PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA

FULL THREE-AXIS MAGNETOMETER AND GYROSCOPE BIAS ESTIMATION USING ANGULAR RATE MEASUREMENTS

SEBASTIÁN A. RODRÍGUEZ MARTÍNEZ

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Advisor:
GIANCARLO TRONI

Santiago de Chile, April, 2023

PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA

FULL THREE-AXIS MAGNETOMETER AND GYROSCOPE BIAS ESTIMATION USING ANGULAR RATE MEASUREMENTS

SEBASTIÁN A. RODRÍGUEZ MARTÍNEZ

Members of the Committee:
GIANCARLO TRONI
MIGUEL TORRES
JOHN ANDREW HAMILTON
DIEGO LÓPEZ-GARCÍA

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Santiago de Chile, April, 2023

To Carmen Rey (Mamina)

ACKNOWLEDGEMENTS

I would not have been able to finish this thesis without the support of many individuals, so I thank all of you very much. Also, thank you for taking the time to read this thesis. If you have any questions or comments, please do not hesitate to contact me at sarodriguez1@uc.cl.

I want to express my gratitude to my advisor, Professor Giancarlo Troni, who supported me in this process in an unconditional manner, always willing to teach me how to be a better engineer and researcher. I would also like to thank the other members serving on the committee: Miguel Torres, Diego López-García, and John Andrew Hamilton, who took the time to be part of this process.

The research presented in this thesis would not have been possible without the support of the Monterey Bay Aquarium Research Institute, who generously provided the necessary in-field data for validating the methods. I am also thankful to all the members of the CoMPAS Lab, who gave me valuable feedback during the different stages of this research.

To my friends, Joaquín, Francisca, and Federico, who encouraged me throughout this journey and supported me in stepping out of my comfort zone to embrace new challenges.

Finally, I would like to thank my family for their support during my university years. My parents, Cristian and María José; my siblings Flo, Benja, and Cristiancito; my grandparents Tomás (Nonno), Carolina (Abuela), and Marcela (Ita); and my uncles Tomás, Diego, and Francisco, who always pushed me to pursue my dreams and goals with unconditional love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Ocean exploration has seen significant advancements with the development of underwater vehicles. However, the high costs of the current high-end vehicles limit their widespread use. Low-cost vehicles are being explored to overcome this limitation, but their limited navigation accuracy restricts their use in low-precision missions. This thesis proposes a novel methodology to improve the navigation precision of low-cost navigation systems for underwater exploration.

The MEMS Attitude Heading Reference Systems (AHRSs) are widely used for determining the attitude of a system. However, the accuracy is limited due to sensor measurement biases. To overcome this, this thesis proposes the Magnetometer and Gyroscope Iterative Calibration (MAGICAL) method, which employs three-axis angular rate measurements from an angular rate gyroscope. Four approaches based on linear and nonlinear iterative least squares and batch and online incremental factor graphs are proposed to implement this method, which is less restrictive regarding instrument movements required for calibration, does not require knowledge of the local magnetic field or instrument's attitude, and can be integrated into factor graph algorithms for Smoothing and Mapping frameworks.

The proposed methods were compared with the state-of-the-art methods in numerical simulations and in-field experimental evaluations with a sensor onboard an underwater vehicle. With MAGICAL, the underwater vehicle's dead reckoning position estimation error was reduced from 10% to 0.5% of the distance traveled. The results show that MAGICAL can significantly improve the accuracy of low-cost navigation systems and pave the way for more widespread use of underwater vehicles in oceanographic missions.

Keywords: **Marine robotics, sensor fusion, magnetometer hard-iron bias and soft-iron bias calibration, doppler navigation, gyroscope bias calibration**.

RESUMEN

La exploración oceánica ha experimentado avances significativos debido al desarrollo de los vehículos submarinos, sin embargo, los elevados costos de los actuales vehículos de gama alta limitan su uso generalizado. Como alternativa, se han explorado vehículos de bajo costo, pero su limitada navegación restringe su uso a misiones de baja precisión, para lo cual esta tesis propone una metodología novedosa que busca una navegación precisa.

Los sistemas de referencia de actitud y rumbo (AHRS, por sus siglas en inglés) basados en sistemas microelectromecánicos (MEMS, por sus siglas en inglés) son ampliamente utilizados para determinar la orientación de un sistema, sin embargo, su precisión se ve limitada por los sesgos en las medición de los sensores que lo componen. Esta tesis propone el método de calibración iterativa de magnetómetro y giróscopo, denominado MAGICAL, por sus siglas en inglés, el cual utiliza medidas de velocidad angular, permitiéndole ser menos restrictivo en cuanto a los movimientos necesarios para la calibración, no requiere conocimiento del campo magnético local ni de la orientación del sistema, y puede integrarse en algoritmos de grafos de factores. Este método puede ser implementado de cuatro formas, utilizando mínimos cuadrados iterativos lineales y no lineales, y utilizando grafos de factores, tanto en forma incremental como incluyendo todas las mediciones.

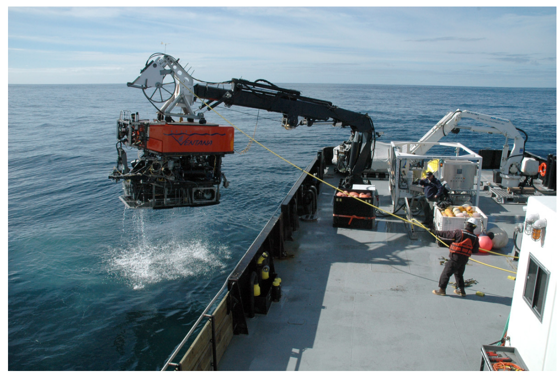
El método propuesto se compara con el estado del arte tanto en simulaciones numéricas como en evaluaciones experimentales en terreno, utliizando un sensor a bordo de un vehículo submarino. Con MAGICAL, el error de estimación de la posición del vehículo submarino se redujo del 10% al 0,5% de la distancia recorrida, demostrando que MAGICAL puede mejorar significativamente la precisión de los sistemas de navegación de bajo costo y dar pie a uso más generalizado de los vehículos submarinos en misiones oceanográficas.

Palabras clave: **Robótica marina, fusión de sensores, calibración de hard-iron y soft-iron en magnetómetros, navegación doppler, calibración del sesgo del giróscopo**.

1. INTRODUCTION

The oceans constitute 71% of the Earth's surface and are vital for sustaining all known life (NOAA, 2023). They serve as a significant source of energy, food, and natural resources and are critical in regulating the climate and weather conditions and preserving the biodiversity of the planet's ecosystem. Despite their immense importance, the underwater floor remains mainly uncharted, with only 10% having been explored to date with modern methods such as sonar technology (NOAA, 2023).

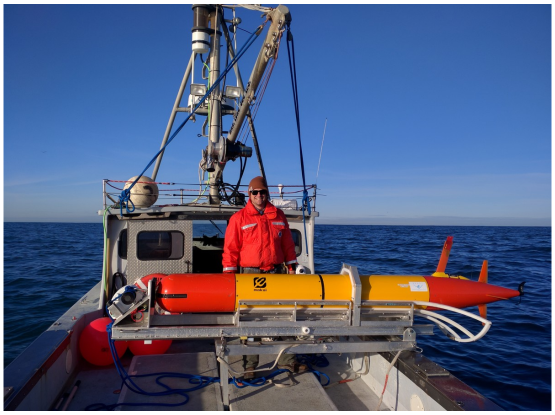
(a) MBARI *Rachel Carson Research Vessel*. Image Source: MBARI.



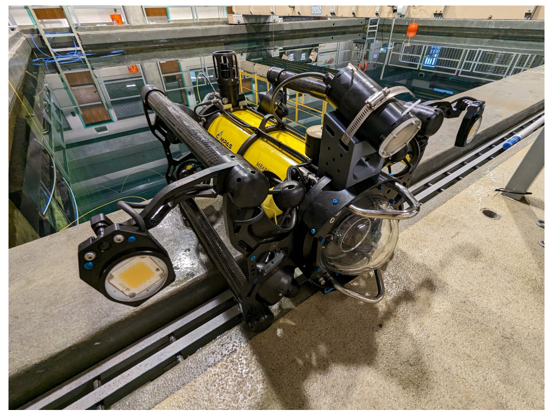
(b) MBARI *Ventana* Remote Operated Vehicle. Image source: MBARI.

Figure 1.1. Monterey Bay Aquarium Research Institute (MBARI) equipment for daily deployments.

Oceanographic exploration using underwater vehicles is advancing rapidly, thanks to continual improvements in precision underwater navigation technology. These advances have expanded the operating range of underwater vehicles, enabling missions that were once deemed impossible. However, many of these developments focus on large-scale operations, requiring significant resources such as remote-operated vehicles (ROVs), autonomous underwater vehicles (AUVs), and research vessels (RVs), leading to high costs that can run into the millions of dollars for regular operations (Fig. 1.1). As an alternative to these costly operations, a new family of relatively low-cost underwater vehicles is emerging, making ocean exploration more accessible. However, these vehicles are limited by the lack of precise low-cost and low-power navigation methods.



(a) MBARI *Long Range* AUV. Image source: MBARI.

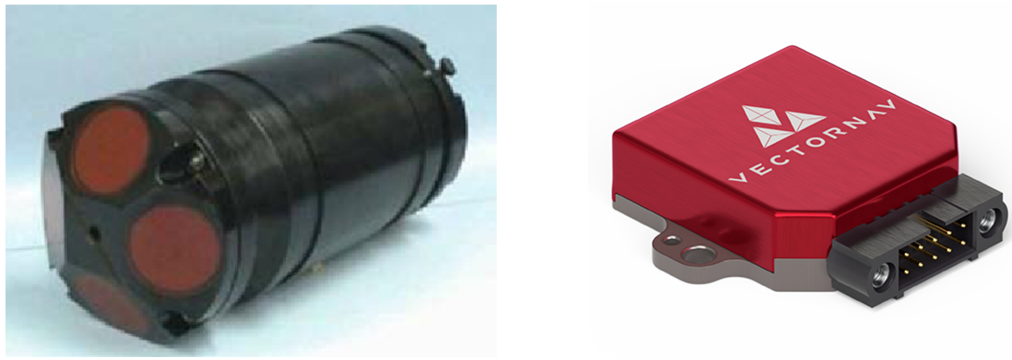


(b) MBARI *Mola* AUV/ROV. Image source: MBARI.

Figure 1.2. Monterey Bay Aquarium Research Institute (MBARI) relative low-cost underwater vehicles.

Accurate sensing and estimation of attitude, including heading, roll, and pitch, are critical components of navigation systems for underwater vehicles. This is especially important for vehicles operating in environments where global positioning system (GPS) signals are unavailable, such as underwater navigation. While high-precision sensors based on optical gyroscopes, like the Kearfott SeaDevil Inertial Navigation System (Fig. 1.3a), are available, they are much more expensive and require more power and space compared to low-cost Micro-Electro-Mechanical System (MEMS) Attitude and Heading Reference Systems (AHRSs), such as the VectorNav VN-100 (Fig. 1.3b), which are typically mounted in low-cost underwater vehicles.

MEMS AHRS generally consists of a three-axis magnetometer, a three-axis accelerometer, a three-axis gyroscope, and a temperature sensor. The magnetometer measures the local Earth's magnetic field vector, enabling the determination of the system's heading. In the absence of considerable external accelerations, the accelerometer is employed to determine the system's inclination relative to the Earth's local gravity vector, thus providing



(a) High-cost attitude sensor: Kearfott SeaDevil Inertial Navigation System. Image source: (Caress et al., 2008).

(b) Low-cost attitude sensor: VectorNav VN-100. Image source: (Vectornav, 2023).

Figure 1.3. Attitude sensors.

information regarding the orientation of the pitch and roll axes. Finally, the gyroscope provides the angular rate of the vehicle, which can be used to smooth the rotation estimation of the system.

The accuracy of the estimation process is critically dependent upon the successful elimination of biases, scale factors, and non-orthogonality that have the potential to impact the functionality of the AHRS components. Specifically, biases pose a challenge to gyroscopes and accelerometers. In contrast, magnetometers are susceptible to two main types of sensor calibration errors induced by any ferrous materials or electric currents near the object, which can bias and distort the background magnetic field, leading to increased errors in the heading estimate. Hard-iron biases result from the constant errors from the permanent magnetic field generated by the vehicle and onboard instruments, leading to a constant bias in the sensor output. On the other hand, soft-iron biases manifest as errors caused by the magnetic permeability of the materials surrounding the sensor, resulting in the distortion or stretching of the magnetic field.

Changes in a vehicle's physical configuration, such as adding or removing sensors or other payloads, commonly occur in underwater vehicles, requiring the AHRS to be recalibrated. However, most previously reported approaches for hard-iron and soft-iron calibration require significant angular motion of the instrument in all rotational degrees of freedom (i.e., roll, pitch, and heading), which can be impractical and even unfeasible on many roll and pitch stable vehicles, leading to a failure in the calibration method. These challenges allow us to propose this thesis hypothesis, which states that the development of a magnetometer calibration method integrated with angular rates allows the estimation of the biases of magnetometer and gyroscopes with improved robustness and versatility.

This thesis presents a novel approach called the Magnetometer and Gyroscope Iterative Calibration (MAGICAL) method, which, taking advantage of the angular-rate measurements, overcomes the limitations of current navigation systems for underwater vehicles. Four different methods based on this approach are proposed, two using iterative least squares and two using factor graphs. The MAGICAL method has the potential to make oceanographic research more accessible by enabling the development of high-accuracy, low-cost navigation systems for underwater vehicles. This will allow for ocean exploration with similar performance to current approaches but at a fraction of the cost.

1.1. Literature Review

Accurate calibration of three-axis magnetometers is crucial for reliable attitude estimation, and several methods have been proposed to estimate calibration parameters without requiring additional reference sensors. One widely used approach proposed by Alonso and Shuster is the TWOSTEP method, which estimates the magnetometer's bias (Alonso & Shuster, 2002b) and later estimates the scale and non-orthogonality factors as well (Alonso & Shuster, 2002a) using an iterative least squares minimization. The full calibration of a magnetometer can also be formulated as an ellipsoid fitting problem, which Vasconcelos et al. (Vasconcelos, Elkaim, Silvestre, Oliveira, & Cardeira, 2011) solved using a maximum likelihood estimate method. Several least squares methods have been reported as well (Fang, Sun, Cao, Zhang, & Tao, 2011; Foster & Elkaim, 2008; Ousaloo, Sharifi, Mahdian, & Nodeh, 2017). However, these methods are not practical in field applications as the device must perform wide ranges of angular motion in all three degrees of freedom, which is infeasible for many devices mounted on full-scale vehicles, such as underwater remote-operated vehicles that are pitch and roll stable. Furthermore, accurately knowing the Earth's local magnetic field is necessary for improved performance, which magnetic field models can calculate (NOAA, 2021), but it can present significant errors due to unmodeled local perturbations.

Given the availability of inertial sensors with magnetometers as a package, some approaches have fused accelerometer and magnetometer measurements to estimate the magnetometer sensor biases, taking advantage of the accelerometer to measure the local gravity vector (Kok & Schon, 2016; Papafotis & Sotiriadis, 2019; Ammann, Derksen, & Heck, 2015). However, these methods are affected by errors in the calibration caused by translational accelerations of the system. Troni and Whitcomb (Troni & Whitcomb, 2020) proposed a novel method using angular velocity measurements, which assumes that the angular velocity sensor is already bias-compensated, to estimate the magnetometer's hard-iron. This method was later extended by Spielvogel and Whitcomb (A. R. Spielvogel & Whitcomb, 2018) to include the estimation of gyroscope and accelerometer biases. However, neither of these methods addresses soft-iron magnetometer calibration.

