

# **Gyroscope and accelerometer bias estimation under non-restrictive conditions for an autonomous Search and Rescue drone**

Master thesis  
of  
Ricardo Martínez García  
born  
23.08.1995 in Cordoba, Spain

August 3, 2020

Supervisor: Dr.-Ing. Thomas Seel  
Second Supervisor: Daniel Laidig

Examiner: Prof. Dr.-Ing. Jörg Raisch  
Control Systems Group  
Institute of Energy and Automation Technology  
Faculty IV - Electrical Engineering and Computer Science  
Technische Universität Berlin



# Abstract

The Mediterranean Sea has become, without doubt, one of the most lethal and dangerous border crossings in the world. Every year, hundreds of humans die on its waters in an attempt to seek for a better life in Europe. Several private organizations struggle, with limited means, to help migrants in danger by performing Search and Rescue missions. Despite these huge efforts, people still drown on the Mediterranean Sea. The Searchwing project is a group of volunteers from Augsburg and Berlin that supports the NGOs that search migrants in distress on the Mediterranean waters. The aim of the project is to build a fully automated drone with a 100 km range that extends the daily search area by a factor of 2. To meet the mission requirements, the drone makes use of accurate sensor fusion and real time motion tracking. However, built-in methods for inertial sensor calibration rely on static periods, which are unlikely to occur on a moving vessel at sea. The object of this Master thesis is to solve the calibration problem on the high-seas, by providing new inertial sensor calibration schemes.

Firstly, this work starts with fundamental concept descriptions and a comprehensive literature research, in order to overview and summarize the state-of-the-art approaches for inertial sensor calibration. Two groups of in-field calibration methods are differentiated: Kalman filter-based and optimization-based. The methods are differentiated regarding the considered parameters and its applicability on the Searchwing drone specific mission. The constraints that each methods imposes, in terms of motion, is assessed, and the research gaps are identified.

Then, two optimization-based methods are derived. In contrast to previous bias estimation methods, the proposed algorithms do not rely on stationary periods to estimate the parameters but leverage the fact that the strap-down integrated accelerations and velocities remain bounded if biases are accurately compensated. Both methods, rely on small time windows for the optimization steps. In addition, two state-of-the-art extended Kalman filter algorithms with gyroscope bias estimation are presented as a baseline.

The performance of the methods is evaluated in two different experimental settings. First, an on-ship calibration scenario is presented. Synthetic IMU data is generated and the limitations of the proposed algorithms are explored within several simulations. Then, the ship motion is simulated with real IMU data in three datasets that emulate the user inputs in moving-vessel. In each of them, the IMU is assumed to stand for a period of approximately 20 seconds on each of its sides on an even surface aboard while a) being static b) oscillating unimodally c) oscillating bimodally. The methods are validated by adding randomized biases to the data on several experiments. For all motions, an average error of 0.06 °/s for gyroscope bias and 0.02 m/s<sup>2</sup> is achieved using simultaneous bias estimation. Pure gyroscope optimization provided an average error of 0.35 °/s. In comparison with the EKF-based methods, the proposed simultaneous-optimization-based method performs more accurately and robustly in all data-sets.

In addition to the calibration scenario, the bias estimation algorithms are evaluated in an on-the-run implementation. A dataset that comprises inertial and optical data from several experiments with different motions and speeds is used. The gyroscope bias estimation accuracy is evaluated when adding randomized gyroscope biases to the measurements. Besides, the methods are evaluated as a modular extension to an orientation estimation filter to simultaneously estimate sensor biases. The analysis showed improvements on the inclination estimation of at least 20 % (between 1 ° - 3 °) for all the analysed measurements, even when very fast motions or disturbances were applied. The modular approach consisting on the proposed optimization-based bias estimation algorithms together

with an analytical orientation estimation filter represent a realistic alternative to extended Kalman filter solutions, with similar accuracy.

The results of the experimental validation on the high-sea calibration scenario are promising, and the algorithms are expected to notably improve accuracy and safety of the the whole Searchwing system, specially on the critical take-off phase. The on-the-run evaluation results showed that the proposed optimization-based algorithms are suitable for an *online* implementation and, consequently, further work regarding optimization of the computational load has to be carried out.

# Acknowledgments

With this Master's Thesis I conclude, not only my studies of Aerospace Engineering, but also an important stage of my academic life. I would like to express deep gratitude to all of those who made it possible and contributed in one or another way. First, I would like to thank to my close friend, Julian Dorner, for introducing me to the Searchwing project. When I was thinking about a thesis topic, he came up with the idea of improving sensor calibration at sea, based on the needs that the drone project had at that moment. I would like to thank to all the Searchwing team for showing interest in this project and supporting it on every possible manner.

This contribution would have never been possible without the great support and guidance of my supervisors of Control Systems Group at TU Berlin: Thomas Seel, who has been a great mentor and did not hesitate to get a bit far out of his research field to supervise this work, Daniel Laidig, who has been incredibly supporting, and especially when regarding technical issues, and Dustin Lehmann, whose great experience with drones was crucial to develop the work at TU Berlin. I feel really thankful for having had the chance to work with all of them.

Ricardo Martínez García  
Berlin, June, 2020.

# Contents

<b>1</b>	<b>Introduction and motivation</b>	<b>13</b>
1.1	Context . . . . .	13
1.2	Overview of the Searchwing project . . . . .	16
1.2.1	Motivation . . . . .	16
1.2.2	Operation area . . . . .	17
1.2.3	The drone: hardware and software description . . . . .	18
1.3	Shortcomings in sensor fusion and calibration . . . . .	19
1.4	Thesis goals and methodology . . . . .	22
<b>2</b>	<b>Fundamentals and state-of-the-art of IMU calibration</b>	<b>24</b>
2.1	Overview of IMU sensors . . . . .	24
2.2	IMU error parameters . . . . .	26
2.2.1	Bias . . . . .	26
2.2.2	Gaussian white noise/Random Walk . . . . .	27
2.2.3	Bias instability . . . . .	28
2.2.4	Scale factors and misalignments . . . . .	28
2.2.5	Summary . . . . .	29
2.2.6	IMU model . . . . .	29
2.3	Review of calibration methods . . . . .	30
2.3.1	Calibration in factory-setting . . . . .	30
2.3.2	In-field calibration . . . . .	32
2.3.3	Discussion . . . . .	37
<b>3</b>	<b>Methods for IMU bias estimation</b>	<b>38</b>
3.1	Model . . . . .	38

3.2	Optimization-based methods for bias estimation . . . . .	39
3.2.1	Quaternions and unit rotations . . . . .	39
3.2.2	Gyroscope bias estimation . . . . .	40
3.2.3	Accelerometer bias estimation . . . . .	44
3.2.4	Computational complexity and tuning parameters . . . . .	46
3.3	Extended Kalman filter with gyroscope bias estimation . . . . .	50
3.3.1	EKF based on gravity observations . . . . .	50
3.3.2	EKF based on velocity observations . . . . .	51
<b>4</b>	<b>On-ship calibration scenario</b>	<b>54</b>
4.1	Simulated-vessel IMU data generation . . . . .	54
4.1.1	Motion . . . . .	54
4.2	Validation on simulated data . . . . .	57
4.2.1	Limitations . . . . .	58
4.3	Validation in ship-like simulated motion . . . . .	61
4.3.1	Gyroscope bias estimation . . . . .	62
4.3.2	Simultaneous gyroscope and accelerometer bias estimation . . . . .	64
4.3.3	EKF-based gyroscope bias estimation results . . . . .	67
4.3.4	Comparison of the methods . . . . .	68
<b>5</b>	<b>On-the-run bias compensation</b>	<b>72</b>
5.1	Data-set description . . . . .	72
5.2	Gyroscope bias estimation . . . . .	74
5.2.1	Linear motion . . . . .	75
5.2.2	Rotational motion . . . . .	76
5.2.3	Arbitrary motion . . . . .	78
5.3	Application to an orientation estimation filter . . . . .	80
5.3.1	Linear motion . . . . .	82
5.3.2	Rotational motion . . . . .	85
5.3.3	Arbitrary motion . . . . .	88
<b>6</b>	<b>Conclusions and recommendations</b>	<b>92</b>
6.1	Outlook and discussion . . . . .	92
6.2	Conclusions and limitations . . . . .	94

6.3 Recommendations and further work . . . . .	95
<b>A Simulation results</b>	<b>97</b>



# List of Figures

1.1	Map showing NGO ships involved in SAR operations in the Mediterranean Sea between 2016 and 1 June 2019. Source: FRA . . . . .	15
1.2	Operation of the Searchwing drone from a ship . . . . .	17
1.3	Search area in Central Mediterranean. Source: [1] . . . . .	18
1.4	Searchwing's X-UAV Mini Talon airframe . . . . .	19
1.5	Extended Kalman filter algorithm. Source: [2]. . . . .	20
1.6	IMU calibration on a GCS in Ardupilot . . . . .	21
1.7	Gyro angle biases estimations by EKF . . . . .	21
2.1	MPU-6050 6 DOF IMU . . . . .	25
2.2	Angle random walk by integrating Gaussian white noise . . . . .	28
3.1	Specific forces on a moving object: change of velocity and gravity . . . . .	40
3.2	Normalized acceleration on the strap-down frame . . . . .	41
3.3	Representation of the time-windows for the estimation. . . . .	43
4.1	Ship in a) rest and b) rolling . . . . .	55
4.2	Synthetic IMU data on a vessel . . . . .	57
4.3	Accuracy of gyroscope bias estimation under different simulated motion parameters . .	59
4.4	Accuracy of accelerometer bias estimation under different simulated motion parameters	59
4.5	Scaling and misalignment effects on the accuracy of the simultaneous bias estimation .	60
4.6	Noise effects on simultaneous accelerometer and gyroscope bias estimation. Gyroscope error is represented on the left, blue $y$ -axis and accelerometer error on the right, red $y$ -axis. On dashed black lines, the typical MEMS IMU values for each noise magnitude are represented. . . . .	61
4.7	Accelerometer and gyroscope measurements in each of the motions . . . . .	62
4.8	Gyro cost function projections on the $XY$ and $XZ$ planes with $b_{g,z} = \hat{b}_{g,z}$ and $b_{g,y} = \hat{b}_{g,y}$ , respectively. . . . .	63

