitively shown to improve upon many aspects of the established techniques.

7.2 Future Work

The body of work presented in this thesis represents the paths of research that were well-developed, implemented, and tested. However, as is the case with most research and development, interesting alternate avenues of research were considered or even briefly explored. These ideas form possible directions of future research and are discussed in this section. Furthermore, the results of experiments often reveal possible areas of improvements to be considered in future iterations. Consequently, recommended future work is listed below.

- A comparison of the axis mapping results that employs all variations of the factor graphs illustrated in Figure 4.3 on page 48 (as well any other possible variations) could be performed. The importance of this comparison is that it allows for a better understanding of the sensor requirements needed to perform accurate axis mapping. This includes both the quality and possible omission of some sensors.

- The accuracy of the LiDAR compass for two-dimensional heading estimation permits the possibility of decoupling heading and position estimation, which makes the position estimation completely linear. By assuming the heading is known, batch estimation of the positions of the vehicle could be solved in closed form. A lightweight online implementation of SLAM which takes advantage of this decoupling would be an interesting avenue of research.

- A three-dimensional implementation of the LiDAR compass could be investigated. It was shown that the LiDAR compass enables simple and accurate heading estimation in two dimensions. It would be interesting to evaluate how well
the measurement of the axes of planar surfaces could be incorporated into an orientation estimation algorithm, especially when non-gravitational accelerations or electromagnetic interference impede the ability of an IMU to serve in this role.

- A common problem in joint orientation estimation is that joint sets whose orientations are nearly perpendicular to the direction of measurement tend to be measured more frequently than those that are nearly parallel. To accommodate for this discrepancy, the Terzaghi correction [90] can be applied to the data, which assigns a weighting factor to each measurement based on its orientation. However, because the mobility of axis mapping allows for the direction of measurement to easily vary, it would be interesting to instead apply a correction that is proportional to the probability of detecting specific joint sets in the LiDAR measurements. It is conceivable that detection statistics [91] could be used for this purpose.

- A new prototype for joint orientation estimation could be designed and tested. The prototype pictured in Figure 6.3 on page 121 is a proof-of-concept device whose design was not intended to be compact and user-friendly. It is conceivable that a compact, lightweight, and user-friendly iteration of the hardware could demonstrate that axis mapping for joint orientation could be used as a drop-in alternative to manual measurements.
Bibliography


Point Theorem,” *The American Mathematical Monthly*, vol. 85, no. 7, pp. 521–524,


for estimation of angular systems based on circular distributions,” in *Proc. of the

els for orientation estimation based on directional statistics,” in *Proc. of the 18th


[15] I. Y. Bar-Itzhack and Y. Oshman, “Attitude determination from vector observa-
tions: Quaternion estimation,” *IEEE Transactions of Aerospace and Electronic Sys-

[16] I. Y. Bar-Itzhack, J. Deutschmann, and F. L. Markley, “Quaternion normalization in
additive EKF for spacecraft attitude determination,” in *Proc. of the AIAA Guidance,


[62] M. Sturzenegger and D. Stead, “Quantifying discontinuity orientation and persistence on high mountain rock slopes and large landslides using terrestrial re-


Appendix A

Axioms of Manifold Operators

For all unit axes \( \mathbf{m}, \mathbf{n} \) and axis vectors \( \phi_1, \phi_2 \), Hertzberg et al. [6] specify that the four axioms outlined in (2.47) on page 27 to ensure that the \( \boxplus \) and \( \boxminus \) operators are locally homeomorphic to \( \mathbb{R}^n \). This appendix demonstrates that the definitions of \( \boxplus \) and \( \boxminus \) for two-dimensional and three-dimensional axes abide by these axioms.

**Proposition A.1:** For every unit axis \( \mathbf{m} \in S^1 \) and null axis vector \( 0 \in h^1 \), \( \mathbf{m} \boxplus 0 = \mathbf{m} \).

**Proof:** Using the definitions of \( \boxplus \) and the axis exponential, \( \mathbf{m}^+ \exp(0) = \mathbf{m}^+ 0 = \mathbf{m} \). ■

**Proposition A.2:** For every unit axis \( \mathbf{m} \in S^2 \) and null axis vector \( 0 \in h^2 \), \( \mathbf{m} \boxplus 0 = \mathbf{m} \).

**Proof:** The proof proceeds identically to the proof of proposition A.1. ■

**Proposition A.3:** For all unit axes \( \mathbf{m}, \mathbf{n} \in S^1 \), \( \mathbf{m} \boxplus (\mathbf{n} \boxminus \mathbf{m}) = \mathbf{n} \).