The previously discussed methods are limited to batch calibrations, where all the measurements must be collected beforehand. However, it is desirable to perform calibration in the field when pre-calibration is not possible, for instance, due to changes in the vehicle's configuration deployed in highly disturbed environments. Crassidis et al. (Crassidis, Lai, & Harman, 2005) presented an extension to the TWOSTEP method that incorporates an Extended Kalman Filter (EKF) and an Unscented Kalman Filter (UKF). Later, Ma and Jing (Ma & Jiang, 2005), and Soken and Sakai (Soken & Sakai, 2019) proposed alternative approaches for the UKF method, while Guo et al. (Guo, Qiu, Yang, & Ren, 2008) presented an alternative EKF method. Additionally, Han et al. (Han, Han, Wang, & Xu, 2017) and Spielvogel et al. (A. Spielvogel, Shah, & Whitcomb, 2022) proposed a gyroscope-aided EKF, with the latter also incorporating gyroscope biases in the estimation. However, these approaches demand substantial angular motion for accurate calibration and may experience difficulties in accurately estimating the actual values if the angular motion is insufficient.

Previous studies have highlighted that existing methods for calibrating magnetometer and gyroscope biases are limited by at least one of the following: (i) the requirement for a broad range of movements, which may not be attainable in many vehicles, (ii) the need for accurate knowledge of the local Earth's magnetic field, or (iii) the inability to determine the magnetometer's soft-iron error. This thesis introduces a novel approach, called Magnetometer and Gyroscope Iterative Calibration (MAGICAL), which overcomes these limitations and proposes four methods based on this approach: two based on iterative least squares and two based on factor graphs. The MAGICAL approach offers several advantages, including (i) fewer restrictions on angular movement requirements, (ii) no need for information about the local magnetic field or system's attitude, (iii) a complete calibration for both magnetometer and gyroscope and (iv) integration into factor graph algorithms for Smoothing and Mapping (SAM) frameworks with tools such as GTSAM or iSAM (BORG Lab, 2023) for problems such as Simultaneous Localization and Mapping (SLAM).

1.2. Thesis Outline

The thesis is structured as follows: Chapter 2 briefly overviews the required mathematical background and the benchmark methods used in this thesis. Chapter 3 details the sensor model, system model, and the proposed MAGICAL methods. Chapter 4 describes the evaluation methodology of MAGICAL. Chapter 5 reports on the numerical simulation evaluation of the MAGICAL approach, while chapter 6 reports on the field experimental evaluation of MAGICAL in a full-scale ROV field survey in Monterey Bay, California, USA. Chapter 7 summarizes and concludes the thesis.

2. BACKGROUND

This chapter aims to provide readers with the necessary mathematical and methodological foundations to understand the research presented in subsequent chapters. The first section offers the required mathematical resources for upcoming chapters, while the second section outlines benchmark methods used to evaluate the proposed MAGICAL method. Overall, this chapter provides the essential context to fully comprehend the thesis's contributions.

2.1. Mathematical Background

2.1.1. Rotation Matrix

The special orthogonal group or also known as the group of rotation matrices, $SO(3)$, is the set of all 3×3 real matrices R describing the orientation of a frame V fixe to the vehicle with respect to the inertial world W, that satisfy

$$SO(3) = \{R : \ R \in \mathcal{R}^{3\times 3}, \ R^T R = I, \ det(R) = 1\}.$$

2.1.2. Kronecker Product

The Kronecker product, also known as tensor product or direct product, of the matrix $A \in \mathbb{M}^{p\times q}$ with the matrix $B \in \mathbb{M}^{r\times s}$ is defined as an $p \times q$ block matrix whose (i, j) block is the $r \times s$ matrix $a_{ij}B$

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1q}B \\ \vdots & & \vdots \\ a_{p1}B & \cdots & a_{pq}B \end{bmatrix}.$$

2.1.3. Operators

2.1.3.1. Skew-Symmetric

We denote $[\,\cdot\,]_\times : \mathbb{R}^3 \rightarrow \mathbb{R}^{3\times 3}$ as the usual skew-symmetric operator, which maps an angular rates vector $w \in \mathbb{R}^3 : w = [w_x \; w_y \; w_z]^T$ into a 3×3 matrix

$$[w]_x = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix}.$$

2.1.3.2. vec-operator

For any matrix $A \in \mathbb{M}^{m\times n}$ the vec-operator is defined as

$$vec(A) = (a_{11}, \cdots, a_{m1}, a_{12}, \cdots, a_{m2}, a_{1n}, \cdots, a_{mn})^T,$$

i.e., the entries of A are stacked columnwise, forming a vector of length $m \times n$.

2.2. Attitude Calculation

The system coordinate frames are defined so that the x axis points forward, the y axis points right, and the z axis points down. Based on this coordinate frame, the instantaneous estimated roll $\hat{\phi}$ and pitch $\hat{\theta}$ (depicted in side view Fig. 2.1) angles are given by (Troni, 2013)

$$\hat{\phi} = atan2(-a_y, -a_z), \tag{2.1a}$$

$$\hat{\theta} = atan2(a_x, \sqrt{a_y^2 + a_z^2}), \tag{2.1b}$$

where a_x, a_y and a_z are the x, y and z, respectively, accelerometer measurements components. The measured magnetic field in the instrument frame, ${}^{i}m \in \mathbb{R}^3$, can be transformed to the local frame by the relation ${}^{l}m = {}^{l}_{i}R(t){}^{i}m$, where ${}^{l}_{i}R(t) \in \mathbb{R}^{3\times3}$ is a rotation matrix using pitch and roll estimates. Then the instantaneous estimated heading $\hat{\psi}$ (depicted in upper view Fig. 2.1) can be computed as (Troni, 2013)

$$\hat{\psi} = atan2(-{}^{l}m_y, {}^{l}m_x) - \psi_0, \tag{2.2}$$

where ψ_0 is the known local magnetic field variation and ${}^{l}m_x$, ${}^{l}m_y$ and ${}^{l}m_z$ are the x, y, and z, respectively, local frame magnetometer measurements components.

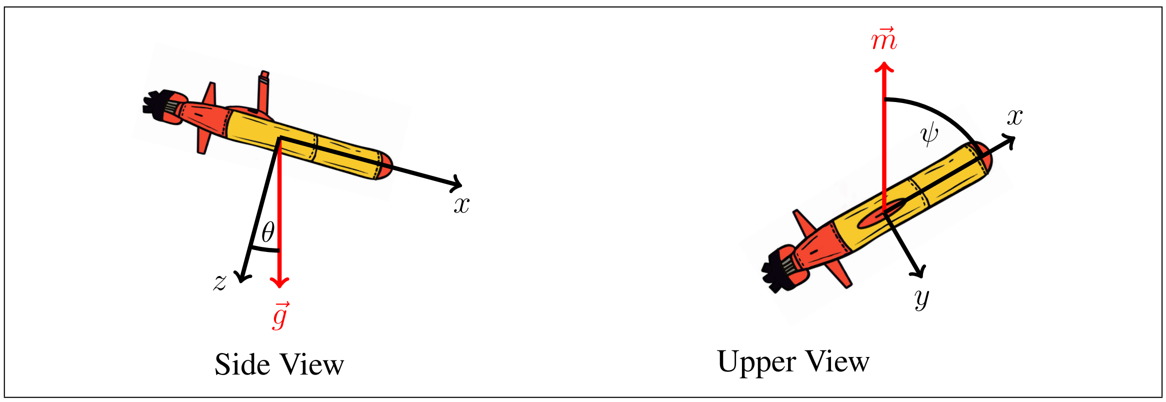


Figure 2.1. Diagram illustrating the attitude of an underwater vehicle from both a side and top view. The side view displays the system's pitch, while the top view shows the system's heading.

2.3. Doppler Dead Reckoning Navigation

Doppler Velocity Loggers (DVLs) are commonly used on underwater vehicles to measure three-axis linear velocity in the sensor frame, provided with respect to the seafloor when the DVL has a bottom lock. Based on the attitude of the system and the mounting offset of the sensor with respect to the vehicle, the velocity measurements can be transformed into the world frame by (Troni, 2013)

$$ {}^{w}v(t) = {}^{w}_{v}R(t)\, {}^{v}_{d}R\, {}^{d}v(t), \tag{2.3} $$

where ${}^{v}_{d}R$ is the constant rotation matrix from the instrument coordinate frame to the vehicle coordinate frame, ${}^{w}_{v}R(t)$ is the rotation matrix from the vehicle coordinate frame to the inertial world coordinate frame, computed with the pitch, roll, and heading, ${}^{d}v(t)$ is the vehicle's velocity in the DVL's coordinate frame, and ${}^{w}v(t)$ is the world frame vehicle velocity. This velocity can later be integrated to provide the dead reckoning position

$$ {}^{w}p(t) = {}^{w}p(t_0) + \int_{t_0}^{t} v(\tau) d\tau. \tag{2.4} $$

2.4. Factor Graphs

Factor graphs are graphical representations of probabilistic or graphical models that consist of nodes representing unknown random variables ($x_i \in \mathcal{X}$) in the model and edges representing the dependencies or relationships between these variables. The edges are associated with factors ($f_i \in \mathcal{F}$), which are probabilistic constraints derived from measurements, prior knowledge, or relationships between variables. Factors can be categorized as unary factors when connecting to a single node or binary factors if they connect two or more nodes (Dellaert, 2012).

As a demonstrative example, let us consider the simple factor graph in Fig. 2.2, which can model the state of a system. This factor graph has variables x_1, x_2, and x_3, which represent the system's state over time and are rendered in the figure as circle nodes. In yellow, we have one unary factor on the first state x_1 that encodes our prior knowledge about x_1, while in red, we also have unary factors that represent the belief about the state of the system given an external measurement, e.g., if the state encodes the position of the system, this factor can represent a GPS measurement. Finally, in green, we have binary factors that encode the relationship between successive states, e.g., the relative rotation in the plane computed by the change in the heading.

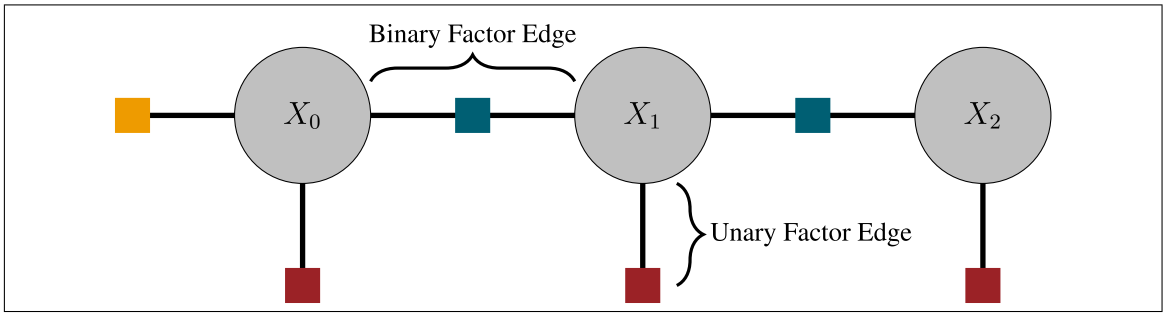


Figure 2.2. Factor graph example.

The factor graph is a concise and intuitive way of representing a model that is well-suited for performing various types of probabilistic inference tasks, such as bias estimation given a set of measurements for sensor calibration. Compared to traditional filtering methods, factor graphs have been shown to be better equipped to handle nonlinear process and measurement models, with a significant advantage in processing time for large-scale problems (Dellaert & Kaess, 2006; Dai, Liu, Hao, Ren, & Yang, 2022).

As a bipartite graph, the value of the factor will be modeled as (2.5), which encodes the connectivity of a factor graph for each factor f_i connected to a subset of variables $\mathcal{X}_i$.

$$f(X_0, \cdots X_n) = \prod f_i(\mathcal{X}_i) \tag{2.5}$$

Given the random set of variables $\mathcal{X}$ and a set of measurements $\mathcal{Z}$, we can define a factor graph as a Gaussian probabilistic distribution that represents the posterior density $P(\mathcal{X}_i|\mathcal{Z}_i)$ for a subset of variables $\mathcal{X}_i$ and measurements $\mathcal{Z}_i$. Unary factors can be represented as measurement likelihood with a Gaussian noise model (Dellaert & Kaess, 2006), which is only evaluated as a function of q since the measurement m is considered known

$$L(x; m) = \exp\left\{-\frac{1}{2}||h(x) - m||^2_\Sigma\right\}. \tag{2.6}$$

Where m is the measurement, x is the unknown variable, $h(x)$ is a nonlinear measurement model, $|| \cdot ||$ is the Mahalanobis distance, and Σ is the noise covariance matrix.

Based on the unary factor and a prior belief over the unknown variable $P(x_0)$, we can rewrite the joint probability model of the factor graph as follows.

$$P(X, Z) = P(x_0) \prod_{i=1}^{N} P(z_i|x_i) = P(x_0) \prod_{i=1}^{N} L(x_i, z_i) \tag{2.7}$$

To find the optimal set of parameters $\mathcal{X}^*$, we aim to obtain the maximum a posteriori (MAP) estimate by maximizing the joint probability $P(X, Z)$ (2.7), which leads us to a nonlinear least squares problem (2.8). Using the factor graph structure and sparse connections in this estimation process reduces the computational complexity (Dellaert & Kaess, 2006; Dellaert, 2012).

$$\mathcal{X}^* \triangleq \underset{x}{\operatorname{argmax}}\ P(\mathcal{X}|\mathcal{Z}) = \underset{x}{\operatorname{argmin}}\ -\log P(\mathcal{X}, \mathcal{Z}) \tag{2.8}$$

2.4.1. Computational Implementation

The factor graph concept can be implemented in Python programs using the wrapper available from the C++ GTSAM/iSAM library[1]. This section covers the basic concepts for constructing and optimizing factor graphs, providing a foundation for understanding the different pseudocode algorithms used to explain the implemented methods.

In GTSAM, a factor graph is just a specification of the probability density $P(\mathcal{X}|\mathcal{Z})$. The corresponding FactorGraph class and its derived classes, such as NonLinearFactorGraph, do not contain ”solutions”. Instead, a separate type Values is used to specify specific values for the different states of the system (x_i), which can then be used to evaluate the probability, or, more commonly, the error, associated with particular values. The Values class holds values in different moments in a defined key-value pair structure, and these values must be initialized with the initial guess for each state component.

[1]The GTSAM source code is available through https://github.com/borglab/gtsam/blob/develop/gtsam.

Once the factor graph object has been created, we can add the different model factors. For unary factors, GTSAM requires as arguments the keys of the state values affected by the factor, the measurement model to compute the factor's residual, and the noise model, which can be defined as a diagonal matrix that gives the diagonal values of the covariance matrix.

The optimization of the factor graph can be computed with the factor graph with all the measurements included (GTSAM), or it can be optimized incrementally (iSAM), computing the optimal values every predefined amount of added factors. The optimization can be performed using different optimization options, such as Gauss-Newton, Levenberg-Marquardt, Dogleg, among others.

2.5. Methods

This section presents the methods that will be used as benchmarks for evaluating the proposed MAGICAL algorithm in Chapters 5 and 6.

2.5.1. TWOSTEP

Alonso and Shuster presented the widely-cited TWOSTEP method for estimating a magnetometer's bias (Alonso & Shuster, 2002b), which they later extended to estimate the scale and non-orthogonality factors as well (Alonso & Shuster, 2002a) using an iterative least squares minimization. The TWOSTEP algorithm is an improvement and extension of Gambhir's centering algorithm (Gambhir, 1975), which did not properly account for the correlations introduced by the centering process or attempt to correct for the potentially significant amount of data discarded during the centering process. The TWOSTEP algorithm overcomes these drawbacks.

