Linköping Studies in Science and Technology
Licentiate Thesis No. 2003

On Indoor Localization Using Magnetic Field-Aided Inertial Navigation Systems

Chuan Huang

LINKÖPING
UNIVERSITY

Linköping studies in science and technology. Licentiate Thesis
No. 2003

On Indoor Localization Using Magnetic Field-Aided Inertial Navigation Systems

Chuan Huang

LINKÖPING
UNIVERSITY

This is a Swedish Licentiate's Thesis.

Swedish postgraduate education leads to a Doctor's degree and/or a Licentiate's degree.
A Doctor's Degree comprises 240 ECTS credits (4 years of full-time studies).
A Licentiate's degree comprises 120 ECTS credits,
of which at least 60 ECTS credits constitute a Licentiate's thesis.

Linköping studies in science and technology. Licentiate Thesis
No. 2003

On Indoor Localization Using Magnetic Field-Aided Inertial Navigation Systems

Chuan Huang

chuan.huang@liu.se
www.control.isy.liu.se
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping
Sweden

ISBN 978-91-8075-783-6 (print)
ISBN 978-91-8075-784-3 (PDF)
ISSN 0280-7971

Printed by LiU-Tryck, Linköping, Sweden 2024

To my mother who loves and cares me the most!

Abstract

Localization and navigation technologies have become integral to modern society, playing crucial roles in daily life. They enable efficient and safe travel, allow emergency services to reach and assist individuals quickly, and are indispensable components of autonomous systems. Indoor localization technology, aimed at enabling precise location determination in indoor environments, has garnered significant research interest. One intriguing research direction is magnetic field-based localization technology, which exploits spatial variations in indoor magnetic fields to provide position information.

This thesis investigates how indoor magnetic fields can be used for localization and develops a magnetic field-aided localization system that does not rely on any preinstalled infrastructures, such as electric coils, or external localization information. To achieve this, a sensor platform consisting of a planar magnetometer array and an *inertial measurement unit* (IMU) was built. The array captures the spatial variations of the magnetic field, from which odometry information can be inferred. This odometry information is then used to aid an *inertial navigation system* (INS) constructed around the IMU on the array.

The thesis addresses three key challenges faced when realizing a magnetic field-based INS using the developed sensor platform. The first challenge is the calibration of the sensors to ensure their measurements are accurate enough for the localization system. The second challenge is to create a magnetic field model that can be used to realize a magnetic field-aided INS. The final challenge is to design a state estimation algorithm that provides consistent estimates so that the perceived uncertainties match the true estimation errors as closely as possible.

To address the first challenge, an easy-to-use and efficient calibration method is proposed to correct the misalignment of the IMU's and magnetometer's sensitivity axes, sensor biases, and scale factors. The second challenge is met by proposing a polynomial magnetic field model to construct a local small-scale magnetic field map and a tightly integrated magnetic field-aided INS. The proposed system was evaluated on simulation and real-world datasets, demonstrating a significant reduction in position drift compared to a stand-alone INS and showing performance comparable to state-of-the-art magnetic field odometry. Additionally, the system offers flexibility in sensor configurations, including sensor placement and the number of sensors involved. Finally, an observability-constrained magnetic field-aided INS is proposed to address the inconsistencies identified in the developed magnetic field-aided INS. This new system maintains the yaw angle unobservable, and demonstrates improved performance and consistency compared to the initial system.

The results show that the proposed magnetic field-aided INS can be realized by low-cost sensors and appropriate signal-processing algorithms. It could be integrated into magnetic field *simultaneous localization and mapping* (SLAM) systems to extend their exploration phase. Most importantly, it showcases the possibility of building self-contained, accurate, and consistent indoor localization systems with magnetic fields.

Populärvetenskaplig sammanfattning

I en värld där tekniken ständigt utvecklas, är exakt lokalisering och navigering avgörande för många av dem system som vi idag tar som självskrivna. Dessa tekniker är hjärtat i system som autonoma fordon, drönare och personliga navigationsenheter. Från att ha använt jordens övergripande magnetfält och enkla magnetiska kompasser för att hitta rätt, använder vi nu avancerade sensorer och signalbehandlingstekniker för att utnyttja lokala variationer i jordens magnetfält för lokalisering och navigering.

Denna avhandling tar sig an tre utmaningar kopplade till inomhus lokalisering och navigering med hjälp av variationer i jordens magnetfält. Först undersöks hur man kan modellera det magnetiska fältet för att skapa ett magnetfälts stöttat tröghetsnavigeringssystem. Sedan undersöks hur informations fusionen i det stöttade tröghetsnavigeringssystemet ska designas för att osäkerheten måtten som produceras från navigeringssystemet ska vara konsistenta med storleken hos det faktiska felet. Detta är mycket viktigt om beslut ska fattas utifrån navigeringslösningen, så som styrandet av autonoma systems. Slutligen undersöks en metod för att kalibrerasensorerna i systemet. Detta är nödvändigt för att kunna bygga magnetfälts stöttat tröghetsnavigeringssystem med hjälp av lågkostnads sensorer och skapa för kommersiellt bruk.

Sammanfattningsvis så bidrar den presenterade forskningen till utvecklingen av viktiga förmågor och insikter som behövs nästa generations inomhusnavigeringssystem. En typ av navigeringssystems som kan revolutionera hur vi interagerar med vår omvärld och möjliggöra nya typer av robotar och autonoma system.

Acknowledgments

First of all, I would like to thank my supervisor Assoc. Prof. Isaac Skog, and co-supervisor Assoc. Prof. Gustaf Hendeby. You have provided me with interesting research topics and ideas to work on and helped me along the way. Your expertise and kind help are invaluable to me, and I would not make it this far without them.

Thanks to Assoc. Prof. Martin Enqvist, head of the division. You are an open-minded, responsible manager who maintains a friendly, open workspace. Also, thanks to Ninna Stensgård for taking care of all practicalities. Your work makes studying and working in our division very convenient and pleasant.

Thank you to all my colleagues for the friendly environment, interesting social activities, and research discussions. I especially want to thank Yuxuan Xia, Jakob Åslund, Daniel Bossér, and Reza Jafari for proofreading the thesis. I appreciate Peng Liu, Jian Zhou, and Jianan Bai for picking me up at the train station when I arrived at Linköping for the first time. I also appreciate the help and advice from Anton Kullberg, and Daniel Bossér about Swedish culture, life tips, and so on.

Thank you to all the teachers and teaching assistants in the courses I studied. Your hard work and knowledge greatly benefited me and inspired me to do good research.

Thank you to all my friends. Lunch break, badminton, swimming, and the Spring Festival dinner are my most enjoyable activities.

This work is supported by the Security Link and the Swedish Research Council (Vetenskapsrådet). Thank you for financing my pursuit of a PhD.

Lastly, I'd like to thank my parents and my girlfriend Ziwei for their support and love.

Linköping, August 2024
Chuan Huang

Contents

I Overview

II Localization using IMU and Magnetometer Array

Notation

Some sets

Notation	Meaning
$\mathbb{R}$	Real numbers
$\mathbb{R}^n$	Column vectors of dimension n
$\mathrm{SO}(3)$	3D orientation matrix group
S^2	Sphere
S^3	Unit quaternion group
$\mathbb{H}_p$	Pure Quaternion group
$\mathcal{M}$	Manifold
$\mathbf{T}_x\mathcal{M}$	Tangent space of $\mathcal{M}$ at x

Operators

Notation	Meaning
$\arg\min$	Minimizing argument
$\cdot^{\top}$	Matrix transpose
$\cdot^{-1}$	Matrix inverse
$\cdot^{-\top}$	Matrix transposed inverse
$\cdot^{\dagger}$	Moore–Penrose inverse
blkdiag	Block diagonal matrix
Cov	Covariance
$\|\cdot\|_{\Sigma}$	Weighted norm
$\otimes$	Quaternion multiplication
conj	Quaternion conjugation
vec	Vectorize operator
tr	Trace
$\otimes_K$	Kronecker product
$\nabla\times$	Curl
$\nabla\cdot$	Divergence
∇	Gradient
Exp	Capitalized exponential map
Log	Capitalized logarithmic map
$[\,\cdot\,]^{\wedge}$	Mapping a vector to an element in the tangent space
$[\,\cdot\,]^{\vee}$	Mapping an element in the tangent space to a vector
$\oplus$	Addition on $\mathcal{M}$
$\ominus$	Subtraction on $\mathcal{M}$

Acronyms

Acronym	Description
ANEES	average normalized estimation error squared
EKF	extended Kalman filter
ESKF	error-state Kalman filter
GPS	global positioning system
IMU	inertial measurement unit
INS	inertial navigation system
KF	Kalman filter
MAINS	magnetic field-aided inertial navigation system
OC-MAINS	observability-constrained magnetic field-aided inertial navigation system
RMSE	root mean square error
SLAM	simultaneous localization and mapping

Part I

Overview

1

Introduction

1.1 Background and Motivation

Navigation and localization are important techniques in both civilian and military applications. For example, when one opens Google Maps to find the route to a booked hotel, the software first tries to locate the user's current position and then provides navigation information. Another example is when submarines are cruising in deep water, they rely on accurate navigation and localization techniques to navigate safely and reach their destinations.

Despite the significant advancements in navigation and localization technology, many challenges remain. Take one of the most widely used localization and navigation systems, the *global positioning system* (GPS) as an example. First, the environment in which GPS operates is increasingly complex. Tall buildings and narrow streets can obstruct GPS signals, leading to inaccurate localization results. Second, indoor, underwater, and underground environments may render GPS useless since the signals are typically weak or unavailable. Third, the GPS signal can be easily jammed or spoofed, which poses a security risk to the navigation system.

The thesis focuses on developing robust and high-performing magnetic field-based localization techniques to complement GPS for indoor environments. The indoor magnetic field offers several appealing properties, making it a strong candidate for localization. First, because the magnetic field obeys Maxwell's equations, it can be easily modeled using well-established physical principles. Second, the magnetic field is generally stable, as it comprises the Earth's relatively constant magnetic field and disturbances from ferromagnetic materials and man-made sources, such as current-carrying coils. These disturbances are often static or weak enough to be negligible. Third, unlike visual-based localization techniques that rely on image capture, sensing the indoor magnetic field does not

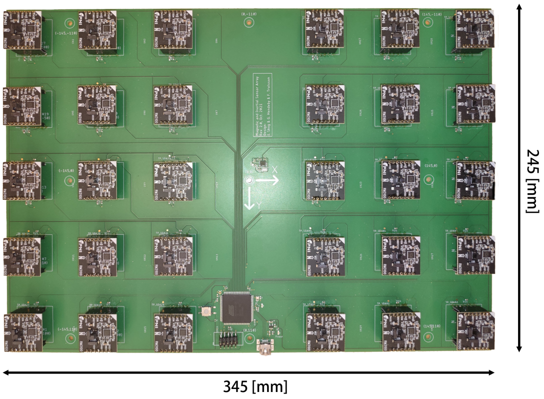


Figure 1.1: *The sensor board used in the thesis. It has 30 PNI RM3100 magnetometers and an Osmium MIMU 4844 IMU mounted on the bottom side.*

compromise privacy. Lastly, the variations in indoor magnetic fields are rich enough to provide useful position information, making magnetic field-based localization techniques well-suited for indoor applications.

Inspired by visual odometry techniques, which sense the environment through thousands of pixels, the magnetic field environment can be recorded similarly. To achieve this, a magnetometer planar array consisting of 30 magnetometers and one IMU, see in Figure 1.1, was built to capture the spatially varying magnetic field. The localization system based on the array is designed to be self-contained and is expected to outperform a stand-alone INS. Additionally, this system has the potential to be integrated into existing magnetic field-based SLAM systems, enhancing their robustness and usability.

1.2 Research Questions

There are several research questions relating to magnetic field-based localization given the magnetic sensor arrays, such as the one in Figure 1.1. Foremost among these is how indoor magnetic fields can be effectively leveraged to enhance localization accuracy and support advanced system requirements. Addressing this pivotal inquiry entails exploring the following sub-questions.

To begin with, sensor calibration is a prerequisite for high-performing multi-sensor systems, such as the magnetic field-aided INS. The system relies on several low-cost sensors. They generally have reduced precision due to sensor biases and scaling factors. Furthermore, there is a certain degree of misalignment between the axes of the magnetometers and the IMU. To account for these imperfections, calibration of these sensor errors is necessary to maintain a good system performance. One should also consider calibration speed for commercial applications, given the necessity to calibrate thousands of products. Which calibration method can be accurate and efficient?

Next, given that the sensors are calibrated to the desired accuracy, how can the magnetometer array measurements be used to aid the INS? One way is to model magnetic fields using prior physics knowledge. The model can then be fitted to the measurements and predict nearby magnetic fields. Then the question boils down to what model is suitable for modeling the magnetic field and how the model can be integrated into INS thereby reducing inherent positional drift.

Lastly, localization systems are often part of large systems, such as advanced driver-assistance systems, missile guidance systems, etc. For these types of systems, not only accurate localization results are needed, but also consistent uncertainties, since they affect high-level decision-making, The question of maintaining consistent uncertainty estimates in the magnetic field-aided INS will be investigated.

1.3 Contributions

The key contributions of the thesis are as follows:

1. An efficient sensor calibration method for sensor arrays consisting of multiple magnetometers and an IMU was proposed. It allows the measurements from these low-cost sensors to be processed by the algorithm used in the proposed magnetic field-aided INS.

2. A magnetic field-aided INS was developed and realized with a magnetometer array and one IMU. The system has such small localization errors that it opens up the possibility of building magnetic field SLAM systems using the presented system.

3. An observability-constrained magnetic field-aided INS was proposed to solve the original system's inconsistent perceived uncertainty in yaw. Compared to the original system the perceived uncertainty in yaw reported by the new system is more consistent with the true estimates error. The consistent perceived uncertainty is crucial if the observability-constrained magnetic field-aided INS is to be incorporated into larger localization or control systems.

1.4 Publications

A: A Tightly-Integrated Magnetic-Field aided Inertial Navigation System.

> Chuan Huang, Gustaf Hendeby, and Isaac Skog. A tightly-integrated magnetic-field aided inertial navigation system. In *Proc. 2022 25th Int. Conf. on Information Fusion (FUSION)*, pages 1–8, Linköping, Sweden, July 2022. doi: 10.23919/FUSION49751.2022.9841304.

In paper A, a tightly integrated magnetic field-aided inertial navigation system is presented. The system uses a magnetometer sensor array to measure spatial variations in the local magnetic field. The variations in the field are — via a recursively updated polynomial magnetic-field model — mapped into displacement and orientation changes of the array, which in turn are used to aid the inertial navigation system. Simulation results show that the resulting navigation system has three orders of magnitude lower position error at the end of a 40-second trajectory as compared to a standalone inertial navigation system.

In paper A, Isaac Skog and Gustaf Hendeby had the research idea and provided help with improving the manuscript. Chuan Huang implemented the idea, carried out the simulation experiments, and wrote the paper.

B: MAINS: A Magnetic-Field-Aided Inertial Navigation System for Indoor Positioning.

> Chuan Huang, Gustaf Hendeby, Hassen Fourati, Christophe Prieur, and Isaac Skog. MAINS: A magnetic-field-aided inertial navigation system for indoor positioning. *IEEE Sensors Journal*, 24(9):15156–15166, 2024. doi: 10.1109/JSEN.2024.3379932.

In paper B, the proposed method in paper A was evaluated on real-world datasets for the first time. Experiments show that the magnetic field-aided INS significantly outperforms the stand-alone INS, demonstrating a remarkable two orders of magnitude reduction in position error. Furthermore, when compared to the state-of-the-art magnetic field-aided navigation approach, the proposed method exhibits slightly improved horizontal position accuracy. The experimental results show that the position error after 2 minutes of navigation in most cases is less than 3 meters when using an array of 30 magnetometers. Thus, the proposed navigation solution has the potential to solve one of the key challenges faced by the current magnetic-field SLAM solutions — the very limited allowable length of the exploration phase during which unvisited areas are mapped.

In paper B, Isaac Skog and Gustaf Hendeby helped collect data and improved the manuscript and filter tuning. Hassen Fourati and Christophe Prieur provided the code of the algorithm used for comparison and helped improve the manuscript. Chuan Huang participated in the data collection, conducted experiments, and wrote the paper.

C: An Observability-Constrained Magnetic Field-Aided Inertial Navigation System.

> Chuan Huang, Gustaf Hendeby, and Isaac Skog. An observability-constrained magnetic field-aided inertial navigation system — extended version. *arXiv preprint arXiv:2406.02161*, abs/2406.02161, 2024. (Accepted to IPIN 2024)

In paper C, the inconsistent perceived yaw uncertainty in the magnetic field-aided INS was identified, and a method to construct an observability-constrained magnetic field-aided INS is proposed. The proposed method builds upon the previously proposed observability-constrained extended Kalman filter and extends it to work with a magnetic field-based odometry-aided INS. The proposed method is evaluated using simulation and real-world data, showing that (i) the system observability properties are preserved, (ii) the estimation accuracy increases, and (iii) the perceived uncertainty calculated by the EKF is more consistent with the true uncertainty of the filter estimates.

In paper C, Isaac Skog and Gustaf Hendeby helped with problem formulation, provided research discussion, and improved the manuscript. Chuan Huang had the idea, implemented it, and wrote the paper.

1.5 Thesis Outline

This thesis consists of two parts. Part I contains the background material of the applications presented in Part II.

Part I discusses the motivation behind the magnetic field-based localization technology and provides relevant background knowledge, including coordinate frame definitions, sensor models, optimization, and state estimation.

Part II presents the magnetic field-aided INS and its variant, the observability-constrained magnetic field-aided INS, along with the sensor calibration required to ensure optimal performance. Chapter 3 is about calibrating the magnetometer array used in developing the magnetic field-aided INS. Chapter 4 presents the proposed magnetic field-aided INS and its performance. Chapter 5 introduces its variant, the observability-constrained magnetic field-aided INS, and compares its performance with that of the magnetic field-aided INS. Finally, Chapter 6 concludes the thesis and gives possible directions for future work.

2 Background

This chapter offers essential background knowledge for understanding the methods developed in the thesis. Additionally, the notations introduced here will be used in subsequent chapters.

2.1 Coordinate Frames

To present the localization problem clearly, it is important to define the coordinate frames to express the physical quantities properly.

First, the body frame (b-frame) is an orthogonal coordinate frame that is aligned with the roll, pitch, and yaw axes of the vehicle (sensor board). Second, the navigation frame (n-frame) is a local geographic frame with its origin attached to a fixed point on Earth. Note the Earth's rotation rate is approximately $7.29 \cdot 10^{-5}$ rad/s [4] and its effects cannot be observed by the low-cost sensors used in this work, so it is neglected. Therefore, the navigation frame and inertial frame are also approximated to be the same. The relation of the coordinate frames can be seen in Figure 2.1.

2.2 Orientation Representations

The relative rotation between different coordinate frames can be defined by one of the four common representations: the orientation matrix, the rotation vector, Euler angles, and the unit quaternion. In the following definitions, all coordinate frames share the same origin, i.e., the relative translation is pre-compensated.

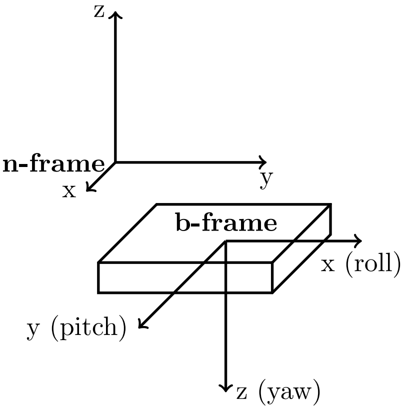


Figure 2.1: The relation of the body frame and navigation frame. The body frame's origin is attached to a fixed point on the sensor board and moves along with it, while the navigation frame remains static, with its origin fixed at a point on Earth.

2.2.1 Orientation Matrix

The orientation matrix is a 3 × 3 matrix whose columns contain the coordinates of the unit vectors of the rotated frame in the reference frame. For example, if the reference frame is the n-frame and the rotated frame is the b-frame, then the orientation matrix of the b-frame is denoted by $R_{\mathrm{b}}^{\mathrm{n}}$ and its columns are the unit vectors of the b-frame's x-axis, y-axis, and z-axis expressed in the n-frame. Therefore, $R_{\mathrm{b}}^{\mathrm{n}}$ can also be used to transform the coordinates of a vector from the b-frame to the n-frame, i.e., $x^{\mathrm{n}} = R_{\mathrm{b}}^{\mathrm{n}} x^{\mathrm{b}}$, where $x \in \mathbb{R}^3$ and the superscript indicates the coordinate frame in which the vector is expressed. Further, all orientation matrices in 3-D space make up the special orthogonal group $\mathrm{SO}(3) \triangleq \{R \in \mathbb{R}^{3\times 3} | R^\top R = I, \det(R) = 1\}$.

2.2.2 Rotation Vector

The rotation vector is a three-dimensional vector representing both the direction and magnitude of the rotation. Its direction corresponds to the axis of rotation, while its magnitude represents the angle by which the reference frame must rotate to align with the rotated frame, see Figure 2.2.

2.2.3 Euler Angles

The Euler angles representation is a triplet whose elements are the angles by which the reference frame needs to rotate around each of its three axes consecutively to align with the rotated frame. A common choice to execute the rotations is to rotate around the z-axis, followed by the y-axis and x-axis, see Figure 2.3. One drawback of this representation is that gimbal lock issues exist, i.e., when two of the three axes of rotation of a 3-D object align, the rotations cannot be executed independently, resulting in a loss of one degree of freedom.

2.2.4 Unit Quaternion

The unit quaternion is a four-element unit-norm vector for representing orientations. Compared to the Euler angles representation, it does not have the gimbal lock issue [5]. An overview of quaternions can be found in [6]. Below are some fundamental concepts that are essential for developing this thesis.

A quaternion can be conveniently denoted by $q \triangleq [q_w \ q_v^\top]^\top$, where q_w is a scalar, known as the real part, and $q_v = [q_x \ q_y \ q_z]^\top$ is a 3-dimensional vector, known as the imaginary part. The unit quaternions encode rotation vectors and the conversion is straightforward. Note that a unit-norm quaternion q can always be written as

$$q = \begin{bmatrix} \cos(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2})\breve{v} \end{bmatrix}, \tag{2.1}$$

where θ is a scalar and $\breve{v} \in \mathbb{R}^3$ is a unit-norm vector, and the corresponding rotation vector is $\theta\breve{v}$. Thus, the identity quaternion, denoted by $q_I = [1 \ 0 \ 0 \ 0]^\top$, represents no rotation.

Quaternion conjugation is defined as

$$\text{conj}(q) \triangleq \text{conj}\left(\begin{bmatrix} q_w \\ q_v \end{bmatrix}\right) = \begin{bmatrix} q_w \\ -q_v \end{bmatrix}. \tag{2.2}$$

The conjugated unit quaternion represents an opposite rotation as q represents, which can be seen by negating θ in (2.1).

The quaternion product of two unit quaternions q^1 and q^2 is defined as

$$q^1 \otimes q^2 = \begin{bmatrix} q_w^1 q_w^2 - (q_v^1)^\top q_v^2 \\ q_w^1 q_v^2 + q_w^2 q_v^1 + q_v^1 \times q_v^2 \end{bmatrix}, \tag{2.3}$$

where × denotes the vector cross product. The product implies composite rotations: the rotation represented by q^2 followed by that represented by q^1.

All unit quaternions constitute the hypersphere $\mathrm{S}^3 = \{q \mid q \in \mathbb{R}^4, \|q\| = 1\}$. All pure quaternions, i.e., the ones whose real part is 0, constitute a set

$$\mathbb{H}_p = \{q \mid q_w = 0\}. \tag{2.4}$$

2.3 Smooth Manifolds

This section provides a brief introduction to smooth manifolds, covering the essential knowledge needed to understand optimization on manifolds. This understanding is crucial for the developed sensor calibration method. For more details on smooth manifolds, interested readers may refer to [7, 8].

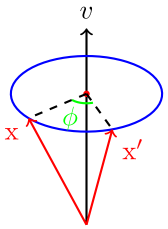


__Figure 2.2:__ The rotation vector v. It has a length ϕ and its direction is the rotation direction. The vector x rotates around the vector v by ϕ degrees to x'.

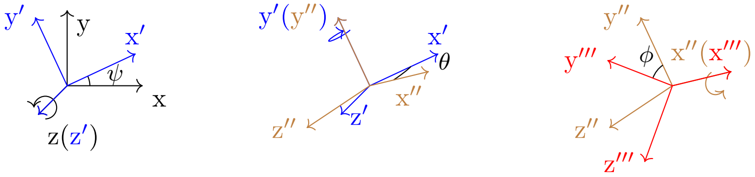


__Figure 2.3:__ The Euler angles (ψ, θ, ϕ) following the z-y'-x'' convention. The original coordinate frame (black) rotates around its z-axis, y'-axis, and x''-axis sequentially to arrive at the final rotated coordinate frame (red).

Roughly speaking, a smooth manifold $\mathcal{M}$, in most cases, can be imagined as a smooth surface with curvature, e.g., a 3-dimensional sphere as shown in Example 2.1. For each point $x \in \mathcal{M}$, the neighborhood of x resembles a Euclidean space. There exists a tangent space at $x \in \mathcal{M}$, denoted by $\mathrm{T}_x\mathcal{M}$. The tangent space is a Euclidean space, and its dimension coincides with the dimension of the manifold, i.e., $\dim(\mathcal{M}) = \dim(\mathrm{T}_x\mathcal{M})$. For the manifolds discussed in the thesis, each one forms a group under certain binary operations [7]. For the identity element denoted by I in the group, the vectors in the tangent space $\mathrm{T}_I\mathcal{M}$ can be mapped to the manifold $\mathcal{M}$ via the exponential map *exp* and mapped back via the logarithm map *log*; these two maps will be defined later. Furthermore, the increment to an element and the difference between elements in $\mathcal{M}$ can be represented by a vector in $\mathbb{R}^{\dim(\mathcal{M})}$, thanks to the two operators *oplus* and *ominus* to be defined for each type of smooth manifolds. With these two operators, defining derivatives properly for the manifold $\mathcal{M}$ is convenient.

Example 2.1

The 3-dimensional sphere $\mathrm{S}^2 = \{x \mid x \in \mathbb{R}^3, \|x\| = 1\}$ is a smooth manifold, see Figure 2.4. The tangent space at x is a tangent plane $\mathrm{T}_x\mathrm{S}^2 = \{v \mid x^\top v = 0, v \in \mathbb{R}^3\}$. The dimension of the manifold is 2.

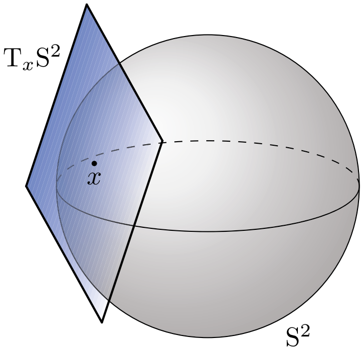


Figure 2.4: *The manifold S^2 and its tangent space. The manifold S^2 is a sphere in 3-dimensional space, and its tangent space T_xS^2, a subspace of $\mathbb{R}^3$, contains all vectors that are perpendicular to x. T_xS^2 can be viewed as the tangent plane to the sphere passing through x, therefore, the dimension of S^2 and T_xS^2 are both 2.*

2.3.1 Vector Manifold

The n-dimensional vector space $\mathbb{R}^n$ is a trivial manifold. The tangent space at $x \in \mathbb{R}^n$ is $\mathbb{R}^n$ itself. The exponential map *exp* and the logarithmic map *log* are identity maps. Further, the operators *oplus* and *ominus* correspond to the normal addition and subtraction, respectively. That is, for $x, y, v \in \mathbb{R}^n$,

$$\oplus : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n; \quad x \oplus v = x + v, \tag{2.5a}$$

$$\ominus : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n; \quad x \ominus y = x - y. \tag{2.5b}$$

2.3.2 Orientaiton Matrix Manifold

The special orthogonal group SO(3) is a 3-dimensional smooth manifold with a tangent space associated with each element in the group [8]. Specifically, for the tangent space associated with the identity matrix I, also known as $\mathfrak{so}(3)$, there are linear maps $\wedge$ and $\vee$ which map from $\mathbb{R}^3$ to $\mathfrak{so}(3)$ and vice versa. They are defined as [7]

$$\wedge : \mathbb{R}^3 \to \mathfrak{so}(3); \quad \omega \mapsto [\omega]^\wedge = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}^\wedge = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}, \tag{2.6a}$$

$$\vee : \mathfrak{so}(3) \to \mathbb{R}^3; \quad [\omega]^\wedge \mapsto \left[[\omega]^\wedge\right]^\vee = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}^\vee = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}. \tag{2.6b}$$

The element in $\mathfrak{so}(3)$ and SO(3) can be mapped to each other via the *exp* and *log* mappings as follows [7]

$$\exp_R : \mathfrak{so}(3) \rightarrow \mathrm{SO}(3);$$

$$[\omega]^\wedge \mapsto \exp_R([\omega]^\wedge) = \begin{cases} I + \frac{\sin(\|\omega\|)}{\|\omega\|}[\omega]^\wedge + \frac{1-\cos(\|\omega\|)}{\|\omega\|^2}([\omega]^\wedge)^2, & \text{if } \|\omega\| \neq 0 \\ I, & \text{otherwise} \end{cases} \tag{2.7a}$$

$$\log_R : \mathrm{SO}(3) \rightarrow \mathfrak{so}(3);$$

$$R \mapsto \log_R(R) = \begin{cases} \frac{\varphi \cdot (R-R^\top)}{2\sin(\varphi)} \text{ with } \varphi = \cos^{-1}\left(\frac{\mathrm{tr}(R)-1}{2}\right), & \text{if } \sin(\varphi) \neq 0 \\ [0\ 0\ 0]^\top, & \text{otherwise} \end{cases} \tag{2.7b}$$

Here $\|\cdot\|$ denotes the norm of a vector and $\mathrm{tr}(\cdot)$ denotes the trace of a matrix.