**Proof:** Using the definitions of \( \boxplus \) and the axis exponential and logarithm,

\[
\mathbf{m} \boxplus (\mathbf{n} \boxminus \mathbf{m}) = \mathbf{m}^+ \exp(\log(\mathbf{m}^{-} \mathbf{n})) \\
= \mathbf{m}^+ \mathbf{m}^{-} \mathbf{n} \\
= \mathbf{n}.
\]

(A.1)
**Proposition A.4:** For all unit axes $m, n \in S^2$, $m \equiv (n \equiv m) = n$.

**Proof:** The proof proceeds identically to the proof of proposition A.3.

**Proposition A.5:** For every unit axis $m \in S^1$ and axis vector $\phi \in h^1$, $(m \equiv \phi) \equiv m = \phi$.

**Proof:** Using the definitions of $\equiv$, $\equiv$, the axis exponential and the axis logarithm,

\[
(m \equiv \phi) \equiv m = \log((m^+ \exp(\phi))^*)
\]

\[
= \log(\exp(\phi))
\]

\[
= \phi.
\]

**Proposition A.6:** For every unit axis $m \in S^2$ and axis vector $\phi \in h^2$, $(m \equiv \phi) \equiv m = \phi$.

**Proof:** The proof proceeds identically to the proof of proposition A.5.

**Proposition A.7:** For every unit axis $m \in S^1$ and axis vectors $\phi_1, \phi_2 \in h^1$, $\|(m \equiv \phi_1) \equiv (m \equiv \phi_2)\| \leq \|\phi_1 - \phi_2\|$.

**Proof:** Using the definitions of $\equiv$, $\equiv$, the axis exponential and the axis logarithm, as well as Propositions 2.1 and 2.5,

\[
\|(m \equiv \phi_1) \equiv (m \equiv \phi_2)\| = \|\log((m^+ \exp(\phi_2))^* m^+ \exp(\phi_1))\|
\]

\[
= \|\log\left[\begin{bmatrix} \cos(\phi_m + \phi_2) \\ \sin(\phi_m + \phi_2) \end{bmatrix} - \begin{bmatrix} \cos(\phi_m + \phi_1) \\ \sin(\phi_m + \phi_1) \end{bmatrix}\right]\|
\]

\[
= \|\log\left[\begin{bmatrix} \cos(\phi_2 - \phi_1) \\ \sin(\phi_2 - \phi_1) \end{bmatrix}\right]\|
\]

\[
= \|\phi_2 - \phi_1\|
\]

where $\phi_m$ is the angle of $m$. ■
Proposition A.8: For every unit axis \( m \in S^2 \) and axis vectors \( \phi_1, \phi_2 \in h^2 \), \( \| (m \uplus \phi_1) \ominus (m \uplus \phi_2) \| \leq \| \phi_1 - \phi_2 \| \).

Proposition A.8 states that the spherical distance between \( m \uplus \phi_1 \) and \( m \uplus \phi_2 \) is less than or equal to the distance between \( \phi_1 \) and \( \phi_2 \) on the plane tangent to the sphere at \( m \). A proof demonstrating that Proposition A.8 is true is detailed in [6, Lemma 11].
Appendix B

Processing Two-Dimensional Sensor Measurements

The measurements required for two-dimensional axis mapping and LiDAR compass localization are described generically in Chapters 4 and 5, respectively. This appendix describes how the raw measurements of various specific sensors are converted into these measurements.

B.1 Rotary Encoders

A rotary encoder is a electro-mechanical device that transforms the absolute or relative (incremental) angular position of a rotating shaft to a digital or analog signal. Installing rotary encoders on the driveshaft and/or steering column of a vehicle provides a way to estimate the motion of the vehicle. This section outlines how to estimate the motion of a skid-steer and tricycle vehicle.

B.1.1 Skid-Steer

On a skid-steer vehicle, incremental encoders can be installed on the the right and left wheels to estimate the motion of the vehicle. A encoder measurement is the signed number of ticks recorded since the previous measurement, which is a measurement of the rotation of the wheel. Suppose at time step \( k \) that \( n_r, n_l \in \mathbb{Z} \) are the number of ticks measured by the right and left encoders since time step \( k-1 \), respectively. Then
the translation $\Delta d_k \in \mathbb{R}$ and rotation $\Delta q_k \in S^1$ of the vehicle expressed in the vehicle coordinate frame $\mathcal{F}_v$ during this time step is

$$
\Delta d_k = \frac{n_r + n_l}{2b}, \quad \Delta q_k = \exp \left( \frac{n_r - n_l}{2bw} \right),
$$

(B.1)

where $b \in \mathbb{Z}$ is the number of ticks per unit distance of travel (determined via calibration), and $w \in \mathbb{R}$ is the track width of the vehicle (i.e., the distance between the right and left wheels). The covariance matrix $Q_k$ of the measurement is

$$
Q_k = \begin{bmatrix}
\alpha_1 \Delta d_k^2 + \alpha_2 \log(\Delta q_k)^2 & 0 \\
0 & \alpha_3 \Delta d_k^2 + \alpha_4 \log(\Delta q_k)^2
\end{bmatrix},
$$

(B.2)

where $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \in \mathbb{R}$ are weights that are determined empirically. This approach of calculating $Q_k$ is based on a similar approach described by Thrun et al. [47, p. 127].