4.9	Estimated and ground truth values for gyroscope biases with the gyroscope optimization-based method on the three datasets . . . . .	64
4.10	Estimated and ground truth values for gyroscope biases with the simultaneous optimization method on the three datasets . . . . .	65
4.11	Estimated and ground truth values for accelerometer biases with the simultaneous optimization method on the three datasets . . . . .	65
4.12	Average error and standard deviation of accelerometer bias error estimation on the three datasets (moving average of 20 seconds) on 20 experiments with randomized biases. . . . .	66
4.13	Estimated and ground truth values for gyroscope biases with the gravity measurements-based EKF on the three datasets . . . . .	67
4.14	Estimated and ground truth values for gyroscope biases with the pseudo-velocity measurements-based EKF on the three datasets . . . . .	68
4.15	Average error and standard deviation of gyroscope bias error (4.14) estimation for all methods on the three datasets after $t = 110$ s. . . . .	70
4.16	Average and standard deviation (shadowed area) of gyroscope bias estimation error (4.14) for the optimization based methods and the 7-states EKF over time. . . . .	70
4.17	Average convergence time to acceptable values . . . . .	71
5.1	Accelerometer and gyro measurements on three recordings of the dataset . . . . .	73
5.2	Optimization results on a single experiment in medium linear motion (2_linear_medium_b0) of gyroscope bias optimization (left) and simultaneous optimization (right). The vertical bias components is not identifiable. . . . .	75
5.3	EKF bias estimation output on a single experiment in medium linear motion (2_linear_medium_b0). The vertical bias component drifts away as it is non-identifiable. . . . .	75
5.4	Average gyroscope bias estimation error ( $x$ - $y$ components) on linear motions for the three methods . . . . .	76
5.5	EKF bias estimation output on a single experiment in medium linear motion (experiment1_rotational_fast_b0) . . . . .	77
5.6	EKF bias estimation output on a single experiment in medium linear motion (experiment3_rotational_slow_b0) . . . . .	77
5.7	Average gyroscope bias estimation error on rotational motions for the three methods . . . . .	78
5.8	EKF bias estimation output on a single experiment in an arbitrary motion (experiment4_arbitrarily_mo_2x60) . . . . .	78
5.9	Average gyroscope bias estimation error on rotational motions with the three methods. . . . .	79
5.10	Average inclination RMSE on linear motions with EKF, and oriEstIMU filter before and after optimization-based bias compensation . . . . .	84
5.11	Average and standard deviation (shadowed area) of orientation estimation RMSE in a medium-speed linear measurement . . . . .	84
5.12	Average and standard deviation (shadowed area) of orientation estimation RMSE in a slow-speed linear measurement . . . . .	84

5.13	Inclination estimation error (5.5) with OriEstIMU on a slow linear motion, before and after applying optimization-based bias compensation. On blue coloured area, the first three overlapping windows of the estimation. The estimation quickly converges to near-ground-truth values after the biases have been identified. . . . .	85
5.14	Average inclination RMSE (5.6) on rotational motions with EKF, and oriEstIMU filter before and after optimization-based bias compensation . . . . .	86
5.15	Average and standard deviation (shadowed area) of orientation estimation RMSE (5.6) in a slow rotational measurement . . . . .	87
5.16	Average and standard deviation (shadowed area) of orientation estimation RMSE (5.6) in a fast rotational measurement . . . . .	87
5.17	Inclination estimation error (5.5) with OriEstIMU on a medium-speed rotational motion, before and after applying optimization-based bias compensation. On blue coloured area, the first three overlapping windows of the simultaneous estimation method. The filter quickly converges to near-ground-truth values after the biases have been identified and compensated. . . . .	88
5.18	Average inclination RMSE (5.6) on arbitrary motions with EKF, and oriEstIMU filter before and after optimization-based bias compensation . . . . .	90
5.19	Average and standard deviation (shadowed area) of orientation estimation RMSE (5.6) in a slow arbitrary measurement (experiment4_arbitrarily_240s_slow) . . . . .	90
5.20	Average and standard deviation (shadowed area) of orientation estimation RMSE (5.6) in a fast arbitrary measurement(experiment4_arbitrarily_mo_2x60) . . . . .	91
5.21	Inclination estimation error (5.5) with OriEstIMU on a slow arbitrary motion, before and after applying optimization-based bias compensation. On blue coloured area, the first three overlapping windows of the estimation. The estimation quickly converges to near-ground-truth values after the biases have been identified. . . . .	91

# List of Tables

1.1	Deaths and arrivals of migrants in the Mediterranean.Source: [3] . . . . .	14
1.2	Achievable search areas and sweep width [1] . . . . .	18
2.1	Summary of MEMS inertial sensor errors . . . . .	29
2.2	Comprehensive summary of latest in-field IMU calibration techniques . . . . .	36
3.1	Computational speed benchmarking of cost function modifications . . . . .	49
3.2	List of tuning parameters . . . . .	49
4.1	Euler angles in the simulated orientations . . . . .	56
4.2	Average (standard deviation) bias estimation error (4.14) under different motion settings	58
4.3	Average (standard deviation) bias estimation error (4.14) of all the methods under different motion settings after a certain time $t$ in 20 experiments with randomized biases between $0.8^\circ/\text{s}$ and $1.5^\circ/\text{s}$ . . . . .	69
4.4	Convergence time [s] to acceptable values of all methods for the three datasets . . . .	69
5.1	List of measurements of the dataset and averages of gyroscope rate and change of velocity	74
5.2	Gyroscope bias estimation error ( $x$ - $y$ components) for linear motions after $t = 90$ s. . .	76
5.3	Gyroscope bias estimation error for rotational motions after $t = 90$ s. . . . .	77
5.4	Gyro bias estimation error for arbitrary motions after $t = 90$ s. . . . .	79
5.5	Inclination RMSE (5.6) for each of the experiments with OriEstIMU filter. A bias of $0.5^\circ/\text{s}$ around the $(1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})$ axis was added to the gyroscope measurements. Parametrization of the filter was: $\tau_{\text{acc}} = 4$ s and $\text{acc}_{\text{rating}} = 0$ . . . . .	82
5.6	Inclination estimation RMSE (5.6) for linear motions from $t_0 = 60$ s to the end of the measurement. The parametrization of OriEstIMU filter was $\tau = 4$ s and $\text{acc}_{\text{rating}} = 0$ . The noise covariance of the EKF was set as $\sigma_w = 10^7$ . . . . .	83
5.7	Relative improvement on the RMSE (5.6) of the inclination in linear motions after applying bias compensation with the optimization-based methods. . . . .	83
5.8	Inclination estimation RMSE (5.6) for rotational motions from $t_0 = 60$ s to the end of the measurement. The parametrization of OriEstIMU filter was $\tau = 4$ s and $\text{acc}_{\text{rating}} = 0$ . The noise covariance of the EKF was set as $\sigma_w = 10^7$ . . . . .	86

5.9	Relative improvement on the RMSE (5.6) of the inclination in rotational motions after applying bias compensation with the optimization-based methods. . . . .	86
5.10	Inclination estimation RMSE (5.6) for arbitrary motions with OriEstIMU filter, before and after applying optimization-based bias estimation, and 6DOF EKF. . . . .	89
5.11	Relative improvement on the RMSE (5.6) of the inclination in arbitrary motions after applying bias compensation with the optimization-based methods. . . . .	89
A.1	Gyroscope optimization: Average (Maximum) error of gyroscope bias [ $^{\circ}/s$ ] . . . . .	97
A.2	Simultaneous optimization: Average (Maximum) error of gyroscope bias [ $^{\circ}/s$ ] . . . . .	97
A.3	Simultaneous optimization: Average (Maximum) error of accelerometer bias [ $m/s^2$ ] . . . . .	98

# 1 | Introduction and motivation

This chapter gives an insight on the motivations of the Searchwing project, an overview over it and its shortcomings regarding inertial sensor-fusion. Firstly, a brief review of the migratory crisis of the XXI century in Europe is addressed, in order to provide a comprehensive review on the context. A summary of the Search and Rescue (SAR) missions and the main participants is provided, together with relevant data from the last years of migratory crisis at sea. Later, the motivations to provide the SAR private organizations with additional intelligence both in air and on board are addressed together with the requirements for the mission. In section 1.2, a summary of the low-cost software and hardware used on the drone is presented. The need of accurate motion tracking for the mission is highlighted, and the instruments, algorithms and short-comings on the drone's sensor-fusion are summarized. To conclude the chapter, section 1.4 defines the goals and research questions covered in this Master's thesis, as well as the structure and methodology of the work carried out.

## 1.1 Context

Since the early 1990s, the illegalization of migration has transformed the Mediterranean into one of the most dangerous and lethal border zones of the planet. The European "refugee" crisis could be addressed, certainly more accurately, as a crisis of the European migration policies [4]. In March 2001, the EU Council Regulation 539/2001 was issued, providing a list of countries that require a visa for entering any of the European countries of the Schengen area. This regulation avoids any chance of a national from any of the listed countries to travel by any mean of public transport to European lands. Thus, inhumane and live risking routes both by land and sea are used as the only option available for migrants. Rubber boats with a payload that surpasses two or even three times its capacity are used to cross the Mediterranean from Turkey to Greece and Cyprus, from Morocco to Spain, and from Lybia, Tunisia and Argelia to Italy or Malta. Migrants pay big amounts of money to smugglers, in their attempt to cross to European land in fishing trawlers.

In [3], a rigorous and quantitative analysis of the number of individuals displaced since 2014 is provided, together with demographic statistics for some of the routes. Syrians and Afghans represent more than 40 percent of the migrants, followed by Moroccans, Algerians and Iraqis. As the data shows, 2015 was the busiest year with more than one million displacements and more than 3,700 deaths. Even though displacements decreased, deaths continue rising, and the ratio almost doubled from 2015 to 2016. Table 1.1 shows the data of arrivals and deaths on the sea from 2014 onwards.