The capitalized exponential and logarithmic map can be defined to map vectors from $\mathbb{R}^3$ to group elements SO(3) and in the reverse direction. That is,

$$\mathrm{Exp}_R : \mathbb{R}^3 \rightarrow \mathrm{SO}(3); \quad \omega \mapsto \exp_R([\omega]^\wedge), \tag{2.8a}$$

$$\mathrm{Log}_R : \mathrm{SO}(3) \rightarrow \mathbb{R}^3; \quad R \mapsto [\log_R(R)]^\vee. \tag{2.8b}$$

Due to the non-commutativity of matrix multiplication, there are two possible ways to define the *oplus* and *ominus* operators. One way is to define them as

$$\oplus : \mathrm{SO}(3) \times \mathbb{R}^3 \rightarrow \mathrm{SO}(3); \quad R \oplus v = R \cdot \mathrm{Exp}_R(v), \tag{2.9a}$$

$$\ominus : \mathrm{SO}(3) \times \mathrm{SO}(3) \rightarrow \mathbb{R}^3; \quad R \ominus S = \mathrm{Log}_R(S^{-1} \cdot R), \tag{2.9b}$$

for elements $R, S \in \mathrm{SO}(3)$ and elements $v \in \mathbb{R}^3$. Another way is to define them as

$$\oplus : \mathrm{SO}(3) \times \mathbb{R}^3 \rightarrow \mathrm{SO}(3); \quad R \oplus v = \mathrm{Exp}_R(v) \cdot R, \tag{2.10a}$$

$$\ominus : \mathrm{SO}(3) \times \mathrm{SO}(3) \rightarrow \mathbb{R}^3; \quad R \ominus S = \mathrm{Log}_R(R \cdot S^{-1}), \tag{2.10b}$$

for elements $R, S \in \mathrm{SO}(3)$ and elements $v \in \mathbb{R}^3$.

2.3.3 Unit Quaternion Manifold

The unit quaternion manifold is a 3-dimensional manifold. For the tangent space associated with the identity quaternion q_I, there are linear maps $\wedge$ and $\vee$ which map from $\mathbb{R}^3$ to $\mathbb{H}_p$ and vice versa through [7]

$$\wedge : \mathbb{R}^3 \rightarrow \mathbb{H}_p; \quad \omega \mapsto [\omega]^\wedge = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}^\wedge = \begin{bmatrix} 0 \\ \frac{\omega_1}{2} \\ \frac{\omega_2}{2} \\ \frac{\omega_3}{2} \end{bmatrix}, \tag{2.11a}$$

$$\vee : \mathbb{H}_p \rightarrow \mathbb{R}^3; \quad [\omega]^\wedge \mapsto \left[[\omega]^\wedge\right]^\vee = \begin{bmatrix} 0 \\ \frac{\omega_1}{2} \\ \frac{\omega_2}{2} \\ \frac{\omega_3}{2} \end{bmatrix}^\vee = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}. \tag{2.11b}$$

The element in $\mathbb{H}_p$ and S^3 can be mapped to each other via mappings $\exp_q$ and $\log_q$ [7],

$$\begin{aligned}
&\exp_q : \mathbb{H}_p \to \mathrm{S}^3;\\
&[\omega]^\wedge \mapsto \exp_q([\omega]^\wedge) = \begin{cases} \begin{bmatrix} \cos(\|\frac{\omega}{2}\|) \\ \frac{\omega}{\|\omega\|}\sin(\|\frac{\omega}{2}\|) \end{bmatrix}, & \text{if } \|\omega\| \neq 0 \\ [1\ 0\ 0\ 0]^\top, & \text{otherwise} \end{cases} && \text{(2.12a)}\\
&\log_q : \mathrm{S}^3 \to \mathbb{H}_p;\\
&q \mapsto \log_q\left(\begin{bmatrix} q_w \\ q_v \end{bmatrix}\right) = \begin{cases} \begin{bmatrix} 0 \\ \frac{q_v}{\|q_v\|}\theta \end{bmatrix} \text{ with } \theta = \operatorname{arctan2}(\|q_v\|, q_w), & \text{if } \|q_v\| \neq 0 \\ [0\ 0\ 0\ 0]^\top, & \text{otherwise} \end{cases} && \text{(2.12b)}
\end{aligned}$$

Here the function $\operatorname{arctan2}(y, x)$ returns the angle θ between the positive x-axis and the point (x, y), taking into account the signs of both arguments to determine the correct quadrant.

The capitalized exponential and logarithmic map can be defined to map vectors from $\mathbb{R}^3$ to group elements S^3 and in the reverse direction,

$$\mathrm{Exp}_q : \mathbb{R}^3 \to \mathrm{S}^3 \quad ; \quad \omega \mapsto \exp_q([\omega]^\wedge), \tag{2.13a}$$

$$\mathrm{Log}_q : \mathrm{S}^3 \to \mathbb{R}^3 \quad ; \quad q \mapsto [\log_q(q)]^\vee. \tag{2.13b}$$

For the same reason as in the orientation matrix manifold, there are also two ways to define $\oplus$ and $\ominus$. For simplicity, here provides one of them, i.e.,

$$\oplus : \mathrm{S}^3 \times \mathbb{R}^3 \to \mathrm{S}^3; \quad q \oplus v = q \otimes \mathrm{Exp}_q(v), \tag{2.14a}$$

$$\ominus : \mathrm{S}^3 \times \mathrm{S}^3 \to \mathbb{R}^3; \quad q \ominus p = \mathrm{Log}_q(\mathrm{conj}(q) \otimes p), \tag{2.14b}$$

for elements $q, p \in \mathrm{S}^3$ and elements $v \in \mathbb{R}^3$.

2.3.4 Product Smooth Manifolds

Product smooth manifolds, abbreviated as product manifolds, are Cartesian products of smooth manifolds. Let $\mathcal{M} = \mathcal{M}_1 \times \mathcal{M}_2 \cdots \mathcal{M}_M$ be a product manifold. Its dimension is the sum of the component manifolds' dimension. The tangent space $\mathrm{T}_x\mathcal{M}$ at $x \in \mathcal{M}$ is the Cartesian product of the tangent space of the component manifolds, therefore, the vectors in the tangent space can be uniquely identified by $\delta x \in \mathbb{R}^{\dim(\mathcal{M}_1)} \times \mathbb{R}^{\dim(\mathcal{M}_2)} \ldots \mathbb{R}^{\dim(\mathcal{M}_M)}$. Furthermore, $\oplus$ and $\ominus$ can be defined for product manifolds by viewing them as applying operators *oplus* and *ominus* already defined on each component manifold and concatenating the results.

In localization and navigation applications, the state space is often a product manifold, e.g., the Cartesian product of several vector manifolds and quaternion manifolds. Example 2.2 shows such manifold.

Example 2.2

Consider the dead-reckoning application. The state vector x_k can be written as

$$x_k = \begin{bmatrix} p_k \\ v_k \\ q_k \end{bmatrix}, \tag{2.15}$$

where $p_k \in \mathbb{R}^3$, $v_k \in \mathbb{R}^3$, and $q_k \in \mathrm{S}^3$ denote the position, velocity, and orientation, respectively.

The state space is a smooth manifold embedded in $\mathbb{R}^{10}$. However, the manifold is only 9-dimensional because it is a product manifold $\mathcal{M} = \mathbb{R}^3 \times \mathbb{R}^3 \times \mathrm{S}^3$.

The operator $\oplus : \mathcal{M} \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \to \mathcal{M}$ is defined as

$$x_k \oplus \delta x_k = \begin{bmatrix} p_k + \delta p_k \\ v_k + \delta v_k \\ q_k \otimes \mathrm{Exp}_q(\delta q_k) \end{bmatrix}, \tag{2.16}$$

and the operator $\ominus : \mathcal{M} \times \mathcal{M} \to \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3$ is defined as

$$x'_k \ominus x_k = \begin{bmatrix} p'_k - p_k \\ v'_k - v_k \\ \mathrm{Log}_q(\mathrm{conj}(q'_k) \otimes q_k) \end{bmatrix}. \tag{2.17}$$

Here $\delta x_k = [\delta p_k^\top \ \delta v_k^\top \ \delta q_k^\top]^\top$ and $x'_k = [p_k'^\top \ v_k'^\top \ q_k'^\top]^\top$.

2.3.5 Derivatives on Manifolds

For function $f : \mathcal{M} \to \mathcal{N}$, where $\mathcal{M}$ and $\mathcal{N}$ are smooth manifolds, the directional derivative $\nabla_{e_i} f(x)$ can be defined as

$$\nabla_{e_i} f(x) \triangleq \lim_{h \to 0} \frac{f(x \oplus h e_i) \ominus f(x)}{h}, \tag{2.18}$$

where e_i is the i-th vector of the natural basis of $\mathbb{R}^{\dim(\mathcal{M})}$, $\oplus$ is the *oplus* operator defined on $\mathcal{M}$, and $\ominus$ is the *ominus* operator defined on $\mathcal{N}$. Furthermore, the Jacobian matrix $J(x) \in \mathbb{R}^{\dim(\mathcal{M}) \times \dim(\mathcal{N})}$, i.e., the derivative of f with respect to (w.r.t.) x, is defined as

$$J(x) = \frac{\partial f(x)}{\partial x} \triangleq [\nabla_{e_1} f(x) \quad \nabla_{e_2} f(x) \quad \cdots \quad \nabla_{e_{\dim(M)}} f(x)]. \tag{2.19}$$

It should be noted that the derivative depends on the chosen $\oplus$ and $\ominus$ operators. Example 2.3 shows how derivative w.r.t. the elements in SO(3) can be computed using one possible choice of operators.

Example 2.3

Consider the function $f : \mathrm{SO}(3) \to \mathbb{R}^3$; $f(R) = Rp$, where $R \in \mathrm{SO}(3)$ and $p \in \mathbb{R}^3$. Using the definition of $\oplus$ in (2.9a) and $\ominus$ in (2.5b), the directional derivative can be computed as

$$\begin{aligned} \nabla_{e_i} f(R) &= \lim_{h\to 0} \frac{f(R \oplus he_i) \ominus f(R)}{h} = \lim_{h\to 0} \frac{(R \cdot \mathrm{Exp}_R(he_i) - R)\, p}{h} \\ &= \lim_{h\to 0} \frac{R\,(I + [he_i]^\wedge - I)\, p}{h} = \lim_{h\to 0} \frac{Rh[e_i]^\wedge p}{h} \\ &= \lim_{h\to 0} \frac{-Rh[p]^\wedge e_i}{h} = -R[p]^\wedge e_i, \end{aligned} \tag{2.20}$$

where the properties $\mathrm{Exp}_R(v) \approx I + [v]^\wedge, \forall v \in \mathbb{R}^3$ and $[a]^\wedge b = -[b]^\wedge a, \forall a, b \in \mathbb{R}^3$ are used. Therefore,

$$\begin{aligned} J(x) = \frac{\partial f(R)}{\partial R} &= \begin{bmatrix} -R[p]^\wedge e_1 & -R[p]^\wedge e_2 & -R[p]^\wedge e_3 \end{bmatrix} \\ &= -R[p]^\wedge \begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix} = -R[p]^\wedge \in \mathbb{R}^{3\times 3}. \end{aligned} \tag{2.21}$$

2.4 Sensors

The sensors used in this thesis are an IMU and magnetometers, see Figure 1.1. Inertial sensors are used to measure an object's acceleration and angular velocity, which are integrated to estimate the position and orientation of the object. The magnetometers are used to measure the magnetic field, which is used to correct the drift in the estimated position and orientation.

In this section, a generic sensor measurement model is introduced, followed by specific models for various sensor types.

2.4.1 Generic Sensor Measurement Model

For an ideal sensor that outputs 3-dimensional measurements, its measurements relate to the sensor's states and inputs as

$$y = h(x, u), \tag{2.22}$$

where $y \in \mathbb{R}^3$ denotes the ideal sensor measurement, x denotes the sensor's state, u denotes the input, and $h(\cdot, \cdot)$ denotes the measurement function that depends on specific types of sensors. Due to manufacturing imperfections and the stochastic nature of the measurement, there will always be a deviation between the actual sensor measurement and the ideal measurement. To capture such deviation, a stochastic linear measurement model

$$\tilde{y} = \tilde{h}(x, u; \theta) + e = Dy + o + e, \tag{2.23}$$

can be used for all types of sensors mentioned. Here, $\tilde{y} \in \mathbb{R}^3$ denotes the actual sensor measurement. In addition, $\theta = \{D, o\}$ denotes the calibration parameters, where $D \in \mathbb{R}^{3\times 3}$ and $o \in \mathbb{R}^3$ denote the distortion matrix and the bias, respectively. Further, $e \in \mathbb{R}^3$ denotes the measurement noise, which is assumed to be Gaussian distributed with zero mean and covariance matrix Σ_e.

2.4.2 Gyroscope

The physical quantity measured by a gyroscope is the angular velocity of the body frame w.r.t. the navigation frame, expressed in the body frame. Implicitly, the angular velocity ω^{b} can be defined such that

$$\frac{dR_{\mathrm{b}}^{\mathrm{n}}}{dt} = R_{\mathrm{b}}^{\mathrm{n}}[\omega^{\mathrm{b}}]^{\wedge}. \tag{2.24}$$

Therefore, the ideal measurement is written as

$$\omega = h(x, u) = \omega^{\mathrm{b}}, \tag{2.25}$$

where $x = R_{\mathrm{b}}^{\mathrm{n}}$, $u = \omega^{\mathrm{b}}$. Using (2.23), the measurement model for a gyroscope is

$$\tilde{\omega} = \tilde{h}(x, u; \theta^{\omega}) + e^{\omega} = D^{\omega}\omega^{\mathrm{b}} + o^{\omega} + e^{\omega}, \tag{2.26}$$

where the superscript ω on D, o, and e distinguishes these quantities from those associated with other sensor types.

For the gyroscope used in the thesis, the distortion effect is negligible. Therefore, D^{ω} is assumed to be the identity matrix.

2.4.3 Accelerometer

The physical quantity measured by an accelerometer is the 'specific force', the difference between the acceleration w.r.t. the n-frame and the acceleration due to gravity, expressed in the body frame. The specific force, denoted by s^{b}, can be described as

$$s^{\mathrm{b}} = R_{\mathrm{b}}^{\mathrm{n}\top}(a^{\mathrm{n}} - \mathrm{g}^{\mathrm{n}}), \tag{2.27}$$

where $a^{\mathrm{n}} \in \mathbb{R}^3$ denotes the acceleration w.r.t. the n-frame expressed in the n-frame, and $\mathrm{g}^{\mathrm{n}} \in \mathbb{R}^3$ denotes the acceleration due to gravity, which is assumed to be known. Therefore, the ideal measurement is written as

$$s = h(x, u) = R_{\mathrm{b}}^{\mathrm{n}\top}(a^{\mathrm{n}} - \mathrm{g}^{\mathrm{n}}), \tag{2.28}$$

where $x = R_{\mathrm{b}}^{\mathrm{n}}$, $u = a^{\mathrm{n}}$. Using (2.23), the measurement model for an accelerometer is

$$\tilde{s} = \tilde{h}(x, u; \theta^{a}) + e^{a} = D^{a}R_{\mathrm{b}}^{\mathrm{n}\top}(a^{\mathrm{n}} - \mathrm{g}^{\mathrm{n}}) + o^{a} + e^{a}. \tag{2.29}$$

For the same reason as in the gyroscope, D^{a} is assumed to be the identity matrix.

2.4.4 Magnetometer

The ideal measurement of a magnetometer is the ambient magnetic field in the body frame, which can be expressed as

$$m = h(x,u) = R_{\mathrm{b}}^{\mathrm{n}\top} m^{\mathrm{n}}, \tag{2.30}$$

where $x = R_{\mathrm{b}}^{\mathrm{n}}, u = 0$, and $m^{\mathrm{n}} \in \mathbb{R}^3$ denotes the magnetic field in the n-frame. Using (2.23), the measurement model for a magnetometer is

$$\tilde{m} = \tilde{h}(x,u;\theta^m) + e^m = D^m R_{\mathrm{b}}^{\mathrm{n}\top} m^{\mathrm{n}} + o^m + e^m. \tag{2.31}$$

2.5 IMU Pre-integration

IMU pre-integration is a technique that combines IMU measurements received in a time interval into a probabilistic description of the pose change in that interval. It suits particularly well in situations where the measurements from low-rate sensors, e.g., cameras, need to be fused with those from an IMU, and when jointly calibrating an IMU and other types of sensors, e.g., magnetometers, that provide direction measurements.

2.5.1 Modeling Uncertainty and Useful Approximations in SO(3)

In terms of modeling uncertainty in SO(3), it is natural to consider the stochastic model

$$\tilde{R} = R\,\mathrm{Exp}_R(\epsilon), \quad \epsilon \sim \mathcal{N}(0,\Sigma), \tag{2.32}$$

where $\tilde{R}, R \in \mathrm{SO}(3)$ denote the random variable and the deterministic value, respectively. Furthermore, $\epsilon \in \mathbb{R}^3$ denotes Gaussian noise with zero mean and covariance matrix $\Sigma \in \mathbb{R}^{3\times 3}$. For small covariances, the distribution of $\tilde{R}$ can be approximated as [9]

$$p(\tilde{R}) = \frac{1}{\sqrt{(2\pi)^3 \det(\Sigma)}} \mathrm{e}^{-\frac{1}{2}\|\mathrm{Log}_R(R^{-1}\tilde{R})\|_\Sigma^2}, \tag{2.33}$$

where $\det(\Sigma)$ denotes the determinant of the matrix Σ, and $\|x\|_\Sigma^2 = x^\top \Sigma^{-1} x$ for $x \in \mathbb{R}^3$.

Two useful approximations are [10]

$$\mathrm{Exp}_R(\tau + \delta\tau) \approx \mathrm{Exp}_R(\tau)\mathrm{Exp}_R(J^r(\tau)\delta\tau), \tag{2.34}$$
$$\mathrm{Log}_R(\mathrm{Exp}_R(\tau)\mathrm{Exp}_R(\delta\tau)) \approx \tau + J^{-r}(\tau)\delta\tau \tag{2.35}$$

Here, $\tau, \delta\tau \in \mathbb{R}^3$, and $\|\delta\tau\| \ll \|\tau\|$. Furthermore, $J^r(\cdot), J^{-r}(\cdot) \in \mathbb{R}^{3\times 3}$ denote the right Jacobian of SO(3) and its inverse, respectively, which can be found in [11, p. 40].

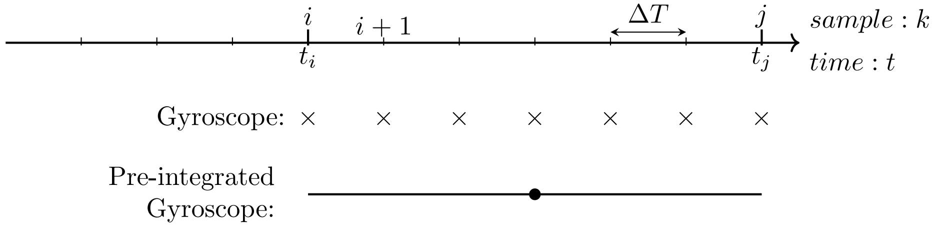


Figure 2.5: *Equal-interval sampling scheme. The i^{th} gyroscope sample is sampled at t_i on the time axis. The gyroscope pre-integration combines the samples from t_i to t_j into one single measurements.*

2.5.2 Pre-integrated Gyroscope Measurements

Consider the equal-interval sampling scheme with sampling interval ΔT as illustrated in Figure 2.5. The goal of gyroscope pre-integration is to derive a probabilistic model for the orientation change, denoted by ΔR_{ij}, between the time instance t_i and t_j, given the gyroscope measurements. To that end, the orientation matrices at two consecutive samples are related via the kinematic model (2.24), and the orientation change between several samples is derived by recursively applying the relation.

Assuming ω^{b} is piece-wise constant during the sample interval $[t_k, t_{k+1}), i \leq k \leq j-1$, the orientation matrix at time t_k and t_{k+1} are then related via

$$R_{\mathrm{b}}^{\mathrm{n}}(t_{k+1}) = R_{\mathrm{b}}^{\mathrm{n}}(t_k)\mathrm{Exp}_R(\omega^{\mathrm{b}}(t_k)\Delta T). \tag{2.36}$$

Recalling the sensor measurement model (2.23) and expressing the angular velocity in terms of the gyroscope measurement, bias, and noise, (2.36) can be rewritten as

$$R_{k+1} = R_k\mathrm{Exp}_R\left((\tilde{\omega}_k - o_k^{\omega} - e_k^{\omega})\Delta T\right), \tag{2.37}$$

where the coordinate frame symbols are dropped, and the notation $(\cdot)_k \triangleq (\cdot)(t_k)$ is introduced for brevity. To compute the orientation change during $[t_i, t_j)$, (2.37) is recursively evaluated to get

$$R_j = R_i \prod_{k=i}^{j-1} \mathrm{Exp}_R\left((\tilde{\omega}_k - o_k^{\omega} - e_k^{\omega})\Delta T\right). \tag{2.38}$$

Here $\prod_{k=i}^{j-1} A_k = A_i A_{i+1} \cdots A_{j-1}, A_{(\cdot)} \in \mathrm{SO}(3)$. Assume the bias is constant over the interval $[t_i, t_j)$, i.e., $o_k^{\omega} \equiv o^{\omega}$. let $\Delta R_{ij} \triangleq R_i^{\top} R_j$, and $J_k^r \triangleq J^r\left((\omega_k - o^{\omega})\Delta T\right)$. Using the approximation in the previous section (2.38) can be rewritten as [9]

$$\Delta\tilde{R}_{ij} = \Delta R_{ij}\mathrm{Exp}_R(\delta\phi_{ij}), \tag{2.39a}$$

where

$$\Delta \tilde{R}_{ij} = \prod_{k=i}^{j-1} \mathrm{Exp}_R \left((\tilde{\omega}_k - o^{\omega}) \Delta T \right), \tag{2.39b}$$

$$\delta \phi_{ij} \approx \sum_{k=i}^{j-1} \Delta \tilde{R}_{k+1j}^{\top} J_k^r e_k^{\omega} \Delta T. \tag{2.39c}$$

Equation (2.39) is referred to as the pre-integrated gyroscope measurement model.

2.6 Inertial Navigation Systems

The basic idea of INS is to determine the position and velocity w.r.t. an inertial reference frame using Newton's laws of motion. The process, known as inertial navigation [12], may be equivalently realized with inertial sensors that sense rotational and translational motion.

An INS framework is shown in Figure 2.6. The body-mounted accelerometer measures the specific force in the b-frame, which is converted to the n-frame using the orientation matrix obtained by integrating the angular velocity measurement. The acceleration in the n-frame is obtained by removing the local gravity vector from the specific force. The acceleration is then integrated twice to get position and velocity estimates.

An INS is an entirely self-contained navigation solution because it does not depend on signals from external sources. However, integrating the IMU measurements will cause errors to accumulate over time so the estimates will become more and more inaccurate. The error growth rate depends on the IMU sensor grade. It can grow to several hundred meters in 1 minute for consumer-grade IMUs like the one in Figure 1.1 [13].

2.7 Magnetic Field Models

The magnetic field is a vector field $M(r,t) : \mathbb{R}^3 \times \mathbb{R} \rightarrow \mathbb{R}^3$, where $r \in \mathbb{R}^3$ denotes the spatial location and t denotes the time. It fulfills the Maxwell's equations [14]

$$\nabla \cdot M(r,t) = 0, \tag{2.40a}$$

$$\nabla \times M(r,t) = \mu_0 \left(J + \epsilon_0 \frac{\partial E}{\partial t} \right), \tag{2.40b}$$

where μ_0 denotes the vacuum magnetic permeability, ϵ_0 denotes the vacuum permittivity, E denotes the electric field, and J denotes the current density. When there is no free current and the electric field is static, which is the default assumption in this thesis, Maxwell's equation simplifies to

$$\nabla \cdot M(r) = 0, \tag{2.41a}$$

$$\nabla \times M(r) = 0. \tag{2.41b}$$

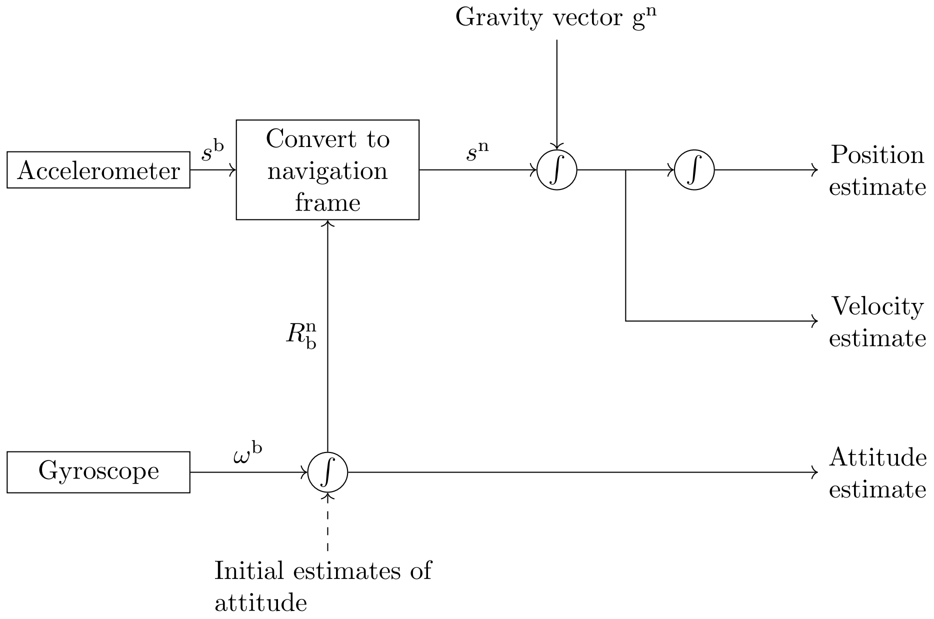


Figure 2.6: *A simplified inertial navigation system.*

Here the dependence on time is dropped. The second condition in (2.41), namely the curl-free field condition, allows for the magnetic field to be defined as the gradient of a scalar magnetic field potential $\phi(r)$, i.e.,

$$M(r) = \nabla\phi(r). \tag{2.42}$$

There are two mainstream approaches to mathematically modeling indoor magnetic fields. These are the Gaussian process model and the polynomial model. The former can represent the magnetic field on multi-scale levels [15, 16], while the latter is more common in modeling the field on a smaller (local) scale.

2.7.1 Gaussian Process Model

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution. A real-valued Gaussian process $f(\mathrm{x})$ with the mean $m(\mathrm{x}) = \mathbb{E}[f(\mathrm{x})]$ and the covariance function $k(\mathrm{x},\mathrm{x}') = \mathbb{E}[(f(\mathrm{x}) - m(\mathrm{x}))(f(\mathrm{x}) - m(\mathrm{x}'))^\top]$ may written as [17]

$$f(\mathrm{x}) \sim \mathcal{GP}\left(m(\mathrm{x}), k\left(\mathrm{x},\mathrm{x}'\right)\right). \tag{2.43}$$

Being a non-parametric model, the Gaussian process is a powerful tool for modeling and inference. The disadvantage is, however, that the computation complexity of regression increases cubically w.r.t. the number of data points.

In the work [18], the covariance functions of the Gaussian process used for modeling the magnetic field are designed in such a way that the process fulfills the divergence- and curl-free conditions.

2.7.2 Polynomial Model

The polynomial model is named after the polynomial function used to model the scalar magnetic field potential $\phi(r)$ [19]. That is,

$$\phi(r;\mu) = h(r)^\top \mu + c. \tag{2.44}$$

Here $h(r)$ is a vector whose elements are given by the product $r_x^i r_y^j r_z^k$ for $\forall i, j, k \in \mathbb{N}$, subject to $1 \leq i + j + k \leq m$, $m = 1, 2, \cdots, l + 1$. Further, $\mu \in \mathbb{R}^L$ is a column vector of dimension $L = (l + 4)(l + 3)(l + 2)/6 - 1$ and c is an arbitrary constant. Let $\Gamma(r) = \nabla_r h(r)^\top$, then $M(r;\mu)$ can be written as

$$M(r;\mu) = \nabla_r \phi(r;\mu) = \Gamma(r)\mu. \tag{2.45}$$

For the model $M(r;\mu)$ to fulfill the second condition in (2.41), the model parameters μ must be selected so that the following holds

$$\nabla_r \cdot \Gamma(r)\mu = \sum_{i=x,y,z} \frac{d[\Gamma(r)\mu]_i}{dr_i} = 0, \quad \forall r \in \Omega. \tag{2.46}$$

Here $\frac{d[\Gamma(r)\mu]_i}{dr_i}$ is an $(l-1)^{\text{th}}$ degree polynomial. For the equality in (2.46) to hold for all r, the coefficients for each term in the polynomial must be 0. This constraint can be written as a linear equation system

$$D_p \mu = 0, \tag{2.47}$$

where D_p is a constant matrix that depends only on the degree of the polynomial function, see [19] for details.

2.8 Optimization

Optimization is an important tool for model parameter and state estimation. It is widely used in sensor calibration [20, 21], visual-inertial navigation [22], and SLAM applications [23, 24]. In these applications, the optimization problems that need to be solved are typically nonlinear least squares problems. Due to the optimization variables involving rotations, special care must be taken when solving these problems. The study of this type of optimization is referred to as optimization on smooth manifolds [8].

2.8.1 Nonlinear Least Squares Problem

A nonlinear least-square problem is a problem of the form [25]

$$x^* = \arg\min_x \frac{1}{2} \sum_{k=1}^{m} f_k^2(x), \tag{2.48}$$

with variable $x \in \mathbb{R}^n$ and where $f_k : \mathbb{R}^n \to \mathbb{R}$. It can be rewritten in a more compact form as

$$x^* = \arg\min_x V(x), \quad V(x) = \frac{1}{2}\|f(x)\|^2, \tag{2.49}$$

where $f(x) = [f_1(x), f_2(x), \cdots, f_m(x)]^\top$.

2.8.2 The Gauss-Newton Method

The Gauss-Newton method is an iterative optimization algorithm to solve (2.49). At each iteration, it replaces $f(x)$ by its first-order Taylor approximation around the current iterate x_k [25]. That is, $f(x)$ is approximated as

$$f(x) \approx f(x_k) + J(x_k)\underbrace{(x - x_k)}_{\triangleq \Delta_k}, \tag{2.50}$$

where $J(x_k) \in \mathbb{R}^{m \times n}$ is the Jacobian matrix defined as

$$J(x_k) = \left.\frac{\partial f(x)}{\partial x}\right|_{x=x_k} \triangleq \begin{bmatrix} \frac{\partial f_1}{\partial x_k^{(1)}} & \cdots & \frac{\partial f_1}{\partial x_k^{(n)}} \\ \vdots & & \vdots \\ \frac{\partial f_m}{\partial x_k^{(1)}} & \cdots & \frac{\partial f_m}{\partial x_k^{(n)}} \end{bmatrix}, \tag{2.51}$$

and Δ_k denotes the increment that will be applied to x_k to form the next iterate. In this way, the optimization problem becomes a linear least squares problem, and the optimal Δ_k^* should fulfill

$$J(x_k)^\top J(x_k)\Delta_k^* = -J(x_k)^\top f(x_k). \tag{2.52}$$

Therefore,

$$\Delta_k^* = -J(x_k)^\dagger f(x_k), \tag{2.53}$$

where $(\cdot)^\dagger$ denote the Moore–Penrose inverse [26]. The algorithm will update the iterate, and repeat the process until it meets some termination conditions, e.g., the increment's norm is smaller than a threshold. The Gauss-Newton method is summarized in Algorithm 1.