### B.1.2 Tricycle

The motion of a tricycle vehicle can be estimated by installing an incremental encoder on the driveshaft and an absolute encoder on the steering column. An illustration of the variables describing a tricycle vehicle is shown in Figure B.1. Suppose at time step $k$, $n \in \mathbb{Z}$ is the number of ticks measured by the incremental encoder since time step $k - 1$, and $\psi \in \mathbb{R}$ is the steering input angle as measured by the absolute encoder. Then the translation $\Delta d_k \in \mathbb{R}$ and rotation $\Delta q_k \in S^1$ of the vehicle expressed in the vehicle coordinate frame $\mathcal{F}_v$ during this time step is

$$
\Delta d_k = \frac{n}{b}, \quad \Delta q_k = \exp \left( \frac{\Delta d_k \cos(\alpha) \tan(\psi)}{l} \right),
$$

(B.3)

where $b \in \mathbb{Z}$ is the number of ticks per unit distance travel of the incremental encoder (determined via calibration), $\alpha$ is the inclination of the steering column, and $l$ is the wheelbase of the vehicle. The covariance matrix $Q_k$ of the measurement is given by (B.2), where $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \in \mathbb{R}$ are determined empirically for each specific vehicle.
B.2 Gyroscopes

A gyroscope measures the rate of rotation in the coordinate frame of the body to which it is fixed. In two-dimensions, a gyroscope fixed to a vehicle is commonly used to measure its angular velocity. Numerically integrating the gyroscope measurements provides a method to estimate the rotation of the vehicle. At time step $k$, let $\omega$ be a gyroscope measurement. Under the assumption that the angular velocity of the vehicle is constant for the duration of a time step, the rotation $\Delta q_k \in S^1$ of the vehicle expressed in the vehicle coordinate frame $\mathcal{F}_v$ during this time step is

$$\Delta q_k = \exp(\omega \Delta t),$$  \hspace{1cm} (B.4)

where $\Delta t$ is the duration of the time step. When a gyroscope is available, (B.4) can simply replace the calculation of $\Delta q_k$ in (B.1) or (B.3). Depending on a number of factors (e.g., the quality of the sensors, the accuracy of the vehicle models, the operating environment), using a gyroscope instead of encoders to estimate the rotation of the vehicle often improves accuracy. In this scenario, the covariance matrix $Q_k$ of the motion when using a gyroscope is still be estimated by (B.2), but the weights $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \in \mathbb{R}$ must be redetermined.
B.2.1 Calibration

Gyroscope measurements have several sources of error, such as bias, scale-factors, output nonlinearities, dead zones, quantization error, and temperature effects. Detailed descriptions of these errors are available in [92]. In particular, biases in gyroscope measurements can be significant and usually warrant some measure of mitigation. In estimation problems, a common solution to dealing with bias is to include it in the estimated state. However, because the rate of change of the bias tends to be small, it is often sufficient to predetermine the bias and simply remove it from each measurement. This is achieved by calculating the mean output of the gyroscope while it is stationary for a couple minutes prior to an experiment.

B.3 LiDAR

A measurement of a two-dimensional scanning LiDAR contains an ordered list of bearings and the measured range at each bearing. Both axis mapping and LiDAR compass localization require measurements of the axes of flat surfaces in the environment. A LiDAR measurement can be transformed into a set of random axes by using the method illustrated in Figure B.2 and enumerated below.

(i) Outliers are filtered from the LiDAR measurement by removing points with large range differences compared to neighbouring points.

(ii) The axis of each individual point is calculated by fitting a line to its \( n \) nearest neighbours (typically \( n = 6 \) to \( 12 \)) using a simple Deming regression [93], which is a linear regression that accounts for uncertainty in both variables (Figure B.2b).

(iii) Points with highly uncertain axes (e.g., points near corners) are discarded.

(iv) The axes of all the remaining points are clustered using DBSCAN [48], which clus-
ters only dense concentrations of similarly-oriented axes of points, and ignores outliers and sparse axes (Figure B.2c).