During the first three months of 2016, Greek Islands (Lesbos and others) were the main entrance way of the more than 150,000 refugees that arrived from Syria. But, since the agreement between Turkey and the European Union, the refugees that arrive from the Egean Islands are deported to Turkey and, in consequence, these entry ways towards Europe from there do not represent an viable option from asylum seekers. Thus, new and more dangerous routes were intensified in Central Mediterranean sea, from Libya and Algeria towards Italy. Before the agreement, the routes were around 9 km length, and the migrants could see their destinations within visual distance, but now, routes are much larger - around 300 km towards Lampedusa, and 500 km towards Sicily, and the boats only carry enough fuel

Table 1.1: Deaths and arrivals of migrants in the Mediterranean. Source: [3]

Year	Arrivals	Dead and missing
2019	123,663	1,319
2018	141,472	2,277
2017	185,139	3,1391
2016	373,652	5,096
2015	1,032,408	3,771
2014	225,455	3,538

to reach international waters. In total the trip can last 1 or 2 days. The Central Mediterranean is, consequently, the most critical for the rescue missions, since it is on the way of the longest and deadliest routes of migrants.

When examining the landscape of those engaged in migration and border governance in the Mediterranean, the humanitarian and human right actors multiplied from 2014 to 2016, with NGOs such as Medecins Sans Frontiers, the Watch the Med alarmphone, SeaWatch, SeaEye, SOS Mediterranee, Open Arms and countless volunteers that have worked in the Greek sea to assist those who had been lucky enough to arrive. In this context, the growing emphasis on humanitarianism coexists with a continued security rationale. The movement of people has defied the security and humanitarian logics of the European border regime [5].

Since the humanitarian crisis started, many actors have been involved in Search & Rescue (SAR) missions. At first place, there are the governments of each of the countries receptors of migrants, supported by Frontex and the European Border Surveillance System (EUROSUR). These two border surveillance organizations have been involved in several militarized rescue operations, such as 'Mare Nostrum' in Italy which lasted for a year, and followed by many others up to date. These organizations count with specialized high-tech military surveillance systems: Satellites, drones, aircrafts and live-streams. Nevertheless, their operations are mainly centred in searching missions with no rescuing involved.

Beyond the rescue missions, another civilian volunteering initiatives to help locating people in distress' boats have risen up as well. A good example of solidarity net is the WatchTheMed Alarm Phone, which provides a 24/7 phone number to which migrants at sea can call at. They provide instructions and recommendations to increase migrants' safety when crossing the sea. By 2019, they have been involved in more than 2,500 communications [6]. Another recent initiative is Space Eye, which aims to detect boats through satellite image processing and documenting human-right violations at sea and land [7].

On the other side, several NGOs conduct humanitarian missions facing the lack of political support from the EU countries. These include, among others, SeaWatch, SeaEye and Proactiva OpenArms, which have been developing SAR missions over the Eastern and Central Mediterranean Sea since 2014. Only in 2018, 2,277 died in distress at the sea, which means an average of more than six people per day. Before 2017, a significant part of migrants were rescued by civilian NGOs, who brought them to safety in European territory. Since 2018, however, these civilian operations were treated with hostility from the governments, accusing them of supporting illegal immigration.

As a reaction, many crew members were arrested, like the case of Carola Rackete in June 2019, who defied the Italian authorities and brought the Sea Watch 3 vessel to the Italian port of Lampedusa. The boat, carrying 50 migrants on-board that were rescued in front of the Libyan coast, had been denied docking before. This and many other cases lead NGOs to be involved in legal procedures against the governments and, consequently, in some cases, rescue vessels were blocked in harbours due to flags. The European Union Agency for Fundamental Rights published a report numbering all the legal procedures in which the NGOs have been involved during the last 4 years [8]. In the report, the stats showing the death ratio are presented as well, showing an increase of 2 % from 2014 to 2018. Figure 1.1 pictures the map with rescue vessels involved in search and rescue operations in June 2019,

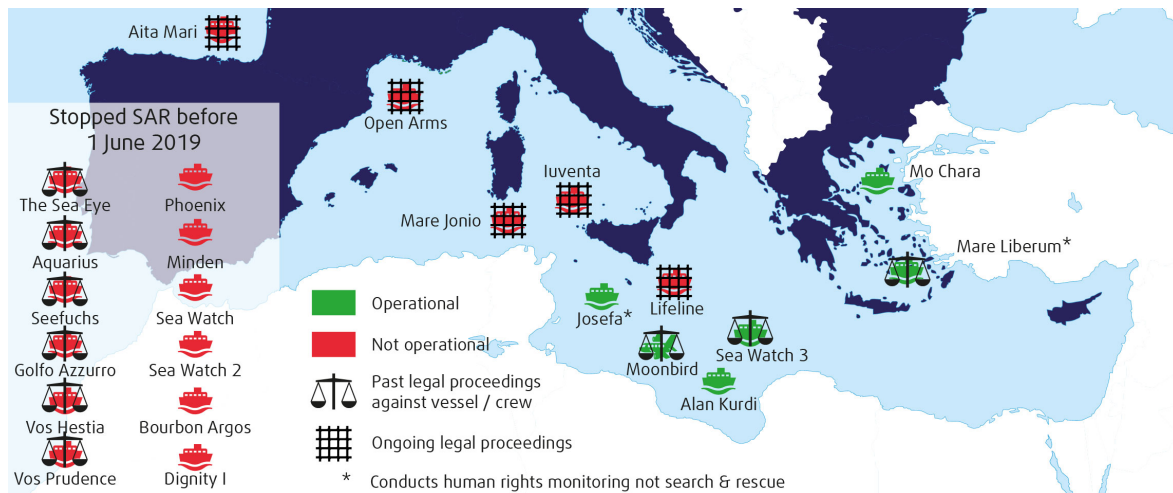


Figure 1.1: Map showing NGO ships involved in SAR operations in the Mediterranean Sea between 2016 and 1 June 2019. Source: FRA

alongside information indicating past or ongoing legal procedures against them.



## 1.2 Overview of the Searchwing project

Private SAR organizations struggle to operate in large areas of the Mediterranean sea with scarce material and financial resources, and sometimes with no institutional support. The Searchwing project aims to construct a low-cost drone to help NGOs to find people in distress at sea. An overview of the motivations of the project together with a brief description of the technical aspects of the drone are presented hereafter. An special emphasis to highlight the shortcomings in sensor fusion and calibration will be taken, so that research and innovation gaps are identified and constitute the main objective of this Master's thesis.

### 1.2.1 Motivation

Despite the great effort made by the civilian rescuers, every year hundreds of people still die on the Mediterranean coasts. For the moment, the principal tools of rescue and surveillance used by the NGOs are, naturally, ships. The operation of a big vessel as a searching tool on the sea has some constraints attached to it. The main equipment is a low resolution RADAR, that can scan up to a radius of 10 km from the ship's position. Nevertheless, the RADAR is not able to detect most of the rubber boats carrying migrants, which makes it a non-viable solution. Thus, binoculars are used to scan visually the horizon, but due to the waves and the Earth curvature, an object of the size of a rubber boat cannot be detected further than 5 km [9].

The experience of the NGOs have shown that there is way to improve some aspects of their searching methods. Currently, their needs are mainly two: First, to detect boat on the sight line of the boat automatically, without needing a person looking through the binoculars continuously. Second, to have an eye in the sky that can reach further areas than the ship's sight line, so the covered search area increases significantly. These problems are to be addressed separately, but in this Master's Thesis only the drone solution is addressed.

Searchwing is a group of volunteers from Augsburg and Berlin that build a low-cost drone specially designated to search people in distress at sea. The main usage framework is to provide support to private rescue organizations such as SeaWatch [10], SeaEye[11] and ResQShip [12]. The work is divided different packages, mainly one regarding flight operations based in Augsburg, and another one focused on image processing, based in Berlin. The goal of the project is the autonomous flight of the drone from start to end, equipped with online image processing so that only the images with actual detections of refugee boats are sent back to the ship to alert the crew.

The description of the mission and the setting provide the necessary requirements for the drone. Ideally, the drone operation should be as automated as possible, requiring the least possible number of humans in the loop, nevertheless, some training is required. These requirements are summarized hereafter:

- **Fully automated flight.** From start to end, the drone should be able to fly over a predefined path that the user sets by using GPS waypoints over a map.
- **More than 50 km range.** In order to satisfactorily extend the search area, a sufficiently long range has to be achieved. Having the restriction of the low power density of LiPo batteries, this is only achievable by using a fixed wing aircraft, with a flight autonomy of 1 to 2 hours.
- **Low cost.** In SAR missions, every minute counts, so there might be the chance that the boat is required to go on the opposite direction of the drone's search area, having to leave it behind. For this reason, the drone should be as cheap as possible so the economical impact of missing a unit is assumable.
- **Detect boats automatically.** In a normal flight, the camera takes thousands of images that, if manually checked, would be a really exhausting task. This is why a image recognition algorithm (either online or offline) is needed so that filters the images in which there are boats appearing alongside GPS coordinates.

- **No false negatives.** The image recognition algorithm should not leave any image with a boat behind. It is better to have false positives than false negatives.
- **Easy to use.** Overall, the drone has to be easily configured and the mission easy to set, so that only small training is required with no special knowledge.

Figure 1.2 shows an illustration of the of the drone operation from the ship. A predefined path is given as GPS-waypoints (in green dotted lines) over an area that surpasses the line-of-sight zone of the rescue ship. The drone scans the area with the mounted cameras and internal image processing and is later retrieved by a RIB boat or lands on a net.