The first step of this method proposes an optimization problem that minimizes the negative-log-likelihood of a cost function derived from the magnetometer's model through the derived scalar measurements and scalar measurement noise, defined as an alternative to avoid the use of the unknown attitude of the system. However, the minimization of this model is complicated by the fact that the negative-log-likelihood function is quartic in the magnetometer bias and, therefore, admits multiple minima and maxima, which does not guarantee convergence of the method when initialized in the trivial solution $\mathbf{m_b} = 0$ and $A = \mathbb{I}_3$. Therefore, any infinite process must start with a reasonable estimate of the bias that the centering approximation provides in a closed-form solution. It is worth mentioning that the centered estimate itself provides a consistent estimate of the magnetometer biases, which will provide adequate accuracy in most cases.

The second step consists of using the centered estimate as an initial value and computing the corrected estimate by applying an iterative optimization based on the Gauss-Newton method to the full negative-log-likelihood function. For further details of this method, refer to the section ”Estimation of the Magnetometer Bias, Scale Factors, and Non-orthogonality Corrections” in the original publication (Alonso & Shuster, 2002a). This method was implemented using Python 3.8, based on (Dinale, 2013), which provides an excellent overview of the TWOSTEP method. Appendix C.1 of (Dinale, 2013) provides a reference Matlab implementation of the TWOSTEP method.

Algorithm 1 TWOSTEP Implementation

B $\leftarrow$ `magnetometer measurements`

H $\leftarrow$ `local magnetic field`

$\Sigma \leftarrow$ `noise covariance`

First Step

$z_k \leftarrow$ SCALAR_MEASUREMENTS(B, H)

$\mu_k \leftarrow$ NOISE_MEAN(Σ)

$[\sigma_k^2, \bar{\sigma}^2] \leftarrow$ NOISE_COVARIANCE(Σ)

$L_k \leftarrow$ MODEL_PARAMETRIZATION(B) $\triangleright$ (Alonso & Shuster, 2002a, eq. 51)

Algorithm 1 TWOSTEP Implementation (continued)

$[\bar{\mu}, \tilde{\mu_k}] \leftarrow$ CENTER_DATA$(\mu_k,\ \sigma_k^2,\ \bar{\sigma}^2)$

$[\bar{z}, \tilde{z_k}] \leftarrow$ CENTER_DATA$(z_k,\ \sigma_k^2,\ \bar{\sigma}^2)$

$[\bar{L}, \tilde{L_k}] \leftarrow$ CENTER_DATA$(L_k,\ \sigma_k^2,\ \bar{\sigma}^2)$

$\tilde{\theta}_0 \leftarrow$ CENTERED_ESTIMATE$(\tilde{L_k},\ \tilde{\mu_k},\ \tilde{z_k},\ \sigma_k^2)$

Second Step

$\theta \leftarrow \theta_0$ ▷ Initial state for iterative step

while `error` $>$ `stop_tol` AND `n_iter` $>$ `n_max` **do**

  $\theta \leftarrow$ GAUSS_NEWTON_OPTIMIZATION(`quadratic_model`, θ)

end while

2.5.2. Ellipsoid Fit

The most widely used calibration method in commercial AHRS is the Ellipsoid Fit. This method can estimate hard-iron and soft-iron values based on the geometric model of an ellipsoid and magnetometer measurements. Since an ellipsoid is a type of concoid, its equation can be expressed as a general equation of a concoid in 3D space (Fang et al., 2011, eq. 6)

$$ax^2 + by^2 + cz^2 + 2dxy + 2exz + 2fyz + 2gx + 2hy + 2iz + j = 0. \tag{2.9}$$

Moreover, we can add the constraint $a + b + c + 3 = 0$, which removes one parameter from the computations. If we apply the change of variable

$$a = \alpha + \beta - 1, \quad b = \alpha - 2\beta - 1, \quad c = \beta - 2\alpha - 1, \tag{2.10}$$

we can convert the general equation into a linear least squares model, which can be solved in closed form:

$$D^T DX = D^T(x^2 + y^2 + z^2), \tag{2.11}$$

where $D = \begin{bmatrix} x^2+y^2-2z^2 & x^2+z^2-2y^2 & 2xy & 2xz & 2yz & 2x & 2y & 2z & 1 \end{bmatrix}$ is formed with the magnetometer measurements and $X = \begin{bmatrix} \alpha & \beta & d & e & f & g & h & i & j \end{bmatrix}^T$. As the undistorted magnetic field measurements belong to the surface of a sphere, based on the computed parameters, we can rewrite 2.9 as

$$\begin{bmatrix} a & d & e & g \\ d & b & f & h \\ e & f & c & i \\ g & h & i & j \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \Leftrightarrow \begin{bmatrix} ax+dy+ez \\ dx+by+fz \\ ex+fy+cz \\ gx+hy+iz \end{bmatrix} = \begin{bmatrix} -g \\ -h \\ -i \\ -j \end{bmatrix}. \tag{2.12}$$

From Equation 2.12, we can compute the first magnetometer calibration parameter, which is the bias or center, by solving the linear system

$$\mathbf{m_b} = -\begin{bmatrix} a & d & e \\ d & b & f \\ e & f & c \end{bmatrix}^{-1} \begin{bmatrix} g \\ h \\ i \end{bmatrix}, \tag{2.13}$$

which is later used to form a transformation matrix with an identity rotation matrix. This will translate the ellipsoid to the center and, finally, from

$$R = \begin{bmatrix} 1 & 0 & 0 & m_{bx} \\ 0 & 1 & 0 & m_{by} \\ 0 & 0 & 1 & m_{bz} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a & d & e & g \\ d & b & f & h \\ e & f & c & i \\ g & h & i & j \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & m_{bx} \\ 0 & 1 & 0 & m_{by} \\ 0 & 0 & 1 & m_{bz} \\ 0 & 0 & 0 & 1 \end{bmatrix}^T = \begin{bmatrix} R_{00} & R_{01} \\ R_{10} & R_{11} \end{bmatrix}, \tag{2.14}$$

we can compute the soft-iron as $A = -R_{00}/R_{11}$, where $R_{00} \in \mathbb{R}^{3\times3}$ and $R_{11} \in \mathbb{R}$. This method was implemented using Python 3.8, based on the implementation proposed by Bazhin et al. (Bazhin, Vrba, & Zogopoulos-Papaliakos, 2022).

Algorithm 2 Ellipsoid Fit Implementation

$[x, y, z] \leftarrow$ `magnetometer measurements`

`D` $\leftarrow [x^2 + y^2 - 2z^2, x^2 + z^2 - 2y^2, 2xy, 2xz, 2yz, 2x, 2y, 2z, 1]$

General Equation Parameters

$[\alpha, \beta, d, e, f, g, h, i, j] \leftarrow \texttt{inv}(D^T D)\texttt{.dot}(D^T(x^2 + y^2 + c^2))$

$a \leftarrow \alpha + \beta - 1$

$b \leftarrow \alpha - 2\beta - 1$

$c \leftarrow \beta - 2\alpha - 1$

$J \leftarrow [[a,\ d,\ e,\ g],\ [d,\ b,\ f,\ h],\ [e,\ f,\ c,\ i],\ [g,\ h,\ i,\ j]]$

Calibration Parameters

`hard_iron` $\leftarrow$ `-inv`$([[a,\ d,\ e],\ [d,\ b,\ f],\ [e,\ f,\ c]])$`.dot`$([[j],\ [h],\ [i]])$

$T \leftarrow$ `identity`(4)

$T[:3, 3] \leftarrow$ `hard_iron`

$R \leftarrow T\texttt{.dot}(J)\texttt{.dot}(T^T)$

`soft_iron` $\leftarrow$ - $R[:3, :3]\ /\ R[3, 3]$

2.5.3. GTSAM MagFactor3

The GTSAM MagFactor3[2] is a pose graph factor that calibrates the local Earth magnetic field and magnetometer bias. The original C++ implementation uses the idea of two unknowns: a scale factor (α) that multiplies the local magnetic field (${}_nM$) and a magnetometer bias (b) that is added to the multiplication. The measurement is then composed of the following equation:

$$m_m(t) = \alpha \cdot {}_bR_w(t) \cdot {}_nM + b. \tag{2.15}$$

[2]The GTSAM MagFactor source code is available through https://github.com/borglab/gtsam/blob/develop/gtsam/navigation/MagFactor.h.

As we can see from Equation (2.15), the drawbacks of this approach do not allow us to compute non-orthogonality. Instead, it assumes that the measured magnetic field belongs to a sphere and requires the vehicle's attitude, which is also affected by the magnetometer biases.

As part of the implementation, it is required to compute the Jacobian of the model, which is as follows:

$$\frac{\partial m_m(t)}{\alpha} = {}_bR_w(t) \cdot {}_nM \qquad \frac{\partial m_m(t)}{b} = \mathbb{I}_3. \tag{2.16}$$

This method was implemented using Python 3.8, based on the C++ code implementation available through the GTSAM MagFactor source code[3], using the Python wrapper that provides the GTSAM module for the factor graph construction.

Algorithm 3 MagFactor3 Implementation

```
Initialize Factor Graph
graph ← NonLinearFactorGraph( )
noise ← NoiseModel(σ_xx, σ_yy, σ_zz)
keys ← [(α(0), 0.0), (b(0), [0.0, 0.0, 0.0])]        ▷ Initial guess for the scale and bias
values ← Values(keys)
function Residual(α, b, m_m, nM, bRw)
    local_magfield ← bRw.rotate(nM)
    h_x ← a * local_magfield + b
    error ← h_x - magfield
    jacobian ← [bRw.rotate(nM), identity(3)]
    return [error, jacobian]
end function
```

[3]https://github.com/borglab/gtsam/blob/develop/gtsam/navigation/MagFactor.h

Algorithm 4 MagFactor3 Implementation (continued)

```
Process Measurements
for (magnetometer_measurement, attitude) do
    factor ← FACTOR(noise, values, RESIDUAL(α, b, m_m, nM, bRw))
    graph.add_factor( )
    if optimization_window then
        values ← graph.optimize( )
    end if
end for
```

3. PROPOSED CALIBRATION APPROACH

In this section, we report four methods based on the novel MAGICAL method to estimate the complete calibration of a three-axis magnetometer, i.e., hard-iron and soft-iron, and a three-axis gyroscope using magnetometer and angular rate measurements in the instrument frame, i.e., the attitude of the instrument is not required.

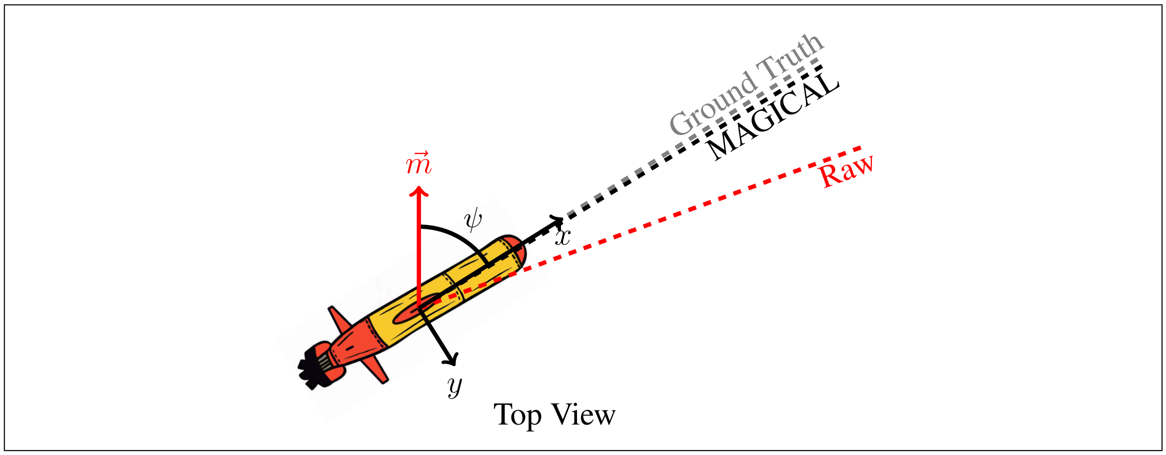


Figure 3.1. Diagram illustrating an underwater vehicle's dead reckoning position using different magnetic field sources for the heading estimation with the corresponding trajectories represented with dashed lines.

3.1. Sensor Error Model

As described in chapter 1, in operational conditions, magnetometers are subject to biases known as hard-iron and soft-iron effects, which can cause inaccuracies in their measurements. We assume these biases are static or vary slowly over time and hence can be treated as constants. The magnetometer model is then given by

$$\mathbf{m_m}(t) = A(\mathbf{m_t}(t) + \mathbf{m_b}), \tag{3.1}$$

where $\mathbf{m_m}(t) \in \mathbb{R}^3$ is the noise-free magnetic field measurement in the sensor's frame, $\mathbf{m_t}(t) \in \mathbb{R}^3$ is the noise-free magnetic field real value in the sensor's frame, $A \in \mathbb{R}^{3\times 3}$ is the soft-iron, represented by a constant fully populated positive definite symmetric (PDS) matrix, and $\mathbf{m_b} \in \mathbb{R}^3$ is a constant pseudo-hard-iron, that once scaled by A will give us the magnetometer's hard-iron.

In contrast, gyroscopes are affected by the constant sensor bias and can be represented as

$$\mathbf{w_m}(t) = \mathbf{w_t}(t) + \mathbf{w_b}, \tag{3.2}$$

where $w_m(t) \in \mathbb{R}^3$ is the noise-free gyroscope measurement in the sensor's frame, $\mathbf{w_t}(t) \in \mathbb{R}^3$ is the noise-free gyroscope real value in the sensor's frame, and $\mathbf{w_b} \in \mathbb{R}^3$ is the constant gyroscope bias.

3.2. System Model

Like other multi-axis field sensors, a magnetometer measures the Earth's local magnetic field in instrument coordinates, which can be considered constant and fixed with respect to the world frame of reference. From (3.1), we can clear the true magnetic field ($\mathbf{m_t}(t)$) and convert it to world coordinates given a rotation matrix R. Then, we can differentiate the equation with respect to time, removing the local magnetic field from the system's model, which yields to

$$\dot{R}(A^{-1}\mathbf{m_m}(t) - \mathbf{m_b}) + RA^{-1}\dot{\mathbf{m}}_\mathbf{m}(t) = 0. \tag{3.3}$$

Using the standard equation $\dot{R}(t) = R(t)\,[w(t)]_\times$ (Lynch & Park, 2017), and also considering the gyroscope bias, for a more comprehensive AHRS calibration that considers both magnetometer and gyroscope calibrations, we obtain a nonlinear system model where the instrument attitude $R(t)$ does not appear.

$$[\mathbf{w_m}(t) - \mathbf{w_b}]_\times (A^{-1}\mathbf{m_m}(t) - \mathbf{m_b}) + A^{-1}\dot{\mathbf{m}}_\mathbf{m}(t) = 0 \tag{3.4}$$

From (3.4), we wish to estimate the hard-iron ($\mathbf{m_b}$), soft-iron (A), and gyroscope bias ($\mathbf{w_b}$) from the gyroscope ($\mathbf{w_m}(t)$) and magnetometer ($\mathbf{m_m}(t)$) measurements, either for the full magnetometer and gyroscope calibration or if we assume that the gyroscope is already bias-compensated, we can write the system model as a linear system over the magnetometer calibration parameters

$$(\mathbf{m_m}(t)^T \otimes [\mathbf{w_m}(t)] + \dot{\mathbf{m}}_\mathbf{m}(t)^T \otimes \mathbb{I}_3) \cdot vec(A^{-1}) - [\mathbf{w_m}(t)] \cdot \mathbf{m_b} = 0. \tag{3.5}$$

To solve the system model, the proposed solutions include four methods based on two approaches to solving the nonlinear model: (i) a least squares-based approach and (ii) a pose graph-based approach.