(v) The axial mean \([1]\) and variance of each cluster are calculated, and clusters with a variance below a threshold are the \(p\) random axes \(\{z_1, \sigma^2_{d\phi_1}\}, \ldots, \{z_p, \sigma^2_{d\phi_p}\}\) (Figure B.2d).
Figure B.2: Transforming a two-dimensional LiDAR measurement into random axes. (a) A LiDAR measurement consists of an ordered list of bearings $\beta \in S^1$ and their respective ranges $r \in \mathbb{R}$ measured in $\mathcal{F}_v$. (b) A Deming regression is used to fit a line to the neighbours of each point. The axis of the resulting line is taken as the axis at that point. (c) DBSCAN is used to cluster points with similar axes, ignoring outliers and sparse axes. (d) The axial mean and variance of the clusters form the random axes $\{z_1, \sigma^2_{\delta \phi_1}\}$ and $\{z_2, \sigma^2_{\delta \phi_2}\}$. The width of wedges are proportional to the variances of the random axes.
Appendix C

Data Association for LiDAR Compass Localization

The motion correction steps of the LCL algorithm require associating axis measurements with entries of an axis map. Specifically, given \( p \geq 0 \) axis measurements \( \{z_1, \sigma^2_{\phi_1}\}, \ldots, \{z_p, \sigma^2_{\phi_p}\} \) expressed in the vehicle coordinate frame \( \mathcal{F}_v \), and an \( m \)-entry axis map \( A = (m_1, \ldots, m_m) \) expressed in the global coordinate frame \( \mathcal{F}_g \), data association determines which (if any) axis measurements are measuring which entries of the axis map. LCL achieves this by finding the largest jointly compatible injective subset of associations among the measurements and map entries.

C.1 Compatibility

An axis measurement is said to be compatible with an axis map entry if the Mahalanobis distance between them is below a threshold. The Mahalanobis distance is a statistical distance in units of standard deviations between two random variables drawn from the same distribution. If the expected axis measurement is considered the mean of this distribution, then the Mahalanobis distance is the number of standard deviations between the expected \( \hat{z} \in S^1 \) and actual \( z \in S^1 \) axis measurements.

The expected measurement \( \hat{z} \) is a prediction of what should be measured given the entries of the LCL state \( x = (p, q, A) \). Therefore, the expected measurement \( \hat{z}_j \) of the \( j \)-th entry of \( A \) is simply calculated by rotating it to \( \mathcal{F}_v \), which is the coordinate
frame in which axis measurements are made, i.e.,

\[
\hat{z}_j = q^+ m_j. \tag{C.1}
\]

The Mahalanobis distance \( d \in \mathbb{R} \) between the expected and actual axis measurement is then

\[
d = \sqrt{\frac{(z \ominus \hat{z}_j)^2}{\sigma^2_{\delta \phi}}}. \tag{C.2}
\]

The squared Mahalanobis distance \( d^2 \) forms a chi-squared distribution. This allows for a statistically meaningful Mahalanobis distance threshold to be selected for determining whether an axis measurement and axis map entry are compatible. In the case where the distributions have a single degree of freedom (e.g., the two-dimensional axes in this scenario), a squared Mahalanobis distance of \( d^2 \leq 3.84 \) indicates with 95% probability that the two axes originate from the same distribution.

### C.2 Joint Compatibility

When there are multiple axis measurements and multiple axis map entries, the joint compatibility is determined between subsets of the measurements and map entries. The method used to produce the axis measurements (described in Section B.3 in Appendix B), allows for the reasonable assumption that each axis map entry is measured at most once by a single LiDAR measurement. Therefore, instead of considering each axis measurement individually, all axis measurements from a single LiDAR measurement are considered simultaneously, where the goal is to determine the largest jointly compatible subset between a subset of the axis measurements and axis map entries.

Suppose a LiDAR measurement generates two axis measurements \( \{z_1, \sigma^2_{\delta \phi_1}\} \), and \( \{z_2, \sigma^2_{\delta \phi_2}\} \), and there are three expected measurements \( \hat{z}_1, \hat{z}_2, \) and \( \hat{z}_3 \) generated by the three entries in an axis map. Then there are 13 possible association scenarios among the actual and expected measurements (including scenarios where measurements are
Table C.1: Possible associations among two axis measurements and three axis map entries.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Expected Measurement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>- - -</td>
</tr>
<tr>
<td>2</td>
<td>(z_1) - -</td>
</tr>
<tr>
<td>3</td>
<td>- (z_1) -</td>
</tr>
<tr>
<td>4</td>
<td>- - (z_1)</td>
</tr>
<tr>
<td>5</td>
<td>(z_2) - -</td>
</tr>
<tr>
<td>6</td>
<td>- (z_2) -</td>
</tr>
<tr>
<td>7</td>
<td>- - (z_2)</td>
</tr>
<tr>
<td>8</td>
<td>(z_1) (z_2) -</td>
</tr>
<tr>
<td>9</td>
<td>(z_1) - (z_2)</td>
</tr>
<tr>
<td>10</td>
<td>(z_2) (z_1) -</td>
</tr>
<tr>
<td>11</td>
<td>(z_2) - (z_1)</td>
</tr>
<tr>
<td>12</td>
<td>- (z_1) (z_2)</td>
</tr>
<tr>
<td>13</td>
<td>- (z_2) (z_1)</td>
</tr>
</tbody>
</table>