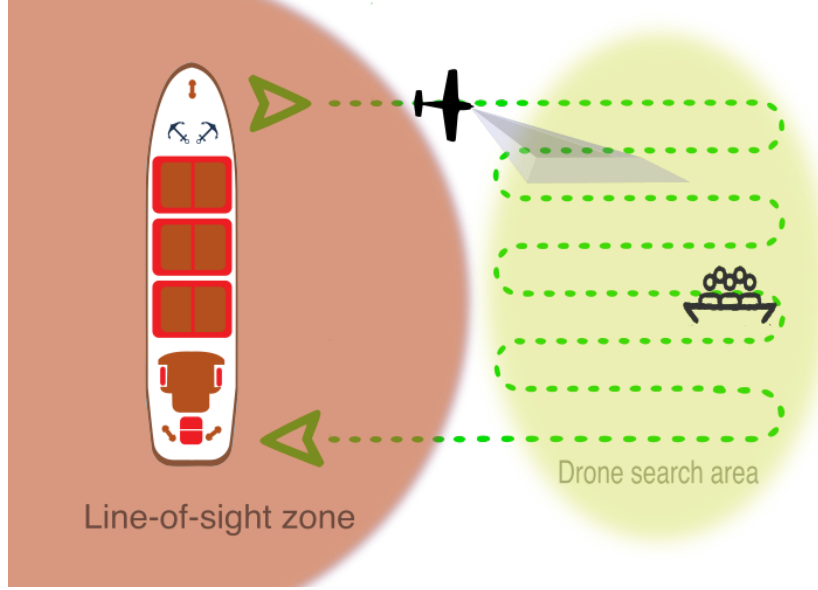


Figure 1.2: Operation of the Searchwing drone from a ship

### 1.2.2 Operation area

To give an overview of the potential improvements on the search of the drone, the search areas of the different tools used by the humanitarian help teams are discussed here. Figure 1.3 shows the search area in the coast of Libya. The distance between Libya and Malta is around 400 km and the distance between Libya and Lampedusa, around 300 km. In this region, the search zone has an approximate area of  $70.624 \text{ km}^2$ .

The Searchwing drone carries a Raspberry Pi v2 camera, which has a focal angle of  $\alpha = 62^\circ$ . Flying at 550 m height, the sweep width would be  $w = 660 \text{ m}$ . the area covered in one flight  $60.7 \text{ km}^2$ . Assuming 10 hours of daylight, 5 flights could be achieved, so the daily area would be  $303 \text{ km}^2$ . The main constricting factor is the low sweep width of the camera at high altitude. This could be improved by placing two cameras with a certain angle of inclination looking at both sides of the drone, so that, approximately the sweep width doubles. In table 1.2, a summary of the covered areas per hour and per day for the drone and the other means of search is presented.



Figure 1.3: Search area in Central Mediterranean. Source: [1]

Table 1.2: Achievable search areas and sweep width [1]

Searcher	Sweep width [km]	Raw area[km <sup>2</sup> /h]	Total area[km <sup>2</sup> ]	Remark
Moonbird	4.3	1225	6243	2500 eur per flight
Seawatch3	9.3	93	930	10 h per day
Searchwing	0.66	60	303	100 km per flight
Searchwing	2	92	920	Double camera

### 1.2.3 The drone: hardware and software description

This section makes a comprehensive description on the drone parts, divided in both software and hardware, in order to provide an overview of the components of the drone and emphasize the relevance of the low-cost requirement. For a better insight of the components, refer to [13]. The approximate total cost of the drone does is less than 600€, excluding the RC controller, which costs around 160 €, although its replacement is less likely. Thus, it represents an economically feasible solution for privately funded SAR organizations. The main features and parts of the drone are summarized hereafter.

- **Airframe:** The main structure of the drone is based on one of the most popular V-tail model low-cost aircrafts, the X-UAV Mini Talon. It is made of polystyrene, which is appropriate to optimize flight durability. See Figure 1.4.
- **Autopilot:** The main controller is board runs the flight control and mission planning algorithms. The Searchwing drone is equipped with the Pixracer, which includes a STM 32 bit processor, 2 IMUs (one 6DOF ICM-20608 and a 9DOF MPU9250), one magnetometer, one barometer, microSD slot, telemetry ports, and 6 PWM out ports.
- **ESC:** The Electronic Speed Controller (ESC) combines the PWM control signal of the micro-controller and the battery power and streams it to the motor.
- **Battery:** A 10,000 mAh battery suitable for long-duration flights.
- **Motor:** A brushless DC motor with 800 - 850 kV (revolutions per Volt).
- **GPS:** An external GPS module for geolocalization.



Figure 1.4: Searchwing's X-UAV Mini Talon airframe

- **Long range telemetry:** An external extended range telemetry, able to send and receive information to and from the drone on distances over 80 km. It is used to send mission commands to the drone as well as retrieving position of people in distress.
- **Camera and image processing:** The pictures are taken by a Raspberry Pi Cam V2, and image analysis software run on a Raspberry Pi. The ultimate function of this set-up is to trigger boat detections and send them back to the boat. The images from detected boats are stored on a separate microSD card together with the position as metadata.
- **RC controller and receiver:** A DC controller complements the drone in order to perform manual landing operations on water, as well as emergency manoeuvres that require pilot intervention. The receiver is connected to the micro-controller through the communication port.
- **Firmware:** The autopilot runs Ardupilot <sup>1</sup> internally, a Free Open Source framework for model airplanes. It allows several modes of manual and automatic flight by receiving only GPS waypoints of a user planned trajectory. It estimates position using an extended Kalman filter (EKF) fusing all the available sensors' information, calculates the aerodynamic surfaces deflection and motor inputs through the internally built-in control laws of the specific model and outputs it as PWM signals.
- **Ground Station:** A Ground Control Software (GCS) runs on the ground computer and is operated by a human for two reasons: In order to send new commands to the drone and monitor the mission, and as a human in the loop on the detection process to avoid false positives. The user plans the mission as a sequence of waypoints on the map on the GCS and then connects with the autopilot to send the information.

### 1.3 Shortcomings in sensor fusion and calibration

Navigation accuracies on the Searchwing drone are a critical operation aspect. In order to perform landing on a net of a moving vessel, both position estimation and control accuracies have to be as optimized as possible. Position accuracies are not that relevant in cruise mission, since deviations of 50 meters are acceptable for geolocalizing the boats. However, landing and take-off manoeuvres are critical in the sense that there is a higher crashing risk. Thus, both position estimation and control accuracies must be as accurate as possible there. Nevertheless, some external factors that can't be controlled, such as wind, are limiting factors of the drone's navigation accuracy.

<sup>1</sup><https://ardupilot.org>

### Sensor fusion methodologies

The Pixracer microcontroller is equipped with two redundant IMUs, GPS and barometer, and it fuses the information from these sensors by means of an extended Kalman filter. The IMU works at a high, non-uniform sampling rate of around 500 Hz. The data from the IMU is downsampled to a lower sampling period  $T_s$ , and the sensors providing the measurements have different sampling rates, so the filter works in a delayed time horizon. The estimation is made by integrating the measurements from the IMU: First, the orientation is computed by strap-down integration and then acceleration is transformed to global coordinates to obtain velocity and position estimates. GPS and barometer work at a lower rate than the IMU, so that when a sample of any of these sensors arrives, an update on the filter is performed by updating the gain and covariance matrices. A schematic showing the EKF algorithm is shown in Figure 1.5. The EKF consists of 24 states which are represented by:

$$\mathbf{x} = \begin{bmatrix} \mathcal{S}\mathbf{q} & \mathbf{v}_{NED} & \mathbf{p}_{NED} & \mathbf{b}_g & \mathbf{b}_a & \mathbf{S}_a & \mathbf{M}_N & \mathbf{M}_B & \mathbf{v}_w \end{bmatrix}^T, \quad (1.1)$$

where  $\mathcal{S}\mathbf{q}$  is the orientation quaternion;  $\mathbf{v}_{NED}$  and  $\mathbf{p}_{NED}$  are the velocity and position in NED coordinates, respectively;  $\mathbf{b}_g$  and  $\mathbf{b}_a$  are the gyroscope and accelerometer biases;  $\mathbf{S}_a$  are the gyroscope scale factors;  $\mathbf{M}_N$  and  $\mathbf{M}_B$  are the soft and hard iron effects on the magnetic field, respectively; and  $\mathbf{v}_w$  is the wind speed estimate.

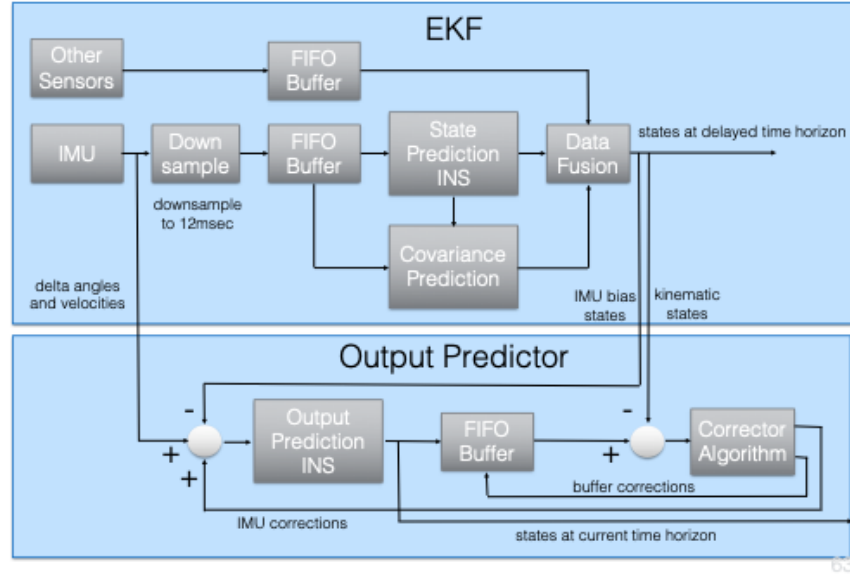


Figure 1.5: Extended Kalman filter algorithm. Source: [2].

The overall accuracy of the entire loop (i.e. position estimate and control) is 3 m, which is sufficient for cruising mode, but lasts short for a landing on the net procedure.

### Sensor calibration

The built-in methods of IMU calibration consist on leaving the sensor still a few seconds, so that the gyroscope biases can be identified by averaging the static signal. Then, the sensor is placed in 6 different static positions, one for each side, in order to identify accelerometer scale factors, and biases. The employed method is known as ellipsoid fitting, and it assumes that under static conditions, the accelerometer output should equal 1  $g$ . A perfect alignment with gravity is not required to achieve an acceptable accuracy, but the method highly depends on the positions achieved by the user. If two positions are switched (i.e. left side with right side), the method is prone to fail. Figure 1.6 shows the calibration guided procedure on a GCS in the Ardupilot platform.