3.3. Iterative Least-Squares Approach

A first approach to compute the calibration parameters is to solve (3.4) using least squares, which, as the model is nonlinear and therefore can not be solved in closed form, uses an iterative approach. The cost function for the least squares minimization problem is defined based on (3.4), or alternatively (3.5), as shown in (3.6). However, the cost function requires the derivative of magnetic field measurements ($\dot{\mathbf{m}}_\mathbf{i}$), which are not directly available from magnetometers and require numerical differentiation, leading to potential noise in the results.

$$SSR(A, m_b, w_b) = \sum_{i=0}^{n} ||\, [\mathbf{w_i} - \mathbf{w_b}]_\times (A^{-1}\mathbf{m_i} - \mathbf{m_b}) + A^{-1}\dot{\mathbf{m}}_\mathbf{i}||^2 \tag{3.6}$$

To solve (3.6), we define the vector of parameters

$$\vec{a} = (A_{00}\ A_{01}\ A_{02}\ A_{11}\ A_{12}\ A_{22})^T, \tag{3.7a}$$

$$\mathbf{x} = (a\ m_b\ w_b)^T, \tag{3.7b}$$

where $\mathbf{a}$ is the vector of the unique six upper triangular terms of the soft-iron matrix, A.

The solution to these nonlinear least squares problems is typically found through trust-region methods (Nocedal & Wright, 2006; Rosen, Kaess, & Leonard, 2014), which restrict the update step within a defined trust region to ensure local linearity of the cost function. Levenberg-Marquardt (Nocedal & Wright, 2006) is a widely used approach, but for bound-constrained minimization problems, the trust-region reflective policy (Branch, Coleman, & Li, 1999) has been shown to be highly reliable.

To maintain the method's scale-invariant property post-optimization, we assume the initial magnetic field lies on a unitary sphere's boundary. A constraint is imposed to ensure the volume of the resulting ellipsoid, due to soft-iron and hard-iron effects, remains constant. It can be demonstrated (demonstration in Appendix A.1) that this constraint is satisfied by scaling the soft-iron matrix by $(\lambda_0\lambda_1\lambda_2)^{-1/3}$, where $\lambda_0, \lambda_1, \lambda_2$ are the eigenvalues of the soft-iron matrix.

This method was implemented using Python 3.8, using the structure outlined in the algorithm 5.

Algorithm 5 MAGICAL Iterative Least Squares Implementation

function COST_FUNCTION(m, $\dot{m}$, w, a, m_b, w_b)

    SI ← [[a_0, a_1, a_2], [a_1, a_3, a_4], [a_2, a_4, a_5]]

    [eigval_0, eigval_1, eigval_2] ← EIGVALS(SI)

    SI_scaled ← SI * ((eigval_0 * eigval_1 * eigval_2) ** ($-1/3$)

    residual_error ← 0.0

Algorithm 5 MAGICAL Iterative Least Squares Implementation (Continued)

for m_i, $\dot{m}_i$, w_i **do**

`sensor_model` $\leftarrow || \left[w_i - w_b\right]_\times (SI_{scaled}{}^{-1} m_i - m_b) + SI_{scaled}{}^{-1} \dot{m}_i ||^2$

`residual_error += sensor_model`

end for

return `residual_error`

end function

`SI` $\leftarrow$ [[1.0, 0.0, 0.0], [0.0, 1.0, 0.0], [0.0, 0.0, 1.0]]

`HI` $\leftarrow$ [0.0, 0.0, 0.0]

`Wb` $\leftarrow$ [0.0, 0.0, 0.0]

while `residual_error` $>$ `stop_tolerance` **do**

`SI, HI, Wb, residual_error` $\leftarrow$ OPTIMIZE(m, $\dot{m}$, w, `SI`, `HI`, `Wb`)

end while

3.4. Factor Graph Approach

The second method for calculating the calibration parameters involves solving (3.4) by modeling the system as a factor graph. As explained in chapter 2.4, the optimization problem can be represented by a single variable node with $2m_f$ unary factors. m_f factors represent the residual error of each measurement, while the other m_f factors relate to the constraint of a unitary norm over the unique upper triangular terms of the soft-iron matrix to prevent the algorithm from converging to the trivial solution $\mathbf{x} = 0$.

To reduce the computational load of the graph, we define m_f as the number of measurements, n, divided by the averaging window's length (θ). By averaging θ measurements before incorporating them into the graph, we can decrease the number of factors and effectively smooth the raw measurements while filtering out high-frequency noise in the signal, preserving the underlying trend. To ensure real-time operation during long-duration tasks, the averaging window's length is set relative to the sensor's frequency, which allows the factor graph to manage the same period, irrespective of the sensor's frequency.

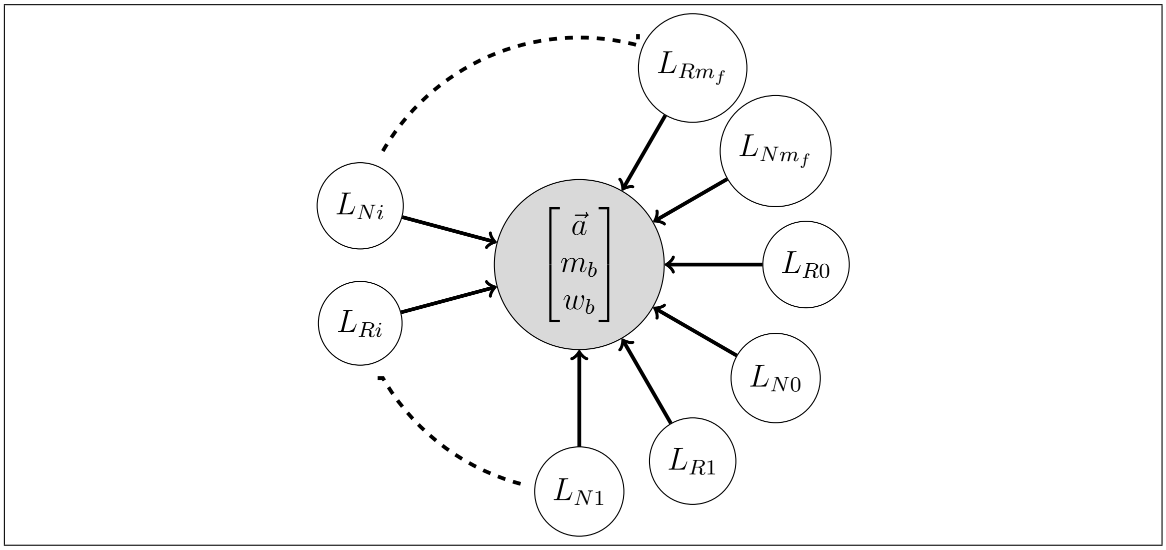


Figure 3.2. A factor graph representation, depicting the residual factor, L_{Ri}, and the soft-iron norm factor, L_{Ni}.

The decision to use a single node instead of one node for each average set, i.e., m_f nodes, is based on the assumption that the calibration parameters remain constant. If multiple nodes were used, a binary factor would be necessary between each state, i.e., $m_f - 1$ in total, to ensure equality, but this may not always be feasible due to the probabilistic nature of factor graphs. Modeling the system as a single node ensures one set of calibration values and spares the use of the $m_f - 1$ constraints. It is worth noting that the same factors introduced in this thesis, namely (3.8) and (3.10), can be added to different value nodes in the graph to enable biases to vary over time and be integrated with SLAM graphs. However, this integration is beyond the scope of this thesis.

The factor graph can be constructed in two ways, for batch or real-time solutions. The first method involves calculating the calibration as a post-processing step, where all n measurements are first collected, and the optimization process is then carried out with all $2m_f$ factors. The second approach involves constructing the graph incrementally, where factors are added to the graph as they become available, and the optimization is performed after a specified number of measurements have been received.

As per (3.4), a unary factor can be defined for the residual of each magnetometer-gyroscope measurement pair. The equation for this factor is shown in (3.8), where $h_r(x)$ represents the residual model (3.4). The model requires the derivative of magnetic field measurements ($\dot{\mathbf{m}}_\mathbf{i}$), which are not directly available from magnetometers and need numerical differentiation. This differentiation process may lead to potential noise in the results.

$$L_R(x; (z_{mag}, z_{gyro}) = \exp\left\{-\frac{1}{2}||h_r(x)||^2_\Sigma\right\} \tag{3.8}$$

We need to compute the algebraic Jacobian of the residual model to use the unary factor defined in (3.8). This can be achieved using the Kronecker product and vec-operator, as shown in (3.9). For conciseness, we do not expand these equations further in this publication. In (3.9), the matrix C represents the inverse of the soft-iron matrix, the vector $\mathbf{c}$ contains the upper triangular terms of matrix C, and i denotes the ith sample. Note that the magnetic field differentiation is not directly available and must be computed numerically.

$$\frac{h_{ri}(x)}{\mathbf{c}} = (\mathbf{m_i}^T \otimes [\mathbf{w_i} - \mathbf{w_b}] + \dot{\mathbf{m}}_\mathbf{i}^T \otimes \mathbb{I}_3) \cdot \frac{\partial vec(C)}{\partial \mathbf{c}} \tag{3.9a}$$

$$\frac{h_{ri}(x)}{\mathbf{m_b}} = [\mathbf{w_b} - \mathbf{w_i}] \cdot \frac{\partial \mathbf{m_b}}{\partial \mathbf{m_b}} \tag{3.9b}$$

$$\frac{h_{ri}(x)}{\mathbf{w_b}} = -\frac{\partial\left(\mathbf{m_i} \otimes [\mathbf{w_b}]\right)}{\partial \mathbf{w_b}} \cdot vec(C) + \frac{\partial [\mathbf{w_b}]}{\partial \mathbf{w_b}} \cdot \mathbf{m_b} \tag{3.9c}$$

The soft-iron norm factors (L_N) are defined as the error given by the difference between the Frobenius norm of the unique soft-iron's upper triangular terms and 1, $N(x) = ||\,\mathbf{c}\,||_F - 1$. The factor is given by

$$L_N(x) = \exp\left\{-\frac{1}{2}\,||N(x)||^2_\Sigma\right\}. \tag{3.10}$$

The Jacobian of this factor can be found using

$$\frac{\partial N(x)}{\partial \mathbf{c}} = \frac{(\mathbf{c})^T}{||\,\mathbf{c}\,||_F}. \tag{3.11}$$

As Section II.D mentioned, constructing the graph requires solving a nonlinear least squares problem with sparsity properties. To address this issue, Rosen et al. (Rosen et al., 2014) proposed RISE, an incremental trust-region method for robust online sparse least-squares estimation. Compared to current state-of-the-art sequential sparse least-squares solvers, RISE offers improved robustness against nonlinearity in the objective function and numerical ill-conditioning and takes advantage of recent advancements in incremental optimization for fast online computation.

This method was implemented using Python 3.8, using the structure outlined in the algorithm 6.

Algorithm 6 MAGICAL Factor Graph Implementation

```
function MResidual(c, m_b, w_b, m, ṁ, w)                          ▷ From 3.8
    C ← [[c_0, c_1, c_2], [c_1, c_3, c_4], [c_2, c_4, c_5]]
    jacobian ← JACOBIAN_MResidual(c, m_b, w_b, m, ṁ, w)
    error ← [w_m − w_b]_× (C m_m − m_b) + C ṁ_m
    return [error, jacobian]
end function
function NResidual(c)                                              ▷ From 3.10
    jacobian ← JACOBIAN_NResidual(c)
    error ← ||c||_F − 1
    return [error, jacobian]
end function
```

Algorithm 6 MAGICAL Factor Graph Implementation (Continued)

function JACOBIAN_MRESIDUAL(`c`, m_b, w_b, m, $\dot{m}$, w) ▷ From 3.9

`J00` $\leftarrow [[\dot{m}_x], [m_x * (-w_{bz} + w_z)], [m_x * (w_{by} - w_y)]]$

`J01` $\leftarrow [[m_x*(w_{bz}-w_z)+\dot{m}_y], [m_y*(w_z-w_{bz})+\dot{m}_x], [m_x*(w_x-w_{bx})+m_y*(w_{by}-w_y)]]$

`J02` $\leftarrow [[m_x*(w_y-w_{by})+\dot{m}_z], [m_x*(w_{bx}-w_x)+m_z*(w_z-w_{bz})], [m_z*(w_{by}-w_y)+\dot{m}_x]]$

`J03` $\leftarrow [[m_y * (w_{bz} - w_z)], [\dot{m}_y], [m_y * (w_x - w_{bx})]]$

`J04` $\leftarrow [[m_y*(w_y-w_{by})+m_z*(w_{bz}-w_z)], [m_y*(w_{bx}-w_x)+\dot{m}_z], [m_z*(w_x-w_{bx})+\dot{m}_y]]$

`J05` $\leftarrow [[m_z * (w_y - w_{by})], [m_z * (w_{bx} - w_x)], [\dot{m}_z]]$

`J0` $\leftarrow$ `[J00, J01, J02, J03, J04, J05]`

`J10` $\leftarrow [[0], [w_{bz} - w_z], [w_y - w_{by}]]$

`J11` $\leftarrow [[w_z - w_{bz}], [0], [w_{bx} - w_x]]$

`J12` $\leftarrow [[w_{by} - w_y], [w_x - w_{bx}], [0]]$

`J1` $\leftarrow$ `[J10, J11, J12]`

`J20` $\leftarrow [[0], [-m_{bz}+c_2*m_x+c_4*m_y+c_5*m_z], [m_{by}-c_1*m_x-c_3*m_y-c_4*m_y]]$

`J21` $\leftarrow [[m_{bz}-c_2*m_x-c_4*m_y-c_5*m_z], [0], [-m_{bx}+c_0*m_x+c_1*m_y+c_2*m_z]]$

`J22` $\leftarrow [[-m_{by}+c_1*m_x+c_3*m_y+c_4*m_z], [m_{bx}-c_0*m_x-c_1*m_y-c_2*m_z], [0]]$

`J2` $\leftarrow$ `[J20, J21, J22]`

return `[J0, J1, J2]`

end function

function JACOBIAN_NRESIDUAL(`c`) ▷ From 3.11

$\alpha \leftarrow ||\, c\, ||_F$

`J` $\leftarrow [c_0/\alpha,\ c_1/\alpha,\ c_2/\alpha,\ c_3/\alpha,\ c_4/\alpha,\ c_5/\alpha]$

return J

end function

Algorithm 6 MAGICAL Factor Graph Implementation (Continued)

```
Initialize Factor Graph
graph ← NONLINEARFACTORGRAPH( )
mnoise ← NOISEMODEL(σ^m_xx, σ^m_yy, σ^m_zz)
nnoise ← NOISEMODEL(σ^n_xx, σ^n_yy, σ^n_zz)
keys ← [(c(0), [0.0, 0.0, 0.0, 0.0, 0.0, 0.0]), (b(0), [0.0, 0.0, 0.0]), (w(0), [0.0, 0.0, 0.0])]
v ← VALUES(keys)
[θ, Γ] ← average_window, optimization_window
[idx ← 0
[sum_m, sum_mdot, sum_w] ← 0.0, 0.0, 0.0  state
while m_i, ṁ_i, w_i do
    if idx % θ != 0.0 then
        [sum_m, sum_mdot, sum_w] ← sum_m + m_i, sum_mdot + ṁ_i, sum_w + w_i
    else
        [ṁ_i, m_i, w_i] ← sum_m / θ, sum_mdot / θ, sum_w / θ
        mf ← FACTOR(mnoise, v, MRESIDUAL(c, m_b, w_b, m, ṁ, w))
        nf ← FACTOR(nnoise, v, NRESIDUAL(c))
        graph.add_factor(mf, nf)
        if online_mode AND idx % Γ != 0.0 then
            values ← graph.optimize( )
        end if
    end if
end while
if NOT online_mode then
    values ← graph.optimize( )
end if
```

4. MAGICAL PERFORMANCE EVALUATION

This chapter introduces the calibration algorithm and performance metrics used for evaluation in this thesis. These algorithms and metrics ensure data accuracy and enable quantitative comparisons of research outcomes.