unassigned), which are listed in Table C.1. For each of these scenarios, the joint Mahalanobis distance can be calculated to determine if the scenario is jointly compatible. For example, the joint Mahalanobis distance \(d \in \mathbb{R}\) of Scenario 13 in Table C.1 is

\[
d = \sqrt{\left[ \begin{array}{c} z_1 \boxplus \hat{z}_3 \\ z_2 \boxplus \hat{z}_2 \end{array} \right]^\top \text{diag}(\sigma_{\delta_1}^2, \sigma_{\delta_2}^2)^{-1} \left[ \begin{array}{c} z_1 \boxplus \hat{z}_3 \\ z_2 \boxplus \hat{z}_2 \end{array} \right]}.
\]  

(C.3)

Because the distribution now has two degrees of freedom, a squared joint Mahalanobis distance of \(d^2 < 5.99\) indicates with 95% probability that the associations originate from the same distribution. Using this threshold, the scenario is said to contain jointly compatible associations when \(d^2 < 5.99\).

The goal of data association is to determine the largest number of jointly compatible associations. Note that the number of possible scenarios grows exponentially with the number of axis measurements and axis map entries. However, LCL tends to operate in environments where the number of scenarios makes a brute force...
search computationally tractable. Regardless, a minor optimization is applied by first checking scenarios with the largest number of associations (i.e., scenarios 8–13 in Table C.1) and not checking scenarios with fewer associations if a jointly compatible scenario is found. There are more elegant algorithms that efficiently eliminate scenarios (e.g., joint compatibility branch and bound [94], which performs a depth-first search through a tree of scenarios); however, these were deemed unnecessary for the low-dimensionality of LCL data association.
Appendix D

Processing Three-Dimensional Sensor Measurements

The measurements required for three-dimensional axis mapping are described generically in Chapter 4 and specifically for joint orientation estimation in Chapter 6. This appendix describes how the raw measurements of various specific sensors are converted into these measurements. This includes calibration and transformation of the measurements into the specific topological spaces required by three-dimensional axis mapping.

D.1 Gyroscope

Assuming it is aligned with the coordinate frame of the mobile platform, a three-axis gyroscope measures the rate of rotation along each of the principal axes, expressed in the mobile platform coordinate frame. Let $\omega \in \mathbb{R}^3$ be the gyroscope measurement in $\mathcal{F}_m$, which is modelled as

$$\omega = \bar{\omega} + b + \delta \omega, \quad \delta \omega \sim \mathcal{N}(0, Q),$$  \hspace{1cm} (D.1)$$

where the true angular velocity $\bar{\omega}$ is offset by a bias $b$ and is corrupted by zero-mean Gaussian noise $\delta \omega$ with covariance matrix $Q$. As mentioned in Section B.2.1, the most prominent source of measurement error for gyroscopes is the bias $b \in \mathbb{R}^3$. However, because the duration of a typical joint orientation estimation experiment is relatively
short (i.e., \( \approx 1-10 \) minutes) compared to the rate of change of the bias, \( b \) is simply estimated before an experiment by calculating the mean output of the gyroscope while it is stationary for a couple minutes. As a result, the estimated unbiased gyroscope measurement is \( \tilde{\omega} = \omega - b \), where

\[
\tilde{\omega} = \tilde{\omega} + \delta \omega, \quad \delta \omega \sim N(0, Q).
\]  

(D.2)

D.2 Accelerometer

The gravitational acceleration experienced by the mobile platform has a known direction in \( \mathcal{F}_g \) (i.e., down). Therefore, its direction measured by the accelerometer in \( \mathcal{F}_m \) provides information about the rotation from \( \mathcal{F}_g \) to \( \mathcal{F}_m \), which is the orientation of the mobile platform by definition. To take advantage of its known direction, it is assumed that only gravitational acceleration is experienced by the mobile platform. Because the violation of this assumption is likely, the heuristics described in Section D.2.1 are used to detect the presence of non-gravitational acceleration and increase the measurement uncertainty if it is detected.