In addition to the calibration process, the EKF is able to estimate the remaining biases of gyroscope and accelerometer on-the-fly, after taking off. However, these estimates take time to stabilize, and

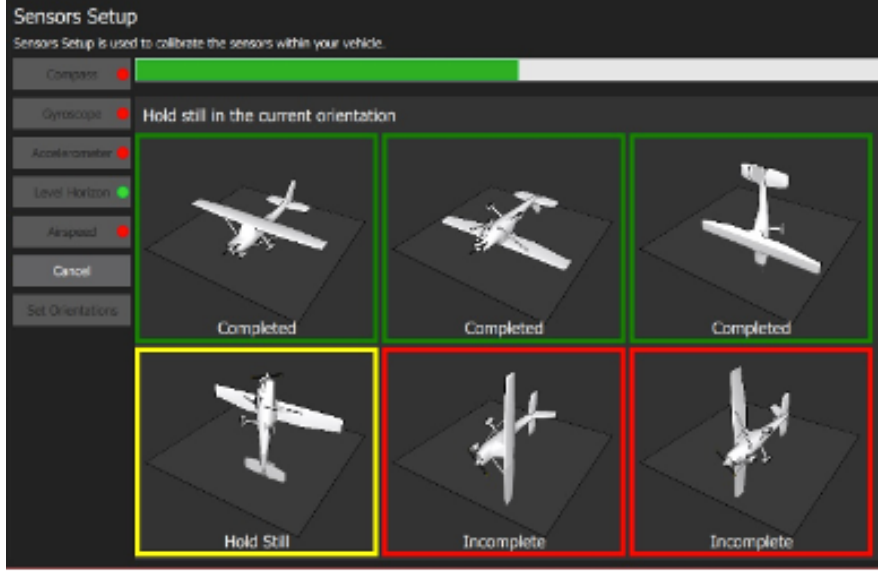


Figure 1.6: IMU calibration on a GCS in Ardupilot

might even diverge if the remaining biases are large. Even with redundant sensors, it is well known that MEMS IMUs are prone to suffer changes in run-to-run biases and variability on the calibration parameters due to ageing and environmental conditions, so that periodic re-calibrations might be needed for an acceptable performance. Moreover, the drone is prone to suffer changes of configuration that may require a complete re-flashing, so that calibration parameters are lost.

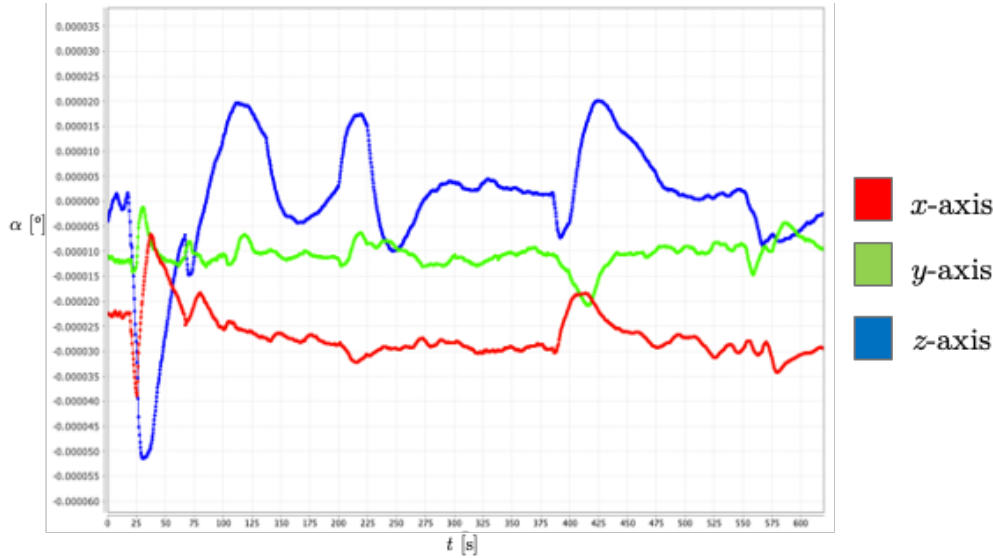


Figure 1.7: Gyro angle biases estimations by EKF

Figure 1.7 shows the EKF estimates of the gyroscope biases. It can be seen that, although small, the estimations take at least 75 seconds to converge. In the case shown at the plot, the biases were not large since a calibration prior flight was performed, as an Ardupilot standard procedure, and the remaining biases were less than 0.1 °/s. On a moving-vessel scenario, this procedure is not possible to perform, since static periods are unlikely. Even if the EKF includes most of sensor error parameters as states, if the sensors are large, this estimate might take time to converge or, might even diverge, which is likely to end up in a crashing. When the sensor spends a few weeks on the sea with adverse environmental conditions, it is likely for IMU sensors to suffer drift on the calibration parameters. Having identified this shortcomings of the Ardupilot framework, which is likely to cause a failure on the Searchwing missions, it will be addressed in this Master's Thesis.

## 1.4 Thesis goals and methodology

The main objective of this Master's Thesis is to provide a reliable non-restrictive tool for compensating sensor biases prior to fly for the Searchwing drone. Bearing in mind the constraints of the high-sea search and rescue missions, the method has to be non-restrictive regarding motions. Thus, performance in different kinds of moving scenarios has to be guaranteed. Additionally, it has to be robust and sufficiently accurate. We understand robustness as the repetitiveness of the results of an experiment under similar conditions; then the algorithm standard deviation of the accuracy under several experiments has to be low enough. Regarding accuracy, we consider an error of 15 % on the magnitude as the lower bound for the acceptance of the algorithm. For typical MEMS biases of 2.5 °/s and 0.4 m/s<sup>2</sup> for rate gyro and accelerometer respectively, acceptable errors would be 0.6 °/s and 0.1 m/s<sup>2</sup>. Overall, the goals of this thesis are summarized as follows.

1. To analyse and evaluate the current approaches for sensor calibration: The restrictiveness of the methods, accuracy and robustness, the additional equipment needed and the level of expertise required.
2. Development an easy-to-use method for sensor bias calibration that can be applied under any condition.
3. Development of a simulation framework for ship-like motion. The simulation framework provides a testing tool for the algorithm, and allows to vary several motion parameters of the motion.
4. Refinement the method under the conduction of several experiments: providing a grid for the tuning parameters, and optimizing the values for specific datasets.
5. Validate the algorithm on real sensor data with added biases and compare with other state-of-the-art methods.
6. Development of EKF-based methods for gyroscope bias calibration, in order to provide different approaches to the problem.
7. Validate the algorithm together with an orientation estimation filter. Compare the accuracy of the orientation estimation by using an optical tracker reference. Compare performance of the developed methods under the different data-sets and simulations.
8. Analyse and discuss the results to identify gaps and the future improvement possibilities.

The presented goals can be interpreted as the research questions that this thesis is tackling with. To be more concise, the questions to be addressed in this thesis are:

1. What are the current state-of-the-art calibration algorithms and how restrictive are regarding a non-static setting? Would any of these be applicable to the Searchwing use-case?
2. How should the characteristics of a calibration method that could be applied under any setting be, and which tuning parameters conform the method?
3. What parameters influence the algorithm's accuracy and robustness and what values might lead the algorithm to fail?
4. What is the accuracy of the algorithm under different assumptions and what are the shortcomings and advantages with respect of other state-of-the-art calibration methods?
5. Under which motions and conditions would the algorithm perform with acceptable accuracy?
6. Could the algorithm be applied in an on-line setting as a complementary tool to orientation estimation filters?

Although out of the scope of this thesis, a relatively user-friendly software tool has to be provided for the correct implementation of the algorithm on the Searchwing missions. This include to develop a software, ideally a GUI or a command-line scripting tool, that uses the drone communication protocols (MAVLink) to log the calibration data from the drone IMUs, and then writes the calibration parameters on the micro-controller board.

In order to accomplish the defined thesis goals, the following methodology has been employed:

- **State of the art research:** To carry an intensive and comprehensive literature research, in order to collect all published methods that could potentially be applied for the use-case, find research gaps and propose how to mitigate them. The research methodology consisted in compiling more than 140 research papers which included certain topic-related keywords, i.e. 'calibration', 'IMU', 'inertial', 'gyroscope'. After reading the abstracts, the most relevant ones were included in the literature review carried in Chapter 2.
- **Prototyping of the algorithm:** To make a draft definition of the physical and intuitive aspects of how a non-restrictive sensor bias estimation method should be based on. The draft prototyping is made by using numerical methods in MATLAB, and tested with several datasets simulating a moving vessel. Several methods are developed and its efficiency is compared in different scenarios. Individual experiments were conducted to find the acceptable grid of the tuning parameters.
- **Testing and analysis through simulations:** Identification of acceptable accuracy boundaries by using simulated IMU data in MATLAB to identify the parameters and motion constraints that would make the method to potentially fail. Additionally, tuning parameters of the algorithm are optimized through intensive simulation tests. The tests conduct to a refinement of the algorithm, in terms of accuracy, robustness and computational complexity.
- **Testing on-the-run calibration scenarios:** Once the algorithm has reached a mature version, possible relevant scenarios where a non-restrictive inertial sensor bias estimator could be useful are identified. This is the case of using such method as a complement to orientation estimation filters. The methods are analysed on a wide range of motions and conditions together with an orientation estimation filter and optical orientation data as reference by using Python code.

The Thesis is structured as follows: In chapter 2, a literature research on inertial sensor calibration methods is presented. Chapter 3 focuses on the derivation of the methods; two novel optimization-based methods for accelerometer and gyroscope bias estimation, and two extended Kalman filter based solutions based on pseudo-measurements of velocity and gravity. In chapter 4, the on-ship calibration scenario is analysed: The simulated-vessel data is described, and the methods are validated by using both synthetic and real ship-like motion IMU data. Then, in chapter 5 the methods are applied to a data set that contains a wide range of different motions, and an experimental evaluation is carried on. The performance of the EKF-based and optimization-based is compared, and a validation of the optimization-based algorithm is made by applying it together with an orientation estimation filter. Lastly, chapter 6 extracts the conclusions based on the analysis over the data sets and identifies gaps and future work packages.