4.1. Calibration Algorithms

We compared the performance of seven methods for three-axis magnetometer calibration and, optionally, for three-axis gyroscope calibration. These methods can be divided into batch and real-time solutions, with the proposed methods highlighted in bold. The batch methods are as follows:

(i) ***MAGICAL-LS***: The calibration parameters are estimated using the least-squares method approach outlined in section 3.3, assuming that the angular velocity sensor is already bias-compensated. The value of $\dot{\mathbf{m}}(t)$ is numerically computed through first-order numerical differentiation of $\mathbf{m}(t)$.

(ii) ***MAGICAL-NLS***: The calibration parameters are estimated using the least-squares approach described in section 3.4.

(iii) ***MAGICAL-BFG***: The calibration parameters are estimated using the batch mode factor graph approach described in section 3.4, where all the factors are added to the factor graph before optimization.

(iv) *TWOSTEP*: The calibration parameters are estimated using the widely cited TWOSTEP method, presented in section 2.5.1.

(v) *Ellipsoid Fit*: The calibration parameters are estimated using the widely used Ellipsoid Fit method, presented in section 2.5.2.

The real-time methods are listed below:

(i) ***MAGICAL-IFG***: The calibration parameters are estimated using the incremental mode factor graph approach described in section 3.4, where the factors are added as they are received, and the graph optimization is performed after a predefined number of added factors.

(ii) *MagFactor3*: The calibration parameters are estimated using the MagFactor3 method, presented in section 2.5.3. Unlike a full soft-iron matrix estimation, this method only estimates a single scale factor uniform across all three axes, as well as the hard-iron. The current attitude of the system and the local magnetic field value must be provided as inputs.

To compare the batch and real-time methods, the calibration parameters estimated using the real-time methods were based on the average of the last 20% of the estimated parameters. Because some of these methods have an up-to-scale nature, we will use the normalized magnetic field for comparison purposes, which will not affect heading comparisons since, as seen in 2.2, heading is computed as the division of two magnetic field components, and any scale factor will have no effect. It should be noted that the TWOSTEP and MagFactor3 methods require knowledge of the local magnetic field magnitude, which was obtained from the World Magnetic Model provided by the National Oceanic and Atmospheric Administration (NOAA) (NOAA, 2023) for in-field evaluations.

The hyper-parameters for the least squares-based approaches were set empirically for both numerical and in-field evaluations. The termination criteria for the iterations was set to 1.0×10^{-6} for the relative error tolerance, and the Jacobian was computed numerically. In the case of the factor graph-based methods, the termination criteria were also set empirically, with both the relative and absolute error tolerances set to 1.0×10^{-7}. A multifrontal Cholesky factorization was used, as it has been shown to outperform the LDL and QR factorizations (Dellaert & Kaess, 2006). The factor graph optimization was computed using the RISE method from GTSAM.

For both least squares-based approaches and factor graph-based approaches, the initial guess for the calibration parameters were

$$A_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad m_{b0} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \text{and} \quad w_{b0} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

4.2. Numerical Simulation Performance Metrics

In Chapter 5, we report the evaluation of these approaches in numerical simulations, where the true simulated bias values and the true simulated heading values are known precisely. To compute the estimation error of the biases, we compare the estimated value (x) to the ground truth value (x^*). For the magnetometer hard-iron and gyroscope bias, which are both 3-dimensional vectors in $\mathbb{R}^3$, we use the Euclidean norm to compute the vector difference as follows:

$$|| \, m_b - m_b^* \, || = \sqrt{(m_{bx} - m_{bx}^*)^2 + (m_{by} - m_{by}^*)^2 + (m_{bz} - m_{bz}^*)^2}, \tag{4.1}$$

$$|| \, w_b - w_b^* \, || = \sqrt{(w_{bx} - w_{bx}^*)^2 + (w_{by} - w_{by}^*)^2 + (w_{bz} - w_{bz}^*)^2}. \tag{4.2}$$

For the magnetometer soft-iron, we use the more general Frobenius norm applicable to $\mathbb{R}^{m \times n}$ matrices, which sums the squared terms of the matrix:

$$|| \, A - A^* \, ||_F = \sqrt{\sum_{i=0}^{m} \sum_{j=0}^{n} |a_{ij}|^2}. \tag{4.3}$$

Using the estimated calibration parameters, we can compute the corrected magnetic field (mf_c) and angular rates (w_c). Based on the corrected magnetic field, we can compute the normalized standard deviation of the magnetic field measurement as follows:

$$\sigma_{mf_c} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{mf_{ci}}{\bar{mf}_c} - 1\right)^2}, \quad \text{where: } \bar{mf}_c = \frac{1}{N}\sum_{i=1}^{N} mf_{ci}. \tag{4.4}$$

Here, N is the number of samples, mf_{ci} is the corrected magnetic field measurement for the i^{th} sample, and $\bar{mf}_c$ is the mean of the corrected magnetic field measurements. The standard deviation will later be scaled by the local magnetic field to recover the mG dimension.

We can also compute the magnetic heading error standard deviation using the corrected magnetic field using 2.2. This metric will be used for evaluation purposes.

4.3. Experimental Performance Metrics

In Chapter 6, we evaluate these approaches in an oceanographic survey using the Monterey Bay Aquarium Research Institute ROV Doc Ricketts. In this scenario, the true bias values and true heading are unknown, but we have access to heading measurements from the Kearfott SeaDevil high-end INS (Kearfott Corporation, 2015), which are considered as ground truth due to the sensor's accuracy. To evaluate the performance of the methods in these circumstances, we can use the same standard deviation of the magnetic heading error as in the simulated experiments.

Additionally, we can estimate their effects on real-world applications, such as dead reckoning navigation, by utilizing DVL measurements from the INS and the attitude of the low-cost IMU, integrating them using 2.4, using the Kearfott SeaDevil (Kearfott Corporation, 2015) as the ground truth, employing the trajectory error in the XY plane as a metric,

$$error_{xy} = \sqrt{(x_{ins} - x)^2 + (y_{ins} - y)^2}. \tag{4.5}$$

5. MAGICAL NUMERICAL SIMULATION EVALUATION

5.1. Simulation Setup

A Monte Carlo numerical simulation was conducted to replicate 10,000 measurements from a MEMS AHRS during sinusoidal motions of a vehicle. Three simulated datasets represented varying degrees of angular motion constraint in all degrees of freedom. The wide angular movement (WAM) dataset (Fig. 5.1a) covered a $\pm 180°$ range in roll, pitch, and heading. The moderate angular movement (MAM) dataset (Fig. 5.1b) had $\pm 5°$ and $\pm 45°$ ranges for roll and pitch, respectively, but the same heading range as WAM. The low angular movement (LAM) dataset (Fig. 5.1c) retained the roll range while reducing the pitch and heading ranges to $\pm 15°$ and $\pm 90°$, respectively. Each experiment lasted 400 seconds, with simulated data generated at a 25 Hz rate and magnetometer measurements ($\sigma_{mag} = 1$ mG) and angular rate sensor ($\sigma_{gyro} = 5$ mrad/s) corrupted by Gaussian noise.

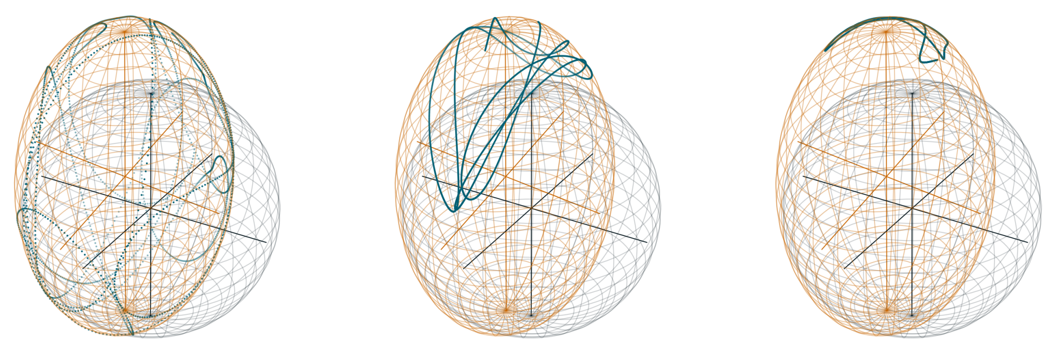

(a) (b) (c)

Figure 5.1. Simulated magnetometer data for three datasets: WAM, MAM, and LAM. The 3D plots show green dots for magnetometer data, gray spheres for the true magnetic field, and orange ellipsoids for the distorted magnetic field.

The true magnetic field vector is $m_0 = [227,\ 52,\ 412]^T$ mG, the soft-iron upper triangular terms are given by $a = [1.10,\ 0.10,\ 0.04,\ 0.88,\ 0.02,\ 1.22]^T$, the hard-iron bias is $m_b = [20,\ 120,\ 90]^T$ mG, and the gyroscope bias is $w_b = [4,\ -5,\ 2]^T$ mrad/s. The magnetic field used in the TWOSTEP and MagFactor3 methods and the attitude used in MagFactor3 were 5% higher than the value used to generate the simulated data.

5.2. Simulation Results

The calibration methods presented in Chapter 4 were calibrated on the three datasets mentioned previously (results summarized in Table 5.1) and later evaluated on a dedicated unique excited evaluation dataset to avoid overfitting. Results in Table 5.2 and Fig. 5.2 show that the proposed MAGICAL methods consistently outperformed the benchmark methods. Specifically, in the WAM dataset calibration, while the TWOSTEP method performed best overall, both MAGICAL-BFG and MAGICAL-IFG demonstrated comparable performance in soft-iron estimation, with errors in hard-iron and gyroscope bias estimation of less than 5 mG and 4 mrad/s, respectively.

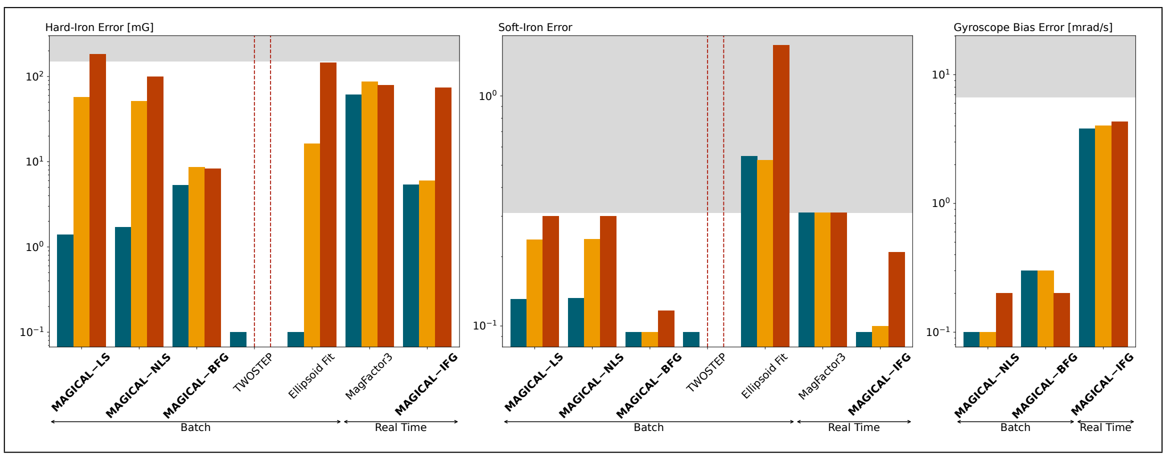


Figure 5.2. Performance comparison of seven calibration methods on three simulated datasets. The hard-iron error, soft-iron error, and gyroscope bias error are analyzed for the WAM (blue), MAM (yellow), and LAM (orange) datasets. Red dashed lines indicate instances where the method failed to estimate the parameters for a particular dataset, and gray-shaded zones show the raw data value.

However, in the MAM and LAM datasets, TWOSTEP failed to converge, and the Ellipsoid Fit method exhibited significant worsening in the results, while MagFactor3 showed more robustness but did not compute non-orthogonality or scale factors in the three axes. In contrast, the four proposed MAGICAL methods showed overall better performance, except for soft-iron estimation in the least squares methods.

Table 5.1. Estimated magnetometer and gyroscope biases for five batch and two real-time calibration methods in three simulated datasets over 100 validation simulations. N/A indicates failure.

WAM for Calibration

		a_0	a_1	a_2	a_3	a_4	a_5	m_{bx} [mG]	m_{by} [mG]	m_{bz} [mG]	w_{bx} [mrad/s]	w_{by} [mrad/s]	w_{bz} [mrad/s]
	Ground Truth	1.100	0.100	0.040	0.880	0.020	1.220	20.000	120.000	90.000	4.000	-5.000	2.000
BATCH	MAGICAL-LS	1.024	0.068	0.031	0.874	0.015	1.125	19.101	119.609	89.008	N/A	N/A	N/A
BATCH	MAGICAL-NLS	1.024	0.068	0.031	0.875	0.015	1.124	19.873	119.238	88.502	4.083	-5.027	1.935
BATCH	MAGICAL-BFG	1.043	0.094	0.038	0.835	0.019	1.161	19.450	114.952	88.328	3.963	-4.800	1.812
BATCH	TWOSTEP	1.045	0.095	0.038	0.836	0.019	1.159	19.944	120.034	90.029	N/A	N/A	N/A
BATCH	Ellipsoid Fit	0.981	-0.096	-0.041	1.153	0.133	0.907	19.946	120.013	90.042	N/A	N/A	N/A
REAL TIME	MAGICAL-IFG	1.043	0.094	0.038	0.836	0.019	1.161	18.954	115.106	88.004	1.775	-2.136	0.852
REAL TIME	MagFactor3	1.000	0.000	0.000	1.000	0.000	1.000	21.733	64.467	63.398	N/A	N/A	N/A

MAM for Calibration

		a_0	a_1	a_2	a_3	a_4	a_5	m_{bx} [mG]	m_{by} [mG]	m_{bz} [mG]	w_{bx} [mrad/s]	w_{by} [mrad/s]	w_{bz} [mrad/s]
	Ground Truth	1.100	0.100	0.040	0.880	0.020	1.220	20.000	120.000	90.000	4.000	-5.000	2.000
BATCH	MAGICAL-LS	1.037	0.040	0.009	0.945	0.004	1.022	36.233	126.694	144.367	N/A	N/A	N/A
BATCH	MAGICAL-NLS	1.034	0.039	0.009	0.946	0.004	1.024	37.295	126.736	138.168	3.930	-4.980	1.972
BATCH	MAGICAL-BFG	1.046	0.094	0.040	0.834	0.018	1.160	16.839	115.233	83.569	3.761	-4.771	1.946
BATCH	TWOSTEP	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
BATCH	Ellipsoid Fit	0.976	-0.090	-0.048	1.127	0.149	0.937	19.711	120.385	106.324	N/A	N/A	N/A
REAL TIME	MAGICAL-IFG	1.051	0.093	0.039	0.842	0.017	1.143	17.023	114.880	90.611	1.557	-1.967	0.938
REAL TIME	MagFactor3	1.000	0.000	0.000	1.000	0.000	1.000	26.068	34.451	105.456	N/A	N/A	N/A

LAM for Calibration

		a_0	a_1	a_2	a_3	a_4	a_5	m_{bx} [mG]	m_{by} [mG]	m_{bz} [mG]	w_{bx} [mrad/s]	w_{by} [mrad/s]	w_{bz} [mrad/s]
	Ground Truth	1.100	0.100	0.040	0.880	0.020	1.220	20.000	120.000	90.000	4.000	-5.000	2.000
BATCH	MAGICAL-LS	1.012	0.007	0.002	0.991	0.000	0.997	70.928	155.958	261.968	N/A	N/A	N/A
BATCH	MAGICAL-NLS	1.009	0.007	0.003	0.992	0.000	0.999	72.205	147.713	169.811	4.095	-4.929	2.099
BATCH	MAGICAL-BFG	1.078	0.086	0.024	0.837	0.012	1.119	12.191	117.941	88.120	3.904	-4.791	2.030
BATCH	TWOSTEP	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
BATCH	Ellipsoid Fit	0.744	-0.041	0.232	0.503	0.023	2.762	66.652	133.566	228.005	N/A	N/A	N/A
REAL TIME	MAGICAL-IFG	1.104	0.082	0.015	0.898	0.008	1.016	24.034	121.612	163.677	1.594	-1.581	1.216
REAL TIME	MagFactor3	1.000	0.000	0.000	1.000	0.000	1.000	-6.282	57.361	130.686	N/A	N/A	N/A

Table 5.2. Mean heading RMSE and magnetic field standard deviation metrics for five batch and two real-time calibration methods in three simulated datasets over 100 validation simulations. N/A indicates failure. The best two results in each column are bolded

		WAM for Calibration		**MAM for Calibration**		**LAM for Calibration**	
		Mean Heading RMSE [deg]	Magnetic Field Std [mG]	Mean Heading RMSE [deg]	Magnetic Field Std [mG]	Mean Heading RMSE [deg]	Magnetic Field Std [mG]
	Raw	28.864	60.330	28.864	60.330	28.864	60.330
BATCH	MAGICAL-LS	3.703	13.516	5.300	23.997	14.368	61.142
	MAGICAL-NLS	3.761	13.680	5.521	24.607	9.320	41.981
	MAGICAL-BFG	2.659	9.794	**2.735**	**9.875**	**3.812**	**16.016**
	TWOSTEP	**2.518**	**9.421**	N/A	N/A	N/A	N/A
	Ellipsoid Fit	18.232	62.522	16.235	60.211	50.842	157.802
REAL TIME	MAGICAL-IFG	**2.654**	**9.762**	**2.791**	**10.371**	**3.528**	**17.552**
	MagFactor3	15.915	45.192	16.917	49.603	14.431	48.784

These results suggest that the MAGICAL methods are more resilient to constrained angular movements than state-of-the-art methods, particularly in the MAM and LAM datasets that emulate actual operating conditions. Furthermore, the MAGICAL methods show consistent performance even under conditions favorable to the benchmark methods, such as in the WAM dataset. Additionally, the MAGICAL-BFG method demonstrated greater robustness compared to MAGICAL-IFG, but both performed better than the benchmark methods. Furthermore, for the real-time MAGICAL-IFG method, from the 40% of the data and on, converged to a constant bias estimation. Notably, the performance of the MAGICAL-LS method indicated that least squares methods could converge even when the gyroscope bias is not computed.