An accelerometer measurement \( a \in \mathbb{R}^3 \) is modelled as

\[
a = \tilde{a} + \delta a, \quad \delta a \sim N(0, R^{(a)}),
\]  

(D.3)

where the true acceleration \( \tilde{a} \in \mathbb{R}^3 \) is corrupted by zero-mean Gaussian noise \( \delta a \in \mathbb{R}^3 \) with covariance matrix \( R^{(a)} \in \mathbb{R}^{3 \times 3} \). Assuming only gravitational acceleration is experienced by the mobile platform, \( \tilde{a} \approx [0 \ 0 \ -9.81 \text{ m/s}^2]^T \) when it is expressed in \( \mathcal{F}_g \). To only measure the direction of gravitational acceleration, each accelerometer measurement is transformed into a three-dimensional unit direction \( \tilde{a} \in S^2 \). Unit directions and unit axes share the properties described in Section 2.3 because they belong to the same topological space, with the exception that \( d \neq -d \) for a unit direction \( d \in S^2 \). However, because the manifold encapsulation of axes described in
Section 2.4.2 enforces the property \( \mathbf{m} \equiv -\mathbf{m} \) for a unit axis \( \mathbf{m} \in S^2 \), a concise description of the changes needed for the manifold encapsulation of three-dimensional directions is provided in Appendix E.

Let \( f : \mathbb{R}^3 \to S^2 \), where

\[
f(\mathbf{v}) = \frac{\mathbf{v}}{\|\mathbf{v}\|},
\]

transforms the vector \( \mathbf{v} \in \mathbb{R}^3 \) to a unit direction through normalization. Because \( f \) is nonlinear, the accelerometer measurements are mapped to unit directions by using an unscented transformation \[84\] through \( f \). First, a set of sigma points \( \alpha_1, \ldots, \alpha_6 \in \mathbb{R}^3 \) are sampled from the measurement; i.e.,

\[
\alpha_i = \begin{cases} 
\mathbf{a} + \left( \sqrt{3\mathbf{R}^{(a)}} \right)_i & \text{for } i = 1, 2, 3 \\
\mathbf{a} - \left( \sqrt{3\mathbf{R}^{(a)}} \right)_{i-3} & \text{for } i = 4, 5, 6,
\end{cases}
\]

where \( (\mathbf{A})_i \) is the \( i \)-th column of the matrix \( \mathbf{A} \). Each of these sigma points is then passed through \( f \); i.e.,

\[
\tilde{\alpha}_i = f(\alpha_i), \quad \text{for } i = 1, \ldots, 6,
\]

which transforms each sigma point into a unit direction \( \tilde{\alpha}_i \in S^2 \). The unit direction \( \tilde{\mathbf{a}} \in S^2 \) of the accelerometer measurement is the normalized mean of the transformed sigma points; i.e.,

\[
\tilde{\mathbf{a}} = f\left( \frac{1}{6} \sum_{i=1}^{6} \tilde{\alpha}_i \right).
\]

The covariance matrix \( \tilde{\mathbf{R}}^{(a)} \in \mathbb{R}^{2 \times 2} \) of \( \tilde{\mathbf{a}} \) is then

\[
\tilde{\mathbf{R}}^{(a)} = \frac{1}{6} \sum_{i=1}^{6} (\tilde{\alpha}_i \ominus \tilde{\mathbf{a}}) (\tilde{\alpha}_i \ominus \tilde{\mathbf{a}})^\top,
\]

where the \( \ominus \) operator maps the difference between two unit directions to its unconstrained parameterization (i.e., it is analogous to the \( \sqcup \) operator, only for directions instead of axes). This operator is formally defined in Appendix E.
Figure D.1: Acceleration measured by the accelerometer. (a) In the absence of external acceleration, the magnitude of measured $a$ and gravitational $g$ acceleration are the same. (b) The magnitude of the measured acceleration $a$ is different than the magnitude of gravitational acceleration $g$ due to the external acceleration $a_{ext}$. (c) Despite the presence of the external acceleration $a_{ext}$, the magnitude of the measured $a$ and gravitational $g$ accelerations are approximately the same. However, because $a_{ext}$ is not usually maintained over several measurements, the jerk in this scenario tends to be nonzero.