Algorithm 1 Gauss-Newton method

```
Input: the function $f$ whose squared norm is to be minimized
Output: the minimizer $x^*$
  Initialisation : initial value $x_0$, iterate step $k = 0$
  While the user-specified termination condition is NOT met
      Calculate the Jacobian $J(x_k)$ and the function value $f(x_k)$
      Calculate the increment $\Delta_k^*$ according to (2.53)
      Set the next iterate to $x_{k+1} = x_k + \Delta_k^*$
      Set iterate step $k = k + 1$
  end while
  Set $x^* = x_k$
```

2.8.3 Optimization on Smooth Manifolds

When the optimization variables contain rotation components, e.g., rotation matrices, they will cause several issues for the Gauss-Newton method. First, one may not calculate the Jacobian matrices as easily as in Euclidean space, since the rotation components live in a nonlinear space. Second, the increment calculated in each step may not easily be applied to the rotation matrices, as matrix additions will lead to them no longer being rotation matrices. Fortunately, in most cases, the optimization variables live on smooth manifolds, and the study of optimization on smooth manifolds provides good tools for solving these issues.

In the case where the optimization variables in (2.49) belong to a smooth manifold $\mathcal{M}$, the optimization problem can be reparameterized using a vector $\delta x_k \in \mathbb{R}^{\dim(\mathcal{M})}$ [9]. That is,

$$\arg\min_{x \in \mathcal{M}} \quad \frac{1}{2}\|f(x)\|^2 \quad \Rightarrow \quad \arg\min_{\delta x_k \in \mathbb{R}^{\dim(\mathcal{M})}} \quad \frac{1}{2}\|f(x_k \oplus \delta x_k)\|^2. \tag{2.54}$$

Here $x_k \in \mathcal{M}$ denotes the current iterate. Note that the optimization variable δx is in Euclidean space and that $f(x_k \oplus \delta x_k) \approx f(x_k) + J(x_k)\delta x_k$, where $J(x_k)$ is defined in (2.19). Therefore, the standard Gauss-Newton method can be applied to the right-hand side of (2.54). In each iterate, the optimal δx_k^* is sought after and the next iterate is computed as $x_{k+1} = x_k \oplus \delta x_k^*$.

2.9 State Estimation

State estimation is a technique for estimating the states of a system from the measurements of its output. It is one of the most important parts of the magnetic field-aided INS.

2.9.1 State-Space Models

The state-space model describes the evolution of the state and the relation between the measurement and the state. Two commonly used state-space models are linear and nonlinear state-space models. Linear state-space models can be written as

$$x_{k+1} = F_k x_k + G_k u_k + w_k, \tag{2.55a}$$
$$y_k = H_k x_k + e_k. \tag{2.55b}$$

Here $x_k \in \mathbb{R}^n$ and $y_k \in \mathbb{R}^p$ denote the state at time k and the measurement, respectively. Moreover, $u_k \in \mathbb{R}^m$, $w_k \in \mathbb{R}^n$, and $e_k \in \mathbb{R}^p$ denote the system input, the process noise, and the measurement noise, respectively. Lastly, $F_k \in \mathbb{R}^{n\times n}$, $G_k \in \mathbb{R}^{n\times m}$, and $H_k \in \mathbb{R}^{p\times n}$ are matrices of proper size. Nonlinear state-space models can be written as

$$x_{k+1} = f_k(x_k, u_k) + w_k, \tag{2.56a}$$
$$y_k = h_k(x_k) + e_k. \tag{2.56b}$$

Here $f_k(x_k, u_k)$ and $h_k(x_k)$ denote the nonlinear state dynamics and the measurement equation, respectively.

Additionally, in both cases the random variables $\{x_0, w_k, e_k\}$ are assumed to satisfy

$$\mathbb{E}\left(\begin{bmatrix} x_0 \\ w_i \\ e_i \end{bmatrix}\begin{bmatrix} x_0 & w_j & e_j & 1 \end{bmatrix}\right) = \begin{bmatrix} \Pi_0 & 0 & 0 & 0 \\ 0 & Q_i\delta_{ij} & 0 & 0 \\ 0 & 0 & R_i\delta_{ij} & 0 \end{bmatrix}, \tag{2.57}$$

where δ_{ij} denotes the Kronecker delta.

2.9.2 Kalman Filter and Extended Kalman Filter

When x_0, w_k, and e_k are Gaussian random variables, the least-mean-squares estimator for (2.55) is the well-celebrated *Kalman filter* (KF) [27]. The Kalman filter gives the estimate $\hat{x}_{k|k}$ and its covariance matrix $P_{k|k}$ given the measurement up to time k, and its procedures are given in Algorithm 2. For the nonlinear state-space model (2.56), the Kalman filter can be applied to the linearized state-space model, resulting in the *extended Kalman filter* (EKF), whose procedures are given in Algorithm 3.

2.9.3 Error-State Kalman Filter

The *error-state Kalman filter* (ESKF) is introduced to circumvent the problem that the filter correction cannot be directly added to the state when the state contains rotation components, i.e., the state space is a manifold. It is essentially an EKF, except the filtering happens in the error state space [28, p. 198]. The error state is a vector in a linear space representing the difference between two state vectors. The main advantage of introducing the error state is that the uncertainty in the error state is well-defined, especially when the state space is a manifold. However, for the sake of simplicity, the basic idea will be presented with the state space being Euclidean space.

Algorithm 2 Kalman filter

```
Input: ${u_k, y_k}_{k=0}^{N}$
Output: ${\hat{x}_{k|k}, P_{k|k}}_{k=0}^{N}$
  Initialisation : estimated state $\hat{x}_{0|-1}$, covariance matrix $P_{0|-1}$
  For $k = 0$ to $N - 1$ do
    Measurement update:
    $S_k = H_k P_{k|k-1} H_k^\top + R_k$
    $K_k = P_{k|k-1} H_k^\top S_k^{-1}$
    $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - H_k \hat{x}_{k|k-1})$
    $P_{k|k} = P_{k|k-1} - K_k H_k P_{k|k-1}$
    Time update:
    $\hat{x}_{k+1|k} = F_k \hat{x}_{k|k} + G_k u_k$
    $P_{k+1|k} = F_k P_{k|k} F_k^\top + Q_k$
  end for
```

Algorithm 3 Extended Kalman filter

```
Input: ${u_k, y_k}_{k=0}^{N}$
Output: ${\hat{x}_{k|k}, P_{k|k}}_{k=0}^{N}$
  Initialisation : estimated state $\hat{x}_{0|-1}$, covariance matrix $P_{0|-1}$
  For $k = 0$ to $N - 1$ do
    Measurement update:
    $H_k = \frac{\partial h_k(x_k)}{\partial x_k}\Big|_{x_k = \hat{x}_{k|k-1}}$
    $S_k = H_k P_{k|k-1} H_k^\top + R_k$
    $K_k = P_{k|k-1} H_k^\top S_k^{-1}$
    $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - h_k(\hat{x}_{k|k-1}))$
    $P_{k|k} = P_{k|k-1} - K_k H_k P_{k|k-1}$
    Time update:
    $F_k = \frac{\partial f_k(x_k, u_k)}{\partial x_k}\Big|_{x_k = \hat{x}_{k|k}}$
    $\hat{x}_{k+1|k} = f_k(\hat{x}_{k|k}, u_k)$
    $P_{k+1|k} = F_k P_{k|k} F_k^\top + Q_k$
  end for
```

Consider the nonlinear state space model (2.56). Let $\delta x_k \triangleq x_k^{\text{true}} - \bar{x}_k$ denote the difference between the true state and some linearization point, then the linearization can be carried out with Taylor's expansion

$$\begin{aligned} x_{k+1}^{\text{true}} &= f_k(x_k^{\text{true}}, u_k) + w_k \\ &\approx \underbrace{f_k(\bar{x}_k, u_k)}_{\bar{x}_{k+1}} + \frac{\partial f_k(x_k, u_k)}{\partial x_k}\bigg|_{x_k=\bar{x}_k} \underbrace{(x_k^{\text{true}} - \bar{x}_k)}_{\triangleq \delta x_k} + w_k, \end{aligned} \tag{2.58a}$$

$$\begin{aligned} y_k &= h(x_k^{\text{true}}) + e_k \\ &\approx \underbrace{h(\bar{x}_k)}_{\bar{y}_k} + \frac{\partial h_k(x_k)}{\partial x_k}\bigg|_{x_k=\bar{x}_k} \underbrace{(x_k^{\text{true}} - \bar{x}_k)}_{\triangleq \delta x_k} + e_k. \end{aligned} \tag{2.58b}$$

Move the first term on the right hand side of (2.58a) and (2.58b) to the left, respectively, and using the definition $\delta y_k \triangleq y_k - h(\bar{x}_k)$ the error-state-space model can be identified as

$$\delta x_{k+1} = F_k \delta x_k + w_k, \tag{2.59a}$$

$$\delta y_k = H_k \delta x_k + e_k, \tag{2.59b}$$

where

$$F_k = \frac{\partial f_k(x_k, u_k)}{\partial x_k}\bigg|_{x_k=\bar{x}_k}, \tag{2.59c}$$

$$H_k = \frac{\partial h_k(x_k)}{\partial x_k}\bigg|_{x_k=\bar{x}_k}. \tag{2.59d}$$

The standard Kalman filter can be used to estimate δx_k and the error state estimate $\delta \hat{x}_{k|k}$ will add to $\bar{x}_k$ to form the posterior estimate $\hat{x}_{k|k}$. The posterior estimate will become the linearization point for propagating the state to the next time step. This is equivalent to linearizing around the current state estimate, which makes it a de facto EKF filter. Since the error state estimate is used to form $\hat{x}_{k|k}$ to make it as close to the true state as possible, the prior estimate of the error state is always 0. Hence, in the error state Kalman filter, only the error state's covariance is propagated. The error state Kalman filter algorithm is given in Algorithm 4.

As said, the ESKF is useful when the state space is a manifold, which is common in localization and navigation applications. In these applications, the state vector contains an orientation component. Furthermore, the input to the system is usually unknown and can only be measured with some noise. The linearization is much more involved than presented here, interested readers can refer to [6] for the derivation and actual implementation.

Algorithm 4 Error state Kalman filter algorithm

```
Input: ${u_k, y_k}_{k=0}^N$
Output: ${\hat{x}_{k|k}, P_{k|k}}_{k=1}^N$
  Initialisation : nominal state $\bar{x}_0$, covariance matrix $P_{0|-1}$
  For $k = 0$ to $N - 1$ do
      Error state observation
      $S_k = H_k P_{k|k-1} H_k^\top + R_k$
      $K_k = P_{k|k-1} H_k^\top S_k^{-1}$
      $\delta\hat{x}_{k|k} = K_k(y_k - H_k \bar{x}_k)$
      $P_{k|k} = P_{k|k-1} - K_k H_k P_{k|k-1}$
      Form the estimated state and update the nominal state
      $\hat{x}_{k|k} = \bar{x}_k \oplus \delta\hat{x}_{k|k}$
      $\bar{x}_k = \hat{x}_{k|k}$
      State propagation
      $\bar{x}_{k+1} = f(\bar{x}_k, u_k)$
      Error state uncertainty propagation
      $P_{k+1|k} = F_k P_{k|k} F_k^\top + Q_k$
  end for
```

2.9.4 Observability

In the deterministic context, i.e., the noise-free state-space model, observability determines whether the state can be inferred from the system's output. Therefore, to understand the behavior of the model it is important to conduct an observability analysis on the state-space model. The observability analysis may be carried out in several ways, one of which is constructing the observability matrix and checking if it has full column rank. If it has a full column rank the system is observable. For the linear state-space model (2.55), the observability matrix $\mathcal{O}_{k_0:k_f}$ associated within time window $[k_0, k_f]$ is defined as [29]

$$\mathcal{O}_{k_0:k_f} \triangleq \begin{bmatrix} H_{k_0} \\ H_{k_0+1}\Phi(k_0+1, k_0) \\ \vdots \\ H_{k_f}\Phi(k_f, k_0) \end{bmatrix}, \tag{2.60a}$$

where

$$\Phi(k_f, k_0) = F_{k_f-1} F_{k_f-2} \cdots F_{k_0}. \tag{2.60b}$$

For the nonlinear state-space model (2.56), it is convenient to consider the concept of local observability [30]. In this definition, the state-space model is linearized around a nominal trajectory, and the observability matrix is constructed with the Jacobians computed from the linearization. That is,

$$\bar{\mathcal{O}}_{k_0:k_f} = \begin{bmatrix} \bar{H}_{k_0} \\ \bar{H}_{k_0+1}\bar{\Phi}(k_0+1,k_0) \\ \vdots \\ \bar{H}_{k_f}\bar{\Phi}(k_f,k_0) \end{bmatrix}, \tag{2.61a}$$

where

$$\bar{H}_k = \left.\frac{\partial h_k(x_k)}{\partial x_k}\right|_{x_k=\bar{x}_k}, \tag{2.61b}$$

$$\bar{\Phi}(k,k_0) = \bar{F}_{k-1}\bar{F}_{k-2}\cdots\bar{F}_{k_0}, \tag{2.61c}$$

$$\bar{F}_k = \left.\frac{\partial f_k(x_k,u_k)}{\partial x_k}\right|_{x_k=\bar{x}_k, u_k=\bar{u}_k}. \tag{2.61d}$$

Here $\{\bar{x}_k | k = k_0 : k_f\}$ denotes the nominal trajectory, which is the solution of (2.56) with the control sequence $\bar{u}_{k_0:k_0+N-1}$ and the process noise turned off.

Part II

Localization using IMU and Magnetometer Array

3 Sensor Calibration

Sensor calibration involves identifying and compensating for sensor imperfections. This process is crucial for INS, which estimate the position and orientation of a moving object by integrating acceleration and angular velocity measurements from an IMU. The accuracy of these estimates is influenced by several factors, including the quality of calibration. For example, the bias in the measurements from uncalibrated accelerometers and gyroscopes can individually cause drift in the estimated position to grow at a rate of t^2 and t^3, respectively [13]. Therefore, it is essential to address sensor errors carefully.

Additionally, the hardware used in magnetic field-aided INS includes a 2-D magnetometer array and an IMU. Since each magnetometer in the array may exhibit slight differences and the sensitivity axes might not align with those of the IMU, it is vital to calibrate the magnetometers and IMU jointly. This ensures that each magnetometer's imperfections are compensated for and that the sensitivity axes of all sensors are properly aligned.

3.1 Sensor Models

Consider the measurement from an accelerometer, a gyroscope, and a magnetometer triad at time k,

$$y_k = \begin{bmatrix} \tilde{s}_k \\ \tilde{\omega}_k \\ \tilde{m}_k \end{bmatrix} \in \mathbb{R}^9. \tag{3.1}$$

If the sensor system is slowly rotated, with a minimum of translational movement, in a homogenous magnetic field, then the measurement can be modeled as [31]

$$y_k = \tilde{h}(x_k, u_k; \theta) + e_k, \tag{3.2}$$

where $e_k = [e_k^{a\top} \; e_k^{\omega\top} \; e_k^{m\top}]^\top \in \mathbb{R}^9$ is the additive white Gaussian noise with the covariance matrix $\Sigma = \mathrm{Cov}(e_k) = \mathrm{blkdiag}(\Sigma_a, \Sigma_m, \Sigma_\omega)$. Here $\Sigma_{(\cdot)}$ denote the noise covariance of each type of sensor. Further,

$$\tilde{h}(x_k, u_k; \theta) = \begin{bmatrix} -R_k^\top \mathrm{g}^\mathrm{n} + o^a \\ \omega_k + o^\omega \\ D^m R_k^\top m(\alpha) + o^m \end{bmatrix}, \tag{3.3a}$$

where

$$\theta = \{o^a, o^\omega, D^m, o^m, \alpha\}, \tag{3.3b}$$

$$x_k = R_k, \tag{3.3c}$$

$$u_k = \omega_k. \tag{3.3d}$$

Here $R_k \in \mathrm{SO}(3)$ denotes the orientation matrix of the body frame, i.e., $R_k \equiv R^\mathrm{n}_{\mathrm{b}_k}$. Moreover, $\mathrm{g}^\mathrm{n} = [0 \; 0 \; -\mathrm{g}_0]^\top$, where g_0 is the amplitude of the local gravity. Furthermore, $D^m \in \mathbb{R}^{3\times 3}$ denotes the magnetometer distortion matrix, which encodes the compound effects of sensor imperfections and the misalignment between the axes of the magnetometer frame and inertial sensors' frame. Lastly, the magnetic field is parameterized by its dip angle $\alpha \in [-\pi, \pi)$, which is the angle formed by the field and the horizontal plane, and can be written as $m(\alpha) = [0 \; \cos(\alpha) \; -\sin(\alpha)]^\top$. When the sample interval ΔT is small enough so that the angular velocity is nearly constant in that interval, the orientation matrices at two consecutive instances are related to each other via the difference equation $R_{k+1} = R_k \, \mathrm{Exp}_R(\omega_k \Delta T)$.

3.2 Nonlinear Least Squares Problem

A nonlinear least square problem can be set up to find the best fit θ^* and $R^*_{0:N-1}$ for the measurements. The reason why ω_k is not sought after is that it will be determined by the orientation matrices according to the difference equation $R_{k+1} = R_k \, \mathrm{Exp}_R(\omega_k \Delta T)$.

Let $y_{0:N-1}$ denote a sequence of measurements $\{y_0, y_1, \cdots, y_{N-1}\}$. Then the optimization problem is set up as

$$\{\theta^*, R^*_{0:N-1}\} = \underset{\theta, R_{0:N-1}}{\arg\min} \, V(\theta, R_{0:N-1}), \tag{3.4a}$$

where

$$\begin{aligned} V(\theta, R_{0:N-1}) =& \frac{1}{2}\sum_{0}^{N-1} \|\tilde{s}_k + R_k^\top \mathrm{g}^\mathrm{n} - o^a\|^2_{\Sigma_a} + \frac{1}{2}\sum_{0}^{N-1} \|\tilde{m}_k - D^m R_k^\top m(\alpha) - o^m\|^2_{\Sigma_m} \\ &+ \frac{1}{2}\sum_{0}^{N-2} \|\tilde{\omega}_k - \frac{1}{\Delta T}\mathrm{Log}_R(R_k^\top R_{k+1}) - o^\omega\|^2_{\Sigma_\omega}. \end{aligned} \tag{3.4b}$$

Here $\tilde{\omega}_{N-1}$ is discarded since the orientation R_N is needed to construct the residual term, whose information is unavailable in the measurements.

The cost function (3.4b) can be rewritten into a form more commonly used to solve using the Gauss-Newton method, i.e.,

$$\begin{aligned}
V(\theta, R_{0:N-1}) =& \frac{1}{2}\sum_{0}^{N-1} \| \underbrace{L_a^\top(\tilde{s}_k + R_k^\top \mathrm{g}^{\mathrm{n}} - o^a)}_{\triangleq r_k^a} \|^2 + \frac{1}{2}\sum_{0}^{N-1} \| \underbrace{L_m^\top \left(\tilde{m}_k - D^m R_k^\top m(\alpha) - o^m\right)}_{\triangleq r_k^m} \|^2 \\
&+ \frac{1}{2}\sum_{0}^{N-2} \| \underbrace{L_\omega^\top \left(\tilde{\omega}_k - \frac{1}{\Delta T}\mathrm{Log}_R(R_k^\top R_{k+1}) - o^\omega\right)}_{\triangleq r_k^\omega} \|^2 \\
=& \frac{1}{2} \| f(\theta, R_{0:N-1}) \|^2,
\end{aligned} \tag{3.5a}$$

$$f(\theta, R_{0:N-1}) \triangleq \begin{bmatrix} r_0^a \\ \vdots \\ r_{N-1}^a \\ r_0^m \\ \vdots \\ r_{N-1}^m \\ r_0^\omega \\ \vdots \\ r_{N-2}^\omega \end{bmatrix}, \tag{3.5b}$$

where $L_{(\cdot)}$ is the lower triangular matrix from Cholesky decomposition of the covariance matrix $\Sigma_{(\cdot)}^{-1}$, and r_k^a, r_k^m, and r_k^ω denote the normalized residuals for accelerometer, magnetometer, and gyroscope measurement, respectively.

The optimization problem (3.4) is a standard optimization problem on smooth manifolds. Consider the manifold,

$$\begin{aligned}
\mathcal{M} &= \{(\theta_{\mathrm{vec}}, R_{0:N-1}) \mid \theta_{\mathrm{vec}} \in \mathbb{R}^{19}, R_{0:N-1} \in \mathrm{SO}(3)^N\}, \\
\theta_{\mathrm{vec}} &= [(o^a)^\top \; (o^\omega)^\top \; {D_{\mathrm{vec}}^m}^\top \; (o^m)^\top \; \alpha]^\top,
\end{aligned} \tag{3.6}$$

where D_{vec}^m denotes the vectorized matrix, i.e.,

$$D_{\mathrm{vec}}^m = [D_{1,1}^m \; D_{2,1}^m \; D_{3,1}^m \cdots D_{2,3}^m \; D_{3,3}^m]^\top. \tag{3.7}$$

Here $D_{i,j}^m$ denotes the element at (i,j) in D^m. Each normalized residual is a function mapping from $\mathcal{M}$ to $\mathbb{R}^3$. To calculate the derivatives of these function w.r.t. the elements in $\mathcal{M}$, the operator $\oplus$ for $\mathcal{M}$ and $\ominus$ for $\mathbb{R}^3$ are defined as

$$X \oplus \delta X \triangleq (\theta_{\mathrm{vec}} + \delta\theta_{\mathrm{vec}}, \mathrm{Exp}_R(\delta R_0)R_0, \cdots, \mathrm{Exp}_R(\delta R_{N-1})R_{N-1}), \tag{3.8a}$$

$$a \ominus b \triangleq a - b. \tag{3.8b}$$

Here $X = (\theta_{\text{vec}}, R_{0:N-1}) \in \mathcal{M}$, $\delta X = [\delta\theta_{\text{vec}}^\top, \delta R_0^\top, \cdots, \delta R_{N-1}^\top]^\top$, where $\delta\theta_{\text{vec}} \in \mathbb{R}^{19}$ and $\delta R_i \in \mathbb{R}^3$. Furthermore, $a, b \in \mathbb{R}^3$. Then the Jacobian of the residual required for the Gauss-Newton method can be calculated. For the residual term r_k^a,

$$\frac{\partial r_k^a}{\partial R_k} = L_a^\top [R_k^\top \mathrm{g}^{\mathrm{n}}]^\wedge R_k^\top, \tag{3.9a}$$

$$\frac{\partial r_k^a}{\partial o_a} = -L_a^\top; \tag{3.9b}$$

for the residual term r_k^m,

$$\frac{\partial r_k^m}{\partial D_{\text{vec}}^m} = -L_m^\top \left(R_k^\top m(\alpha)\right)^\top \otimes_K I_3, \tag{3.10a}$$

$$\frac{\partial r_k^m}{\partial R_k} = -L_m^\top D^m [R_k^\top m(\alpha)]^\wedge R_k^\top, \tag{3.10b}$$

$$\frac{\partial r_k^m}{\partial \alpha} = L_m^\top D^m R_k^\top [0 \ \sin(\alpha) \ \cos(\alpha)]^\top, \tag{3.10c}$$

$$\frac{\partial r_k^m}{\partial o^m} = -L_m^\top; \tag{3.10d}$$

$$\tag{3.10e}$$

and for the residual term r_k^ω,

$$\frac{\partial r_k^\omega}{\partial R_k} = \frac{1}{\Delta T} L_\omega^\top J^{-r}\left(\mathrm{Log}_R(R_k^\top R_{k+1})\right) R_{k+1}^\top, \tag{3.11a}$$

$$\frac{\partial r_k^\omega}{\partial R_{k+1}} = -\frac{1}{\Delta T} L_\omega^\top J^{-r}\left(\mathrm{Log}_R(R_k^\top R_{k+1})\right) R_{k+1}^\top, \tag{3.11b}$$

$$\frac{\partial r_k^\omega}{\partial o^\omega} = -L_\omega^\top, \tag{3.11c}$$

where $\otimes_K$ denotes the Kronecker product. The Jacobian matrix of $f(\theta, R_{0:N-1})$ can be constructed from these Jacobian matrices of the residuals, and the optimization procedure can be carried out as described in Section.2.8.3.

3.3 Reduce Optimizing Variables by IMU Pre-integration

The calibration data collection usually requires several minutes to complete, because the sensor system needs to be rotated slowly to keep a minimum acceleration while exposing to as many orientations as possible. This leads to a large number of data samples N, and the minimization process in (3.4) becomes time-consuming.

In fact, the calibration parameter θ has a much smaller dimension compared to the rotation trajectory $R_{0:N-1}$, therefore, to speed up the calibration one may consider reducing the number of the orientation matrices to be optimized while keeping them representative of the rotation trajectory. A natural choice is to optimize the orientation matrices at every N_{int} timesteps. This means that the residual terms need to be adjusted accordingly. It is easily done for accelerometer and magnetometer measurements — every N_{int} measurements are kept because they only depend on the orientation of the sensor at the measuring time instance. However, it is not straightforward to do so for gyroscope measurements because the original gyroscope residual term involves orientation matrices at consecutive timestamps. Hence a method for combining gyroscope measurements in a time interval so that the combined measurement only concerns the orientation at the start and the end is needed. The pre-integrated gyroscope measurement model proposed in [9] suits this aim well.

Recalling the result in Section 2.5, the pre-integrated gyroscope measurement model on the time interval $[t_i, t_j]$ is

$$\Delta\tilde{R}_{ij} = \Delta R_{ij}\mathrm{Exp}_R(\delta\phi_{ij}), \tag{3.12a}$$

where

$$\Delta\tilde{R}_{ij} = \prod_{k=i}^{j-1} \mathrm{Exp}_R\left((\tilde{\omega}_k - o^{\omega})\Delta T\right), \tag{3.12b}$$

$$\Delta R_{ij} = R_i^\top R_j, \tag{3.12c}$$

$$\delta\phi_{ij} \approx \Sigma_{k=i}^{j-1}\Delta\tilde{R}_{k+1j}^\top J_k^r e_k^{\omega}\Delta T, \tag{3.12d}$$

$$J_k^r \triangleq J^r((\tilde{\omega}_k - o^{\omega})\Delta T). \tag{3.12e}$$

It is natural to construct the residual for the pre-integrated gyroscope measurement as

$$r_{ij}^{\text{p.}} = L_{ij}^\top \mathrm{Log}_R(\Delta\tilde{R}_{ij}^\top \Delta R_{ij}), \tag{3.13}$$

where L_{ij} is the lower triangular matrix from Cholesky decomposition of the inverse of the covariance matrix $\Sigma_{\text{p.}}^{ij} = \mathrm{Cov}(\delta\phi_{ij})$. The residual can be further tailored to suit the calibration task by reparameterizing the gyroscope bias as

$$o^{\omega} = \bar{o}^{\omega} + \Delta o^{\omega}, \tag{3.14}$$

where $\bar{o}^{\omega} \in \mathbb{R}^3$ denotes the nominal gyroscope bias and $\Delta o^{\omega} \in \mathbb{R}^3$ denotes the difference in the nominal and true gyroscope bias. That is, instead of seeking a good fit $o^{\omega *}$ for the gyroscope bias, the goal is to find an optimal $\Delta o^{\omega *}$ that gives minimum squared error.

Reparameterizing the gyroscope bias offers several benefits. First, the nominal bias can be easily obtained by averaging measurements when the sensor is stationary. This nominal bias is typically close to the true bias, which helps initialize the orientation matrices and thereby reduces the likelihood of optimization failure. Secondly, the pre-integrated measurement $\Delta\tilde{R}_{ij}$ is a function of the gyroscope bias o^{ω}, whose value is changed in each iteration in the Guass-Newton method. This means it needs to be re-evaluated each iteration, which is computationally inefficient. With the nominal gyroscope bias $\bar{o}^{\omega}$, the re-evaluation, given an updated Δo^{ω}, can be approximated by [9]

$$\Delta\tilde{R}_{ij}(\bar{o}^{\omega}+\Delta o^{\omega}) \approx \underbrace{\prod_{k=i}^{j-1} \mathrm{Exp}_R\left((\tilde{\omega}_k-\bar{o}^{\omega})\Delta T\right)}_{\Delta\bar{R}_{ij}} \mathrm{Exp}_R\left(\left.\frac{\partial\Delta\tilde{R}_{ij}}{\partial o^{\omega}}\right|_{o^{\omega}=\bar{o}^{\omega}} \Delta o^{\omega}\right), \tag{3.15}$$

where $\Delta\tilde{R}_{ij}(\bar{o}^{\omega}+\Delta o^{\omega})$ denotes the value of $\Delta\tilde{R}_{ij}$ when $o^{\omega}=\bar{o}^{\omega}+\Delta o^{\omega}$. Note that $\Delta\bar{R}_{ij}$ and the Jacobian matrix $\left.\frac{\partial\Delta\tilde{R}_{ij}}{\partial o^{\omega}}\right|_{o^{\omega}=\bar{o}^{\omega}}$ remain constant at each iteration, so they can be precomputed before the optimization starts[1]. Lastly, the covariance matrix $\Sigma_{\mathrm{p.}}^{ij}$ is also a function of o^{ω}. Given that the nominal bias $\bar{o}^{\omega}$ is close enough to the true bias, then $\tilde{\omega}_k - o^{\omega} \approx \tilde{\omega}_k - \hat{o}^{\omega}$. Further, due to the sensor system being slowly rotated, $(\tilde{\omega}_k-\bar{o}^{\omega})\Delta T$ is so small that $J^r((\tilde{\omega}_k-\bar{o}^{\omega})\Delta T)$ is close to the identity matrix. Therefore, $\Sigma_{\mathrm{p.}}^{ij}$ can be further simplified to

$$\Sigma_{\mathrm{p.}}^{ij} \approx \mathrm{Cov}\left(\sum_{k=i}^{j-1} \Delta\tilde{R}_{k+1j}^{\top} e_k^{\omega}\Delta T\right). \tag{3.16}$$

When the gyroscope noise on the three axes is assumed to be uncorrelated and identically distributed, i.e., $\Sigma_{\omega}=\sigma_{\omega}^2 I_3$, it can be shown that

$$\Sigma_{\mathrm{p.}}^{ij} = \mathrm{Cov}\left(\sum_{k=i}^{j-1} e_k^{\omega}\Delta T\right) = (j-i)\Delta T^2\sigma_{\omega}^2 I_3. \tag{3.17}$$

In this way, the pre-integrated noise covariance matrix is approximated to a constant matrix, so it can be pre-calculated and does not need to be updated in each iteration. With slight abuse of notation, $L_{\mathrm{p.}}$ is used to denote the lower triangle matrix from the Cholesky decomposition of the covariance matrix (3.17) when pre-integration stride $N_{\mathrm{int}}=j-i$ is decided.