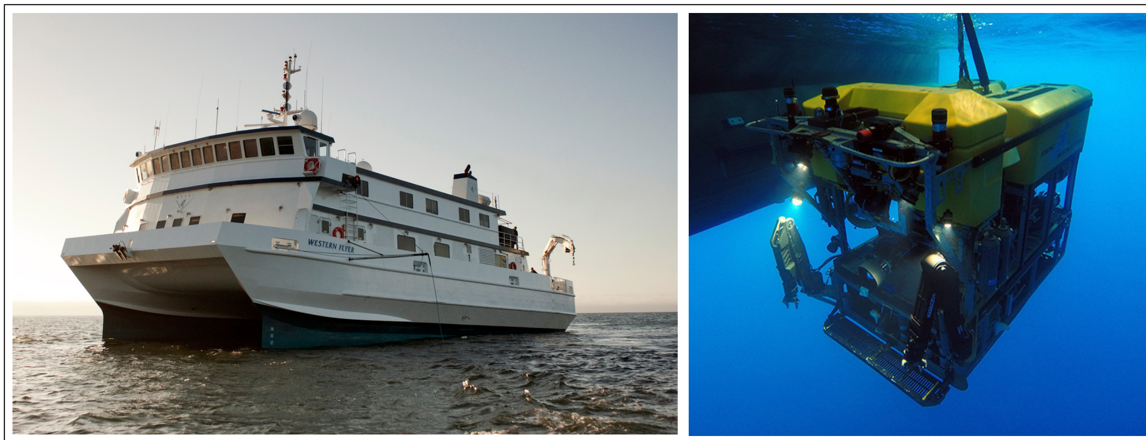
Regarding processing time, all MAGICAL methods run much faster than in real-time. The MAGICAL-LS and MAGICAL-NLS methods exhibit a constant processing time across the four datasets, with an average computation time of around 1.4 seconds per simulation, each lasting 400 s. Meanwhile, the MAGICAL-IFG method has a computation time of around 0.8 seconds for all datasets, demonstrating the feasibility of the real-time operation. In contrast, the MAGICAL-BFG method shows a slight increase in processing time with a decrease in the movement range, rising from around 0.3 seconds to 0.8 seconds due to longer iterations to reach the termination criteria.

These simulation results support that all proposed MAGICAL methods show competitive or better performance than the benchmark methods, indicating their effectiveness for post-processing and real-time navigation applications.

6. MAGICAL FIELD EXPERIMENTAL EVALUATION

6.1. Field Experimental Setup

This chapter evaluates the in-field performance of the proposed and benchmark methods. The navigation data employed in this study was obtained during a seafloor mapping survey dive conducted by the Monterey Bay Aquarium Research Institute (MBARI) in December 2014. The survey was conducted at a depth of 2,800 m in Monterey Bay, using the *Doc Rickets* ROV (shown in Fig. 6.1b), which weighs 5,000 kg and is rated to a depth of 4,000 m and was operated from the *R/V Western Flyer* (shown in Fig. 6.1a), both owned and operated by MBARI.



(a) MBARI *Western Flyer Research Vessel*. Image source: MBARI.

(b) MBARI *Doc Ricketts* Remote Operated Vehicle. Image source: MBARI.

Figure 6.1. Monterey Bay Aquarium Research Institute (MBARI) equipment deployed during December 2014 Monterey Bay seafloor mapping expedition.

6.1.1. *Doc Ricketts* Navigation Sensors

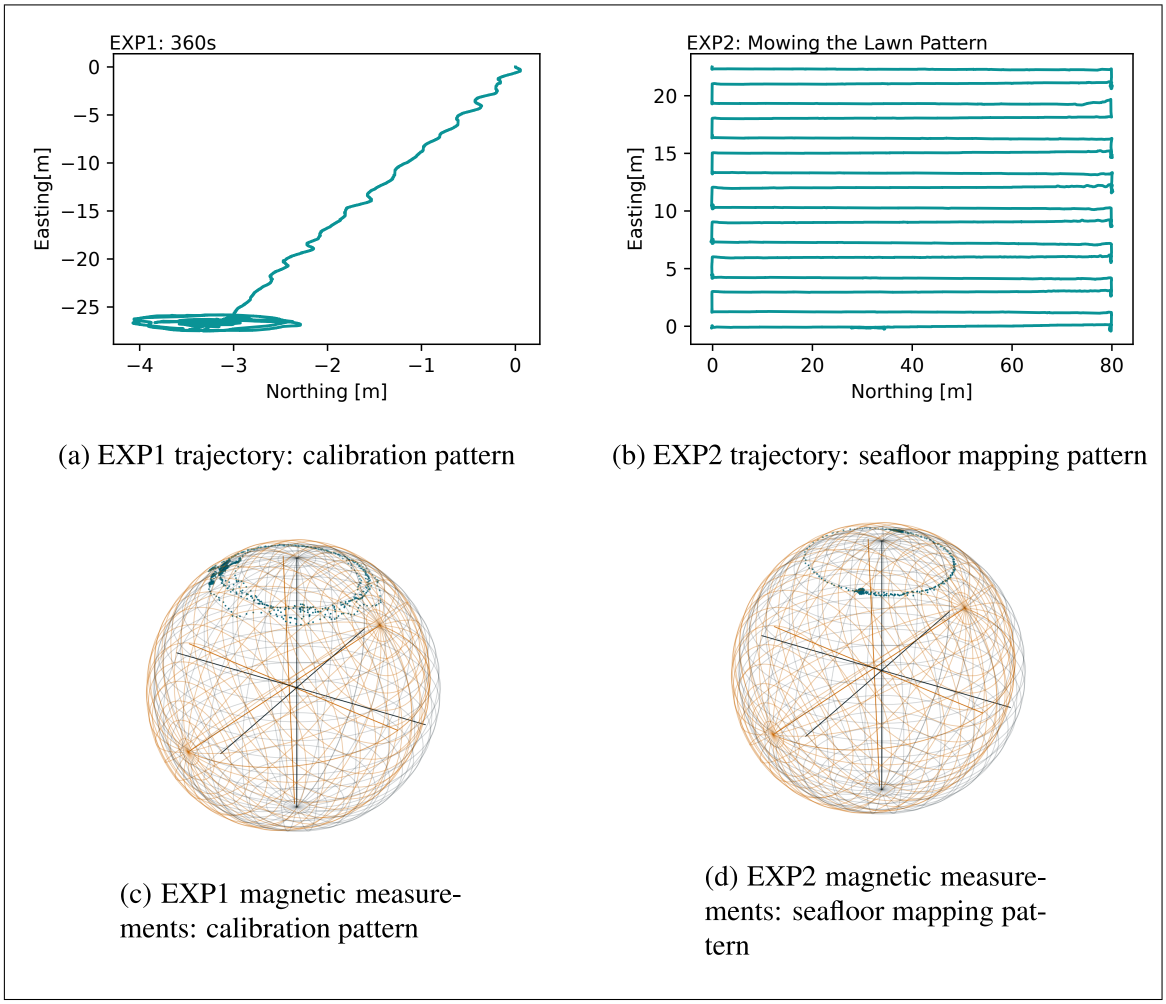
The *Doc Ricketts* ROV is equipped with an extensive suite of sensors, including acoustic, physical oceanographic, geophysical, optical, and navigation sensors. For this evaluation, we will focus on the VectorNav VN100 MEMS-based IMU operated at a sampling rate of 80 Hz, with a magnetometer noise level of $\sigma_{mag} = 1$ mG and an angular-rate gyroscope noise level of $\sigma_{gyro} = 0.5$ mrad/s (Vectornav, 2023), and in the Kearfott SeaDevil high-end Inertial Navigation System (INS) operating at a sampling rate of 25 Hz, which includes a Doppler Velocity Logger (DVL) and a ring-laser gyro, providing a precision of $0.05°$ and $0.03°$ in heading and pitch/roll, respectively, with a real-time position accuracy of 0.1% of the total distance traveled when the DVL continuously tracks the seafloor (Kearfott Corporation, 2015). Both sensors' accuracy and operation rate are summarized in Table 6.1.

Table 6.1. *Doc Ricketts* ROV sensors precision and update rate.

Sensor	Instrument	Variable(s)	Accuracy	Operation Rate
High-end INS	Kearfott SeaDevil	Heading	0.05^o RMS	25 Hz
		Pitch/Roll	0.03^o RMS	
		Position	0.1% DT	
		Velocity	0.3 m/s RMS	
Low-cost AHRS	VectorNav VN100	Heading	2.00^o RMS	80 Hz
		Pitch/Roll	1.00^o RMS	

6.1.2. Dive Description

Two field experiments were conducted on the same day in the Monterey Bay in California, USA, where the local magnetic field had a magnitude of 479 mG (NOAA, 2023). The first experiment, denoted as EXP1 (Fig. 6.2a, Fig. 6.2c), involved a series of $360°$ heading rotations, with the pitch and roll configurations changing to produce $5°$ pitch and roll movements. The second experiment, denoted as EXP2 (Fig. 6.2b, Fig. 6.2d), consisted of a standard survey of 1395 m length, where the vehicle maintained a stable pitch and roll, resembling the pattern of ”mowing a lawn” for a standard survey. The calibration parameters for the magnetometer and gyroscope were estimated using data from both EXP1 and EXP2 with each method and evaluated with EXP1.



(a) EXP1 trajectory: calibration pattern

(b) EXP2 trajectory: seafloor mapping pattern

(c) EXP1 magnetic measurements: calibration pattern

(d) EXP2 magnetic measurements: seafloor mapping pattern

Figure 6.2. *Doc Rickett*'s December 2014 Monterey Bay dive.

6.2. Heading Estimation Performance

We used the previously described VectorNav VN100 MEMS-based IMU and Kearfott SeaDevil high-end INS to evaluate the heading estimation performance, using the latter as the ground truth for heading comparison. We interpolated the MEMS IMU data to the INS sampling time to estimate the vehicle's heading. The heading error, defined as the standard deviation between the measured heading from the INS and the calculated heading from the bias-compensated magnetometer data for each evaluated method, was used as the evaluation metric in order to isolate alignment errors.

As shown in Fig. 6.3, the proposed MAGICAL methods exhibit outstanding performance when calibrated using a standard ROV magnetometer calibration routine (EXP1), reducing the original heading error from $6.33°$ to less than $1.53°$. The factor graph-based methods improve in heading error for calibration with EXP2, while the benchmark methods fail to converge to a solution except for MagFactor3, which is still outperformed by both proposed factor graph-based methods.

For this particular case, only minimal magnetometer calibration was required, with the soft-iron value close to the identity and hard-iron values in the order of a few mG. The gyroscope biases were also found to be in the order of a few $mrad/s$. While this is an uncommon scenario, it emphasizes the importance of magnetometer calibration even in less critical cases.

Furthermore, the results demonstrate the feasibility of both proposed factor graph methods to improve the calibration for highly constrained implementations, converging even under challenging scenarios in batch or real-time mode, such as calibration with

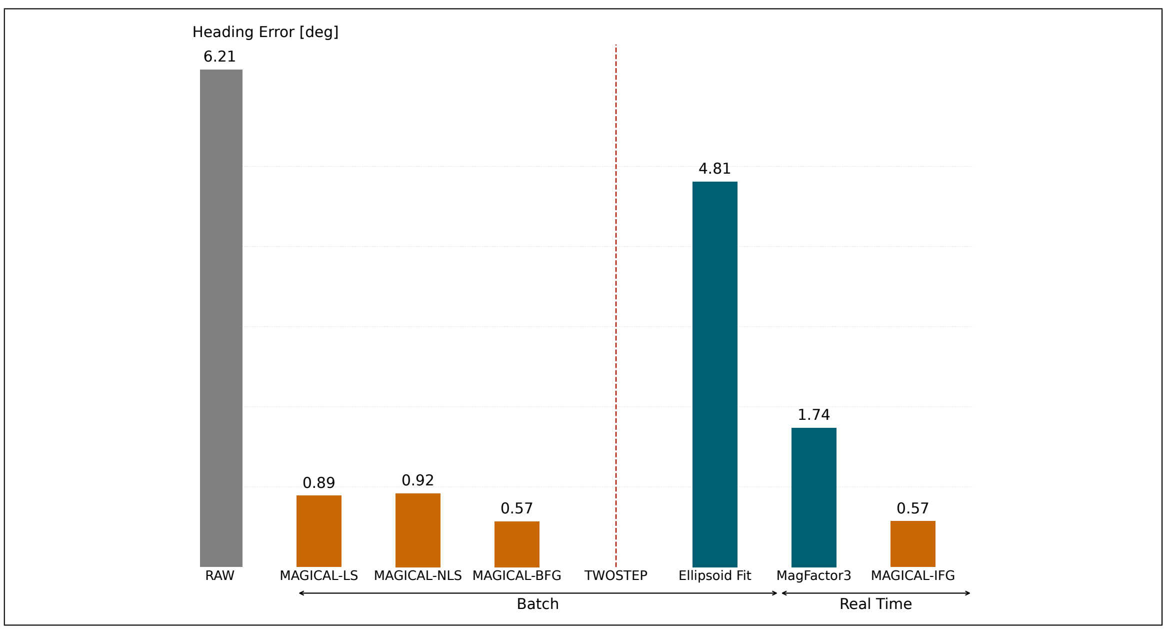


Figure 6.3. Heading error for calibration parameters estimated with EXP1 and evaluated with EXP2. Red dashed lines indicate parameter estimation failures.

Table 6.2. Estimated magnetometer and gyroscope biases for five batch and two real-time calibration methods in three simulated datasets over 100 validation simulations. N/A indicates failure.