# 2 | Fundamentals and state-of-the-art of IMU calibration

The cheapening of micro-electromechanical (MEMS) based sensors manufacturing techniques have allowed these products to become a standard feature in many user applications. Inertial sensors have been an active topic of research since the 1960s, with publications including sensor fusion algorithms, calibration methods and sensor modeling, among others. However, since the expansion of MEMS sensors in the late 1990s, the number of publications per year has multiplied until reaching about 200 papers and dissertations about Inertial sensors yearly [14]. Among these research topics, in-field calibration techniques have been extensively developed, and in the last 20 years diverse methods and algorithms that consider different approaches, restrictions and tools have been studied. This chapter provides an insight on different methods of IMU calibration. The review will be preceded by a comprehensive description of the different sources of IMU errors and will focus especially on in-field calibration techniques with minimal equipment needed. In-field calibration methods have a special relevance since are those to be applied by end-users in MEMS inertial sensor triads. Special attention on the boundary conditions and assumptions made by each of the algorithms will be given, in order to evaluate the applicability of each them in non-restrictive scenarios, like, for example, a moving vessel.

## 2.1 Overview of IMU sensors

An Inertial measurement unit (IMU) comprises accelerometer and gyroscope, which provides a vital role in giving information about navigation (i.e. velocity, position, attitude) for different applications. Accelerometers and gyroscopes are usually mounted in orthogonal triads of sensors to measure specific forces and angular velocities along each of the axes, respectively. Sometimes, the inertial sensors are mounted on the same platform in a redundant way, forming inertial sensor arrays, which provides higher accuracy and reliability. The application fields of IMU sensors is very wide, to cite some of the fields and applications [15]:

- **Robotics:** Wheel-slip detection, legged robots, etc.
- **Medical rehabilitation:** Exoskeleton rehabilitation, post-traumatic motion analysis, arm-posture correction tools.
- **Navigation systems:** GPS faults detection, INS/GPS systems, quadrotor stabilizers.
- **Sports learning:** Measuring spinning dynamics and hand dynamics.
- **Manufacturing quality control:** Inertial-head and fastening-tool trackers.
- **Augmented reality systems:** Personal Outdoor Navigation.

The information from the IMUs is usually combined with other sources such as GNSS systems, barometric sensors, airspeed sensors or cameras used for optical flow navigation. Complex sensor fusion techniques to determine position and orientation that use inertial sensors have been extensively developed, such as extended Kalman filters [16] or optimization-based orientation estimation filters [17, 18]. These algorithms are able to combine the information from different sensors and compensate the errors of one with the advantages of another. The accuracies of the estimates from inertial sensors depend, however, on the sensor quality and the error embedded. These errors accumulate with time after integrating the sensor measurements, resulting in large errors in orientation and velocity estimations. The absolute error given by the sensor is a determinant factor of the application of the sensor, and it spans from low-accuracy commercially available micro-electromechanical systems (MEMS) sensors, to tactical grade, expensive, optical sensors.

Advancements in silicon chips manufacturing have allowed MEMS inertial sensors to become a standard feature in many devices, such as smart-phones, Unmanned Air Vehicles (UAVs) or video-games controllers. Usually, these sensors are commercialized mounted on the same chip, forming an inertial sensor triad, or 6 degrees-of-freedom (6 DOF) IMU. Figure 2.1 shows a popular MEMS 6 DOF IMU which consists in a 3-axis gyroscope and accelerometer. The cost of these modules is usually less than 10 €, and the size is very reduced, of around  $20 \times 15$  mm. Despite its cost and dimensions, MEMS sensors have considerably lower performance than the optical counterparts, mainly due to axes-misalignment, changing bias due to thermo-electrical effects, and a high-variance random noise. Moreover, the errors present in MEMS sensors change with time and are very dependant on environmental conditions, such as temperature. In practice, the information from MEMS sensors is treated by means of extended Kalman filters (EKF) in which the navigation states are extended to estimate calibration parameters, such as sensor biases. But convergence of the filter might take time, or might even diverge if the initial estimates are not good enough. Thus, performing periodic re-calibration to compensate these errors is needed to achieve a correct performance of the Inertial Navigation System (INS) for a sufficiently long period of time. Research has proved that even fiber-optic gyroscopes change their calibration parameters over time, due to environmental disturbances and ageing of the device [19].

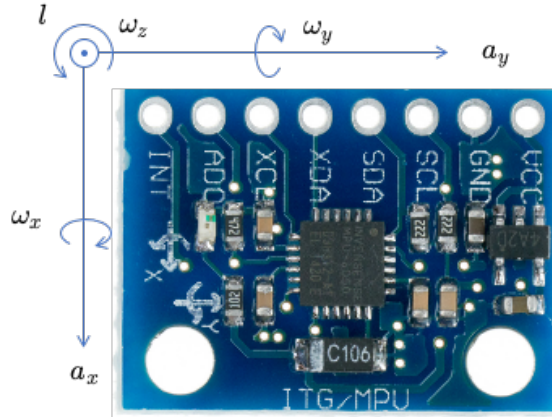


Figure 2.1: MPU-6050 6 DOF IMU

Calibration is understood as the compensation of deterministic errors, such as bias offsets, scale factors or G-sensitivities. The traditional way of calibrating inertial sensors is to use a turntable precision table where to perform several known orientations and angular velocities as reference values, to then estimate the error parameters after comparing the IMU signals. This procedure, although effective, is really time-consuming and costly due to the precision equipment needed, and for MEMS inexpensive sensors might not make sense to use it, for obvious reasons.

Recalibration scheme was initially and almost exclusively developed, for MEMS sensors which parameters change over time [20]. In-field calibration methods that avoid the use of expensive tools have been extensively developed in the recent years, and represent a more realistic and user-friendly solution for MEMS sensors calibration. These methods usually require the user to perform a set of different specific motions or rotations and to leave the IMU standing still in different orientations. Two big groups of algorithms may be defined regarding in-field calibration schemes: Multi-position based, and Kalman

filter based. First group's approach is usually to define a set of variables (i.e. the sensor errors) and a cost function. Then, the parameters are estimated by defining an optimization problem, so that the estimated values minimize the cost function. Optimization techniques are quite diverse, ranging from classical gradient-based Gauss-Newton methods to more modern algorithms such as Particle Swarm Optimization [21, 22], or Maximum Likelihood approaches [23, 24]. The second group, uses a Kalman filter to estimate the calibration parameters together with the navigation states in one framework. Some Kalman filtering techniques require external equipment to provide a reference trajectory, like a GPS receiver, and some others do not require any specific equipment. A third group of calibration techniques have arose in recent years with the addition of cameras to the INS, i.e. an inertial-optical framework. These methods use a optical reference from the camera to estimate the error parameters by means of either least-square methods, or shape-from-motion methods.

To estimate the error parameters, these algorithms use, as a reference, the gravity acceleration and Earth's rotation always present in nature. The approaches are diverse: some require a perfect alignment of the gravity vector with the sensor axes whereas some others rely on the fact that, under static conditions, the output of the accelerometer should always be equal to the magnitude of the local gravity vector. In order to estimate the gyroscope bias offsets, most of the algorithms rely on the same principle: under static conditions (thus, no external rotations applied) the output of the gyroscope should be zero on average. Some other algorithms are based on quasi-static detectors [25]. These consist of finding instants of time on the motion where the sensor is in a almost static position, i.e. the standard deviation of a cluster of samples from the sensor is under a certain threshold value. However, these detectors do not seem to work well in case of detecting quasi-static gyroscope measurements in practice [26].

## 2.2 IMU error parameters

In this section the different sources of errors for MEMS accelerometers and gyroscopes are presented. IMU errors are differentiated in two main groups, deterministic and non-deterministic errors. The deterministic errors affect to the system performance and are estimated in the calibration procedure. Deterministic errors include scale factors, misalignments and constant biases, as well as G-sensitivities, although the latter will not be considered in this overview. The other group of errors are non-deterministic. Thus, its values over the time are random variables rather than mathematical relations and, hence, cannot be eliminated from the readings. However, random errors can be modeled by statistical parameters that fit a specific stochastic model. The analysis of this errors can be done by using tools such as a Power Spectral Density (PSD) plot and the Allan-Variance method [27], which identifies several sources of noise on the whole frequency spectrum by analysing a large number of samples and dividing it into variable length clusters. These errors mainly include Gaussian white noise and low-frequency noises such as random-walk and bias instabilities. For the sake of simplicity, other errors such as temperature-derived, and quantization or resolution errors are neglected since its relevance for system accuracy is less than the above mentioned.

### 2.2.1 Bias

The bias is the non-zero value that the sensor outputs when having no input. A bias error  $b$ , when integrated causes an error that grows linearly over time as  $\epsilon(t) = bt$ . If bias errors are not compensated for, it will accumulate and lead to drifts in attitude and position. It is usually expressed in degrees/s for gyroscopes and  $\text{m/s}^2$  for accelerometers. Bias error is divided a constant part, which is deterministic and hence, identifiable by calibration, and a variable part consisting in a random drift (i.e. the bias instability) and a temperature drift, which is also deterministic.

### 2.2.2 Gaussian white noise/Random Walk

The output of a MEMS sensor will be perturbed by some thermo-mechanical noise which fluctuates at a rate much greater than the sampling rate of the sensor. As a result the samples obtained from the sensor are perturbed by a white noise sequence, which is simply a sequence of zero-mean uncorrelated random variables, with a variance  $\sigma$ . Thus the noisy signal can be represented as:

$$\hat{y}_t = y_t + \epsilon_t \quad (2.1)$$

where  $\epsilon_t$  is the normal distribution with zero mean and variance  $\sigma$ :

$$X_t \sim \mathcal{N}(0, \sigma). \quad (2.2)$$

This noise is usually modelled by the noise density parameter,  $ND$ , which is related to the variance by:

$$\sigma = \sqrt{\frac{F_s}{2}} ND. \quad (2.3)$$

The effects of the Gaussian white noise are visible on the integrated signal as a low frequency noise Random Walk. To see the effect that it has on the integrated signal, let  $N_i$  be the the  $i^{th}$  random variable of the noise assumed to have a normal distribution with mean  $E(N_i) = E(N) = 0$  and variance  $\text{Var}(N) = \sigma^2$ . The result of integrating that signal is:

$$\int_0^{\tau=t} \epsilon_\tau d\tau = T_s \sum_{k=0}^n = N_i, \quad (2.4)$$

where  $n$  is the number of samples and  $T_s$  the sampling time. Then the mean and the variance of the integrated signal are as follows:

$$E\left(\int_0^t \epsilon(\tau) d\tau\right) = T_s \cdot n \cdot E(N) = 0 \quad (2.5)$$

$$\text{Var}\left(\int_0^t \epsilon(\tau) d\tau\right) = T_s \cdot n \cdot \sigma^2. \quad (2.6)$$

Therefore, the random walk noise of the signal is a zero mean process whose standard deviation is:

$$\sigma_{rw}(t) = \sigma \sqrt{T_s \cdot t}, \quad (2.7)$$

and grows over time.