Now consider the optimization problem after pre-integrating every N_{int} measurements. Let two index sets S_1 and S_2 be defined as

$$S_1 \triangleq \left\{0, N_{\mathrm{int}}, \cdots, \left\lfloor\frac{N-1}{N_{\mathrm{int}}}\right\rfloor N_{\mathrm{int}}\right\}, \tag{3.18a}$$

$$S_2 \triangleq \left\{0, N_{\mathrm{int}}, \cdots, \left(\left\lfloor\frac{N-1}{N_{\mathrm{int}}}\right\rfloor - 1\right) N_{\mathrm{int}}\right\}. \tag{3.18b}$$

The measurements and their covariance in the optimization problem are

$$\{\tilde{s}_{k'} | k' \in S_1\}, \quad \mathrm{Cov}(\tilde{s}_{k'}) = \Sigma_a, \tag{3.19a}$$

$$\{\tilde{m}_{k'} | k' \in S_1\}, \quad \mathrm{Cov}(\tilde{m}_{k'}) = \Sigma_m, \tag{3.19b}$$

$$\{y_{k'}^{\mathrm{p.}} | k' \in S_2\}, \quad \mathrm{Cov}(y_{k'}^{\mathrm{p.}}) = \Sigma_{\mathrm{p.}}, \tag{3.19c}$$

[1] The exact expression of the Jacobian matrix can be found in Appendix in [9, p. 17].

where

$$y_{k'}^{\mathrm{p.}} \triangleq \Delta\tilde{R}_{k'k'+N_{\mathrm{int}}}. \tag{3.20}$$

Let $\theta^{'} \triangleq \{o^a, \Delta o^\omega, D^m, o^m, \alpha\}$ and $\{R_{k'}\} \triangleq \{R_k \mid k \in S_1\}$. A new optimization problem can be set up as

$$\begin{aligned} V^{'}(\theta^{'},\{R_{k'}\}) =& \frac{1}{2}\sum_{k'\in S_1} \| \underbrace{L_a^\top(\tilde{s}_{k'} + R_{k'}^\top \mathrm{g}^{\mathrm{n}} - o^a)}_{\triangleq r_{k'}^a} \|^2 + \frac{1}{2}\sum_{k'\in S_1} \| \underbrace{L_m^\top\left(\tilde{m}_{k'} - D^m R_{k'}^\top m(\alpha) - o^m\right)}_{\triangleq r_{k'}^m} \|^2 \\ &+ \frac{1}{2}\sum_{k'\in S_2} \| \underbrace{L_{\mathrm{p.}}^\top \mathrm{Log}_R(y_{k'}^{\mathrm{p.}^\top} R_{k'}^\top R_{k'+N_{\mathrm{int}}})}_{\triangleq r_{k'}^{\mathrm{p.}}} \|^2 \\ =& \frac{1}{2}\left\| f^{'}(\theta', \{R_{k'}\}) \right\|^2, \end{aligned} \tag{3.21a}$$

where

$$f'(\theta^{'},\{R_{k'}\}) \triangleq \begin{bmatrix} r_0^a \\ \vdots \\ r^a_{\lfloor \frac{N-1}{N_{\mathrm{int}}} \rfloor N_{\mathrm{int}}} \\ r_0^m \\ \vdots \\ r^m_{\lfloor \frac{N-1}{N_{\mathrm{int}}} \rfloor N_{\mathrm{int}}} \\ r_0^{\mathrm{p.}} \\ \vdots \\ r^{\mathrm{p.}}_{(\lfloor \frac{N-1}{N_{\mathrm{int}}} \rfloor - 1) N_{\mathrm{int}}} \end{bmatrix}. \tag{3.21b}$$

The Jacobian matrices of $r_{k'}^a$ and $r_{k'}^m$ are of the same form as in Section 3.2. The Jacobian matrices of $r_{k'}^{\mathrm{p.}}$ w.r.t. $R_{k'}$, $R_{k'+N_{\mathrm{int}}}$, and Δo^ω are

$$\frac{\partial r_{k'}^{\mathrm{p.}}}{\partial R_{k'}} = -L_{\mathrm{p.}}^\top J^{-r}(\mathrm{Log}_R(Q)) R_{k'+N_{\mathrm{int}}}^\top, \tag{3.22a}$$

$$\frac{\partial r_{k'}^{\mathrm{p.}}}{\partial R_{k'+N_{\mathrm{int}}}} = L_{\mathrm{p.}}^\top J^{-r}(\mathrm{Log}_R(Q)) R_{k'+N_{\mathrm{int}}}^\top, \tag{3.22b}$$

$$\frac{\partial r_{k'}^{\mathrm{p.}}}{\partial \Delta o^\omega} = -L_{\mathrm{p.}}^\top J^{-r}(\mathrm{Log}_R(Q)) Q^\top J^r \left(\left.\frac{\partial y_{k'}^{\mathrm{p.}}}{\partial o^\omega}\right|_{o^\omega = \bar{o}^\omega} \Delta o^\omega \right) \left.\frac{\partial y_{k'}^{\mathrm{p.}}}{\partial o^\omega}\right|_{o^\omega = \bar{o}^\omega}, ^2 \tag{3.22c}$$

where

$$Q = y_{k'}^{\mathrm{p.}^\top} R_{k'}^\top R_{k'+N_{\mathrm{int}}}. \tag{3.22d}$$

[2] The derivation can be found in [9, p. 19, eq. 81].

The calibration parameters can be obtained by solving the optimization problem with the Gauss-Newton method given the derived Jacobians. The procedures described in this section constitute the proposed method to calibrate the sensor system.

3.4 Evaluation

The proposed method was evaluated using both simulation and real-world datasets. To measure calibration accuracy, the percentage error of calibration parameters was computed for the simulation datasets. For sensor bias, the percentage error was calculated as the ratio of the norm of the error vector and that of the true bias vector. For distortion matrix D, the error was determined as the ratio of the norm of the vectorized error matrix and that of the vectorized true distortion matrix. Lastly, the dip angle percentage error was computed as the ratio of the absolute value of the angle error and the true dip angle.

3.4.1 Simulation Setup and Results

In the simulation, the datasets are generated according to the sensor model (3.3a). To limit memory usage in generating the datasets, the sensor sampling rate is set at 100 Hz, and each dataset consists of 32182 measurement samples, corresponding to a data collection duration of approximately 5 minutes. Furthermore, the true parameters used to generate the measurements are drawn as follows,

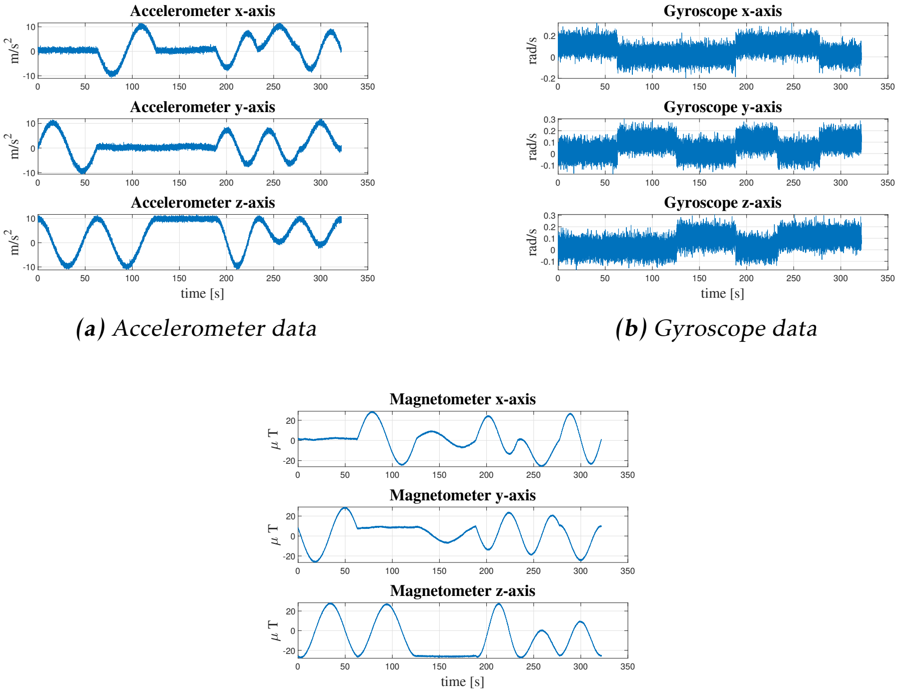
$$\begin{aligned}
o^a &\sim N(0, 0.5^2 I_3) && (\mathrm{m/s^2}) \\
o^\omega &\sim N([0.01\ 0.01\ 0.01]^\top, 0.004^2 I_3) && (\mathrm{rad/s}) \\
D^m_{i,j} &\sim \begin{cases} N(27, 0.5^2), & \text{if } i = j \\ N(0, 0.5^2), & \text{otherwise} \end{cases} \\
o^m &\sim N(0, 2^2 I_3) && (\mu\mathrm{T}) \\
\alpha &\sim N(1.3, 0.1^2) && (\mathrm{rad})
\end{aligned} \tag{3.23}$$

The standard deviation of the Gaussian white measurement noise on each axis of the accelerometer, gyroscope, and magnetometer is set to 0.5 m/s^2, 0.001 rad/s, and 0.3 μT, respectively. Figure 3.1 shows an example of the simulated dataset.

The initial value for the calibration parameter was chosen as the mean value of the distribution, and the Gauss-Newton method was applied to the optimization problem in (3.21). The magnetometer data was then calibrated with the obtained calibration parameters as follows,

$$\breve{m}_k = \hat{D}^{m^{-1}}(\tilde{m}_k - \hat{o}^m). \tag{3.24}$$

The norm of the calibrated data is scaled by the true magnetic field magnitude because the magnetic field is assumed to have a unit norm in calibration. The scaled norm is plotted against the norm of the uncalibrated data, as shown in



*(**a**) Accelerometer data*

*(**b**) Gyroscope data*

*(**c**) Magnetometer data*

***Figure 3.1:** The simulated dataset used in the evaluation.*

Figure 3.2. The results indicate that the norm of the calibrated magnetometer measurements exhibits lower variance and is more concentrated around the true magnetic field magnitude of 27 μT, demonstrating the effectiveness of the proposed calibration algorithm. Additionally, the experiment was repeated 10 times, and the calibration parameter error box plots are presented in Figure 3.3. The percentage errors for all parameters are within a reasonable range (less than 10%), further confirming the effectiveness of the proposed algorithm.

3.4.2 Real-world Experiment Setup and Results

A real-world experiment was also conducted on the calibration dataset collected by the sensor array board in Figure 1.1. The dataset was collected outdoors in Linköping, Sweden. When collecting these datasets, the sensor board was placed in an environment where the local magnetic field was approximately homogenous. The data collection lasted approximately 4 minutes, during which the sensor board was slowly rotated to minimize the acceleration induced by rotation and to cover as many orientations as possible. The data collected by all sensors were synchronized and the sampling rate was 500 Hz. The IMU data and magne-

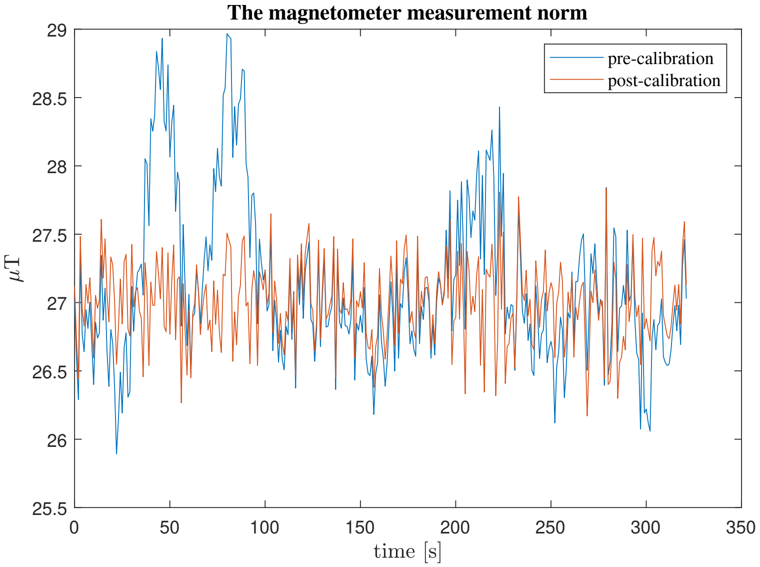
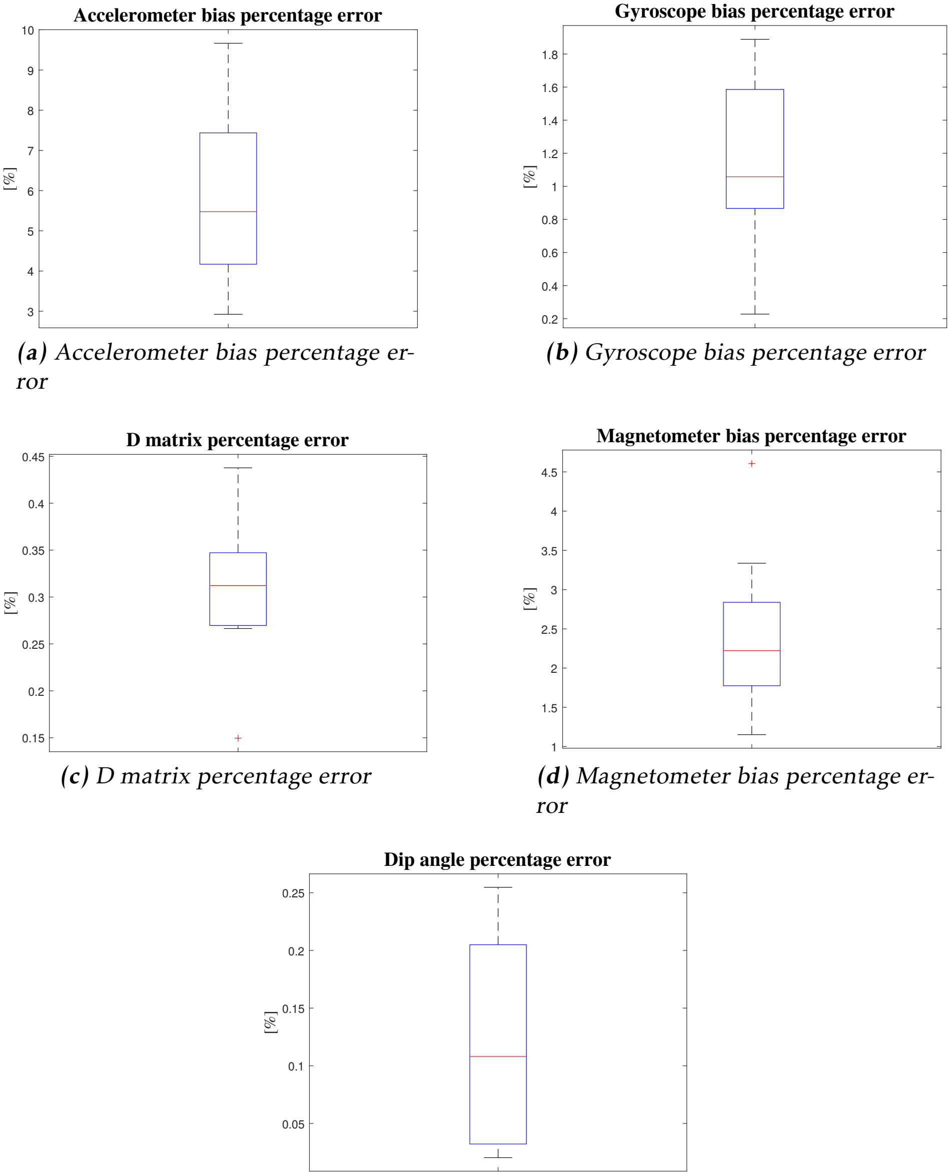


Figure 3.2: *The magnetometer measurement norm before and after calibration. The calibrated measurements' norm fluctuates around the expected magnetic field magnitude and has less variance.*

tometer measurements from one of 30 magnetometers are shown in Figure 3.4.

To calibrate all sensors on board, the proposed algorithm was used to calibrate a magnetometer and IMU pair one by one — the calibration process was repeated 30 times before the calibration task was completed. For each dataset, 30 pair-wise calibration parameters are collected, enabling consistency checks. Consistency is indicated by minimal variation in the inertial sensor biases and the dip angle across the pair-wise calibrations.

The calibration results are presented in Figure 3.5 and Figure 3.6. It can be seen from Figure 3.5 that the reported IMU biases are consistent since they have small variances and are concentrated around the peak value. Furthermore, the dip angle histogram plot also indicates that the algorithm is consistent because the estimated dip angles are very close to the reference value given by the World Magnetic Model [32]. Lastly, Figure 3.6 shows that the calibrated magnetometer measurements exhibit significantly less variance in norm across sensors compared to the uncalibrated data.



*(**a**) Accelerometer bias percentage error*

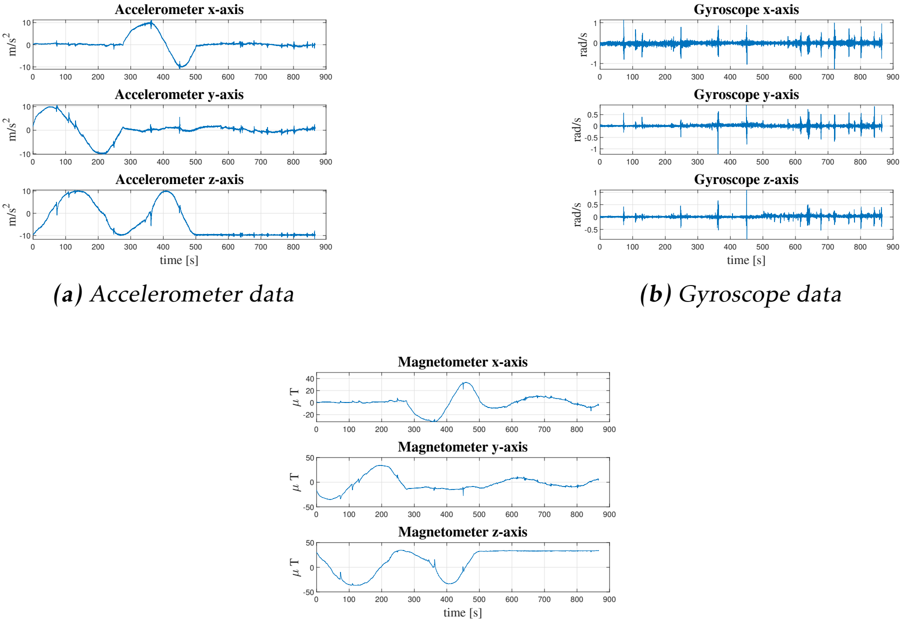
*(**b**) Gyroscope bias percentage error*

*(**c**) D matrix percentage error*

*(**d**) Magnetometer bias percentage error*

*(**e**) Dip angle percentage error*

Figure 3.3: *The summary of the calibration parameter errors in 10 trials. Here, the bottom and top of each box are the 25th and 75th percentiles of the errors, respectively. Furthermore, the red line in the middle of each box is the median value. Lastly, the horizontal bar at the top and the bottom are the maximum and minimum values, respectively.*

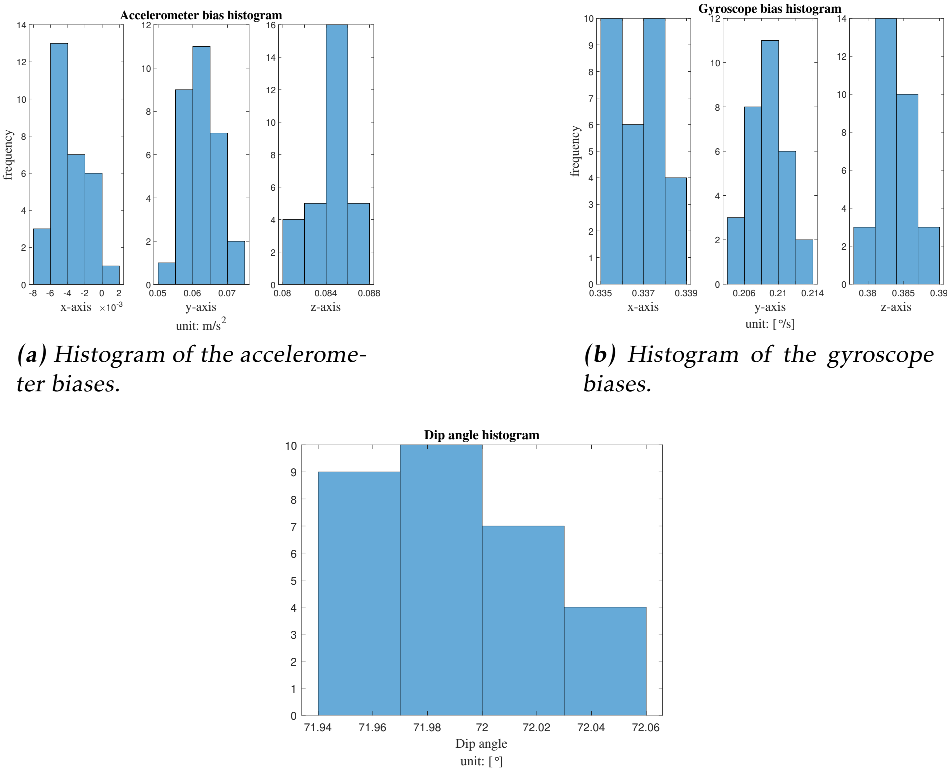


(a) *Accelerometer data*

(b) *Gyroscope data*

(c) *Magnetometer data*

Figure 3.4: *The real-world dataset used in the evaluation.*



(a) *Histogram of the accelerometer biases.*

(b) *Histogram of the gyroscope biases.*

(c) *Histogram of the dip angle.*

Figure 3.5: *The histogram of calibration parameters reported in 30 pair-wise experiments. Since the true IMU bias and dip angle should be the same across all IMU-MAG calibration pairs, a concentrated histogram indicates the calibration's consistency.*

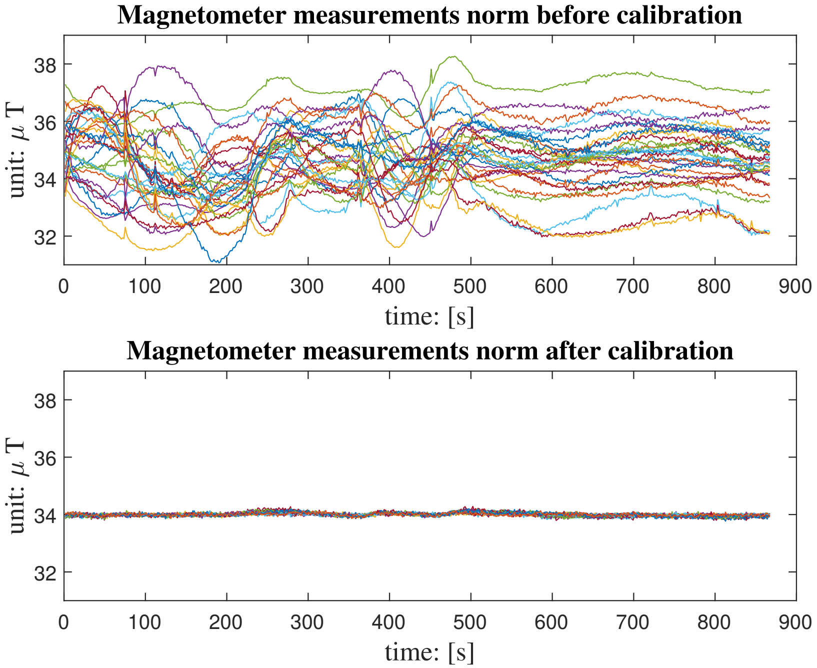


Figure 3.6: *The norm of measurements from the 30 magnetometers before calibration (upper) and after calibration (lower).*

4

The Magnetic Field-Aided Inertial Navigation System

A magnetic field-aided INS for indoor navigation is presented in this chapter. The system leverages an array of magnetometers to measure spatial variations in the magnetic field, which are then used to estimate the displacement and orientation changes of the system, thereby aiding the INS. A block diagram of the system is shown in Figure 4.1.

This chapter is adapted from the material presented in

> Chuan Huang, Gustaf Hendeby, Hassen Fourati, Christophe Prieur, and Isaac Skog. MAINS: A magnetic-field-aided inertial navigation system for indoor positioning. *IEEE Sensors Journal*, 24(9):15156–15166, 2024. doi: 10.1109/JSEN.2024.3379932,

and

> Chuan Huang, Gustaf Hendeby, and Isaac Skog. A tightly-integrated magnetic-field aided inertial navigation system. In *Proc. 2022 25th Int. Conf. on Information Fusion (FUSION)*, pages 1–8, Linköping, Sweden, July 2022. doi: 10.23919/FUSION49751.2022.9841304.

4.1 System Modeling

Consider the problem of estimating the position and orientation of the sensor platform in Figure 1.1 in an indoor environment where the magnetic field variation is similar to that in Figure 4.2. To that end, a state-space model will be presented to realize a magnetic field-aided INS.

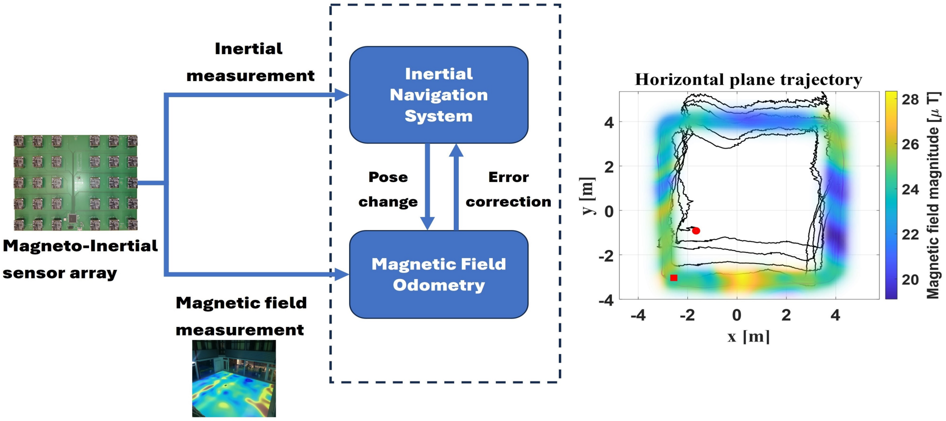

Inertial measurement
Magneto-Inertial sensor array
Magnetic field measurement
Inertial Navigation System
Pose change
Error correction
Magnetic Field Odometry
Horizontal plane trajectory
y [m]
x [m]
4
2
0
-2
-4
-4
-2
0
2
4
28
26
24
22
20
Magnetic field magnitude [μ T]

Figure 4.1: *Overview of the proposed magnetic field-aided INS and localization result. The proposed system is a tightly integrated inertial navigation system and magnetic field odometry. In the experiment, the sensor array moves along a square with its borders in the colored region, and the black curve is the estimated position.*

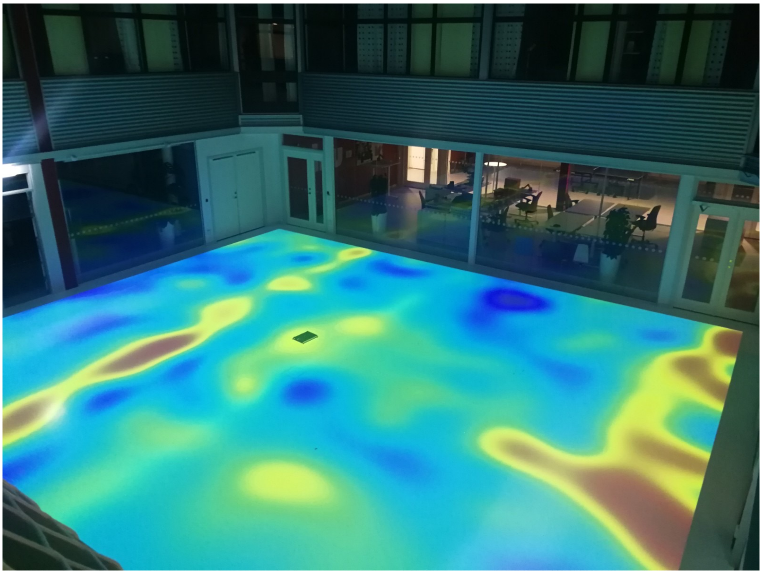


Figure 4.2: *Illustration of the magnetic-field magnitude variations inside a building. The field near the floor was measured with a magnetometer, whose location was tracked by camera-based tracking systems. The field measurement was then interpolated, and the field magnitude was projected on the floor.*

4.1.1 Inertial Navigation Equations

Let the INS navigation state x_k^{ins}, the inertial measurements $\tilde{u}_k$, and the process noise w_k^{ins} be defined as

$$x_k^{\text{ins}} \triangleq \begin{bmatrix} p_k^{\text{n}} \\ v_k^{\text{n}} \\ q_k \\ o_k^a \\ o_k^\omega \end{bmatrix}, \quad \tilde{u}_k \triangleq \begin{bmatrix} \tilde{s}_k \\ \tilde{\omega}_k \end{bmatrix}, \quad \text{and} \quad w_k^{\text{ins}} \triangleq \begin{bmatrix} w_k^a \\ w_k^\omega \\ w_k^{o^a} \\ w_k^{o^\omega} \end{bmatrix}, \tag{4.1}$$

respectively. Here, $p_k^{\text{n}} \in \mathbb{R}^3$, $v_k^{\text{n}} \in \mathbb{R}^3$, and $q_k \in \text{S}^3$ denote the position, velocity, and orientation (parameterized as a unit quaternion) at time k, respectively. The superscript n indicates that the vector is represented in the navigation frame. Moreover, $o_k^a \in \mathbb{R}^3$ and $o_k^\omega \in \mathbb{R}^3$ denote the accelerometer and gyroscope bias, respectively. Lastly, w_k^a and w_k^ω denote the accelerometer and gyroscope measurement noise, respectively. And $w_k^{o^a}$ and $w_k^{o^\omega}$ denote the random walk process noise for the accelerometer and gyroscope biases, respectively. For an INS that uses low-cost sensors and moves at moderate velocities such that the effects of the transport rate, earth rotation, etc., can be neglected, the navigation equations are given by [6]

$$x_{k+1}^{\text{ins}} = f^{\text{ins}}(x_k^{\text{ins}}, \tilde{u}_k, w_k^{\text{ins}}), \tag{4.2a}$$

where

$$f^{\text{ins}}(x_k^{\text{ins}}, \tilde{u}_k, w_k^{\text{ins}}) = \begin{bmatrix} p_k^{\text{n}} + v_k^{\text{n}}\Delta T + (R_{\text{b}_k}^{\text{n}} s_k + \text{g}^{\text{n}})\frac{\Delta T^2}{2} \\ v_k^{\text{n}} + (R_{\text{b}_k}^{\text{n}} s_k + \text{g}^{\text{n}})\Delta T \\ q_k \otimes \text{Exp}_q(\omega_k \Delta T) \\ o_k^a + w_k^{o^a} \\ o_k^\omega + w_k^{o^\omega} \end{bmatrix}, \tag{4.2b}$$

and

$$s_k = \tilde{s}_k - o_k^a - w_k^a, \tag{4.2c}$$

$$\omega_k = \tilde{\omega}_k - o_k^\omega - w_k^\omega. \tag{4.2d}$$

Here, the subscript b_k denotes the body frame at time k. Further, w_k^{ins} is modeled as a zero-mean white Gaussian noise process with covariance matrix $\Sigma_{w_k^{\text{ins}}} = \text{blkdiag}(\Sigma_a, \Sigma_\omega, \Sigma_{o^a}, \Sigma_{o^\omega})$, where $\Sigma_{(\cdot)}$ denotes the covariance matrix of the corresponding noise component.