EXP1 for Calibration

		a_0	a_1	a_2	a_3	a_4	a_5	m_{bx} [mG]	m_{by} [mG]	m_{bz} [mG]	w_{bx} [mrad/s]	w_{by} [mrad/s]	w_{bz} [mrad/s]
BATCH	MAGICAL-LS	0.875	-0.036	0.061	1.038	-0.103	1.116	-30.309	73.232	-83.238	N/A	N/A	N/A
	MAGICAL-NLS	0.875	-0.036	0.061	1.038	-0.104	1.117	-30.421	74.108	-82.123	-0.56	0.581	1.788
	MAGICAL-BFG	1.007	0.000	0.011	1.030	0.004	0.964	-7.509	25.211	-0.732	0.175	-0.101	0.490
	TWOSTEP	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Ellipsoid Fit	0.725	-0.006	-0.001	1.861	0.041	0.742	-3.003	14.301	332.575	N/A	N/A	N/A
REAL TIME	MAGICAL-IFG	1.013	0.000	0.011	1.035	0.004	0.954	-7.431	25.107	5.238	0.162	-0.103	0.515
	MagFactor3	1.000	0.000	0.000	1.000	0.000	1.000	-5.926	19.544	-29.264	N/A	N/A	N/A

EXP2 for Calibration

		a_0	a_1	a_2	a_3	a_4	a_5	m_{bx} [mG]	m_{by} [mG]	m_{bz} [mG]	w_{bx} [mrad/s]	w_{by} [mrad/s]	w_{bz} [mrad/s]
BATCH	MAGICAL-LS	0.717	-0.078	0.074	1.204	-0.114	1.184	-43.266	126.702	-236.364	N/A	N/A	N/A
	MAGICAL-NLS	0.717	-0.078	0.074	1.204	-0.114	1.184	-43.070	126.849	-236.796	0.284	-0.003	0.948
	MAGICAL-BFG	0.973	-0.007	0.029	1.009	-0.040	1.021	-16.870	45.694	-47.413	0.265	-0.070	0.746
	TWOSTEP	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Ellipsoid Fit	0.725	-0.006	-0.001	1.861	0.041	0.742	-3.003	14.301	332.575	N/A	N/A	N/A
REAL TIME	MAGICAL-IFG	0.970	-0.008	0.030	1.006	-0.041	1.028	-17.167	46.534	-51.592	0.271	-0.013	0.729
	MagFactor3	1.000	0.000	0.000	1.000	0.000	1.000	2.717	2.464	-21.951	N/A	N/A	N/A

EXP2 (Fig. 6.4). Additionally, from the 25% of the data and on, MAGICAL-IFG converged to a constant bias estimation. Overall, these findings suggest that the MAGICAL methods can significantly enhance the accuracy of underwater vehicle navigation.

6.3. Navigation Performance

Accurate sensing and estimation of attitude, including heading, roll, and pitch, is a critical component of navigation systems for underwater vehicles, especially for vehicles operating in environments where global positioning system (GPS) signals are unavailable, such as underwater navigation. Bottom-lock Doppler sonar navigation is a common method for high-precision near-bottom underwater vehicle navigation. Doppler sonar navigation typically employs a 3-axis Doppler Velocity Logger (DVL), a precision pressure

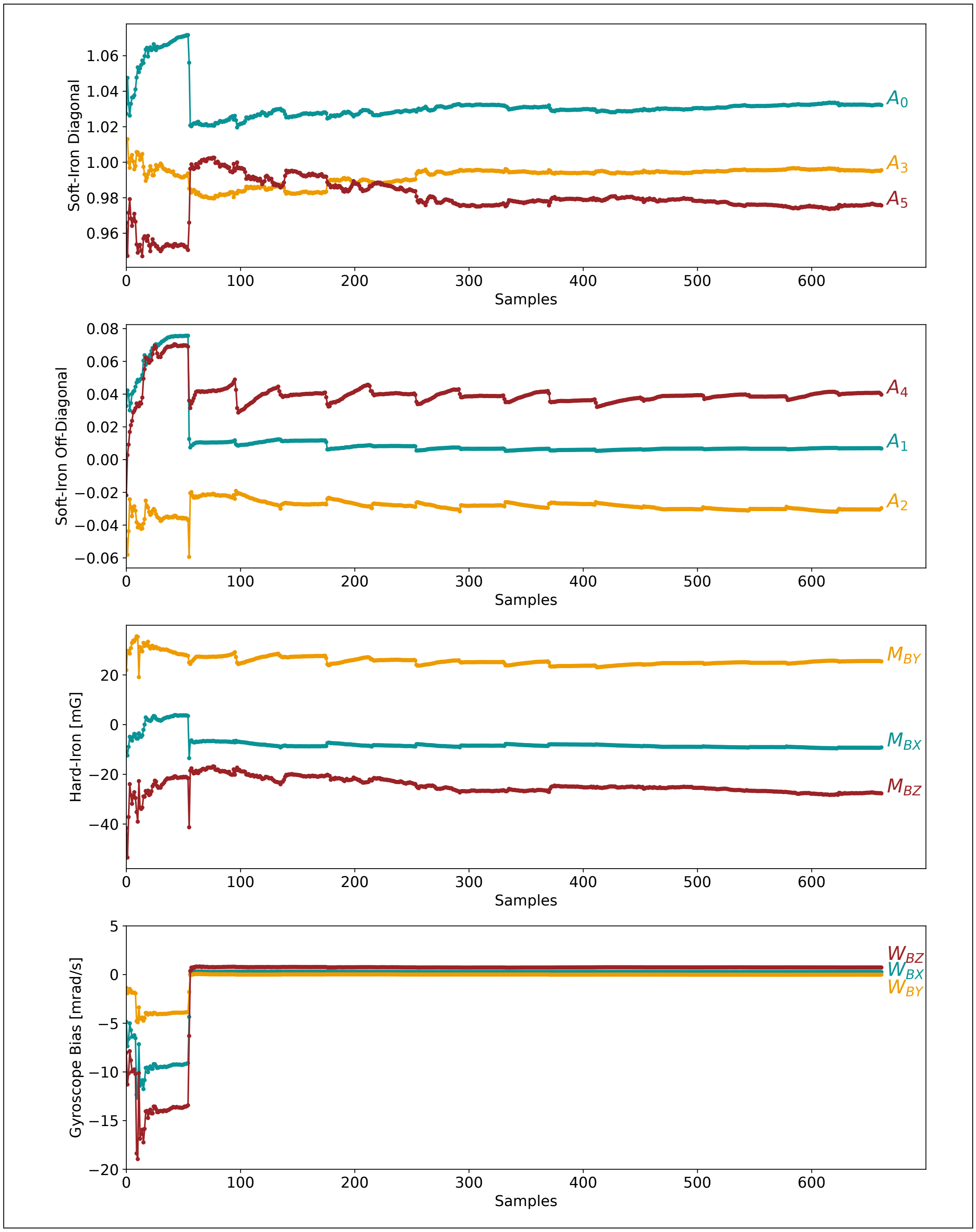
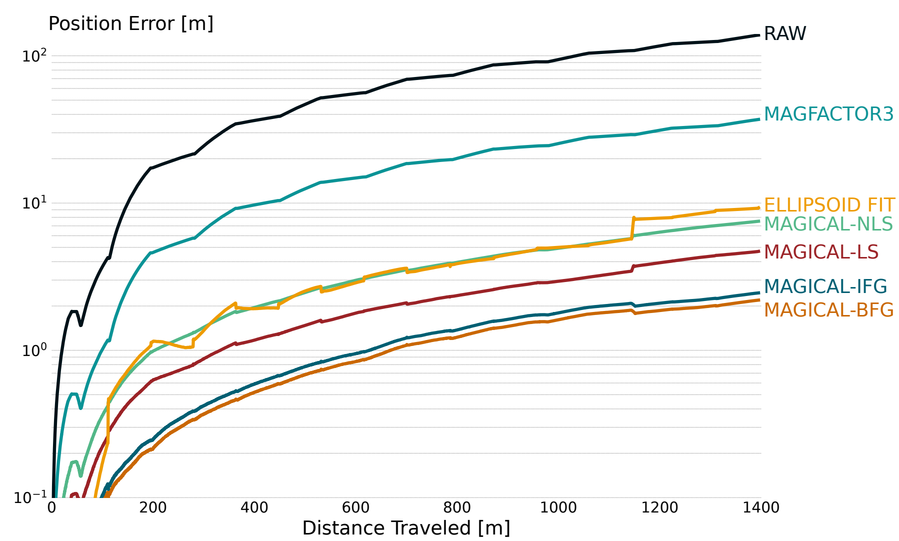


Figure 6.4. Soft-Iron, Hard-Iron, and Gyroscope Bias calibration parameters estimation convergence using the MAGICAL-IFG method with data from EXP2.

depth sensor, and a 3-axis attitude sensor, such as AHRS (Whitcomb, Yoerger, & Singh, 1999).

Biases in the AHRS measurements can cause heading errors, which are a critical factor in accurate position estimation (Troni & Whitcomb, 2012). To address this, we used the calibration obtained from EXP1, and for each method, we computed the dead reckoning using the accelerations and corrected magnetic field measurements from the VectorNav VN100 MEMS IMU to estimate the vehicle's attitude and the velocity reported by the Kearfott SeaDevil INS, which served as ground truth for the vehicle's position, which as shown in Table 6.1 has an error of 0.1% of the total distance traveled. We measured the position error as the difference between the estimated and ground truth positions.



RAW	MAGICAL-LS	MAGICAL-NLS	MAGICAL-BFG	Ellipsoid Fit	MagFactor3	MAGICAL-IFG
138.25 m (9.91 %)	4.70 m (0.34 %)	7.53 m (0.54 %)	2.19 m (0.16 %)	9.28 m (0.67 %)	36.97 m (2.65 %)	2.45 m (0.18 %)

Batch Real Time

Figure 6.5. Norm of XY position error, with method-wise total error and error relative to the distance traveled, summarized in the table below.

Fig. 6.5 shows the norm of the XY position error for each calibration method. When the magnetometer is not calibrated (RAW), the position error after traveling 1,395 meters is 138 meters, which represents a significant 10% error. However, the four proposed

MAGICAL methods show a significant improvement, with position errors of less than 7.5 meters, representing only 0.54% of the full distance traveled. Of the benchmark methods, only the Ellipsoid Fit produced comparable results with an error of 0.67%, while the TWOSTEP method failed to converge.

Furthermore, when we analyze the trajectory output produced by dead reckoning using each method, we find that navigation becomes unfeasible when the magnetometer is not calibrated, as shown in Fig. 6.6a. Even small biases in the calibration parameters (see Table 6.2) can lead to significant divergence in position estimation during straight sections of the trajectory. Among the benchmark methods, we observe that the MagFactor3 method shows some improvement in trajectory compared to raw measurements (Fig. 6.6c) but still exhibits drift that leads to significant divergence from the ground truth trajectory. In contrast, for the Ellipsoid Fit method (Fig. 6.6b), inaccuracies in calibration are reflected in trajectory divergence during turns. However, the MAGICAL methods closely follow the ground truth trajectory, accumulating error only during turns and in a significantly smaller proportion than the benchmark methods.

The results presented in this section demonstrate that the proposed MAGICAL methods can significantly improve position estimation accuracy in real-time underwater vehicle navigation scenarios. In particular, the MAGICAL approach can enhance the accuracy of low-cost navigation systems and pave the way for more widespread use of underwater vehicles in oceanographic missions.

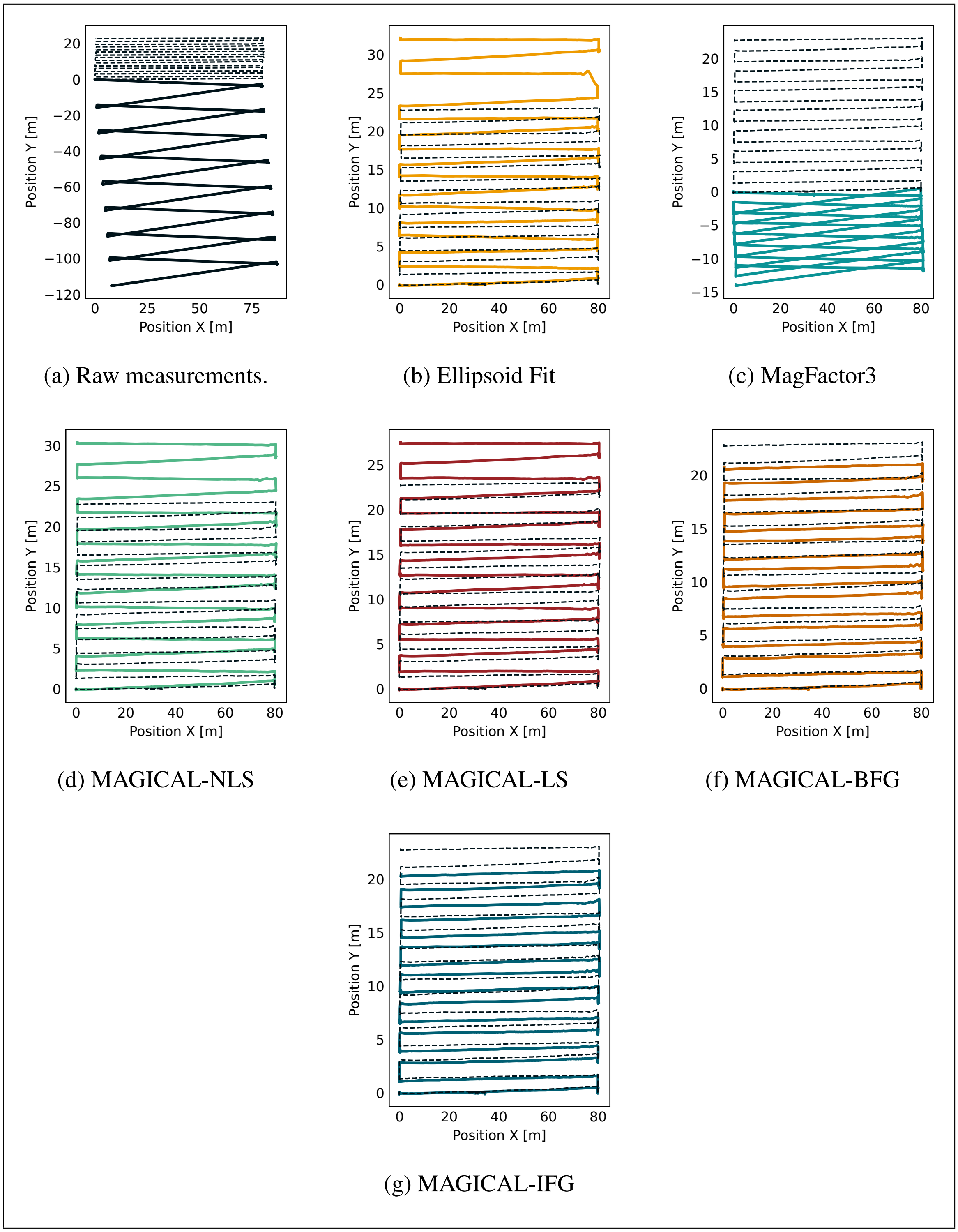


Figure 6.6. Comparison of XY trajectories estimated through dead reckoning using different magnetometer calibration methods. The calibrated trajectories are shown in color, while dashed lines represent the ground truth trajectory.

7. CONCLUSIONS

The Magnetometer and Gyroscope Iterative Calibration (MAGICAL) methods proposed in this study have been demonstrated to significantly improve the performance of Attitude and Heading Reference System (AHRS) sensors in both simulated and in-field scenarios. Our results indicate that the MAGICAL methods outperform or are comparable to previously reported methods, such as TWOSTEP and Ellipsoid Fit, which failed to converge in constrained range scenarios, even when accurate knowledge of the local magnetic field vector was available.