D.2.1 Detecting Non-Gravitational Acceleration

Although it is assumed, it is unlikely that only gravitational acceleration is experienced by the mobile platform at all times. To compensate for non-gravitational acceleration, heuristics are used to detect its presence and proportionally increase the uncertainty of the measurement (i.e., $R^{(a)}$). More specifically, the uncertainty is increased if the measured magnitude of acceleration differs from gravitational acceleration (herein called magnitude error), and if there is a nonzero measured rate of change of the magnitude of acceleration (herein called nonzero jerk). These scenarios are illustrated in Figure D.1. The update to the covariance matrix of the accelerometer measurement is then

$$R^{(a)} \leftarrow R^{(a)} + \left( \alpha (||a|| - ||g||)^2 + \beta j^2 \right) I_3,$$

where $a$ is the measured acceleration, $||g|| \approx 9.81 \text{ m/s}^2$ is the magnitude of gravitational acceleration, and $j \in \mathbb{R}$ is the jerk. The positive scaling factors $\alpha$ and $\beta$ control how much uncertainty the magnitude error and jerk add to the accelerometer measurements.
Estimating the Jerk

The jerk \( j \) is estimated by examining the rate of change of the magnitude of the measured accelerations; i.e.,

\[
j = \frac{d \|a\|}{dt},
\]

which is calculated numerically. To calculate the jerk at time step \( k \), a line is fit to \( n \) neighbours before and after \( k \) (e.g., for \( n = 4 \), this would be \( k - 2, k - 1, k + 1, \) and \( k + 2 \)). The slope of this line is taken as an estimate of the jerk at time \( k \). This is illustrated in Fig. D.2.

One issue with estimating the jerk numerically is that noise (especially outliers) can significantly affect the slope calculation. To help reduce the effects of noise, the measurements are first smoothed using a mean filter. For each measurement magnitude \( \|a\|_k \), its corresponding smoothed point \( \|a_{\text{smooth}}\|_k \) is calculated by averaging the value of its \( m \) neighbours. The smoothed values are then used to calculate the jerk.

D.3 Magnetometer

Like gravitational acceleration, the Earth’s magnetic field surrounding the mobile platform has a known direction in \( \mathcal{F}_g \) (i.e., magnetic north). Therefore, its direction measured by the magnetometer in \( \mathcal{F}_m \) provides information about the rotation from \( \mathcal{F}_g \) to \( \mathcal{F}_m \), which is the orientation of the mobile platform by definition. To take ad-
vantage of its known direction, it is assumed that only the Earth’s magnetic field is measured by the magnetometer. The calibration procedure described in Section D.3.1 is performed prior to experiments to compensate for other magnetic fields (e.g., from the other components of the mobile platform).

A magnetometer measurement \( m \in \mathbb{R}^3 \) is modelled as

\[
m = \bar{m} + \delta m, \quad \delta m \sim \mathcal{N}\left(0, R^{(m)}\right),
\]

where the true magnetic field vector \( \bar{m} \in \mathbb{R}^3 \) is corrupted by zero-mean Gaussian noise \( \delta m \in \mathbb{R}^3 \) with covariance matrix \( R^{(m)} \in \mathbb{R}^{3 \times 3} \). Assuming only the Earth’s magnetic field is measured by the magnetometer, the value of \( \bar{m} \) is known when expressed in \( \mathbb{F}_g \). To only measure the direction of the Earth’s magnetic field, each magnetometer measured is transformed into a three-dimensional unit direction \( \hat{m} \in \mathbb{S}^2 \).

To transform the magnetometer measurement into a unit direction, the procedure outlined in Section D.2 (i.e., an unscented transformation) is applied to the sensor data. That is,

\[
m_i = \begin{cases} m + \left(\sqrt{3}R^{(m)}\right)_i & \text{for } i = 1, 2, 3 \\ m - \left(\sqrt{3}R^{(m)}\right)_{i=3} & \text{for } i = 4, 5, 6 \end{cases}
\]

\[
\hat{m}_i = f(m_i), \quad \text{for } i = 1, \ldots, 6
\]

\[
\hat{m} = f\left(\frac{1}{6} \sum_{i=1}^{6} \hat{m}_i\right)
\]

\[
\hat{R}^{(m)} = \frac{1}{6} \sum_{i=1}^{6} (\hat{m}_i \otimes \hat{m}) (\hat{m}_i \otimes \hat{m})^\top,
\]

where \( f \) is the normalization function defined in (D.4), \( m_i \in \mathbb{R}^3, \hat{m}_i \in \mathbb{S}^2, \hat{m} \in \mathbb{S}^2 \), and \( \hat{R}^{(m)} \in \mathbb{R}^{2 \times 2} \).
D.3.1 Calibration

It is assumed that only the Earth’s magnetic field is measured by the magnetometer. As a result, compensation must be made for other sources of magnetic fields (e.g., electronics and ferrous materials onboard the mobile platform). A calibration routine is performed prior to experiments to detect and compensate for these extraneous magnetic field sources. Off-the-shelf software is used to perform this calibration, but its underlying algorithm is rather simple.