In the case of a gyroscope at rest, for example, the output should be zero, but there is always some white Gaussian noise added on. Sometimes the noise will take the output above zero, and sometimes below. On average (for white noise) the time above and below zero will even out. If we integrate the signal to get the angle, we would hope to have the integration equal zero, since we know the sensor is at rest. The noise, however, will make the integration bounce around, and the longer is the integration time, the higher this uncertainty is. Thus, the random walk can be understood as the error that provides a limit on any angle measurement that solely relies on rate integration.

The random walk standard deviation parameter is often given at a specific time by MEMS sensor manufacturers, say  $\sigma_{rw}(1)$ , and it is usually expressed in  $^\circ/\sqrt{h}$  for gyroscopes and  $m/s/\sqrt{h}$  for accelerometers.

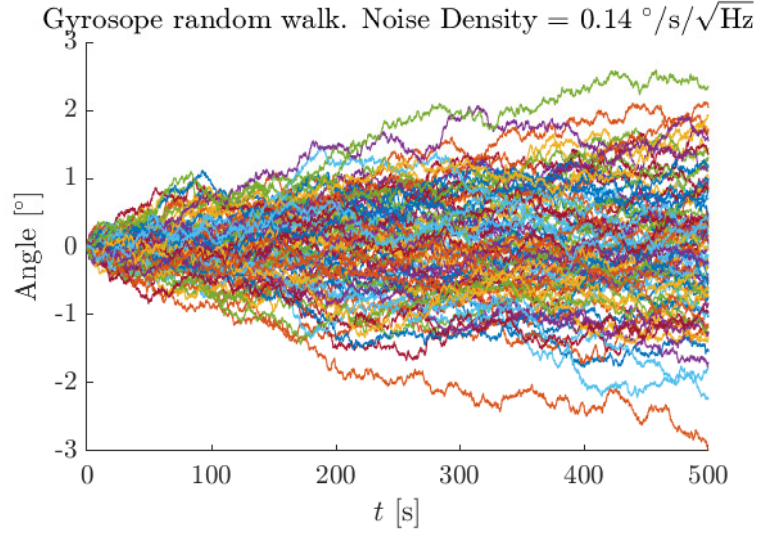


Figure 2.2: Angle random walk by integrating Gaussian white noise

### 2.2.3 Bias instability

The constant offset of the MEMS sensors wanders over time due to flicker noise in the electronics and in other components susceptible to random flickering. Flicker is a noise which effects are usually observed at low-frequencies in electronic components. At high frequencies, flicker noise tends to be overshadowed by Gaussian white noise. A bias instability measurement describes how the bias of a device may change over a specified period of time, in fixed conditions. Bias instability is usually specified as a  $\sigma_b$  value with units  $^{\circ}/h$ , or  $^{\circ}/s$  for less accurate devices. This is interpreted as follows: The bias offset,  $b_t$  is known at an instant  $t$ , then the bias at an instant  $t + \tau$  is a random variable with mean  $b_t$  and variance  $\sigma_b$ .

$$b_{t+\tau} = b_t + X_t, \quad (2.8)$$

with  $X_t \sim \mathcal{N}(b_t, \sigma_b)$ .

### 2.2.4 Scale factors and misalignments

These types of errors are due to small mechanical deficiencies on the assembly or calibration inaccuracies. Scale factor error is the relation between the output and the input. For an input of, say 100%, the expected output should be also 100%, ideally, but if there is a 1% scale factor error, the output will be 101%.

The three gyroscopes and accelerometers are mounted orthogonally to each other, but this orthogonality is, however, not perfect, so the angles are not exactly  $90^{\circ}$ . This leads to a correlation between sensors, so that part of the magnitude that should be measured in one axis is measured by the other, and vice-versa. For triaxial MEMS accelerometers and gyroscopes the effect of these errors on the signal can be represented in a matrix form

$$\hat{\mathbf{y}} = \begin{bmatrix} S_x & c_{xy} & c_{xz} \\ c_{yx} & S_y & c_{yz} \\ c_{zx} & c_{zy} & S_z \end{bmatrix} \mathbf{y}, \quad (2.9)$$

where,  $S_i$  is the scale factor in the  $i^{th}$  axis, and  $c_{ij}$  represents the misalignments between the axes  $i$  and  $j$ .

Scale factors and non-orthogonalities are, unlike Gaussian noise and bias instability described above, deterministic errors, so it can be compensated with a careful assembly process and a correct calibration. The influence of these errors are only visible when there is some input on the sensor, so the calibration is only possible then. In tri-axial accelerometers, the value of gravity is the best reference signal to compensate these errors and determine the parameters in (2.9). For tri-axial gyroscopes, on the contrary, this process may require of expensive precise turn tables, since the Earth rotation is too weak to determine the parameters accurately.

### 2.2.5 Summary

The main sources of errors are summarized in Table 2.1. The relative importance of every of each error depends on the type of sensor and the manufacturer. Regarding deterministic errors, a good correction of the constant bias offset is crucial for a good performance of both sensors, and on the case of tri-axial accelerometers the scale factor and non-orthogonalities parameters are of special importance, and it is also easier to calibrate due to the natural observability of the gravity signal. For gyroscopes, a good bias stability performance is generally a rule of thumb to decide the goodness of a sensor, whereas for an accelerometer is not that important when it comes to strap-down inertial navigation systems applications. As discussed, angle random walk of the gyroscope can be considered as an uncertainty bound for angle integration, and so it does for accelerometers and velocities.

Error type	Description	Parameter gyro/accel	units
Constant bias	A constant offset on the measured signal	$[\circ/\text{s}]$ / $[\text{m}/\text{s}^2]$	
Gaussian noise	Random white noise with zero mean and standard deviation $\sigma$ . Described by the noise density parameter.	$[\circ/\text{s}/\sqrt{\text{Hz}}]$ / $[\mu\text{g}/\sqrt{\text{Hz}}]$	
Random Walk	Result of the integration of gaussian noise, with growing with time standard deviation $\sigma_{rw}(t) = \sigma\sqrt{T_s \cdot t}$	$[\circ/\text{s}\sqrt{\text{Hz}}]$ / $[\mu\text{g}\sqrt{\text{Hz}}]$	
Bias instability	Low frequency flickering noise on the bias offset with variance $\sigma_b$	$[\circ/\text{h}]$ / $[\mu\text{g}]$	
Scale factors and misalignments	Deterministic errors due to scaling and non-orthogonalities that can be compensated through calibration	-	

Table 2.1: Summary of MEMS inertial sensor errors

The effect of these sources of errors and its influence on the bias estimation algorithm will be studied, by simulating IMU measurements and progressively incrementing the values of the errors, so the upper limit of each source is calculated.

### 2.2.6 IMU model

Adding-up all sensor error definitions from above, a mathematical model to represent both gyroscope and accelerometer measurements as function of angular velocities and accelerations is shown in (2.10) and (2.11).

$$\begin{bmatrix} \hat{\omega}_x \\ \hat{\omega}_y \\ \hat{\omega}_z \end{bmatrix} = \begin{bmatrix} b_{G,x} \\ b_{G,y} \\ b_{G,z} \end{bmatrix} + \begin{bmatrix} S_{G,x} & c_{G,xy} & c_{G,xz} \\ c_{G,yx} & S_{G,y} & c_{G,yz} \\ c_{G,zx} & c_{G,zy} & S_{G,z} \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} + \begin{bmatrix} \epsilon_{G,x} \\ \epsilon_{G,y} \\ \epsilon_{G,z} \end{bmatrix}, \quad (2.10)$$

$$\begin{bmatrix} \hat{a}_x \\ \hat{a}_y \\ \hat{a}_z \end{bmatrix} = \begin{bmatrix} b_{A,x} \\ b_{A,y} \\ b_{A,z} \end{bmatrix} + \begin{bmatrix} S_{A,x} & c_{A,xy} & c_{A,xz} \\ c_{A,yx} & S_{A,y} & c_{A,yz} \\ c_{A,zx} & c_{A,zy} & S_{A,z} \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} + \begin{bmatrix} \epsilon_{A,x} \\ \epsilon_{A,y} \\ \epsilon_{A,z} \end{bmatrix}. \quad (2.11)$$

Here,  $\omega_{x,y,z}$  and  $a_{x,y,z}$  are the angular velocities and accelerations applied to each axis, and  $\hat{\omega}_{x,y,z}$  and  $\hat{a}_{x,y,z}$  are the gyroscope and accelerometer outputs, respectively. The scale factors on each axis are designated by  $S_{G,x,y,z}$  for the gyroscope and  $S_{A,x,y,z}$  for the accelerometer. The cross-axis sensitivities for gyroscope and accelerometer are represented in the off-diagonal terms of the same matrix as  $c_{G,x,y,z}$  and  $c_{A,x,y,z}$ . The terms  $\epsilon_{G,x,y,z}$  and  $\epsilon_{A,x,y,z}$  represent the random noises on each of the sensor axes, for gyroscope and accelerometer, respectively.

## 2.3 Review of calibration methods

The calibration of a sensor is a process to compare its output with a known reference value. It allows to determine a set of parameters that comprise the mathematical model of sensor errors stated in the previous section. The estimated parameters, i.e., bias, scale factors and misalignments, attempt to form an exact relationship between the expected and the output values. IMU calibration techniques can be widely divided in two groups: a) Those that require specialized equipment and are performed in a laboratory or factory setting, and b) Those that do not require precise equipment and can be performed by the user itself in-field. The first group uses precise equipment turntables and other tools to align every axis of the sensor with gravity and compare the output of the sensor with the expected value. The latter, define a set of boundary conditions on the user motion (i.e. pseudo-static moments or approximate rotations) and estimates the parameters either by means of Kalman filtering or by optimization-based methods. This review will focus especially in the second group of calibration methods, due to its applicability to low-cost MEMS sensors, and focusing special attention in the restrictiveness of the methods. We understand restrictiveness of a calibration method as the number of boundary conditions needed for the method to succeed. Thus, a method which needs a specific number of rotations and a specific number of static positions is more restrictive than another that allows the user to perform arbitrary motions. Additionally, in-factory calibration methods will be briefly reviewed for contextual reasons.