4.1.2 Magnetic Field Modeling

Recalling the polynomial model in Section 2.7.2, equations (2.45) and (2.47) define the magnetic field model, which writes as

$$M(r;\mu) = \Gamma(r)\mu, \quad D_p\mu = 0. \tag{4.3}$$

Note that $M(r;\mu)$ can be reparameterized by introducing the matrix $D_p^{\perp} \triangleq \text{null}\{D_p\}$ whose columns span the nullspace of D_p, and then setting $\mu = D_p^{\perp}\theta$, where θ is a column vector of dimension equal to that of the nullspace of D_p. The reparameterized model is given by

$$M(r;\theta) = \Phi(r)\theta, \tag{4.4}$$

where $\Phi(r) \triangleq \Gamma(r)D_p^{\perp} \in \mathbb{R}^{3\times\kappa}$ is the regression matrix defined in [19] and $\theta \in \mathbb{R}^{\kappa}$ is the coefficient of the polynomial model; for a l^{th} order polynomial the model has $\kappa = \dim(\theta) = l^2 + 4l + 3$ unknown parameters [16][1]. Note that the model (4.4) can be defined in either the body frame or navigation frame. Within this work, it will be defined in the body frame. Next, a procedure for transforming the model from body frame α to body frame β will be presented.

4.1.3 Transforming Models Between Body Frames

Let the magnetic field model (4.4) be associated with the body frame, which means $M(r;\theta)$ accepts locations r expressed in the current body frame and outputs magnetic field vector in the same frame. Then the magnetic field can be represented in the two body frames α and β, i.e.,

$$\begin{aligned} M^{\alpha}(r^{\alpha};\theta^{\alpha}) &= \Phi(r^{\alpha})\theta^{\alpha}, \\ M^{\beta}(r^{\beta};\theta^{\beta}) &= \Phi(r^{\beta})\theta^{\beta}. \end{aligned} \tag{4.5}$$

Here the superscripts on M and r denote the corresponding body frame in which they are resolved and the superscript on θ denotes the frame with which the model coefficients are associated.

By expressing the magnetic field vector at a given location with the two models and aligning them in the same frame, the two models can be related as

$$M^{\beta}(r^{\beta};\theta^{\beta}) = R_{\alpha}^{\beta} M^{\alpha}(r^{\alpha};\theta^{\alpha}), \tag{4.6a}$$

where

$$r^{\alpha} = (R_{\alpha}^{\beta})^{\top} r^{\beta} + \Delta p^{\alpha}. \tag{4.6b}$$

Here Δp^{α} denotes the translation vector expressed in body frame α. An illustration of the geometric relationship between the two frames and the magnetic field vector is shown in Figure 4.3.

Now consider transforming the magnetic field model from body frame b_k to b_{k+1}. Let the relative body frame change be encoded by

$$\psi_k = \begin{bmatrix} \Delta p_k^{\mathrm{b}_k} \\ \Delta\phi_k \end{bmatrix}, \tag{4.7}$$

where $\Delta p_k^{\mathrm{b}_k} \in \mathbb{R}^3$ and $\Delta\phi_k \in [0, 2\pi)^3$ denote the translation and orientation change from the body frame b_k to b_{k+1}, respectively. Replacing α and β with

[1] The 1st and 2nd order models are provided in Appendix 4.A.

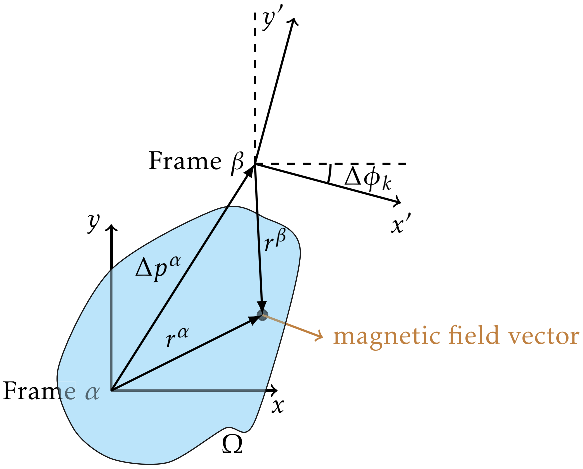


Figure 4.3: *A 2D illustration of the geometric relationship between the body frames at two consecutive times. The applicable region Ω of the magnetic field model at time k is in blue, and the black dot indicates the location where the two models output the corresponding magnetic field in their coordinate frames.*

b_k and b_{k+1} respectively and using θ_k and θ_{k+1} to denote the corresponding coefficients, the following holds

$$M^{\mathrm{b}_{k+1}}(r^{\mathrm{b}_{k+1}};\theta_{k+1}) = R_{\mathrm{b}_k}^{\mathrm{b}_{k+1}} M^{\mathrm{b}_k}(r^{\mathrm{b}_k};\theta_k), \tag{4.8a}$$

where

$$r^{\mathrm{b}_k} = (R_{\mathrm{b}_k}^{\mathrm{b}_{k+1}})^\top r^{\mathrm{b}_{k+1}} + \Delta p_k^{\mathrm{b}_k}. \tag{4.8b}$$

The rotation matrix $R_{\mathrm{b}_k}^{\mathrm{b}_{k+1}}$ and translation $\Delta p_k^{\mathrm{b}_k}$ are given by

$$R_{\mathrm{b}_k}^{\mathrm{b}_{k+1}} = (\mathrm{Exp}_R([\Delta\phi_k]))^\top, \tag{4.9a}$$

$$\Delta p_k^{\mathrm{b}_k} = R_{\mathrm{b}_k}^{\mathrm{n}\top}\left(v_k^{\mathrm{n}}\Delta T + (R_{\mathrm{b}_k}^{\mathrm{n}} s_k + \mathrm{g}^{\mathrm{n}})\frac{\Delta T^2}{2}\right). \tag{4.9b}$$

Here $\Delta\phi_k = \omega_k \Delta T$.

Next, substituting the generic magnetic model in (4.8) with the proposed polynomial model (4.4) yields the equality

$$\Phi(r^{\mathrm{b}_{k+1}})\theta_{k+1} = R_{\mathrm{b}_k}^{\mathrm{b}_{k+1}} \Phi(r^{\mathrm{b}_k})\theta_k. \tag{4.10}$$

Note that for a given $\{r^{\mathrm{b}_{k+1}}, \theta_k, \psi_k\}$, (4.10) represents 3 linear equations. Since θ_{k+1} is of dimension κ, which is greater than 3, it is necessary to use more than one location vector r to solve the equation system. In general, $S = \lceil \kappa/3 \rceil$ location vectors can be used to construct the equation system

$$A\theta_{k+1} = B(\psi_k)\theta_k, \tag{4.11a}$$

where

$$A = \begin{bmatrix} \Phi(r_1^{\mathrm{b}_{k+1}}) \\ \vdots \\ \Phi(r_S^{\mathrm{b}_{k+1}}) \end{bmatrix}, \quad B(\psi_k) = \begin{bmatrix} R_{\mathrm{b}_k}^{\mathrm{b}_{k+1}} \Phi(r_1^{\mathrm{b}_k}) \\ \vdots \\ R_{\mathrm{b}_k}^{\mathrm{b}_{k+1}} \Phi(r_S^{\mathrm{b}_k}) \end{bmatrix}. \tag{4.11b}$$

If the vectors $r_1^{\mathrm{b}_{k+1}}, \cdots, r_S^{\mathrm{b}_{k+1}}$ can be chosen such that A has full column rank, it holds that

$$\theta_{k+1} = A^{\dagger} B(\psi_k)\theta_k. \tag{4.12}$$

Since the magnetic field model should describe the magnetic field locally, the update in (4.12), which with time implicitly expands the applicable space Ω of the model, will inevitably introduce modeling errors. To account for those errors, with a slight abuse of notation, the update of the polynomial coefficients of the magnetic field as the body frame change is modeled as

$$\theta_{k+1} = f^{\theta}(\theta_k, x_k^{\text{ins}}, \tilde{u}_k, w_k^{\text{ins}}, w_k^{\theta}), \tag{4.13}$$

where

$$f^{\theta}(\theta_k, x_k^{\text{ins}}, \tilde{u}_k, w_k^{\text{ins}}, w_k^{\theta}) = A^{\dagger} B(\psi_k)\theta_k + w_k^{\theta}. \tag{4.14}$$

Here w_k^{θ} is assumed to be a white Gaussian noise process with zero mean and covariance matrix Σ_{θ}. Note that ψ_k is a function of x_k^{ins}, $\tilde{u}_k$, and w_k^{ins}.

4.1.4 Magnetometer Array Measurement Model

Given the magnetic field model in (4.4), the measurement $y_k^{(i)} \in \mathbb{R}^3$ from the i^{th} sensor in the magnetometer array at time k can be modeled as

$$y_k^{(i)} = \Phi(r_{m_i})\theta_k + e_k^{(i)}, \tag{4.15}$$

where $r_{m_i} \in \mathbb{R}^3$ denotes the location of the i^{th} magnetometer in the array. Further, $e_k^{(i)} \in \mathbb{R}^3$ denotes the measurement error, which includes both the measurement noise and the imperfections of the magnetic-field model. The error is assumed to be white and Gaussian distributed with covariance matrix $\Sigma_{e_k^{(i)}}$.

4.2 Complete System

Given the presented navigation equations and the magnetic-field model, the dynamics and observations of the full system can be described by the following state-space model.

Let the state vector x_k, the process noise vector w_k, and the measurement noise e_k be defined as

$$x_k \triangleq \begin{bmatrix} x_k^{\text{ins}} \\ \theta_k \end{bmatrix}, \quad w_k \triangleq \begin{bmatrix} w_k^{\text{ins}} \\ w_k^{\theta} \end{bmatrix}, \quad \text{and} \quad e_k \triangleq \begin{bmatrix} e_k^{(1)} \\ \vdots \\ e_k^{(N)} \end{bmatrix}, \tag{4.16}$$

respectively. Combining the models in (4.2), (4.13), and (4.15) gives the state-space model

$$x_{k+1} = f(x_k, \tilde{u}_k, w_k), \tag{4.17a}$$

$$y_k = Hx_k + e_k, \tag{4.17b}$$

where

$$f(x_k, \tilde{u}_k, w_k) = \begin{bmatrix} f^{\text{ins}}(x_k^{\text{ins}}, \tilde{u}_k, w_k^{\text{ins}}) \\ f^{\theta}(\theta_k, x_k^{\text{ins}}, \tilde{u}_k, w_k^{\theta}) \end{bmatrix}, \tag{4.17c}$$

and

$$H = \begin{bmatrix} 0_{3\times16} & \Phi(r_{m_1}) \\ \vdots & \vdots \\ 0_{3\times16} & \Phi(r_{m_N}) \end{bmatrix}. \tag{4.17d}$$

Here, the process noise covariance Q_k and the measurement noise covariance R_k are

$$Q_k \triangleq \text{Cov}(w_k) = \text{blkdiag}(\Sigma_a, \Sigma_\omega, \Sigma_{o^a}, \Sigma_{o^\omega}, \Sigma_\theta), \tag{4.17e}$$

$$R_k \triangleq \text{Cov}(e_k) = \text{blkdiag}(\Sigma_{e_k^{(1)}}, \Sigma_{e_k^{(2)}}, \cdots, \Sigma_{e_k^{(N)}}), \tag{4.17f}$$

respectively.

4.3 State Estimation

This section discusses the state estimation technique used in the magnetic field-aided INS. The nonlinear state-space model is linearized after introducing the error state, and the state estimate is done by the ESKF, as mentioned in Section 2.9.3.

4.3.1 Error State Definition

The error state δx_k is defined as

$$\delta x_k \triangleq \begin{bmatrix} \delta x_k^{\text{ins}} \\ \delta\theta_k \end{bmatrix} \quad \text{and} \quad \delta x_k^{\text{ins}} = \begin{bmatrix} \delta p_k^{\text{n}} \\ \delta v_k^{\text{n}} \\ \epsilon_k \\ \delta o_k^{a} \\ \delta o_k^{\omega} \end{bmatrix}. \tag{4.18}$$

Here, the standard additive error definition is used for the position, velocity, sensor biases, and magnetic field model parameters (e.g., $\delta p_k^{\text{n}} = p_k^{\text{n}} - \hat{p}_k^{\text{n}}$). On the other hand, the orientation error $\epsilon_k \in \mathbb{R}^3$ satisfies the equation $R_{\text{b}_k}^{\text{n}} \approx \hat{R}_{\text{b}_k}^{\text{n}}(I_3 + \epsilon_k^{\wedge})$. The true state x_k and the estimated state $\hat{x}_k$ relate to each other via

$$x_k = \hat{x}_k \oplus \delta x_k, \tag{4.19}$$

where the operator $\oplus$ is defined by

$$p_k^{\text{n}} = \hat{p}_k^{\text{n}} + \delta p_k^{\text{n}}, \tag{4.20a}$$
$$v_k^{\text{n}} = \hat{v}_k^{\text{n}} + \delta v_k^{\text{n}}, \tag{4.20b}$$
$$q_k = \hat{q}_k \otimes [1 \quad \frac{1}{2}\epsilon_k^{\top}]^{\top}, \tag{4.20c}$$
$$o_k^a = \hat{o}_k^a + \delta o_k^a, \tag{4.20d}$$
$$o_k^{\omega} = \hat{o}_k^{\omega} + \delta o_k^{\omega}, \tag{4.20e}$$
$$\theta_k = \hat{\theta}_k + \delta \theta_k. \tag{4.20f}$$

4.3.2 Inertial Error State Dynamics

The dynamics of δx_k^{ins} has been derived in [6] and are given by

$$\delta x_{k+1}^{\text{ins}} = F_k^{\text{ins}} \delta x_k^{\text{ins}} + G_k^{\text{ins}} w_k^{\text{ins}}, \tag{4.21a}$$

where

$$F_k^{\text{ins}} = \begin{bmatrix} I_3 & I_3\Delta T & 0 & 0 & 0 \\ 0 & I_3 & -\hat{R}_{\text{b}_k}^{\text{n}}[\hat{s}_k]^{\wedge}\Delta T & -\hat{R}_{\text{b}_k}^{\text{n}}\Delta T & 0 \\ 0 & 0 & \text{Exp}_R(\Delta\hat{\phi}_k)^{\top} & 0 & -I_3\Delta T \\ 0 & 0 & 0 & I_3 & 0 \\ 0 & 0 & 0 & 0 & I_3 \end{bmatrix}, \tag{4.21b}$$

$$G_k^{\text{ins}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \hat{R}_{\text{b}_k}^{\text{n}}\Delta T & 0 & 0 & 0 \\ 0 & I_3\Delta T & 0 & 0 \\ 0 & 0 & I_3\sqrt{\Delta T} & 0 \\ 0 & 0 & 0 & I_3\sqrt{\Delta T} \end{bmatrix}. \tag{4.21c}$$

Here, $\hat{s}_k = \tilde{s}_k - \hat{o}_k^a$ and $\Delta\hat{\phi}_k = (\tilde{\omega}_k - \hat{o}_k^{\omega})\Delta T$.

4.3.3 Magnetic Field Subsystem Error State Dynamics

To the first order, the errors in (4.13) propagate according to

$$\delta\theta_{k+1} = A^{\dagger}\Big[B(\hat{\psi}_k) \quad \tfrac{d}{d\psi_k}\Big(B(\hat{\psi}_k)\theta_k\Big)\Big]\begin{bmatrix}\delta\theta_k \\ \delta\psi_k\end{bmatrix} + w_k^{\theta}, \tag{4.22a}$$

where

$$\delta\psi_k = \psi_k - \hat{\psi}_k. \tag{4.22b}$$

However, instead of expressing the error development in terms of $\delta\psi_k$, we would like to express it in terms of the orientation error ϵ_k, velocity error δv_k^{n}, accelerometer bias estimation error δo_k^a, and gyroscope bias estimation error δo_k^ω. To do this, note that from (4.9b) we have

$$\Delta\hat{p}_k^{\mathrm{b}_k} = \hat{R}_{\mathrm{n}}^{\mathrm{b}_k}\Delta T(\hat{v}_k^{\mathrm{n}} + \mathrm{g}^{\mathrm{n}}\Delta T/2) + \hat{s}_k t_s^2/2, \tag{4.23a}$$

where

$$\hat{R}_{\mathrm{n}}^{\mathrm{b}_k} = (I_3 - [\epsilon_k]^\wedge)^{-1} R_{\mathrm{n}}^{\mathrm{b}_k}, \tag{4.23b}$$

which gives that

$$\begin{aligned} \delta\Delta p_k &= \Delta p_k^{\mathrm{b}_k} - \Delta\hat{p}_k^{\mathrm{b}_k} \\ &\approx -[\epsilon_k]^\wedge \hat{R}_{\mathrm{n}}^{\mathrm{b}_k}\Delta T(\hat{v}_k^{\mathrm{n}} + \mathrm{g}^{\mathrm{n}}\Delta T/2) + \hat{R}_{\mathrm{n}}^{\mathrm{b}_k}\delta v_k^{\mathrm{n}}\Delta T \\ &= [\underbrace{\hat{R}_{\mathrm{n}}^{\mathrm{b}_k}\Delta T(\hat{v}_k^{\mathrm{n}} + \mathrm{g}^{\mathrm{n}}\Delta T/2)}_{\eta(\hat{R}_{\mathrm{n}}^{\mathrm{b}_k},\hat{v}_k^{\mathrm{n}})}]^\wedge \epsilon_k + \hat{R}_{\mathrm{n}}^{\mathrm{b}_k}\delta v_k^{\mathrm{n}}\Delta T. \end{aligned} \tag{4.24}$$

Here, the second and higher-order error terms have been neglected. Moreover, it holds that

$$\delta\phi_k = \Delta\phi_k - \Delta\hat{\phi}_k = -(\delta o_k^\omega + w_k^\omega)\Delta T. \tag{4.25}$$

Bringing it all together gives the following expression for the magnetic field subsystem error state propagation

$$\delta\theta_{k+1} = F_k^\theta \delta\hat{x}_k + G_k^\theta \begin{bmatrix} w_k^{\text{ins}} \\ w_k^\theta \end{bmatrix}, \tag{4.26a}$$

where

$$F_k^\theta = A^\dagger \begin{bmatrix} B(\hat{\psi}_k) & J_1 & J_2 \end{bmatrix} C(\hat{x}_k), \tag{4.26b}$$

$$G_k^\theta = \begin{bmatrix} 0 & -A^\dagger J_2 \Delta T & 0 & 0 & I_\kappa \end{bmatrix}. \tag{4.26c}$$

Here,

$$J_1 = \frac{d}{d\Delta p_k}\Big(B(\hat{\psi}_k)\theta_k\Big), \tag{4.26d}$$

$$J_2 = \frac{d}{d\Delta\phi_k}\Big(B(\hat{\psi}_k)\theta_k\Big), \tag{4.26e}$$

and

$$C(\hat{x}_k) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & I_\kappa \\ 0 & \hat{R}_{\mathrm{n}}^{\mathrm{b}_k}\Delta T & [\eta(\hat{R}_{\mathrm{n}}^{\mathrm{b}_k}, \hat{v}_k^{\mathrm{n}})]^\wedge & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -I_3\Delta T & 0 \end{bmatrix}. \tag{4.26f}$$

Combining (4.21) and (4.26) gives the complete state-space model of the error state, i.e.,

$$\delta x_{k+1} = F_k \delta x_k + G_k w_k, \tag{4.27a}$$

$$\delta y_k = H_{\delta x} \delta x_k + e_k, \tag{4.27b}$$

where

$$F_k = \begin{bmatrix} F_k^{\text{ins}} & 0 \\ & F_k^{\theta} \end{bmatrix}, \quad G_k = \begin{bmatrix} G_k^{\text{ins}} & 0 \\ & G_k^{\theta} \end{bmatrix}, \tag{4.27c}$$

$$H_{\delta x} = \begin{bmatrix} 0_{3\times 15} & \Phi(r_{m_1}) \\ \vdots & \vdots \\ 0_{3\times 15} & \Phi(r_{m_N}) \end{bmatrix}. \tag{4.27d}$$

Here $\delta y_k \triangleq y_k - H\hat{x}_k$.

The ESKF can then be applied to (4.27) to estimate state, the detailed algorithm is listed in Algorithm 4.

4.3.4 Adaption of the Measurement Noise Covariance

As previously mentioned, the polynomial magnetic field model is not perfect. The model imperfections will vary with the complexity of the magnetic field and the covariance R_k should vary accordingly. One possibility to make R_k adapt to the complexity of the field is to assume $R_k = \sigma_k^2 I_{3N}$ and then fit the magnetic-field model to the current observations y_k and estimate σ_k^2 from the residual. That is, σ_k^2 is estimated as [33]

$$\hat{\sigma}_k^2 = \frac{1}{3N} \|(I_{3N} - XX^{\dagger}) y_k\|^2, \tag{4.28a}$$

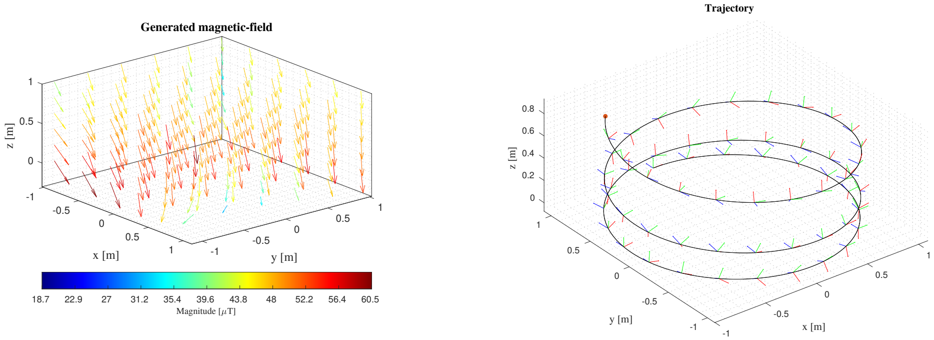
where X is given by

$$X = \begin{bmatrix} \Phi(r_{m_1}) \\ \vdots \\ \Phi(r_{m_N}) \end{bmatrix}. \tag{4.28b}$$

Note that this adaption was not used in the following simulations but in the real-world experiments. The reason is that real-world magnetic fields are more complex and difficult to be represented by the polynomial model.

4.4 Evalutation

To evaluate the performance of the magnetic field-aided INS, Monte Carlo simulations were conducted.



(a) *Illustration of the magnetic field used in the simulations. The field model is based on real measurements to which a high-order dipole model has been fitted.*

(b) *The orientation of the body frame and trajectory used in the simulations. The red dot indicates the end of the trajectory (black line) and the blue, red, and green lines indicate the x-, y-, and z- coordinates axes of the navigation platform.*

Figure 4.4: *The navigation state error and uncertainty.*

4.4.1 Simulation Setup and Evaluation Metrics

The magnetic field data was collected using a magnetometer within a volume of approximately 4 m^3 in the room shown in Figure 4.2. A high-order dipole reference model was then fitted to the data. The field of the resulting reference model is shown in Figure 4.4a. Using the reference model, 1000 Monte Carlo simulations where a sensor array moved along a 60-second spiral trajectory through the field, were conducted. During the spiral motion, the Euler angles representing body frame orientation were changing at a constant rate. The body frame pose trajectory is shown in Figure 4.4b. The geometry of the simulated array was similar to the one shown in Figure 1.1. That is, the array consisted of 30 magnetometers placed in a 6 times 5 grid with 64 mm and 55 mm spacing in the x- and y-axis directions, respectively. For comparison, during the first 20 seconds position measurements were also provided, making the system a position and magnetic field-aided INS. Then the position measurements were removed, leaving only magnetometer array measurements available for the last 40 seconds. The settings used in the simulation are summarized in Table 4.1.

The *root mean square error* (RMSE) of the estimates and *perceived* uncertainty

Table 4.1: *Parameter settings used in the simulations*

Parameter	Description	Value	Unit
Σ_a	accelerometer noise covariance	$0.05^2 I_3$	m/s^2
Σ_ω	gyroscope noise covariance	$0.1^2 I_3$	°/s
Σ_{o^a}	accelerometer bias noise covariance	$(10^{-8})^2 I_3$	m/s^2
Σ_{o^ω}	gyroscope bias noise covariance	$(10^{-8})^2 I_3$	°/s
Σ_θ	coefficient noise covariance	manually tuned	-
$b^{(s)}$	initial accelerometer bias	$\mathcal{N}(0, 0.1^2 I_3)$	m/s^2
$b^{(\omega)}$	initial gyroscope bias	$\mathcal{N}(0, 0.05^2 I_3)$	°/s
σ_e	st.d. of magnetometer noise	0.01	μT
σ_p	st.d. of position measurement noise	0.01	m
p_0^n	initial position	$[0, 1, 0]^T$	m
v_0^n	initial velocity	$[1, 0, 0]^T$	m/s
q_0	initial orientation	$[1, 0, 0, 0]^T$	-
o_0^a	initial accelerometer bias estimate	$[0, 0, 0]^T$	m/s^2
o_0^ω	initial gyroscope bias estimate	$[0, 0, 0]^T$	°/s
θ_0	initial coefficients estimate	$X^\dagger y_0$	-
l	the order of polynomial model	2	-

is calculated as

$$\begin{aligned} \mathrm{RMSE} &= \sqrt{\frac{1}{N_\mathrm{sim}} \sum_{i=1}^{N_\mathrm{sim}} \|\hat{\rho}_{k,i} - \rho_{k,i}\|^2}, \\ \textit{perceived}\ \text{uncertainty} &= \sqrt{\frac{1}{N_\mathrm{sim}} \sum_{i=1}^{N_\mathrm{sim}} P_{k,i}^{\rho}}, \end{aligned} \tag{4.29}$$

Here ρ denotes the quantity for which RMSE is computed, N_sim is the number of simulations, and the subscript k and i denote the timestep and index of the simulation, respectively. Furthermore, $P_{k,i}^{\rho}$ denotes the covariance of the posterior estimate of the quantity ρ. Lastly, the *average normalized estimation error squared* (ANEES) was also calculated as in [34].

4.4.2 Simulation Results and Discussion

The results from the Monte Carlo simulations are shown in Figure 4.5–4.8. From the figures, the following can be observed. Firstly, during the first 20 seconds, the position error and associated uncertainty are very small, which is expected as position measurements were provided during this period. However, when the position aiding is removed after 20 seconds the position error grows much slower for the magnetic field-aided INS than for the stand-alone INS. This confirms the effectiveness of the proposed method. The reduction in the position error growth rate is from cubic to linear in time. Secondly, the velocity and orientation RMSEs

Table 4.2: *Information about the datasets*

Data	Length*(m)	Duration*(s)	Avg. height (m)	Board orientation
LP-1	138.72	272	0.49	parallel
LP-2	167.07	286	0.52	parallel
LP-3	194.41	332	0.55	parallel
NP-1	136.23	177	0.85	parallel
NP-2	134.66	165	0.84	parallel
NP-3	137.76	154	0.79	parallel
NT-1	164.62	185	0.73	tilted
NT-2	137.87	151	0.74	tilted

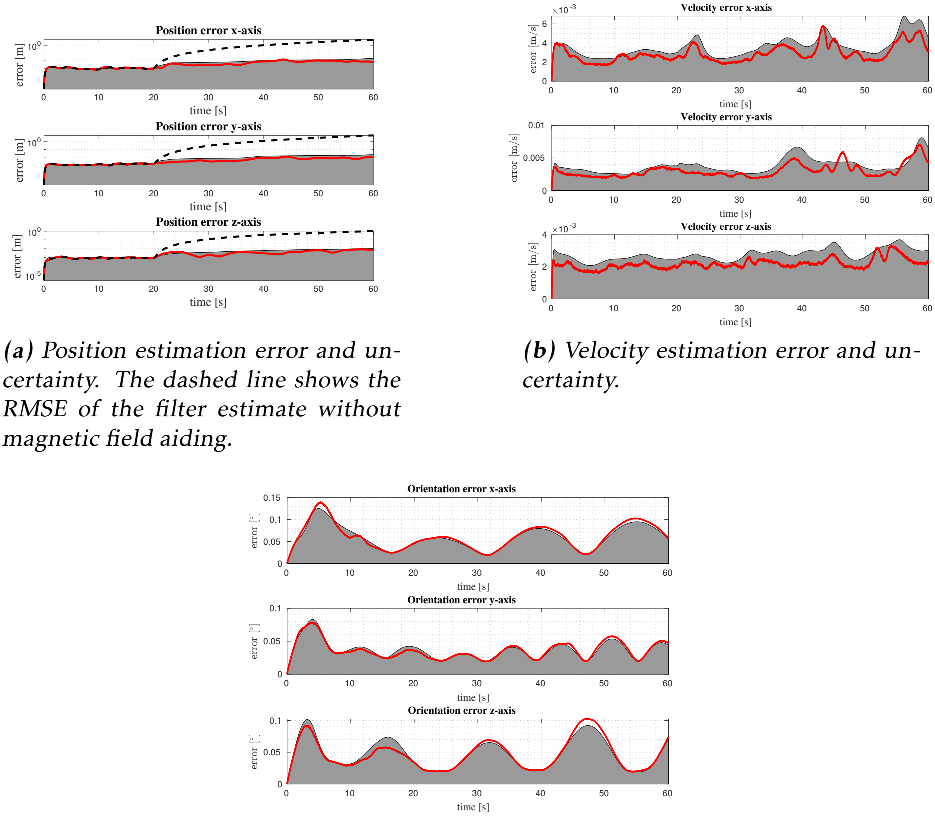
* Including the initial part of the trajectory where the position-aiding is turned on.
LP: low height and parallel NP: normal height and parallel NT: normal height and tilted.

are consistent with the perceived uncertainty, and all sensor bias RMSEs converge in the end. Thirdly, the plot of polynomial coefficient error also suggests that the second-order polynomial model is sufficient to model the magnetic field. Finally, the calculated ANEES remains below the lower bound of the 99% interval except for the transience in the beginning, indicating the filter is conservative. One possible cause is that the process noise covariance, which corresponds to the random changes to polynomial model coefficients, was not tuned well enough to match the true changes. Currently, the process noise of the polynomial model coefficients is assumed to be white and the covariance matrix fixed, the variance of which is set to approximately (by visual inspection) the variance of the prediction error given true coefficients at two consecutive timesteps. In practice, model error grows along with the displacement between two timesteps and there may exist a correlation in process noise, hence, it may need to be adjusted accordingly.