A vector representing the magnitude and direction of the Earth’s magnetic field at a particular time and location can be obtained from various databases given the latitude, longitude, and date. Although this vector changes over time, its change over the duration of an experiment is negligible. As a result, it is reasonable to assume that the direction and magnitude of the Earth’s magnetic field vector is constant and known in the global coordinate frame for the duration of an experiment. As the mobile platform is rotated, the magnetometer measurement of the Earth’s magnetic field traces out a sphere in the mobile platform coordinate frame. Constant disturbances to the magnetic field distort the sphere into an ellipsoid, and a transformation from the ellipsoid to the ideal disturbance-free sphere is applied to the raw data. This transformation is calculated by recording magnetometer data while moving the mobile platform through a series of rotations.

D.4 LiDAR

A three-dimensional LiDAR measurement is a set of points in $\mathbb{R}^3$ called a point cloud. For ToF cameras, each point corresponds to a depth measurement at one pixel of a captured image that has been transformed to 3D through application of the intrinsic parameters of the sensor. The point cloud is transformed into a set of three-dimensional random axes by using what is essentially a three-dimensional version
of the algorithm described in Section B.3 in Appendix B, which transforms a two-
dimensional LiDAR measurement into a set of two-dimensional random axes. The
three-dimensional version of the algorithm is illustrated in Figure D.3 and is enumer-
ated below.

(i) If the mean distance between a point and its \( n \) neighbours is large compared to
the average mean distance between all points and their respective \( n \) neighbours,
then the point is considered an outlier and is removed.

(ii) The axis of each individual point is calculated by using a robust normal esti-
mation algorithm [95]. This algorithm estimates the axis of a point by fitting a
plane to a subset of its neighbours meeting minimum consistency and orthogo-
nal distance criteria.

(iii) A term proportional to the curvature of the surface being measured by a point is
estimated, and points determined to be measuring curved surfaces are removed.

(iv) The axes of all the remaining points are clustered using DBSCAN [48], which clus-
ters only dense concentrations of similarly oriented axes of points, and ignores
outliers and sparsely populated clusters.

(v) The axial mean [86] and covariance matrix of each cluster is calculated, produc-
ing \( p \) random axes \( \{z_1, R^{(\delta\phi_1)}\}, \ldots, \{z_p, R^{(\delta\phi_p)}\} \).

It is important to note that the steps enumerated above contain tuneable param-
eters that must be adjusted based on various environmental factors. For example,
the average size of the planar surfaces in the environment influences the number of
neighbours \( n \) to use in step (i), and the roughness/waviness of the planar surfaces
influences the curvature threshold in step (iii). Failure to properly consider these
factors could result in poor axis extraction.
Figure D.3: Transforming a three-dimensional LiDAR measurement into random axes. (a) After removing outliers, the axis at each point is estimated. (b) Points measuring curved surfaces are removed, leaving only axes of points measuring planar surfaces (shown here in $h^2$). (c) DBSCAN is used to cluster similar axes, ignoring outliers. (d) The axial mean and variance of the clusters form the random axes $\{z_1, R^{(\delta \phi_1)}\}, \ldots, \{z_p, R^{(\delta \phi_p)}\}$. 
Appendix E

Manifold Encapsulation of Three-Dimensional Directions

A unit direction $d \in S^2$ shares the properties of unit axes described in Section 2.3, with the exception that $d \neq -d$. Because the operators defined for the manifold encapsulation of three-dimensional axes outlined in Section 2.4.2 specifically maintain the property $m \equiv -m$ for a unit axis $m \in S^2$, different operators are required for unit directions. In particular, these are

$$\Theta : S^2 \times S^2 \to S^2$$
$$\Theta : S^2 \times S^2 \to s^2,$$

where for $\beta = (\beta_1, \beta_2) \in \mathbb{R}^2$,

$$s^2 := \{ \beta : \| \beta \| < \pi \} \cup \{ (\pi, 0) \}$$

is an open disk of radius $\pi$ centred at the origin in union with a single point at $(\pi,0)$, which is illustrated in Figure E.1. Like $h^2$ for unit axes, $s^2$ is required to maintain the injectivity of $\Theta$, which ensures that the axioms outlined in Section 2.4.3 are not violated. For unit directions $c, d \in S^2$ and $\beta \in s^2$, the operators are defined as follows:

$$d = c \oplus \beta := c^+ \exp(\beta)$$
Figure E.1: A geometric representation of $s^2$ is an open disk of radius $\pi$ centred at the origin in union with a single point at $(\pi,0)$. This disk is the domain of $\beta = (\beta_1, \beta_2) \in \mathbb{R}^2$ in (E.3) and (E.4).

$$\beta = c \ominus d := \begin{cases} \log(d^{-1}c) & \text{for } c^\top d > -1 \\ (\pi,0) & \text{otherwise.} \end{cases}$$ \hfill (E.4)

Proofs that these definitions abide by the axioms outlined in Section 2.4.3 proceed similarly to those provided by Hertzberg et al. [6].