Our numerical simulations revealed that the MAGICAL methods consistently outperformed benchmark methods, including the MagFactor3 method, which requires accurate knowledge of the local magnetic field and system attitude in different ranges of movements. Specifically, MAGICAL methods demonstrated improved robustness in low ranges of movements, which are common scenarios in oceanographic expeditions. In our in-field experiments, we observed that the MAGICAL methods significantly reduced heading error compared to previously reported methods. In particular, the MAGICAL-BFG and MAGICAL-IFG methods showed promising results, with a strong convergence even in scenarios where the system is stable in roll/pitch, such as in standard oceanographic surveys. These results were also complemented by the improved navigation performance, where the MAGICAL methods reduced the underwater vehicle's dead reckoning position estimation error from 10% to 0.5% of the distance traveled. Our findings suggest that MAGICAL can significantly improve the accuracy of low-cost navigation systems and pave the way for more widespread use of underwater vehicles in oceanographic missions.

In conclusion, our study has demonstrated that the MAGICAL methods provide a viable and effective approach to calibrating magnetometers and gyroscopes for attitude estimation. Our findings can benefit the development of low-cost navigation systems and improve the performance of ground, marine, and aerial vehicles in real-world scenarios.

REFERENCES

Alonso, R., & Shuster, M. D. (2002a). Complete Linear Attitude Independent Magnetometer Calibration. *Journal of the Astronaitical Sciences*, *50*(4), 477–490.

Alonso, R., & Shuster, M. D. (2002b). TWOSTEP: A fast robust algorithm for attitude-independent magnetometer-bias determination. *Journal of the Astronautical Sciences*, *50*(4), 433–451. doi: 10.1007/bf03546247

Ammann, N., Derksen, A., & Heck, C. (2015). A novel magnetometer-accelerometer calibration based on a least squares approach. *2015 International Conference on Unmanned Aircraft Systems, ICUAS 2015*, 577–585. doi: 10.1109/ICUAS.2015.7152338

Bazhin, A., Vrba, V., & Zogopoulos-Papaliakos, G. (2022). *Ellipsoid Fit Python.* Retrieved from `https://github.com/aleksandrbazhin/ellipsoid_fit_python`

BORG Lab. (2023). *GTSAM.* Retrieved from `https://github.com/borglab/gtsam`

Branch, M. A., Coleman, T., & Li, Y. (1999). A Subspace, Interior, and Conjugate Gradient Method for Large-Scale Bound-Constrained Minimization Problems. *SIAM Journal on Scientific Computing*, *21*(1), 1–23.

Caress, D., Thomas, H., Kirkwood, W., McEwen, R., Henthorn, R., Clague, D., ... Maier, K. (2008). High-Resolution Multibeam, Sidescan, and Subbottom Surveys Using the MBARI AUV D. Allan B. *Marine Habitat Mapping Technology for Alaska*(November 2014), 47–70. doi: 10.4027/mhmta.2008.04

Crassidis, J., Lai, K.-L., & Harman, R. (2005). Real-time attitude-independent three-axis magnetometer calibration. *Journal of Guidance, Control, and Dynamics*, *28*(1), 115—-120.

Dai, J., Liu, S., Hao, X., Ren, Z., & Yang, X. (2022). UAV Localization Algorithm Based on Factor Graph Optimization in Complex Scenes. *Sensors*, *22*(15). doi: 10.3390/s22155862

Dellaert, F. (2012). *Factor graphs and GTSAM: A hands-on introduction* (Tech. Rep.). Georgia Institute of Technology. Retrieved from `http://tinyurl.com/gtsam.`

Dellaert, F., & Kaess, M. (2006). Square root SAM: Simultaneous localization and mapping via square root information smoothing. *International Journal of Robotics Research*, *25*(12), 1181–1203. doi: 10.1177/0278364906072768

Dinale, J. P. (2013). *Magnetic Test Facility - Sensor and Coil Calibrations* (Master's Thesis). University of Adelaide.

Fang, J., Sun, H., Cao, J., Zhang, X., & Tao, Y. (2011). A novel calibration method of magnetic compass based on ellipsoid fitting. *IEEE Transactions on Instrumentation and Measurement*, *60*(6), 2053–2061. doi: 10.12733/jics20101833

Foster, C. C., & Elkaim, G. H. (2008). Extension of a two-step calibration methodology to include nonorthogonal sensor axes. *IEEE Transactions on Aerospace and Electronic Systems*, *44*(3), 1070–1078. doi: 10.1109/TAES.2008.4655364

Gambhir, B. (1975). Determination of magnetometer biases using module RESIDG. *Computer Sciences Corporation*, *3000-32700*, 01.

Guo, P., Qiu, H., Yang, Y., & Ren, Z. (2008). The soft iron and hard iron calibration method using extended kalman filter for attitude and heading reference system. *Record - IEEE PLANS, Position Location and Navigation Symposium*(20070851011), 1167–1174. doi: 10.1109/PLANS.2008.4570003

Han, K., Han, H., Wang, Z., & Xu, F. (2017). Extended Kalman Filter-Based Gyroscope-Aided Magnetometer Calibration for Consumer Electronic Devices. *IEEE Sensors Journal*, *17*(1), 63–71. doi: 10.1109/JSEN.2016.2624821

Hilbert, D., & Cohn-Vossen, S. (1999). The second-order surfaces. *Geometry and the Imagination*, 12–19.

Kearfott Corporation. (2015). *Kearfott SeaDevil INS* (Tech. Rep.). Kearfott Corporation.

Kok, M., & Schon, T. B. (2016). Magnetometer calibration using inertial sensors. *IEEE Sensors Journal*, *16*(14), 5679–5689. doi: 10.1109/JSEN.2016.2569160

Lynch, K., & Park, F. (2017). *Modern Robotics* (No. May). Cambridge University Press.

Ma, G. F., & Jiang, X. Y. (2005). Unscented kalman filter for spacecraft attitude estimation and calibration using magnetometer measurements. *2005 International Conference on Machine Learning and Cybernetics, ICMLC 2005*(August), 506–511. doi: 10.1109/icmlc.2005.1526998

NOAA. (2021). *The World Magnetic Model.* Retrieved from `https://www.ngdc.noaa.gov/geomag/WMM/`

NOAA. (2023). *How much of the ocean have we explored?* Retrieved from `https://oceanservice.noaa.gov/facts/exploration.html`

Nocedal, J., & Wright, S. (2006). *Numerical Optimization* (2nd ed.). New York: Springer New York. doi: 10.1007/978-0-387-40065-5

Ousaloo, H. S., Sharifi, G., Mahdian, J., & Nodeh, M. T. (2017). Complete Calibration of Three-Axis Strapdown Magnetometer in Mounting Frame. *IEEE Sensors Journal*, *17*(23), 7886–7893. doi: 10.1109/JSEN.2017.2766200

Papafotis, K., & Sotiriadis, P. P. (2019). Computationally Efficient Calibration Algorithm for Three-Axis Accelerometer and Magnetometer. *2019 8th International Conference on Modern Circuits and Systems Technologies, (MOCAST)*, 1–4. doi: 10.1109/MOCAST.2019.8741694

Rosen, D. M., Kaess, M., & Leonard, J. J. (2014). RISE: An incremental trust-region method for robust online sparse least-squares estimation. *IEEE Transactions on Robotics*, *30*(5), 1091–1108. doi: 10.1109/TRO.2014.2321852

Soken, H., & Sakai, S.-I. (2019). TRIAD+ filtering approach for complete magnetometer calibration. In *2019 9th international conference on recent advances in space technologies (rast)* (pp. 703—-708). IEEE.

Spielvogel, A., Shah, A., & Whitcomb, L. (2022). Online Three-Axis Magnetometer Hard-Iron and Soft-Iron Bias and Angular Velocity Sensor Bias Estimation Using Angular Velocity Sensors for Improved Dynamic Heading Accuracy. *Field Robotics*, *2*(1), 1001–1027. doi: 10.55417/fr.2022033

Spielvogel, A. R., & Whitcomb, L. L. (2018). Adaptive Sensor Bias Estimation in Nine Degree of Freedom Inertial Measurement Units: Theory and Preliminary Evaluation. *IEEE International Conference on Intelligent Robots and Systems*, 5555–5561. doi: 10.1109/IROS.2018.8594439

Troni, G. (2013). *Advances in precision navigation of underwater vehicles* (Phd. Thesis). Johns Hopkins University.

Troni, G., & Whitcomb, L. L. (2012). Experimental evaluation of a MEMS inertial measurements unit for Doppler navigation of underwater vehicles. *OCEANS 2012 MTS/IEEE: Harnessing the Power of the Ocean*. doi: 10.1109/OCEANS.2012.6405003

Troni, G., & Whitcomb, L. L. (2020). Field sensor bias calibration with angular-rate sensors: Theory and experimental evaluation with application to magnetometer calibration. *IEEE/ASME Transactions on Mechatronics*, *24*(4), 1698–1710. doi: 10.1109/TMECH.2019.2920367

Vasconcelos, J. F., Elkaim, G., Silvestre, C., Oliveira, P., & Cardeira, B. (2011). Geometric approach to strapdown magnetometer calibration in sensor frame. *IEEE Transactions on Aerospace and Electronic Systems*, *47*(2), 1293–1306. doi: 10.1109/TAES.2011.5751259

Vectornav. (2023). *VN-100 User Manual* (Tech. Rep.). Vectornav Technologies.

Whitcomb, L., Yoerger, D., & Singh, H. (1999). Advances in Doppler-based navigation of underwater robotic vehicles. *Proceedings - IEEE International Conference on Robotics and Automation*, *1*, 399–406. doi: 10.1109/robot.1999.770011

Zwillinger, D. (2018). *CRC standard mathematical tables and formulas*. Chapman and Hall/CRC.

APPENDIX

A. QUADRATIC SURFACES

For a second-order algebraic surface given by the general equation (Hilbert & Cohn-Vossen, 1999)

$$ax^2 + by^2 + cz^2 + 2fyz + 2gzx + 2hxy + 2px + 2qy + 2rz + d = 0, \quad \text{(A.1)}$$

we can get the matrix form for a quadratic surface

$$x^T Sx + 2Px + d = 0, \quad \text{(A.2)}$$

where

$$S = \begin{bmatrix} a & h & f \\ h & b & g \\ f & g & c \end{bmatrix} \qquad P = \begin{bmatrix} p \\ q \\ r \end{bmatrix}.$$

The general magnetometer equation, which represents an ellipsoid affected by non-orthogonality and scale factors, which can be modeled as a quadratic surface by

$$h_m = Ah + b. \quad \text{(A.3)}$$

From (A.3), we can get the local magnetic field and compute the norm as $||h||^2 = h^T h$, or equivalently

$$||h||^2 = h_m^T M h_m - 2b^T M h_m - b^T Mb, \quad \text{(A.4)}$$

where $M = (A^{-1})^T(A^{-1})$. Based on (A.2) and (A.4) we can find the following equivalencies

$$\begin{aligned}
S &= (A^{-1})^T(A^{-1}), \\
P &= -b^T(A^{-1})^T(A^{-1}), \\
d &= -(b^T(A^{-1})^T(A^{-1})b + ||h||^2).
\end{aligned}$$

Using S, P, and d, we can define the block matrix

$$E = \left[\begin{array}{c|c} S & P \\ \hline P^T & d \end{array}\right], \tag{A.5}$$

which combined with S, gives us the following set of conditions that a quadratic surface must meet to be an ellipsoid in the real numbers domain (Zwillinger, 2018).

(i) Ranking of the S matrix must be equal to 3.
(ii) Ranking of the E matrix must be equal to 4.
(iii) The sign of the determinant of E must be negative.
(iv) The eigenvalues of matrix S must be all non-negative.

A.1. Soft-iron Regularization

To compare the soft-iron calculated by different calibration methods, it is necessary to regularize them to set all of them on the same scale and remove the up-to-scale property that different methods provide. In other words, we want to remove the scale component and keep the non-orthogonality. Based on the up-to-scale concept, we can consider the uncalibrated magnetic field as the ground truth magnetic field, represented by a unitary sphere, i.e., $||h||^2 = 1$, transformed by the non-orthogonality and displaced due to the bias. This transformation has the property that the volume must be conserved since we are not scaling the magnetic field, and the bias does not affect this property, so we can consider $b = 0$. Hence, we can rewrite (A.4) as

$$h_m^T (A^{-1})^T (A^{-1}) h_m - 1 = 0, \tag{A.6}$$

and rewrite the quadratic surface representation as (A.2), finding $S = (A^{-1})^T (A^{-1})$. As we know, A is a symmetric matrix, and therefore the corresponding eigendecomposition is

$$A = Q \Lambda Q^T, \tag{A.7}$$

where Q is an orthonormal base composed by the eigenvectors of A and Λ is a diagonal matrix, populated with the eigenvalues of A, $(\lambda_0, \lambda_1, \lambda_2)$. Due to the symmetry of A, we can directly get the inverse as $A^{-1} = Q \Lambda^{-1} Q^T$.

As a next step, we want to align the principal axes of the modified magnetic field with the x-, y-, and z-axis of the coordinate frame taking advantage of the symmetry of the S matrix. Based on (A.7),

$$S = \left(Q \left(\Lambda^{-1} \right)^T Q^T \right) \left(Q \Lambda^{-1} Q^T \right) = Q \Lambda^{-2} Q^T. \tag{A.8}$$

Using the change of variable $x = Py$ to remove the cross terms, i.e., non-orthogonality,

$$x^T S x = (Py)^T S (Py) = y^T P^T S P y \;\rightarrow\; x^T S x = y^T \Lambda^{-2} y.$$

An ellipsoid centered in the origin of the $\mathbb{R}^3$ cartesian space modeled as

$$\frac{x^2}{\alpha^2} + \frac{y^2}{\beta^2} + \frac{z^2}{\gamma^2} = 1,$$

has a volume $V = \frac{4}{3}\pi\alpha\beta\gamma$, where $\alpha = \lambda_0$, $\beta = \lambda_1$ and $\gamma = \lambda_2$. As we started from the unitary sphere and want to constrain the volume to stay constant, we have to scale the eigenvalues as $\lambda_i' = \lambda_i/(\lambda_0\lambda_1\lambda_2)^{-1/3}$. Therefore, the volume now is

$$
\begin{aligned}
V &= \frac{4}{3}\pi\frac{\lambda_0}{(\lambda_0\lambda_1\lambda_2)^{-1/3}}\frac{\lambda_1}{(\lambda_0\lambda_1\lambda_2)^{-1/3}}\frac{\lambda_2}{(\lambda_0\lambda_1\lambda_2)^{-1/3}} \\
&= \frac{4}{3}\pi\frac{(\lambda_0\lambda_1\lambda_2)}{(\lambda_0\lambda_1\lambda_2)} \\
&= \frac{4}{3}\pi
\end{aligned}
$$

Based on the eigendecomposition, the scaled version of the S matrix is

$$S' = (\lambda_0\lambda_1\lambda_2)^{-2/3}\left(Q\Lambda^{-2}Q^T\right) \rightarrow S' = (\lambda_0\lambda_1\lambda_2)^{-2/3}S. \tag{A.9}$$

Then, to get from the quadratic surface to the magnetometer model, define the scaled soft-iron as $A' = \alpha A$

$$S' = (\frac{1}{\alpha}A^{-1})^T(\frac{1}{\alpha}A^{-1}) = \frac{1}{\alpha^2}(A^{-1})^T(A^{-1}),$$

and replacing (A.9):

$$(\lambda_0\lambda_1\lambda_2)^{-2/3}S = \frac{1}{\alpha^2}(A^{-1})^T(A^{-1}) \rightarrow \alpha = (\lambda_0\lambda_1\lambda_2)^{-1/3}$$

Therefore, the scaled soft-iron is

$$A' = (\lambda_0\lambda_1\lambda_2)^{-1/3}A.$$