4.4.3 Real-world Experiment Setup

The data collection was in the Visionen arena at Linköping University, see Figure 4.2. The Visionen arena is a large modern research arena with an indoor positioning system with millimeter-level accuracy. It provided the ground truth for evaluation, i.e., the position and rotation of the sensor array w.r.t. the navigation frame.

To evaluate the proposed method multiple experiments were conducted. In each experiment, the magnetometer array was first sitting still on the ground for a few seconds and then picked up by a person. The person then held it in his/her hands parallel to the ground or slightly tilted, see in Figure 4.9, and walked in squares for a few laps before putting the board back on the ground. The true trajectory of the array was measured using a camera-based motion-tracking system. In total 8 datasets were recorded. The main characteristics of the different datasets are summarized in Table 4.2.

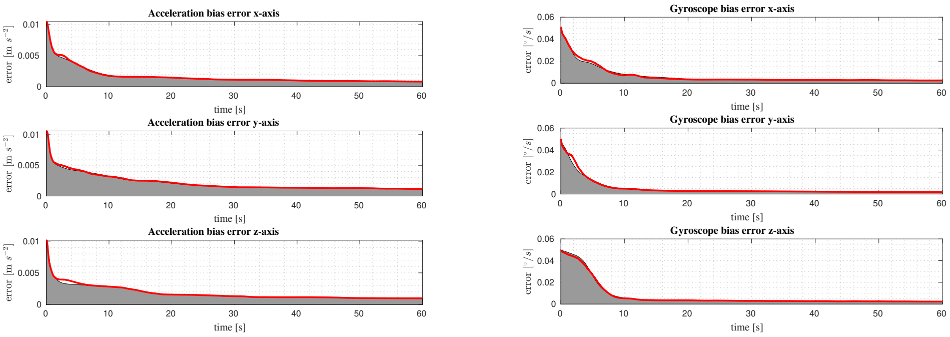


***(a)** Position estimation error and uncertainty. The dashed line shows the RMSE of the filter estimate without magnetic field aiding.*

***(b)** Velocity estimation error and uncertainty.*

***(c)** Orientation estimation error and uncertainty.*

***Figure 4.5:** The navigation state error and uncertainty. The red line shows the RMSE of the filter estimate with magnetic-field aiding, and the gray area shows the perceived uncertainty of the filter.*



(a) *Accelerometer bias estimation error and uncertainty.*

(b) *Gyroscope bias estimation error.*

Figure 4.6: *The IMU bias state error and uncertainty. The red line shows the RMSE of the filter estimate, and the gray area shows the perceived uncertainty of the filter.*

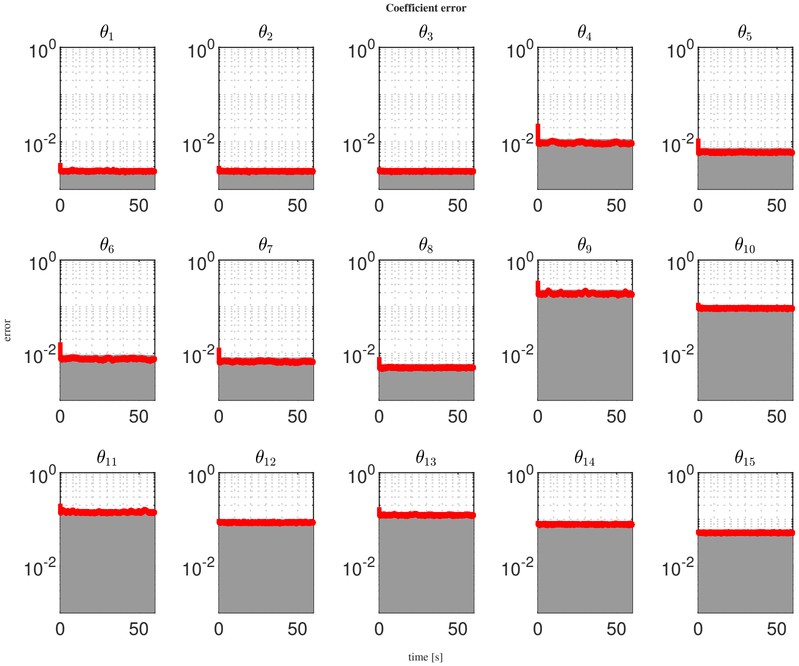


Figure 4.7: *Coefficient estimation error and uncertainty. The red line shows the RMSE of the filter estimate and the gray area shows the perceived uncertainty of the filter. The reference values of θ_k were calculated by fitting the model (4.4) to the generated field.*

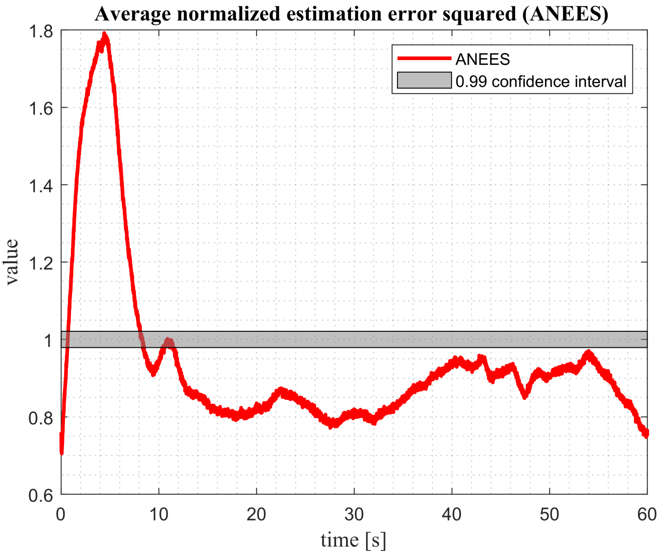
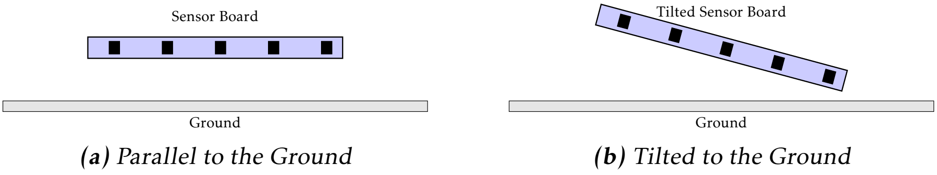


Figure 4.8: *Average normalized estimation error squared.*



*(**a**) Parallel to the Ground* *(**b**) Tilted to the Ground*

Figure 4.9: *Sensor board orientation*

4.4.4 Real-world Experiment Results and Discussion

The datasets were processed with three algorithms: a stand-alone INS; the proposed *magnetic field-aided inertial navigation system* (MAINS) algorithm, and the method proposed in [35]. The positions measured by the motion-tracking system were first made available to all algorithms for 60 seconds to calibrate IMU biases and stabilize state estimates. Then all systems operated without position aiding for the rest of the trajectory. This is similar to the scenario of a user coming into a building where GPS signals are lost. Since the method in [35] by default is designed to use 5 magnetometers, the same sensor configuration (the left in Figure 4.10) as in [35] was used when comparing the two algorithms; only the performance during the non-position-aiding part of the trajectory was evaluated. An example of the estimated trajectories estimated by the three algorithms and the corresponding positional errors are plotted in Figure 4.11. Since the MAINS algorithm supports using other sensor configurations than the square configuration, the performance of the proposed algorithm was also evaluated using all the sensors in the array (see Figure 4.10). An example of the trajectory estimated when using all sensors is shown in Figure 4.12. The results, in terms of root mean square (RMS) position and velocity errors, from processing all 8 datasets with the different sensor configurations and algorithms are summarized in Table 4.3. The position errors at the end of the trajectories are also shown.

It can be seen from Figure 4.11 that both the MAINS and the method in [35] output a trajectory with a similar shape as the true trajectory, while the INS trajectory quickly drifted away. The same conclusion can be drawn from the horizontal and vertical error plots. As expected the position error of the INS grows much faster than those of the other two methods. Looking at the results in Table 4.3, it shows that both the MAINS and the method [35] achieved superior performance in terms of horizontal error, vertical error, and speed error, compared to the stand-alone INS. However, the MAINS has a consistently lower speed error than the method [35] on all datasets. Furthermore, the MAINS has, in general, a lower average horizontal error, which is consistent with the observation of the trajectory shown in Figure 4.11. In terms of vertical error, the method [35] performed poorly on the datasets where the board was tilted, while the MAINS had larger errors on the datasets where the board was close to the ground. The reason for [35] performing worse is that when the board is tilted (so is the body frame), the speed errors in all three axes contributed to a larger vertical error, compared to the case when the board is flat and the vertical error comes mostly from the speed error in the z-axis. Meanwhile, the reason why the MAINS produced trajectories with a large vertical drift when the board was close to the ground is that the magnetic field there was too complex for the polynomial model, which resulted in large fitting residuals and thus large innovations in filtering pulling the estimate away from what it should be. Comparing the performance of the MAINS algorithm with the two different sensor configurations, the benefit of using more sensors is apparent — Vertical error was significantly reduced, and both the horizontal and vertical error at the end of the trajectory were less than 3 meters for most trajectories.

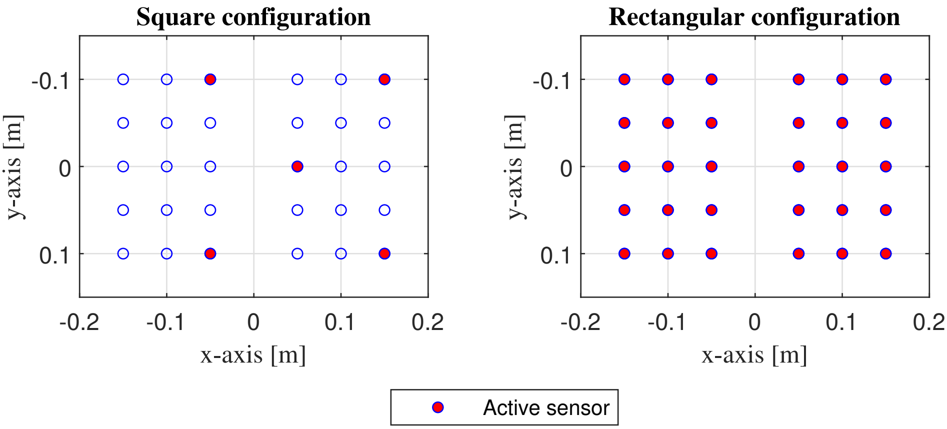


Figure 4.10: *Sensor configurations used in the experiments. Left: Square configuration. Right: Rectangular configuration.*

To help readers better understand the magnetic field in which the experiments were conducted and the full potential of the MAINS algorithm, the trajectory estimated by the MAINS algorithm with rectangular sensor configuration is plotted on top of a magnetic field magnitude plot, as shown in Figure 4.12. It can be seen that the magnitude variance along the trajectory is around 8 μT, and the gradient varies. The MAINS is capable of producing a trajectory that is very close to the true one, and more importantly, the positional error is consistently reflected by the uncertainty. One thing that may raise readers' interest is why the trajectory seems to always "bend inwards" near the top left corner. We cannot offer a precise explanation now, but one of the possibilities is that the MAINS is sensitive to errors in magnetometer calibration parameters, which are difficult to eliminate.

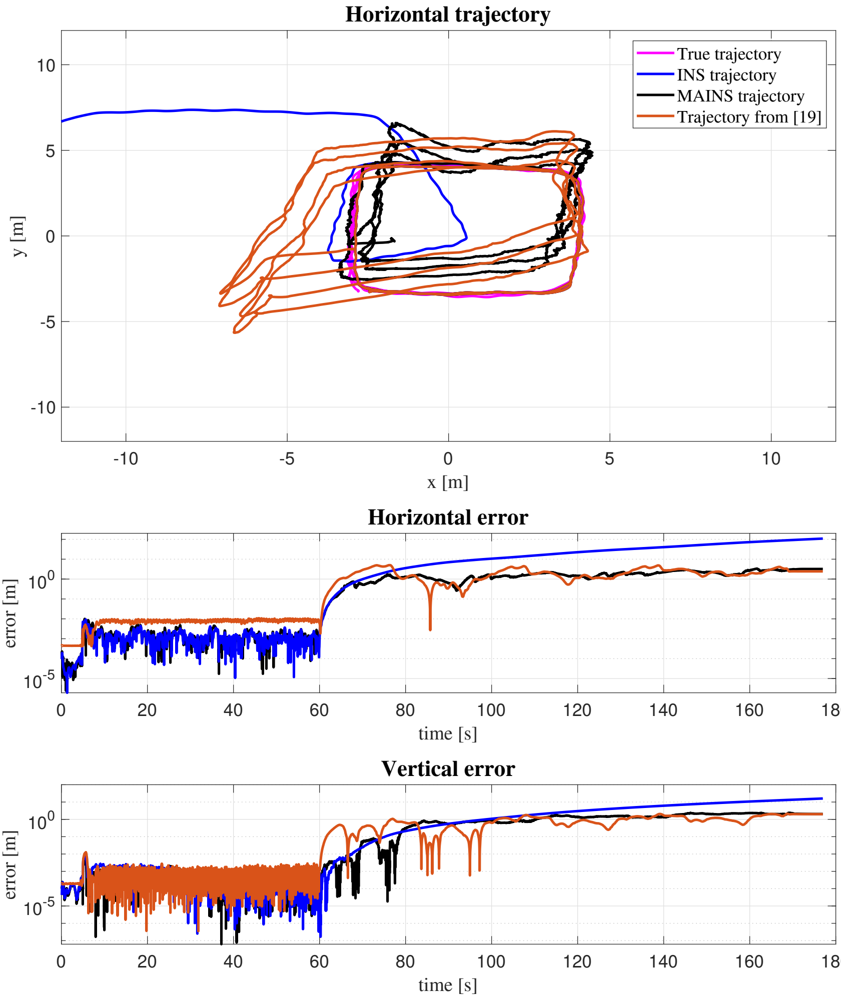


Figure 4.11: *Estimated trajectory and the corresponding positional errors from a stand-alone INS, the MAINS algorithm, and the method proposed in [35]. The square sensor configuration was used in this experiment.*

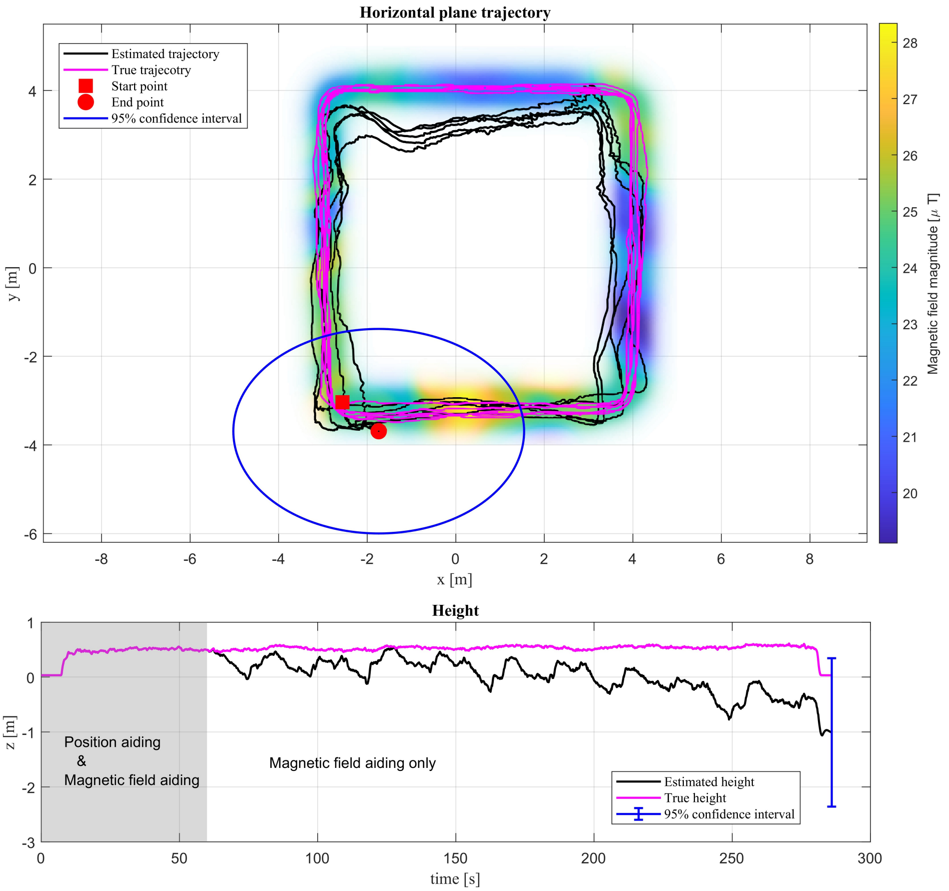


Figure 4.12: *Illustration of the estimated and the true trajectory, as well as the magnetic field magnitude along the trajectory. The magnetic field magnitude plot is created by the interpolated magnetic field measurements using a Gaussian process model.*

Table 4.3: *Summary of results from experimental evaluation*

	LP-1	**LP-2**	**LP-3**	**NP-1**	**NP-2**	**NP-3**	**NT-1**	**NT-2**
Trajectory length*(m)	114.14	139.87	162.30	93.68	89.47	86.43	123.01	87.26
Trajectory duration*(s)	212	226	272	117	105	94	125	91
Stand-alone INS								
RMS Horizontal Error (m)	45.18	140.04	384.41	43.92	28.14	36.0	130.27	36.91
Horizontal Error at the end (m)	75.48	346.84	1090.47	108.53	77.02	84.27	300.99	90.84
RMS Vertical Error (m)	87.13	78.45	110.54	6.48	10.77	5.08	5.72	10.76
Vertical Error at the end (m)	206.12	183.6	256.96	16.02	24.73	12.07	11.09	25.9
RMS Speed Error (m/s)	1.84	2.44	5.96	1.29	1.06	1.09	2.86	1.3
The method [35] with the **square** sensor configuration in Fig. 4.10								
RMS Horizontal Error (m)	2.16	2.61	3.02	2.38	3.88	1.43	4.3	1.18
Horizontal Error at the end (m)	3.89	4.42	5.15	2.44	5.79	1.56	7.15	0.98
RMS Vertical Error (m)	0.96	1.49	2.4	1.09	2.03	0.74	12.24	6.74
Vertical Error at the end (m)	2.0	2.73	4.25	2.0	3.39	0.68	20.6	10.96
RMS Speed Error (m/s)	0.35	0.51	0.58	0.63	0.51	0.56	0.49	0.50
The MAINS with the **square** sensor configuration in Fig. 4.10								
RMS Horizontal Error (m)	0.66	1.29	1.6	1.88	1.62	1.89	2.91	0.81
Horizontal Error at the end (m)	1.25	1.82	2.55	3.22	2.65	2.89	4.51	1.04
RMS Vertical Error (m)	5.59	6.32	6.87	1.42	0.58	1.16	1.27	0.74
Vertical Error at the end(m)	9.61	10.54	11.76	2.04	1.21	1.16	2.22	1.42
RMS Speed Error (m/s)	0.12	0.17	0.14	0.24	0.17	0.29	0.22	0.20
The MAINS with the **rectangular** sensor configuration in Fig. 4.10								
RMS Horizontal Error (m)	0.77	0.84	1.15	2.65	1.24	2.32	0.83	1.73
Horizontal Error at the end (m)	0.61	1.65	2.49	4.21	1.49	3.72	1.82	2.38
RMS Vertical Error (m)	0.49	0.57	0.74	2.09	0.35	2.11	1.91	0.25
Vertical Error at the end (m)	0.56	0.99	1.22	2.78	0.89	2.52	3.16	0.64
RMS Speed Error (m/s)	0.10	0.11	0.12	0.20	0.24	0.24	0.19	0.17

* excluding the initial part of the trajectory where the position-aiding is turned on.

Appendix

4.A The Explicit Form of $\Phi(r)$

The 1st order $\Phi(r)$ is given by

$$\Phi(r) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & r_z & r_y & 2r_x \\ 0 & 1 & 0 & r_z & 2r_y & 0 & r_x & 0 \\ 1 & 0 & 0 & r_y & -2r_z & r_x & 0 & -2r_z \end{bmatrix}. \tag{4.30}$$

The 2nd order $\Phi(r)$ is given by

$$\Phi(r) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & r_z & r_y & 2r_x & 0 & 0 & r_y r_z & r_y^2 - r_z^2 & 2r_x r_z & 2r_x r_y & 3r_x^2 - 3r_z^2 \\ 0 & 1 & 0 & r_z & 2r_y & 0 & r_x & 0 & 2r_y r_z & 3r_y^2 - 3r_z^2 & r_x r_z & 2r_x r_y & 0 & r_x^2 - r_z^2 & 0 \\ 1 & 0 & 0 & r_y & -2r_z & r_x & 0 & -2r_z & r_y^2 - r_z^2 & -6r_y r_z & r_x r_y & -2r_x r_z & r_x^2 - r_z^2 & -2r_y r_z & -6r_x r_z \end{bmatrix}. \tag{4.31}$$

5

The Observability-Constrained Magnetic Field-Aided Inertial Navigation System

A consistent estimate of position and orientation uncertainty, i.e., the perceived uncertainty that truly reflects the error in the estimate, is essential for localization, especially when the localization result should fuse with information from other systems or be used in control or decision systems. This chapter discusses the inconsistent perceived yaw uncertainty by the magnetic field-aided INS presented in Chapter 4. To overcome this inconsistency an observability-constrained magnetic field-aided INS is presented.

This chapter is an adaption of the material in

> Chuan Huang, Gustaf Hendeby, and Isaac Skog. An observability-constrained magnetic field-aided inertial navigation system — extended version. *arXiv preprint arXiv:2406.02161*, abs/2406.02161, 2024. (Accepted to IPIN 2024)

5.1 Inconsistent Perceived Uncertainty in Yaw

In an odometry-aided INS, such as a magnetic field odometry-aided INS [1, 2, 35–38], the uncertainty about the position and yaw of the navigation platform can, if the errors in the initial state are uncorrelated, never become smaller than the initial position and yaw uncertainty. This is because odometry and inertial measurements only provide relative motion information. Hence, for any algorithm used to estimate the navigation state x_k in an odometry-aided INS it should hold that

$$P_k^{\mathrm{p}} \geq P_0^{\mathrm{p}} \quad \text{and} \quad P_k^{\phi} \geq P_0^{\phi}, \tag{5.1}$$

where P_k^{p} and P_k^{ϕ} denote the covariance of the, at time k, estimated position and yaw states, respectively.

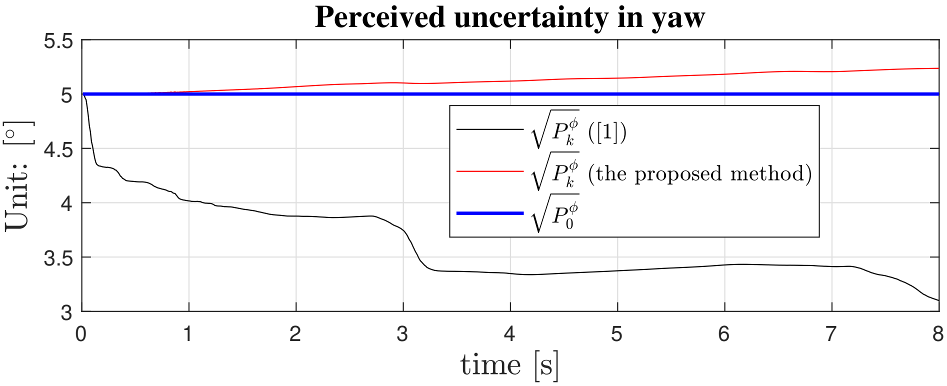


Figure 5.1: *Example of the perceived yaw uncertainty calculated by the EKF used to realize magnetic field-odometry-aided INS in [2]. Also shown is the perceived yaw uncertainty of the proposed observability-constrained EKF algorithm.*

However, the perceived yaw uncertainty in the MAINS will fall below the initial yaw uncertainty. In Figure 5.1, where the square roots of P_0^ϕ and P_k^ϕ as calculated by the EKF in the MAINS are shown. As seen from the figure the square root of P_k^ϕ falls below that of P_0^ϕ. Hence the inequality in (5.1) does not hold and the covariance estimate of the EKF is inconsistent. The inconsistency is because the EKF linearizes the system model around the estimated state, which causes the yaw to be perceived as observable even though it is not. This type of inconsistency effect has been observed in multiple EKF-based implementations of odometry-aided INS, see e.g., [39–41], and is troublesome if the yaw information should be fused with information from other systems or be used in a control or decision system. Next, the observability-constrained EKF [40], which aims to fix this inconsistency, is discussed.

5.2 Preserve Observability Properties

The behavior of odometry-aided INS is commonly described by a nonlinear state-space model of the following form

$$\begin{aligned} x_{k+1} &= f(x_k, u_k, w_k), & x &\in \mathbb{R}^{n_x}, \\ y_k &= h(x_k) + e_k, & y &\in \mathbb{R}^{n_y}. \end{aligned} \tag{5.2}$$

Here f and h are the nonlinear functions, and u_k denotes the control input. Further, w_k and e_k denote the process and measurement noise, respectively. They are assumed to be white noise with covariance Q_k and R_k, respectively.

Several approaches have been proposed to preserve observability properties of the model (5.2) in EKF filtering, see [39, 40, 42–45], one of which is called the observability-constrained EKF [40]. The basic idea of the method in [40] is to modify the Jacobians used in the EKF, such that the basis of the unobservable subspace, evaluated at state estimates, lies in the nullspace of the observability matrix evaluated at state estimates of EKF. The unobservable subspace is a subspace of the nullspace of the local observability matrix [30] associated with (5.2), i.e. the observability matrix constructed as

$$\bar{\mathcal{O}}_k \triangleq \begin{bmatrix} \bar{H}_k \\ \bar{H}_{k+1}\bar{\Phi}(k+1,k) \\ \vdots \\ \bar{H}_{k+n_x-1}\bar{\Phi}(k+n_x-1,k) \end{bmatrix}. \tag{5.3a}$$

Here $\bar{x}_{k:k+n_x-1}$ denotes the nominal trajectory, which is the solution of (5.2) with the control sequence $\bar{u}_{k:k+n_x-1}$ and the process noise turned off. Further,

$$\bar{\Phi}(k+i,k) = \bar{F}_{k+i-1}\bar{F}_{k+i-2}\cdots\bar{F}_k, \tag{5.3b}$$

where

$$\bar{H}_k = \left.\frac{\partial h}{\partial x_k}\right|_{x_k=\bar{x}_k} \quad \text{and} \quad \bar{F}_k = \left.\frac{\partial f}{\partial x_k}\right|_{\substack{x_k=\bar{x}_k \\ u_k=\bar{u}_k \\ w_k=0}}. \tag{5.3c}$$

The unobservable subspace $\mathcal{N}_k$ can be represented as

$$\mathcal{N}_k = \operatorname{span}\{N(\bar{x}_k)\}, \tag{5.4a}$$

where $N : \mathbb{R}^n \to \mathbb{R}^{n\times p}$ is a matrix valued function such that $\bar{\mathcal{O}}_k N(\bar{x}_k) = 0$, and p denotes the dimension of the unobservable subspace. That is, the columns of $N(\bar{x}_k)$ are basis vectors that span the unobservable subspace. The basis of the unobservable subspace is not unique. Hence, right multiplying $N(\bar{x}_k)$ with any full-rank matrix $\mathcal{E}_k \in \mathbb{R}^{p\times p}$ will not change the span of the basis. That is,

$$\operatorname{span}\{N^{\ddagger}(\bar{x}_k)\} = \operatorname{span}\{N(\bar{x}_k)\}, \tag{5.5a}$$

where

$$N^{\ddagger} : \quad \mathbb{R}^n \to \mathbb{R}^{n\times p} \quad \text{and} \quad N^{\ddagger}(x_k) = N(x_k)\mathcal{E}_k. \tag{5.5b}$$

This property, which was not used in [40], will be important when modifying the Jacobians, as it reduces the changes made to the Jacobians to preserve the observability properties.

Once the basis vectors of the unobservable subspace are determined, they are, as will be described next, used to modify the Jacobians used in the EKF. Let

$$\hat{F}_k = \left.\frac{\partial f}{\partial x_k}\right|_{\substack{x_k=\hat{x}_{k|k} \\ u_k=\hat{u}_k \\ w_k=0}} \quad \text{and} \quad \hat{H}_k = \left.\frac{\partial h}{\partial x_k}\right|_{x_k=\hat{x}_{k|k-1}} \tag{5.6}$$

be the Jacobians used in the unmodified EKF. Here $\hat{x}_{k|k}$ and $\hat{x}_{k|k-1}$ denote the posterior and prior estimate of the state, i.e., the estimate of x_k given measurements up to time k and $k-1$, respectively. These Jacobians are modified by making the

Algorithm 5 Observability-constrained EKF

```
Input: $\{\bar{u}_k, y_k\}_{k=1}^{L}$
Output: $\{\hat{x}_{k|k}, P_{k|k}\}_{k=1}^{L}$
  Initialisation : estimated state $\hat{x}_{1|0}$, covariance matrix $P_{1|0}$
  For $k = 1$ to $L$ do
      Measurement update:
      Calculate $\hat{H}_k$ using (5.6) and find $\tilde{H}_k^*$ by solving (5.7b)
      $S_k = \tilde{H}_k^* P_{k|k-1} (\tilde{H}_k^*)^\top + R_k$
      $K_k = P_{k|k-1} (\tilde{H}_k^*)^\top S_k^{-1}$
      $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (y_k - h(\hat{x}_k))$
      $P_{k|k} = P_{k|k-1} - K_k \tilde{H}_k^* P_{k|k-1}$
      Time update:
      $\hat{x}_{k+1|k} = f(\hat{x}_{k|k}, \bar{u}_k, 0)$
      Calculate $\hat{F}_k$ using (5.6) and find $\tilde{F}_k^*$ by solving (5.7a)
      $G_k = \frac{\partial f}{\partial w_k}\Big|_{x_k=\hat{x}_{k|k}, u_k=\bar{u}_k, w_k=0}$
      $P_{k+1|k} = \tilde{F}_k^* P_{k|k} (\tilde{F}_k^*)^\top + G_k Q_k G^\top$
  end for
```

smallest possible (in terms of the Frobenius norm) changes to their entries while still preserving the observability properties. This is done by solving the following optimization problems.

$$\begin{aligned} \tilde{F}_k^* &= \arg\min_{\tilde{F}_k} \|\tilde{F}_k - \hat{F}_k\|_{\mathcal{F}}^2 \\ \text{s.t.} \quad \operatorname{span}\{N(\hat{x}_{k+1|k})\} &= \operatorname{span}\{\tilde{F}_k N(\hat{x}_{k|k-1})\}, \end{aligned} \tag{5.7a}$$

and

$$\begin{aligned} \tilde{H}_k^* &= \arg\min_{\tilde{H}_k} \|\tilde{H}_k - \hat{H}_k\|_{\mathcal{F}}^2 \\ \text{s.t.} \quad \tilde{H}_k N(\hat{x}_{k|k-1}) &= 0. \end{aligned} \tag{5.7b}$$

Here $\|\cdot\|_{\mathcal{F}}$ denotes the Frobenius norm. The constraints in (5.7a) and (5.7b) guarantee that the unobservable subspace is preserved and that the unobservable directions cannot be observed. The observability-constrained EKF algorithm is shown in Algorithm 5.

5.3 Application to the Magnetic Field-Aided INS

The method presented in Section 5.2 will next be used to modify the EKF algorithm used in the MAINS to address the inconsistency illustrated in Figure 5.1. Unless otherwise stated, we adhere to the notations and definitions established in previous chapters, particularly those found in Chapter 4.

5.3.1 State-Space Model

Consider the simplified MAINS where IMU biases are not included in the state vector. The IMU biases are excluded as they only complicate the forthcoming observability analysis and biases never improve the observability. Later, when evaluating the proposed method the biases are included.

Let the state vector x_k, without IMU biases, be defined as

$$x_k \triangleq \left[(p_k^{\mathrm{n}})^\top \ (v_k^{\mathrm{n}})^\top \ (q_k)^\top \ (\theta_k)^\top\right]^\top. \tag{5.8}$$

Based on the state-space model (4.17) developed in Chapter 4, the state dynamics of the MAINS excluding the bias states is given by

$$x_{k+1} = f(x_k, u_k, w_k^\theta), \tag{5.9a}$$

where

$$f(x_k, u_k, w_k^\theta) = \begin{bmatrix} p_k^{\mathrm{n}} + v_k^{\mathrm{n}}\Delta T + (R_k s_k + \mathrm{g}^{\mathrm{n}})\frac{\Delta T^2}{2} \\ v_k^{\mathrm{n}} + (R_k s_k + \mathrm{g}^{\mathrm{n}})\Delta T \\ q_k \otimes \mathrm{Exp}_q(\omega_k \Delta T) \\ A^\dagger B(\psi_k)\theta_k + w_k^\theta \end{bmatrix}. \tag{5.9b}$$

Here $u_k \triangleq [s_k^\top \ \omega_k^\top]^\top$ is the system input. $R_k \in SO(3)$ denotes the orientation matrix corresponding to q_k, i.e., the orientation matrix of the b-frame. Furthermore, $\psi_k \in \mathbb{R}^6$ denotes the pose change between time k and $k+1$, i.e.,

$$\psi_k = \begin{bmatrix} R_k^\top \Delta T(v_k^{\mathrm{n}} + \mathrm{g}^{\mathrm{n}}\Delta T/2) + s_k \Delta T^2/2 \\ \omega_k \Delta T \end{bmatrix}, \tag{5.10}$$

where the first and second elements encode the translation and orientation change, respectively.

The measurements are from the magnetometer array, whose equation is given by

$$y_k = H x_k + e_k, \tag{5.11a}$$

where

$$H = \begin{bmatrix} 0_{3\times 10} & H^\theta(r_1) \\ \vdots & \vdots \\ 0_{3\times 10} & H^\theta(r_m) \end{bmatrix}. \tag{5.11b}$$

Here $H^\theta(r_i) \in \mathbb{R}^{3\times\kappa}$ denotes the measurement matrix for magnetometer sensor location r_i, whose expression is given in Appendix 4.A. Further, m denotes the number of magnetometer sensors.

5.3.2 Linearized Error State Model

Following the same linearization procedures as in Chapter 4, let δx_k denote the error state vector

$$\delta x_k \triangleq \left[(\delta p_k^{\mathrm{n}})^\top \ (\delta v_k^{\mathrm{n}})^\top \ (\epsilon_k)^\top \ (\delta\theta_k)^\top\right]^\top. \tag{5.12a}$$

Further, let $\bar{x}_k$ denote the nominal state vector

$$\bar{x}_k \triangleq \left[(\bar{p}_k^{\mathrm{n}})^\top \; (\bar{v}_k^{\mathrm{n}})^\top \; (\bar{q}_k)^\top \; (\bar{\theta}_k)^\top\right]^\top. \tag{5.12b}$$

Given the measurement input $\tilde{u}_k = [\tilde{s}_k^\top \; \tilde{\omega}_k^\top]^\top$, the linearized error state dynamics is given by

$$\delta x_{k+1} = \bar{F}_k \delta x_k + \bar{G}_k w_k, \tag{5.13a}$$

where

$$\bar{F}_k = \begin{bmatrix} I_3 & I_3 \Delta T & 0 & 0 \\ 0 & I_3 & -\bar{R}_k[\tilde{s}_k]^\wedge \Delta T & 0 \\ 0 & 0 & \mathrm{Exp}_R(\tilde{\omega}_k \Delta T)^\top & 0 \\ 0 & A^\dagger \bar{J}_k \bar{R}_k^\top \Delta T & A^\dagger \bar{J}_k [\eta(\bar{R}_k, \bar{v}_k^{\mathrm{n}})]^\wedge & A^\dagger B(\bar{\psi}_k) \end{bmatrix}, \tag{5.13b}$$

$$\bar{\psi}_k = \begin{bmatrix} \bar{R}_k^\top \Delta T (\bar{v}_k^{\mathrm{n}} + \mathrm{g}^{\mathrm{n}} \Delta T/2) \\ \tilde{\omega}_k \Delta T \end{bmatrix}, \tag{5.13c}$$

$$\eta(\bar{R}_k, \bar{v}_k^{\mathrm{n}}) = \bar{R}_k^\top \Delta T (\bar{v}_k^{\mathrm{n}} + \mathrm{g}^{\mathrm{n}} \Delta T/2), \tag{5.13d}$$

$$\bar{J}_k = \left.\frac{\partial B(\psi_k)\theta_k}{\Delta p_k}\right|_{\psi_k = \bar{\psi}_k, \theta_k = \bar{\theta}_k}. \tag{5.13e}$$

Here $\tilde{s}_k \in \mathbb{R}^3$ and $\tilde{\omega}_k \in \mathbb{R}^3$ denote the measured acceleration and angular velocity, respectively. Further, $w_k \triangleq [(w_k^a)^\top \; (w_k^\omega)^\top \; (w_k^\theta)^\top]^\top$ denote the process noise, where $w_k^a \in \mathbb{R}^3$ and $w_k^\omega \in \mathbb{R}^3$ denote the acceleration and angular velocity measurement noise, respectively. The explicit form of $\bar{G}_k$ is not given as it is relevant in the observability analysis, interested readers can find it in Chapter 4.

The corresponding measurement model is given by

$$\delta y_k = H_{\delta x} \delta x_k + e_k, \tag{5.14a}$$

where

$$H_{\delta x} = \begin{bmatrix} 0_{3\times 9} & H^\theta(r_1) \\ \vdots & \vdots \\ 0_{3\times 9} & H^\theta(r_m) \end{bmatrix} \tag{5.14b}$$

and $\delta y_k \triangleq y_k - H\bar{x}_k$.

5.3.3 Unobservable Subspace and Interpretations

Let

$$N(x_k) \triangleq \begin{bmatrix} I_3 & 0_{3\times 1} \\ 0_{3\times 3} & -[v_k^{\mathrm{n}}]^\wedge \mathrm{g}^{\mathrm{n}} \\ 0_{3\times 3} & R_k^\top \mathrm{g}^{\mathrm{n}} \\ 0_{\kappa\times 3} & 0_{\kappa\times 1} \end{bmatrix}, \tag{5.15}$$

then the basis of the unobservable subspace associated with the linearized error state model is given by the column vectors in $N(\bar{x}_k)$. See Appendix 5.A for a proof.

The first three columns of $N(\bar{x}_k)$ correspond to a body frame translation and the last column with the first three corresponds to a navigation frame rotation around the gravity vector. Interested readers can refer to Appendix 5.B to see why they are interpreted as those effects. Note that the unobservable velocity component is caused by the fact that the yaw angle cannot be determined. Thus the direction of the velocity in the navigation frame is ambiguous.

5.3.4 Suggested Modifications of the Jacobians

Since the measurement model (5.14) is linear and H_{δ_x} fulfills the constraint in (5.7b), it is left unmodified, i.e., $\tilde{H}_k^* = H_{\delta_x}$. Only the Jacobian involved in the state transition, i.e.,

$$\hat{F}_k = \begin{bmatrix} I_3 & I_3 \Delta T & 0 & 0 \\ 0 & I_3 & -\hat{R}_k [\tilde{s}_k]^\wedge \Delta T & 0 \\ 0 & 0 & \mathrm{Exp}_R(\tilde{\omega}_k \Delta T)^\top & 0 \\ 0 & A^\dagger \hat{J}_k \hat{R}_k^\top \Delta T & A^\dagger \hat{J}_k \hat{R}_k^\top \Delta T [\hat{v}_k^{\mathrm{n}} + \mathrm{g}^{\mathrm{n}} \frac{\Delta T}{2}]^\wedge \hat{R}_k & \hat{T}_k^{k+1} \end{bmatrix} \tag{5.16}$$

needs to be modified. Here the accent $\hat{\cdot}$ denotes the posterior estimate of the quantity and $\hat{T}_k^{k+1} \triangleq A^\dagger B(\hat{\psi}_k)$. Furthermore, the equality $[R\xi]^\wedge = R[\xi]^\wedge R^\top, R \in SO(3), \xi \in \mathbb{R}^3$ is used.

Solving the optimization problem in (5.7a) is typically difficult, However, a suboptimal solution can be obtained by narrowing the search space for $\tilde{F}_k$ and transforming the constraint by selecting a specific set of transformations $\{\mathcal{E}_k, \mathcal{E}_{k+1}\}$ such that the basis vectors in the constraint are equal, i.e.,

$$N(\hat{x}_{k+1|k})\mathcal{E}_{k+1} = \tilde{F}_k N(\hat{x}_{k|k-1})\mathcal{E}_k. \tag{5.17}$$

Since the goal is to make minimal changes to the original Jacobian $\hat{F}_k$ but still fulfill the constraints in (5.7a), the sub-blocks of matrices $\hat{F}_k$ that are independent of the linearization points are kept unchanged. Furthermore, the sub-block $\hat{T}_k^{k+1}$ will not affect the constraint since the last κ rows of $N(\hat{x}_{k|k-1})$ are all zeros. Hence, it is kept as it is. The remaining subblocks of $\hat{F}_k$ must be modified to meet the constraint in (5.7a). Therefore, the proposed modified Jacobian $\tilde{F}_k$ has the structure

$$\tilde{F}_k = \begin{bmatrix} I_3 & I_3 \Delta T & 0 & 0 \\ 0 & I_3 & \tilde{F}_k^{(1,2)} & 0 \\ 0 & 0 & \tilde{F}_k^{(2,2)} & 0 \\ 0 & A^\dagger \hat{J}_k \hat{R}_k^\top \Delta T & A^\dagger \hat{J}_k \hat{R}_k^\top \Delta T \tilde{F}_k^{(3,2)} & \hat{T}_k^{k+1} \end{bmatrix}, \tag{5.18}$$

where $\tilde{F}_k^{(1,2)}$, $\tilde{F}_k^{(2,2)}$, and $\tilde{F}_k^{(3,2)}$ are the block matrices to be determined. Further, transformations $\{\mathcal{E}_k, \mathcal{E}_{k+1}\}$ are chosen as

$$\mathcal{E}_k = \begin{bmatrix} I_3 & a_k \\ 0 & 1 \end{bmatrix}, \quad \mathcal{E}_{k+1} = \begin{bmatrix} I_3 & a_{k+1} \\ 0 & 1 \end{bmatrix}. \tag{5.19}$$

Here $a_k, a_{k+1} \in \mathbb{R}^3$ are column vectors. Then the optimization problem (5.7a) can be written as

$$
\begin{aligned}
\tilde{F}_k^* &= \arg\min_{\tilde{F}_k} \|\tilde{F}_k - \hat{F}_k\|_{\mathcal{F}}^2 && (5.20a)\\
\text{s.t.} \quad a_{k+1} &= a_k - [\hat{v}^{\mathrm{n}}_{k|k-1}]^{\wedge} \mathrm{g}^{\mathrm{n}} \Delta T, && (5.20b)\\
[\hat{v}^{\mathrm{n}}_{k+1|k}]^{\wedge} \mathrm{g}^{\mathrm{n}} &= [\hat{v}^{\mathrm{n}}_{k|k-1}]^{\wedge} \mathrm{g}^{\mathrm{n}} - \tilde{F}_k^{(1,2)} \hat{R}^{\top}_{k|k-1} \mathrm{g}^{\mathrm{n}}, && (5.20c)\\
\hat{R}^{\top}_{k+1|k} \mathrm{g}^{\mathrm{n}} &= \tilde{F}_k^{(2,2)} \hat{R}^{\top}_{k|k-1} \mathrm{g}^{\mathrm{n}} && (5.20d)\\
A^{\dagger} \hat{J}_k \hat{R}_k^{\top} \Delta T(-[\hat{v}^{\mathrm{n}}_{k|k-1}]^{\wedge} \mathrm{g}^{\mathrm{n}} &+ \tilde{F}_k^{(3,2)} \hat{R}^{\top}_{k|k-1} \mathrm{g}^{\mathrm{n}}) = 0 && (5.20e)
\end{aligned}
$$

These constraints are obtained by matching the entries on both sides of (5.17).

Since the constraints in (5.20) only contain one unique subblock of $\tilde{F}_k$ each, the optimization problem in (5.20) can be split into three separate optimization problems of the form

$$
\begin{aligned}
\tilde{F}^{*(i,2)} &= \arg\min_{\tilde{F}^{(i,2)}} \|\tilde{F}^{(i,2)} - \hat{F}^{(i,2)}\|_{\mathcal{F}}^2,\\
\text{s.t.} \quad \tilde{F}^{(i,2)} u &= w.
\end{aligned} \tag{5.21}
$$

Here $\hat{F}^{(i,2)}$ denotes the matrix to be modified and $u, w \in \mathbb{R}^3$. For example, the constraint (5.20c) corresponds to $u = \hat{R}^{\top}_{k|k-1} \mathrm{g}^{\mathrm{n}}$, $w = [\hat{v}^{\mathrm{n}}_{k|k-1} - \hat{v}^{\mathrm{n}}_{k+1|k}]^{\wedge} \mathrm{g}^{\mathrm{n}}$. The optimization problem has a closed-form solution [40]

$$
\tilde{F}^{*(i,2)} = \hat{F}^{(i,2)} - (\hat{F}^{(i,2)} u - w)(u^{\top} u)^{-1} u^{\top}. \tag{5.22}
$$

Note that in [40], the optimization problem set for $\tilde{F}^{(2,2)}$ is different, where $\tilde{F}^{(2,2)}$ is constrained to be a rotation matrix and the object is to minimize the squared norm of the difference of the quaternions corresponding to the rotation matrices. In this paper, we also adopted this approach. Furthermore, when dealing with (5.20e), we consider the term in the parenthesis to be 0, although in the general case, it can be any vector in the nullspace of $A^{\dagger} \hat{J}_k \hat{R}_k^{\top} \Delta T$.

5.4 Experimental Evaluation

The proposed *observability-constrained magnetic field-aided inertial navigation system* (OC-MAINS) algorithm is compared with the original MAINS algorithm. Both algorithms are evaluated using Monte Carlo simulations and real-world data. As a performance measure, RMSE is used to evaluate position and yaw estimation errors. Furthermore, the perceived uncertainty of the estimate is compared with the RMSE to evaluate consistency. Specifically, the covariance of the yaw in i^{th} simulation at timestep k is calculated as

$$
P^{\phi}_{k,i} = \nabla_{\text{yaw}}(\hat{q}_{k,i})^{\top} P^{\epsilon}_{k,i} \nabla_{\text{yaw}}(\hat{q}_{k,i}). \tag{5.23}
$$

Here $P^{\epsilon}_{k,i} \in \mathbb{R}^{3\times3}$ denote the block covariance matrix corresponding to the orientation error. Furthermore, $\nabla_{\text{yaw}}(\hat{q}_{k,i}) \in \mathbb{R}^{3\times1}$ denotes the gradient w.r.t. the Euler angles of the function that converts $\hat{q}_{k,i}$ to the yaw angle.

5.4.1 Simulation Setup and Result

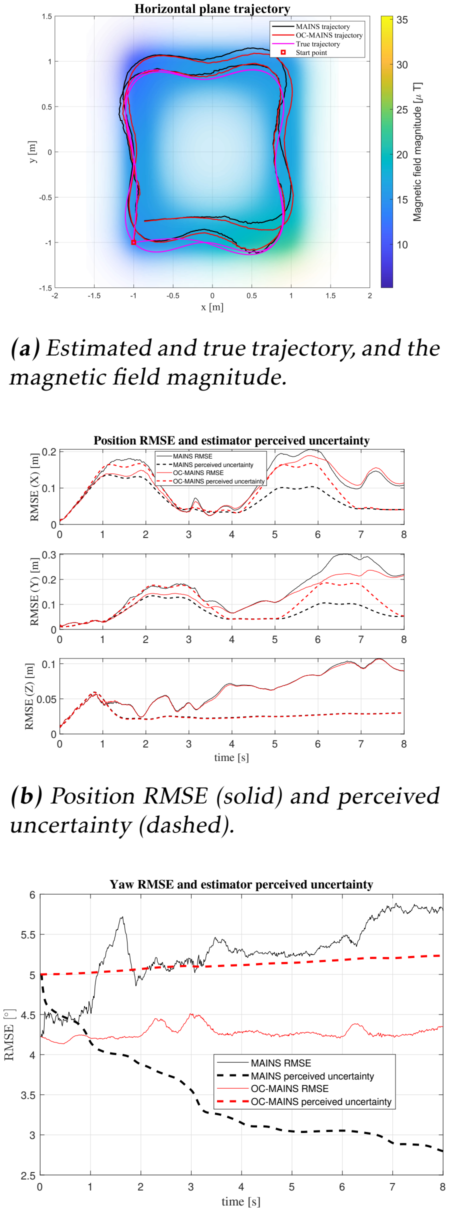
In the simulation, a sensor board, similar to that shown in Figure 1.1, moves in squares of size 2 × 2 meters; see Figure 5.2a. The trajectory's duration was 8 seconds and the data was sampled at 100 Hz. The IMU measurements were generated as the true value corrupted by additive white noise and biases. The magnetometer measurements were generated as the magnetic field from a multi-dipole model corrupted by additive white noise. The magnitude of the simulated magnetic field is shown in Figure 5.2a. In total, 50 simulations with independent noise, bias, and initial state realizations, were used in the Monte Carlo simulation evaluation. The same filter parameters were used in both algorithms.

The results are shown in Figure 5.2b and Figure 5.2c. The figures show that the proposed OC-MAINS algorithm generally has a smaller position and yaw RMSE. Further, the perceived uncertainty of the yaw estimate is more consistent with the true uncertainty, see also in Figure 5.1. Concerning the position estimates, even though the perceived uncertainty of the OC-MAINS is somewhat more consistent with the true uncertainty of the position estimates, significant inconsistency still exists. However, it is worth noting that the perceived uncertainty of OC-MAINS is always higher than the initial uncertainty, thus satisfying the condition in (5.1). The original MAINS algorithm does not meet this condition.

5.4.2 Experimental Setup and Result

In the experiment, a person held the sensor board in Figure 1.1 parallel to the ground, and walked in squares for a few laps. The true trajectory was measured using a camera-based motion-tracking system. The same filter parameters were used in both algorithms.

The results are shown in Figure 5.3b and Figure 5.3c. From Figure 5.3c it can be seen that the yaw error of OC-MAINS is significantly lower than that of the original MAINS. Also, it can be seen that the perceived and true yaw uncertainty agree when using the OC-MAINS algorithm. From Figure 5.3b it can be seen that the position error of the original MAINS and OC-MAINS algorithms are about the same, but the OC-MAINS algorithm has slightly better performance in the y-axis direction. For the OC-MAINS algorithm, the perceived and true uncertainty agrees in the y- and z-axis directions, whereas in the x-axis there is a significant inconsistency. This could result from imperfect IMU calibration. Similar to in the simulations, the perceived uncertainty of OC-MAINS is always higher than the initial uncertainty, thus satisfying the condition in (5.1).

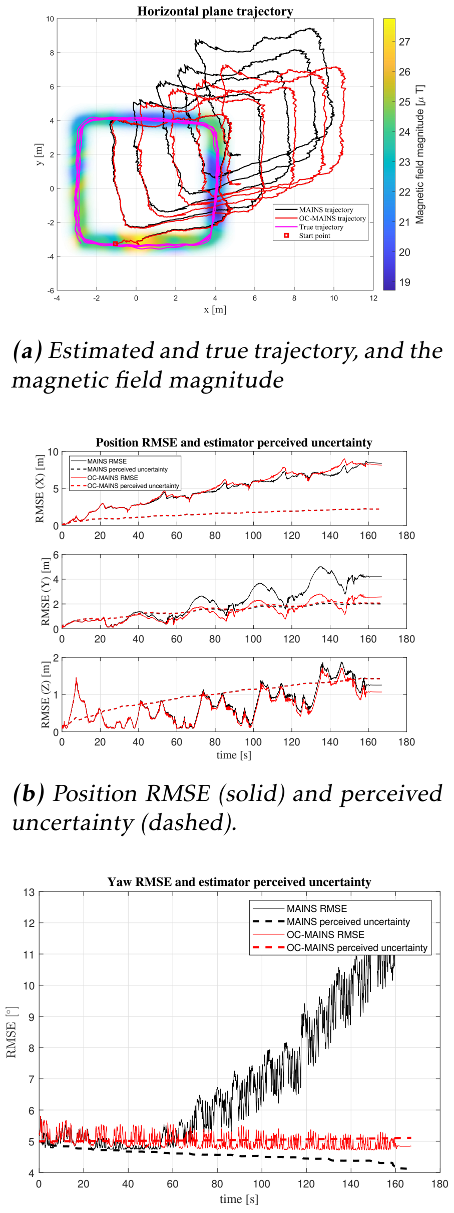


*(**a**) Estimated and true trajectory, and the magnetic field magnitude.*

*(**b**) Position RMSE (solid) and perceived uncertainty (dashed).*

*(**c**) Yaw RMSE (solid) and perceived uncertainty (dashed).*

Figure 5.2: *Monte Carlo simulation results. The RMSE results are the average value calculated from 50 independent simulations.*



*(**a**) Estimated and true trajectory, and the magnetic field magnitude*

*(**b**) Position RMSE (solid) and perceived uncertainty (dashed).*

*(**c**) Yaw RMSE (solid) and perceived uncertainty (dashed).*

Figure 5.3: *Results from real-world experiment. The RMSE results are the average value calculated from using 12 randomly selected initialization states.*

Appendix

5.A The Unobservable Subspace Basis

To derive the unobservable subspace basis for the system defined by (5.13) and (5.14), $\bar{F}_k$ is rewritten. Propagating the nominal state $\bar{x}_k$ with $\tilde{u}_k$ using (5.9b) leads to

$$\bar{v}^{\mathrm{n}}_{k+1} = \bar{v}^{\mathrm{n}}_k + \bar{R}_k \tilde{s}_k \Delta T + \mathrm{g}^{\mathrm{n}} \Delta T, \tag{5.24a}$$

$$\bar{q}_{k+1} = \bar{q}_k \otimes \mathrm{Exp}_q \left(\tilde{\omega}_k \Delta T \right). \tag{5.24b}$$

By rearranging the terms and using the equality $[R\xi]^{\wedge} = R[\xi]^{\wedge}R^{\top}$, where $R \in SO(3)$ and $\xi \in \mathbb{R}^3$, the following holds

$$\bar{R}_k \tilde{s}_k^{\wedge} \Delta T = [\bar{v}^{\mathrm{n}}_{k+1} - \bar{v}^{\mathrm{n}}_k - \mathrm{g}^{\mathrm{n}} \Delta T]^{\wedge} \bar{R}_k, \tag{5.25a}$$

$$\mathrm{Exp}_R(\tilde{\omega}_k \Delta T)^{\top} = \bar{R}_{k+1}^{\top} \bar{R}_k, \tag{5.25b}$$

Let $\bar{T}_k^{k+1} \triangleq A^{\dagger} B(\bar{\psi}_k)$ and $\Delta \bar{p}_k^G = \bar{R}_k \Delta \bar{p}_k$, then it holds that

$$A^{\dagger} \bar{J}_k \bar{R}_k^{\top} \Delta T = \frac{\partial A^{\dagger} B(\bar{\psi}_k) \bar{\theta}_k}{\partial \Delta \bar{p}_k} \bar{R}_k^{\top} \Delta T = \frac{\partial \bar{T}_k^{k+1} \bar{\theta}_k}{\partial \Delta \bar{p}_k^G} \Delta T, \tag{5.26a}$$

and

$$A^{\dagger} \bar{J}_k [\eta(\bar{R}_k, \bar{v}^{\mathrm{n}}_k)]^{\wedge} = \frac{\partial \bar{T}_k^{k+1} \bar{\theta}_k}{\partial \Delta \bar{p}_k^G} \Delta T [\bar{v}^{\mathrm{n}}_k + \mathrm{g}^{\mathrm{n}} \Delta T / 2]^{\wedge} \bar{R}_k. \tag{5.26b}$$

Next, combining (5.25) and (5.26), then $\bar{F}_k$ can be rewritten as

$$\bar{F}_k = \begin{bmatrix} I_3 & I_3 \Delta T & 0 & 0 \\ 0 & I_3 & -[\bar{v}^{\mathrm{n}}_{k+1} - \bar{v}^{\mathrm{n}}_k - \mathrm{g}^{\mathrm{n}} \Delta T]^{\wedge} \bar{R}_k & 0 \\ 0 & 0 & \bar{R}_{k+1}^{\top} \bar{R}_k & 0 \\ 0 & \frac{\partial \bar{T}_k^{k+1} \bar{\theta}_k}{\partial \Delta \bar{p}_k^G} \Delta T & \frac{\partial \bar{T}_k^{k+1} \bar{\theta}_k}{\partial \Delta \bar{p}_k^G} \Delta T [\bar{v}^{\mathrm{n}}_k + \mathrm{g}^{\mathrm{n}} \Delta T / 2]^{\wedge} \bar{R}_k & \bar{T}_k^{k+1} \end{bmatrix}, \tag{5.27}$$

The matrix $\bar{\Phi}(l,k)$ can then, for $l > k$, be written as

$$\Phi(l,k) = \begin{bmatrix} I_3 & \bullet & \bullet & 0 \\ 0 & I_3 & -[\bar{v}_l^{\mathrm{n}} - \bar{v}_k^{\mathrm{n}} - \mathrm{g}^{\mathrm{n}}(l-k)\Delta T]^\wedge \bar{R}_k & 0 \\ 0 & 0 & \bar{R}_l^\top \bar{R}_k & 0 \\ 0 & \sum\limits_{j=k}^{l-1} D_j & \sum\limits_{j=k}^{l-1} D_j[\bar{v}_k^{\mathrm{n}} + \tfrac{2(j-k)+1}{2}\Delta T \mathrm{g}^{\mathrm{n}}]^\wedge \bar{R}_k & \bar{T}_k^l \end{bmatrix}, \tag{5.28a}$$

where

$$D_j = \frac{\partial \bar{T}_k^l \bar{\theta}_k}{\partial \Delta \bar{p}_j^G} \Delta T \quad \text{and} \quad \bar{T}_k^l = \bar{T}_{l-1}^l \bar{T}_{l-2}^{l-1} \cdots \bar{T}_k^{k+1}. \tag{5.28b}$$

Here $\bullet$ denotes subblocks of no interest to the analysis.

Now, it can be verified that the columns of $N(\bar{x}_k)$ in (5.15) are unobservable error states by multiplying every block row in $\bar{\mathcal{O}}_k$ with $N(\bar{x}_k)$. Let $\bar{\mathcal{O}}_k^{(s,l)} \triangleq H_{\delta x}^{(s)} \bar{\Phi}(l,k)$, where $H_{\delta x}^{(s)}$ denotes the block row in $H_{\delta x}$ corresponding to the s-th magnetometer. For $l = k$,

$$\bar{\mathcal{O}}_k^{(k,k)} = \Big[0_{3\times3} \quad 0_{3\times3} \quad 0_{3\times3} \quad H^\theta(r_s) \Big], \tag{5.29}$$

and $\bar{\mathcal{O}}_k^{(s,k)} N(\bar{x}_k) = [0_{3\times3} \quad 0_{3\times1}]$, which can be verified by simple calculation using that $\bar{\Phi}(k,k) = I$.

For $l > k$,

$$\bar{\mathcal{O}}_k^{(s,l)} = H^\theta(r_s) \begin{bmatrix} 0_{3\times3} \\ \left(\sum\limits_{j=k}^{l-1} D_j\right)^\top \\ \left(\sum\limits_{j=k}^{l-1} D_j[\bar{v}_k^{\mathrm{n}} + \tfrac{2(j-k)+1}{2}\Delta T \mathrm{g}^{\mathrm{n}}]^\wedge \bar{R}_k\right)^\top \\ (\bar{T}_k^l)^\top \end{bmatrix}^\top \tag{5.30}$$

and

$$\begin{aligned} \bar{\mathcal{O}}_k^{(s,l)} N(\bar{x}_k) &= H^\theta(r_{m_s}) \left[0 \quad \sum_{j=k}^{l-1} D_j \left(-[\bar{v}_k^{\mathrm{n}}]^\wedge \mathrm{g}^{\mathrm{n}} + \left[\bar{v}_k^{\mathrm{n}} + \frac{2(j-k)+1}{2} \Delta T \mathrm{g}^{\mathrm{n}} \right]^\wedge \bar{R}_k \bar{R}_k^\top \mathrm{g}^{\mathrm{n}} \right) \right] \\ &= H^\theta(r_{m_s}) \left[0 \quad \sum_{j=k}^{l-1} D_j \left(-[\bar{v}_k^{\mathrm{n}}]^\wedge \mathrm{g}^{\mathrm{n}} + [\bar{v}_k^{\mathrm{n}}]^\wedge \mathrm{g}^{\mathrm{n}} + \left[\frac{2(j-k)+1}{2} \Delta T \mathrm{g}^{\mathrm{n}} \right]^\wedge \mathrm{g}^{\mathrm{n}} \right) \right] \\ &= [0_{3\times3} \quad 0_{3\times1}]. \end{aligned} \tag{5.31}$$

Here the equality $[\mathrm{g}^{\mathrm{n}}]^\wedge \mathrm{g}^{\mathrm{n}} = 0$ is used in the last step.

Since $\bar{\mathcal{O}}_k^{(s,l)} N(\bar{x}_k) = 0 \ \forall s, \forall l \geq k$ then $\bar{\mathcal{O}}_k N(\bar{x}_k) = 0$ and the column vectors of $N(\bar{x}_k)$ belong to the null space of $\bar{\mathcal{O}}_k$.

5.B Interpretation of the Unobservable Subspace

The error state vector can be seen as perturbations to the state vector. The proof proceeds by applying a perturbation to the state and then calculating the resulting error state vector. If the error state vector lies in the space spanned by some columns in $N(\bar{x}_k)$ then those unobservable directions correspond to the perturbation applied.

Suppose the initial true state $x = [p^{\mathrm{n}\top} \; v^{\mathrm{n}\top} \; q^\top \; \theta^\top]^\top$, where the time index will be dropped for brevity. Next, suppose that a translation $\Delta \in \mathbb{R}^3$ is applied to the body frame. That will change the state to $x' = [(p^\mathrm{n}+\Delta)^\top \; v^{\mathrm{n}\top} \; q^\top \; \theta^\top]^\top$. The error state $\delta x = [-\Delta^\top \quad 0^\top \quad 0^\top \quad 0^\top]^\top$ lies in the space spanned by the first three columns in $N(\bar{x}_k)$. Thus, the perturbation implied by the first three columns is a body frame translation.

Next, suppose the navigation frame is rotated around the gravity vector by an angle $c \in [0, 2\pi)$ and denote the new navigation frame as the n′-frame, then the changed state x' becomes

$$x' = \begin{bmatrix} R_\mathrm{n}^{\mathrm{n}'} \cdot p^\mathrm{n} \\ R_\mathrm{n}^{\mathrm{n}'} \cdot v^\mathrm{n} \\ q\{R_\mathrm{n}^{\mathrm{n}'}\} \otimes q \\ \theta \end{bmatrix}, \quad R_\mathrm{n}^{\mathrm{n}'} = (R\{c\,\mathrm{g}^\mathrm{n}\})^\top, \tag{5.32}$$

where $R_\mathrm{n}^{\mathrm{n}'}$ denotes the orientation matrix that transforms the coordinate in n-frame to n′-frame, $R\{c\,\mathrm{g}^\mathrm{n}\}$ denotes the orientation matrix corresponding to the axis-angle representation $c\,\mathrm{g}^\mathrm{n}$, and $q\{R_\mathrm{n}^{\mathrm{n}'}\}$ denotes the quaternion corresponding to the orientation matrix $R_\mathrm{n}^{\mathrm{n}'}$.

Let $\epsilon \in \mathbb{R}^3$ denote the orientation error, i.e. the orientation difference in x and x', and c be infinitesimally small. Recalling the definition of orientation error and using $(R\{c\,\mathrm{g}^\mathrm{n}\})^\top \approx I_3 - [c\,\mathrm{g}^\mathrm{n}]^\wedge$, it holds that

$$\begin{aligned} R &\approx (R_\mathrm{n}^{\mathrm{n}'} \cdot R) \cdot (I_3 + [\epsilon]^\wedge) = (R\{c\,\mathrm{g}^\mathrm{n}\})^\top \cdot R \cdot (I_3 + [\epsilon]^\wedge) \\ &\approx (I_3 - [c\,\mathrm{g}^\mathrm{n}]^\wedge) R (I_3 + [\epsilon]^\wedge), \end{aligned} \tag{5.33}$$

where R is the orientation matrix corresponding to the quaternion q. Neglecting the second-order term, then it holds that

$$R \approx R + R[\epsilon]^\wedge - [c\,\mathrm{g}^\mathrm{n}]^\wedge R, \tag{5.34}$$

which leads to

$$[c\,\mathrm{g}^\mathrm{n}]^\wedge \approx R[\epsilon]^\wedge R^\top = [R\epsilon]^\wedge. \tag{5.35}$$

Therefore,

$$\epsilon = cR^\top \mathrm{g}^\mathrm{n}. \tag{5.36}$$

The error state for the position is calculated as

$$\begin{aligned} p^\mathrm{n} - R_\mathrm{n}^{\mathrm{n}'} \cdot p^\mathrm{n} &= p^\mathrm{n} - R^\top\{c\,\mathrm{g}^\mathrm{n}\} \cdot p^\mathrm{n} \\ &\approx (I_3 - (I_3 - [c\,\mathrm{g}^\mathrm{n}]^\wedge)) \cdot p^\mathrm{n}, \\ &= -c[p^\mathrm{n}]^\wedge \mathrm{g}^\mathrm{n}, \end{aligned} \tag{5.37}$$

where $[\mathrm{g}^{\mathrm{n}}]^{\wedge} p^{\mathrm{n}} = -[p^{\mathrm{n}}]^{\wedge}\mathrm{g}^{\mathrm{n}}$ is used. Similarly, the error state for the velocity is $-c[v^{\mathrm{n}}]^{\wedge}\mathrm{g}^{\mathrm{n}}$. Therefore, the error state

$$\delta x = c \begin{bmatrix} -[p^{\mathrm{n}}]^{\wedge}\mathrm{g}^{\mathrm{n}} \\ -[v^{\mathrm{n}}]^{\wedge}\mathrm{g}^{\mathrm{n}} \\ R^{\top}\mathrm{g}^{\mathrm{n}} \\ 0 \end{bmatrix} = c\left(- \begin{bmatrix} I_3 \\ 0_{3\times3} \\ 0_{3\times3} \\ 0_{\kappa\times3} \end{bmatrix} [p^{\mathrm{n}}]^{\wedge}\mathrm{g}^{\mathrm{n}} + \begin{bmatrix} 0 \\ -[v^{\mathrm{n}}]^{\wedge}\mathrm{g}^{\mathrm{n}} \\ R^{\top}\mathrm{g}^{\mathrm{n}} \\ 0 \end{bmatrix} \right) \tag{5.38}$$

lies in the space spanned by all columns, so they correspond to a navigation frame rotation around the gravity vector.

6

Concluding Remarks

This thesis presents an emerging indoor localization technology: the magnetic field-aided INS. Both the theory and practicality of developing such type of systems are discussed in this thesis.

6.1 Conclusions

The results in Chapter 3 and 4 demonstrate that it is feasible to build a magnetic field-aided INS using low-cost sensors given the sensors are well-calibrated. Further, the experimental results show that the localization accuracy of the system allows the exploration phase of current magnetic field SLAM systems to be greatly extended, which indicates the possibility of building self-contained magnetic field SLAM systems. Moreover, the results in Chapter 5 indicate that applying specific observability constraints can reduce yaw inconsistencies and improve the localization system's accuracy. This is fundamental if the system is to be incorporated into some larger localization or control systems because only consistent estimates can be safely used.

6.2 Future work

One intriguing avenue for future research involves integrating the proposed polynomial model with existing Gaussian process models used in magnetic field SLAM systems. While Gaussian processes excel in modeling large-scale magnetic field variations, the polynomial model maintains a small-scale map, i.e., the polynomial coefficients, that is accurate around the sensor board. The challenge here is to ensure that the updates of the large-scale and small-scale maps are seamless and mutually beneficial.

Addressing inconsistency issues remains a critical research objective. The current observability-constrained extended Kalman Filter method, while commonly used, may not always provide optimal solutions. Exploring alternative state-space model modifications or entirely different filtering paradigms such as the invariant extended Kalman filter [46] or equivalent filter [47] could potentially offer enhanced consistency and accuracy in state estimation.

Bibliography

[1] Chuan Huang, Gustaf Hendeby, and Isaac Skog. A tightly-integrated magnetic-field aided inertial navigation system. In *Proc. 2022 25th Int. Conf. on Information Fusion (FUSION)*, pages 1–8, Linköping, Sweden, July 2022. doi: 10.23919/FUSION49751.2022.9841304.

[2] Chuan Huang, Gustaf Hendeby, Hassen Fourati, Christophe Prieur, and Isaac Skog. MAINS: A magnetic-field-aided inertial navigation system for indoor positioning. *IEEE Sensors Journal*, 24(9):15156–15166, 2024. doi: 10.1109/JSEN.2024.3379932.

[3] Chuan Huang, Gustaf Hendeby, and Isaac Skog. An observability-constrained magnetic field-aided inertial navigation system — extended version. *arXiv preprint arXiv:2406.02161*, abs/2406.02161, 2024.

[4] Manon Kok, Jeroen D. Hol, and Thomas B. Schön. *Using Inertial Sensors for Position and Orientation Estimation*. Now Foundations and Trends, 2017. doi: 10.1561/2000000094.

[5] David Hoag. Apollo guidance and navigation: Considerations of apollo imu gimbal lock. *Canbridge: MIT Instrumentation Laboratory*, pages 1–64, 1963.

[6] Joan Solà. Quaternion kinematics for the error-state Kalman filter. *arXiv preprint arXiv:1711.02508*, abs/1711.02508, 2017. URL `http://arxiv.org/abs/1711.02508`.

[7] Joan Sola, Jeremie Deray, and Dinesh Atchuthan. A micro lie theory for state estimation in robotics. *arXiv preprint arXiv:1812.01537*, 2018.

[8] N. Boumal. *An Introduction to Optimization on Smooth Manifolds*. Cambridge University Press, 2023. ISBN 9781009166171.

[9] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-manifold preintegration for real-time visual–inertial odometry. *IEEE Transactions on Robotics*, 33(1):1–21, 2016.

[10] Christiane Sommer, Vladyslav Usenko, David Schubert, Nikolaus Demmel, and Daniel Cremers. Efficient derivative computation for cumulative b-splines on lie groups. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11145–11153, 2020. doi: 10.1109/CVPR42600.2020.01116.

[11] Gregory S Chirikjian. *Stochastic models, information theory, and Lie groups, volume 2: Analytic methods and modern applications*, volume 2. Springer Science & Business Media, 2011.

[12] D. Titterton, J.L. Weston, Institution of Electrical Engineers, American Institute of Aeronautics, and Astronautics. *Strapdown Inertial Navigation Technology*. IEE Radar Series. Institution of Engineering and Technology, 2004. ISBN 9780863413582.

[13] INS error budget. URL `https://www.vectornav.com/resources/inertial-navigation-primer/specifications--and--error-budgets/specs-inserrorbudget`. (Accessed 27-April-2024).

[14] John David Jackson. *Classical electrodynamics*. Wiley, New York, NY, 3rd ed. edition, 1999. ISBN 9780471309321. URL `http://cdsweb.cern.ch/record/490457`.

[15] Clara Menzen, Marnix Fetter, and Manon Kok. Large-scale magnetic field maps using structured kernel interpolation for gaussian process regression. In *2023 26th International Conference on Information Fusion (FUSION)*, pages 1–7, 2023. doi: 10.23919/FUSION52260.2023.10224210.

[16] Isaac Skog, Gustaf Hendeby, and Felix Trulsson. Magnetic-field based odometry – an optical flow inspired approach. In *Int. Conf. on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, Lloret de Mar, Spain, Nov. 2021. doi: 10.1109/IPIN51156.2021.9662626.

[17] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

[18] Niklas Wahlström, Manon Kok, Thomas B. Schön, and Fredrik Gustafsson. Modeling magnetic fields using gaussian processes. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3522–3526, 2013. doi: 10.1109/ICASSP.2013.6638313.

[19] Isaac Skog, Gustaf Hendeby, and Fredrik Gustafsson. Magnetic odometry - a model-based approach using a sensor array. In *Int. Conf. on Information Fusion (FUSION)*, pages 794–798, Cambridge, United Kingdom, July 2018. doi: 10.23919/ICIF.2018.8455430.

[20] Jeroen D. Hol. *Sensor Fusion and Calibration of Inertial Sensors, Vision, Ultra-Wideband and GPS*. PhD thesis, Linköping UniversityLinköping University, Automatic Control, The Institute of Technology, 2011.

[21] Manon Kok. *Probabilistic modeling for sensor fusion with inertial measurements*. PhD thesis, Linköping University Electronic Press, 2016.

[22] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34 (4):1004–1020, 2018. doi: 10.1109/TRO.2018.2853729.

[23] Takafumi Taketomi, Hideaki Uchiyama, and Sei Ikeda. Visual slam algorithms: A survey from 2010 to 2016. *IPSJ transactions on computer vision and applications*, 9:1–11, 2017.

[24] Natalia Pavlasek, Charles Champagne Cossette, David Roy-Guay, and James Richard Forbes. Magnetic navigation using attitude-invariant magnetic field information for loop closure detection. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5251–5257, Detroit, USA, 2023. doi: 10.1109/IROS55552.2023.10342466.

[25] Andersen Martin and Hansson Anders. *Optimization Problems*, chapter 5, pages 94–117. John Wiley & Sons, Ltd, 2023. ISBN 9781119809180. doi: https://doi.org/10.1002/9781119809180.ch5.

[26] Gilbert Strang. *Introduction to Linear Algebra*. CUP, 6 edition, 2023.

[27] Thomas Kailath, Ali H Sayed, and Babak Hassibi. *Linear estimation*. Prentice Hall, 2000.

[28] Fredrik Gustafsson. *Statistical sensor fusion*. Studentlitteratur, 2018.

[29] Wilson J Rugh. *Linear system theory*. Prentice-Hall, Inc., 1996.

[30] Z. Chen, K. Jiang, and J.C. Hung. Local observability matrix and its application to observability analyses. In *[Proceedings] IECON '90: 16th Annual Conference of IEEE Industrial Electronics Society*, pages 100–103 vol.1, Pacific Grove, CA, USA, 1990. doi: 10.1109/IECON.1990.149118.

[31] Manon Kok and Thomas B Schön. Magnetometer calibration using inertial sensors. *IEEE Sensors Journal*, 16(14):5679–5689, 2016.

[32] NCEI Geomagnetic Modeling Team. World magnetic model 2020. Technical report, NOAA National Centers for Environmental Information, 2019. doi : 10.25921/11v3-da71.

[33] S. Theodoridis. *Machine Learning: A Bayesian and Optimization Perspective*. Elsevier Science, 2020. ISBN 9780128188040.

[34] Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[35] Makia Zmitri, Hassen Fourati, and Christophe Prieur. Magnetic Field Gradient-Based EKF for Velocity Estimation in Indoor Navigation. *Sensors*, 20(20):5726, 2020.

[36] Tisheng Zhang, Linfu Wei, Jian Kuang, Hailiang Tang, and Xiaoji Niu. Mag-odo: Motion speed estimation for indoor robots based on dual magnetometers. *Measurement*, 222:113688, 2023. ISSN 0263-2241. doi: https://doi.org/10.1016/j.measurement.2023.113688.

[37] Charles-Ivan Chesneau, Mathieu Hillion, and Christophe Prieur. Motion estimation of a rigid body with an EKF using magneto-inertial measurements. In *Int. Conf. on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–6, Alcalá de Henares, Spain, Oct. 2016.

[38] Eric Dorveaux, Thomas Boudot, Mathieu Hillion, and Nicolas Petit. Combining inertial measurements and distributed magnetometry for motion estimation. In *Proc. 2011 American Control Conference*, pages 4249–4256, San Francisco, CA, USA, June 2011.

[39] Mingyang Li and Anastasios I. Mourikis. High-precision, consistent EKF-based visual-inertial odometry. *The International Journal of Robotics Research*, 32(6):690–711, 2013. doi: 10.1177/0278364913481251.

[40] Joel A. Hesch, Dimitrios G. Kottas, Sean L. Bowman, and Stergios I. Roumeliotis. Consistency analysis and improvement of vision-aided inertial navigation. *IEEE Transactions on Robotics*, 30(1):158–176, 2014. doi: 10.1109/TRO.2013.2277549.

[41] Guoquan P Huang, Nikolas Trawny, Anastasios I Mourikis, and Stergios I Roumeliotis. Observability-based consistent EKF estimators for multi-robot cooperative localization. *Autonomous Robots*, 30:99–122, 2011.

[42] Zheng Huai and Guoquan Huang. Robocentric visual-inertial odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6319–6326, Madrid, Spain, October 2018. doi: 10.1109/IROS.2018.8593643.

[43] Martin Brossard, Axel Barrau, and Silvère Bonnabel. Exploiting symmetries to design EKFs with consistency properties for navigation and SLAM. *IEEE Sensors Journal*, 19(4):1572–1579, 2019. doi: 10.1109/JSEN.2018.2882714.

[44] Guoquan P Huang, Anastasios I Mourikis, and Stergios I Roumeliotis. Observability-based rules for designing consistent EKF SLAM estimators. *The International Journal of Robotics Research*, 29(5):502–528, 2010.

[45] David Caruso, Alexandre Eudes, Martial Sanfourche, David Vissière, and Guy Le Besnerais. Magneto-visual-inertial dead-reckoning: Improving estimation consistency by invariance. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 7923–7930, Nice, France, December 2019. doi: 10.1109/CDC40024.2019.9029283.

[46] Axel Barrau and Silvere Bonnabel. The invariant extended kalman filter as a stable observer. *IEEE Transactions on Automatic Control*, 62(4):1797–1812, 2016.

[47] Pieter van Goor, Tarek Hamel, and Robert Mahony. Equivariant filter (eqf). *IEEE Transactions on Automatic Control*, 68(6):3501–3512, 2023. doi: 10.1109/TAC.2022.3194094.

Licentiate Theses
Division of Automatic Control
Linköping University

P. Andersson: Adaptive Forgetting through Multiple Models and Adaptive Control of Car Dynamics. Thesis No. 15, 1983.

B. Wahlberg: On Model Simplification in System Identification. Thesis No. 47, 1985.

A. Isaksson: Identification of Time Varying Systems and Applications of System Identification to Signal Processing. Thesis No. 75, 1986.

G. Malmberg: A Study of Adaptive Control Missiles. Thesis No. 76, 1986.

S. Gunnarsson: On the Mean Square Error of Transfer Function Estimates with Applications to Control. Thesis No. 90, 1986.

M. Viberg: On the Adaptive Array Problem. Thesis No. 117, 1987.

K. Ståhl: On the Frequency Domain Analysis of Nonlinear Systems. Thesis No. 137, 1988.

A. Skeppstedt: Construction of Composite Models from Large Data-Sets. Thesis No. 149, 1988.

P. A. J. Nagy: MaMiS: A Programming Environment for Numeric/Symbolic Data Processing. Thesis No. 153, 1988.

K. Forsman: Applications of Constructive Algebra to Control Problems. Thesis No. 231, 1990.

I. Klein: Planning for a Class of Sequential Control Problems. Thesis No. 234, 1990.

F. Gustafsson: Optimal Segmentation of Linear Regression Parameters. Thesis No. 246, 1990.

H. Hjalmarsson: On Estimation of Model Quality in System Identification. Thesis No. 251, 1990.

S. Andersson: Sensor Array Processing; Application to Mobile Communication Systems and Dimension Reduction. Thesis No. 255, 1990.

K. Wang Chen: Observability and Invertibility of Nonlinear Systems: A Differential Algebraic Approach. Thesis No. 282, 1991.

J. Sjöberg: Regularization Issues in Neural Network Models of Dynamical Systems. Thesis No. 366, 1993.

P. Pucar: Segmentation of Laser Range Radar Images Using Hidden Markov Field Models. Thesis No. 403, 1993.

H. Fortell: Volterra and Algebraic Approaches to the Zero Dynamics. Thesis No. 438, 1994.

T. McKelvey: On State-Space Models in System Identification. Thesis No. 447, 1994.

T. Andersson: Concepts and Algorithms for Non-Linear System Identifiability. Thesis No. 448, 1994.

P. Lindskog: Algorithms and Tools for System Identification Using Prior Knowledge. Thesis No. 456, 1994.

J. Plantin: Algebraic Methods for Verification and Control of Discrete Event Dynamic Systems. Thesis No. 501, 1995.

J. Gunnarsson: On Modeling of Discrete Event Dynamic Systems, Using Symbolic Algebraic Methods. Thesis No. 502, 1995.

A. Ericsson: Fast Power Control to Counteract Rayleigh Fading in Cellular Radio Systems. Thesis No. 527, 1995.

M. Jirstrand: Algebraic Methods for Modeling and Design in Control. Thesis No. 540, 1996.

K. Edström: Simulation of Mode Switching Systems Using Switched Bond Graphs. Thesis No. 586, 1996.

J. Palmqvist: On Integrity Monitoring of Integrated Navigation Systems. Thesis No. 600, 1997.

A. Stenman: Just-in-Time Models with Applications to Dynamical Systems. Thesis No. 601, 1997.

M. Andersson: Experimental Design and Updating of Finite Element Models. Thesis No. 611, 1997.

U. Forssell: Properties and Usage of Closed-Loop Identification Methods. Thesis No. 641, 1997.

M. Larsson: On Modeling and Diagnosis of Discrete Event Dynamic systems. Thesis No. 648, 1997.

N. Bergman: Bayesian Inference in Terrain Navigation. Thesis No. 649, 1997.

V. Einarsson: On Verification of Switched Systems Using Abstractions. Thesis No. 705, 1998.

J. Blom, F. Gunnarsson: Power Control in Cellular Radio Systems. Thesis No. 706, 1998.

P. Spångéus: Hybrid Control using LP and LMI methods – Some Applications. Thesis No. 724, 1998.

M. Norrlöf: On Analysis and Implementation of Iterative Learning Control. Thesis No. 727, 1998.

A. Hagenblad: Aspects of the Identification of Wiener Models. Thesis No. 793, 1999.

F. Tjärnström: Quality Estimation of Approximate Models. Thesis No. 810, 2000.

C. Carlsson: Vehicle Size and Orientation Estimation Using Geometric Fitting. Thesis No. 840, 2000.

J. Löfberg: Linear Model Predictive Control: Stability and Robustness. Thesis No. 866, 2001.

O. Härkegård: Flight Control Design Using Backstepping. Thesis No. 875, 2001.

J. Elbornsson: Equalization of Distortion in A/D Converters. Thesis No. 883, 2001.

J. Roll: Robust Verification and Identification of Piecewise Affine Systems. Thesis No. 899, 2001.

I. Lind: Regressor Selection in System Identification using ANOVA. Thesis No. 921, 2001.

R. Karlsson: Simulation Based Methods for Target Tracking. Thesis No. 930, 2002.

P.-J. Nordlund: Sequential Monte Carlo Filters and Integrated Navigation. Thesis No. 945, 2002.

M. Östring: Identification, Diagnosis, and Control of a Flexible Robot Arm. Thesis No. 948, 2002.

C. Olsson: Active Engine Vibration Isolation using Feedback Control. Thesis No. 968, 2002.

J. Jansson: Tracking and Decision Making for Automotive Collision Avoidance. Thesis No. 965, 2002.

N. Persson: Event Based Sampling with Application to Spectral Estimation. Thesis No. 981, 2002.

D. Lindgren: Subspace Selection Techniques for Classification Problems. Thesis No. 995, 2002.

E. Geijer Lundin: Uplink Load in CDMA Cellular Systems. Thesis No. 1045, 2003.

M. Enqvist: Some Results on Linear Models of Nonlinear Systems. Thesis No. 1046, 2003.

T. Schön: On Computational Methods for Nonlinear Estimation. Thesis No. 1047, 2003.

F. Gunnarsson: On Modeling and Control of Network Queue Dynamics. Thesis No. 1048, 2003.

S. Björklund: A Survey and Comparison of Time-Delay Estimation Methods in Linear Systems. Thesis No. 1061, 2003.

M. Gerdin: Parameter Estimation in Linear Descriptor Systems. Thesis No. 1085, 2004.

A. Eidehall: An Automotive Lane Guidance System. Thesis No. 1122, 2004.

E. Wernholt: On Multivariable and Nonlinear Identification of Industrial Robots. Thesis No. 1131, 2004.

J. Gillberg: Methods for Frequency Domain Estimation of Continuous-Time Models. Thesis No. 1133, 2004.

G. Hendeby: Fundamental Estimation and Detection Limits in Linear Non-Gaussian Systems. Thesis No. 1199, 2005.

D. Axehill: Applications of Integer Quadratic Programming in Control and Communication. Thesis No. 1218, 2005.

J. Sjöberg: Some Results On Optimal Control for Nonlinear Descriptor Systems. Thesis No. 1227, 2006.

D. Törnqvist: Statistical Fault Detection with Applications to IMU Disturbances. Thesis No. 1258, 2006.

H. Tidefelt: Structural algorithms and perturbations in differential-algebraic equations. Thesis No. 1318, 2007.

S. Moberg: On Modeling and Control of Flexible Manipulators. Thesis No. 1336, 2007.

J. Wallén: On Kinematic Modelling and Iterative Learning Control of Industrial Robots. Thesis No. 1343, 2008.

J. Harju Johansson: A Structure Utilizing Inexact Primal-Dual Interior-Point Method for Analysis of Linear Differential Inclusions. Thesis No. 1367, 2008.

J. D. Hol: Pose Estimation and Calibration Algorithms for Vision and Inertial Sensors. Thesis No. 1370, 2008.

H. Ohlsson: Regression on Manifolds with Implications for System Identification. Thesis No. 1382, 2008.

D. Ankelhed: On low order controller synthesis using rational constraints. Thesis No. 1398, 2009.

P. Skoglar: Planning Methods for Aerial Exploration and Ground Target Tracking. Thesis No. 1420, 2009.

C. Lundquist: Automotive Sensor Fusion for Situation Awareness. Thesis No. 1422, 2009.

C. Lyzell: Initialization Methods for System Identification. Thesis No. 1426, 2009.

R. Falkeborn: Structure exploitation in semidefinite programming for control. Thesis No. 1430, 2010.

D. Petersson: Nonlinear Optimization Approaches to $\mathcal{H}_2$-Norm Based LPV Modelling and Control. Thesis No. 1453, 2010.

Z. Sjanic: Navigation and SAR Auto-focusing in a Sensor Fusion Framework. Thesis No. 1464, 2011.

K. Granström: Loop detection and extended target tracking using laser data. Thesis No. 1465, 2011.

J. Callmer: Topics in Localization and Mapping. Thesis No. 1489, 2011.

F. Lindsten: Rao-Blackwellised particle methods for inference and identification. Thesis No. 1480, 2011.

M. Skoglund: Visual Inertial Navigation and Calibration. Thesis No. 1500, 2011.

S. Khoshfetrat Pakazad: Topics in Robustness Analysis. Thesis No. 1512, 2011.

P. Axelsson: On Sensor Fusion Applied to Industrial Manipulators. Thesis No. 1511, 2011.

A. Carvalho Bittencourt: On Modeling and Diagnosis of Friction and Wear in Industrial Robots. Thesis No. 1516, 2012.

P. Rosander: Averaging level control in the presence of frequent inlet flow upsets. Thesis No. 1527, 2012.

N. Wahlström: Localization using Magnetometers and Light Sensors. Thesis No. 1581, 2013.

R. Larsson: System Identification of Flight Mechanical Characteristics. Thesis No. 1599, 2013.

Y. Jung: Estimation of Inverse Models Applied to Power Amplifier Predistortion. Thesis No. 1605, 2013.

M. Syldatk: On Calibration of Ground Sensor Networks. Thesis No. 1611, 2013.

M. Roth: Kalman Filters for Nonlinear Systems and Heavy-Tailed Noise. Thesis No. 1613, 2013.

D. Simon: Model Predictive Control in Flight Control Design — Stability and Reference Tracking. Thesis No. 1642, 2014.

J. Dahlin: Sequential Monte Carlo for inference in nonlinear state space models. Thesis No. 1652, 2014.

M. Kok: Probabilistic modeling for positioning applications using inertial sensors. Thesis No. 1656, 2014.

J. Linder: Graybox Modelling of Ships Using Indirect Input Measurements. Thesis No. 1681, 2014.

G. Mathai: Direction of Arrival Estimation of Wideband Acoustic Wavefields in a Passive Sensing Environment. Thesis No. 1721, 2015.

I. Nielsen: On Structure Exploiting Numerical Algorithms for Model Predictive Control. Thesis No. 1727, 2015.

C. Veibäck: Tracking of Animals Using Airborne Cameras. Thesis No. 1761, 2016.

N. Evestedt: Sampling Based Motion Planning for Heavy Duty Autonomous Vehicles. Thesis No. 1762, 2016.

H. Nyqvist: On Pose Estimation in Room-Scaled Environments. Thesis No. 1765, 2016.

Y. Zhao: Position Estimation in Uncertain Radio Environments and Trajectory Learning. Thesis No. 1772, 2017.

P. Kasebzadeh: Parameter Estimation for Mobile Positioning Applications. Thesis No. 1786, 2017.

K. Radnosrati: On Timing-Based Localization in Cellular Radio Networks. Thesis No. 1808, 2018.

G. Lindmark: Methods and Algorithms for Control Input Placement in Complex Networks. Thesis No. 1814, 2018.

M. Lindfors: Frequency Tracking for Speed Estimation. Thesis No. 1815, 2018.

D. Ho: Some results on closed-loop identification of quadcopters. Thesis No. 1826, 2018.

O. Ljungqvist: On motion planning and control for truck and trailer systems. Thesis No. 1832, 2019.

P. Boström-Rost: On Informative Path Planning for Tracking and Surveillance. Thesis No. 1838, 2019.

K. Bergman: On Motion Planning Using Numerical Optimal Control. Thesis No. 1843, 2019.

M. Klingspor: Low-rank optimization in system identification. Thesis No. 1855, 2019.

A. Bergström: Timing-Based Localization using Multipath Information. Thesis No. 1867, 2019.

F. Ljungberg: Estimation of Nonlinear Greybox Models for Marine Applications. Thesis No. 1880, 2020.

E. Hedberg: Control, Models and Industrial Manipulators. Thesis No. 1894, 2020.

R. Forsling: Decentralized Estimation Using Conservative Information Extraction. Thesis No. 1897, 2020.

D. Arnström: On Complexity Certification of Active-Set QP Methods with Applications to Linear MPC. Thesis No. 1901, 2021.

M. Malmström: Uncertainties in Neural Networks: A System Identification Approach. Thesis No. 1902, 2021.

K. Nielsen: Robust LIDAR-Based Localization in Underground Mines. Thesis No. 1906, 2021.

H. Haghshenas: Time-Optimal Cooperative Path Tracking for Multi-Robot Systems. Thesis No. 1915, 2021.

A. Kullberg: On Joint State Estimation and Model Learning using Gaussian Process Approximations. Thesis No. 1917, 2021.

J. Nordlöf: On Landmark Densities in Minimum-Uncertainty Motion Planning. Thesis No. 1927, 2022.

S. A. Zimmermann: Data-driven Modeling of Robotic Manipulators—Efficiency Aspects. Thesis No. 1963, 2023.

S. Shoja: On Complexity Certification of Branch-and-Bound Methods for MILP and MIQP with Applications to Hybrid MPC. Thesis No. 1967, 2023.

A. Hellander: On Optimal Integrated Task and Motion Planning with Applications to Tractor-Trailers. Thesis No. 1981, 2023.

J. Wilroth: Exploring Auditory Attention Using EEG. Thesis No. 1993, 2024.

FACULTY OF SCIENCE AND ENGINEERING

Linköping Studies in Science and Technology, Licentiate Thesis No. 2003, 2024
Department of Electrical Engineering

Linköping University
SE-581 83 Linköping, Sweden

www.liu.se

LINKÖPING
UNIVERSITY