### 2.3.1 Calibration in factory-setting

Traditionally, calibration has been done in a factory setting, by means of high-precision tools and equipment. The usage of these tools allow to estimate constant biases, scale factors and misalignments. The standard methodology consists of mounting the IMU on a leveled turntable with its axes aligned with the turntable's axes, and then perform six position motion so that each sensor of the triads is aligned with the vertical axis of the local frame. If the alignment is perfect, then the observations should be equal to the local gravity vector or the earth rotation rate in the case of gyroscopes, and then the scale factor and bias error parameters are directly identifiable as in (2.12):

$$\begin{aligned} y_i^{\text{up}} &= b_i + S_i K \\ y_i^{\text{down}} &= b_i - S_i K, \end{aligned} \quad (2.12)$$

where  $K$  is a known constant, the local observed gravity for accelerometers, and a constant external rotation rate in the case of gyroscopes. This procedure is repeated for each axis of the sensor, but the accuracy highly depends on the accuracy of the alignment of the sensor frame to the global frame, which is hard to achieve [20]. Besides, it does not exploit the non-orthogonality parameters of the model. Thus, an improved method was proposed by Niu in 2002 [28] that allowed to estimate the three types of deterministic errors, following the model in (2.11) and (2.10). The method estimates the parameters by a least-squares estimator:

$$\mathbf{M} = \mathbf{U}\mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}, \quad (2.13)$$

where  $\mathbf{M}$  is the misalignment-scale factor matrix,  $\mathbf{U}$  is the concatenation of the inputs of the accelerometer at each position and  $\mathbf{A}$  the concatenation of the expected values at each position, as given in (2.14)

$$\begin{aligned}\mathbf{A} &= \begin{bmatrix} K & -K & 0 & 0 & 0 & 0 \\ 0 & 0 & K & -K & 0 & 0 \\ 0 & 0 & 0 & 0 & K & -K \end{bmatrix}, \\ \mathbf{U} &= \begin{bmatrix} \mathbf{y}_{+x} & \mathbf{y}_{-x} & \mathbf{y}_{+y} & \mathbf{y}_{-y} & \mathbf{y}_{+z} & \mathbf{y}_{-z} \end{bmatrix}.\end{aligned}\tag{2.14}$$

For MEMS gyroscopes, the projection of the Earth's angular rate in each axes is easily shadowed by the high noise and bias instability errors, which makes the parameters unobservable. Thus, the gyroscope has to be mounted in a high-precision controlled rate turntable where the constant  $K$  is known for each axis. The least-squares method in (2.13) was later modified by El-Diasty et al. in 2008 [29] by including a  $6 \times 6$  weight matrix  $\mathbf{P}$ , which tries to improve the calibration accuracy. Later, Shin and El-Sheimy [30] proposed a multi-position scheme that could also be applied to in-field calibration, which assumes that the sum of the output of each sensor's axis must be equal to the reference value. However, a precision turntable that provides large rotation rates was later introduced by Syed [20]. The multi-position method estimates all parameters only by means of a low-grade single-axis turntable, notably reducing the equipment needed.

Kalman filtering techniques have been extensively used for laboratory calibration. Bekkeng [31] proposed a Kalman filter that calibrates gyroscope's 12 error parameters using a ramp profile calibration maneuver input at all axes. This procedure, although effective may cause additional misalignments since it requires each axis of the IMU to be mounted on the single-axis turntable. The observability of the parameters is only realistic under very controlled laboratory conditions. Nieminen et al. [32] implemented a simple Kalman filter that took advantage of the centripetal acceleration caused by rotating the table. Moreover, the numerical method is simpler and extensible to multiple sensors. Kalman filters also allowed improvements in the multi-position techniques, like the ones proposed in [33] by Wang et al., where they require the IMU to be mounted on an inclined static plane in an alternately up and down fashion for each axis, and integrates the thermal compensation to the filter. Other algorithms also include g-dependent biases on their Kalman filters, like the one proposed by Radi et al. [34]. The g-dependent bias coefficients accounts the effect of the specific force on the gyroscope reading and is obtained by placing the IMU on 27 different positions about three axes.

The number of linearly-independent measurements is limited to the sensor schemes presented above, so new methodologies based on three-dimensional vibration generators have been proposed, which exploit the frequency-dependant errors on the sensors [35]. Choi et al. defined a methodology of calibrating the IMU using pendulum motion rather than the earlier known techniques which employ turntable to generate known references [36], which is based on a vibration generator as well. The movement of the pendulum with sensors located at their bob is measured using high-accuracy angle sensors. This method constructed a non-linear pendulum equation whose coefficients are estimated using neural network and aid in generating the reference dataset. This reference data is compared with the sensor output to obtain the calibration parameters. Renk et al. [37] used a robotic arm to perform an optimization-based calibration of a magnetometer-accelerometer pair. To perform this calibration, the sensor is mounted to the end of a robotic arm. Accelerometer measurements are recorded as the robot moves three of the six joints, performing rotational movements in roll, pitch, and yaw, which correspond to rotations about the x-axis, y-axis, and z-axis of a reference frame, respectively. The magnetometer-accelerometer estimates of the Euler angles are compared to the ones provided by the sensors on the robot joints, which provide direct analogue voltage, and the calibration parameters are optimized by minimizing the error from the reference values.

All methodologies presented up to now are meant to be used in a controlled laboratory setting, with different types of equipment. It has been seen that some of the methods above require precise tooling, like precise 3-axis turntables, to achieve the desired accuracy. Kalman filter-based options seem to be a more realistic option when the equipment available is limited, as some approaches only consider a low-grade single-axis turntable [31, 32]. Nevertheless, laboratory calibration is a time-consuming task since the sensor has to be carefully aligned with the axes of the turntable, and this will make it not very convenient for low-cost sensors. Moreover, the turntable where the motions are performed would need



from special equipment to be calibrated as well. Low-grade sensors error parameters are constantly changing over time due to environmental conditions, and doing periodic laboratory re-calibration does not seem to be wise option. Therefore, in-field calibration methods that are applicable under less-restrictive circumstances and do not require to mount the sensor several times are necessary. The following section makes an overview of these schemes, making emphasis in the circumstances where the methods can be applied.

### 2.3.2 In-field calibration

The rising demand of MEMS sensors and its spread over multiple applications has lead to an intensive research on in-field calibration methodologies. MEMS sensors are prone to suffer variations in the error parameters, needing periodic re-calibration that should be applicable on-site. Calibration techniques are varied, but most of them rely on the same principle: under static conditions the sensors output should be equal to the Earth's spinning rate and the local gravity, for gyroscope and accelerometer, respectively.

One of the first publications about in-field calibration dates back to 1994, from Ferraris et al. [38]. In their work, they propose a six-position method (each of them aligned with gravity) from which accelerometer scale factors and biases can be estimated in an analogous way to (2.12). Gyroscope biases are compensated by averaging the outputs of the static part, and gyroscope scale factors are compensated by integrating the gyroscope outputs and comparing the rotated angle. The biggest shortcoming of this method is that it does not account for a misalignment error between the platform and local gravity. Bachmann et al. in 2003 presented a simple hand calibration method for the accelerometer, gyroscope, and magnetometer by placing these sensors in different orientations [39]. Gyroscope biases are computed by averaging a static position, and accelerometer and magnetometer biases and scale factors are estimated following a similar procedure as in (2.19). Again, the main drawback of this method is that the errors are only approximations since perfect alignment with gravity and magnetic fields is not realistic.

Later in 1998, Lötters et al. [40] proposed a method to calibrate accelerometers that exploited the fact that the norm of the accelerometer should be equal to the local gravity under static conditions. In order to detect static periods in an arbitrary motion, they designed a quasi-static detector, which consists in applying a high-pass filter to the output value of the accelerometer in order to eliminate offsets, followed by a rectifier which gives the effective value, and a low pass filter that smooths the signal. The resulting output is given by:

$$V_{out} = LPF(REC(HPF(V_{in}))). \quad (2.15)$$

Whenever the voltage is under a certain threshold, the accelerometer is said to be in a static position. After collecting several quasi-static periods, a minimum variance estimator is applied to calculate the calibration parameters. The method from Lötters accounted for accelerometer calibration under random positions and was designed to gather data from motion analysis in medical applications. The results were promising, although it did not account for gyroscopes. Shin and El-Sheimy [30] used a similar approach and extended the previous model with misalignment errors. This is based on the assumption that the values of gravity sensed by each accelerometer's axis are, in sensor frame:

$$\begin{aligned} g_x &= g \cos \alpha \\ g_y &= g \cos \beta \\ g_z &= g \cos \gamma, \end{aligned} \quad (2.16)$$

where  $\alpha$ ,  $\beta$  and  $\gamma$  are the angles between the local gravity and each sensor's axis. The total magnitude sensed by the accelerometer should be equal to the gravity module:

$$g_x^2 + g_y^2 + g_z^2 = g(\cos^2 \alpha + \cos^2 \beta + \cos^2 \gamma) = \|\mathbf{g}\|. \quad (2.17)$$

The outputs of the accelerometer axes can be expressed as:

$$\begin{aligned} a_x &= b_{a,x} + (1 + S_{a,x})g_x \\ a_y &= b_{a,y} + (1 + S_{a,y})(-g_x \sin \theta_{yz} + g_y \cos \theta_{yz}) \\ a_z &= b_{a,z} + (1 + S_{a,z})(g_x \sin \theta_{zy} - g_y \sin \theta_{zx} \cos \theta_{zy} + g_z \cos \theta_{zx} \cos \theta_{zy}), \end{aligned} \quad (2.18)$$

and a similar model is used for the gyroscope:

$$\begin{aligned} \omega_x &= b_{\omega,x} + (1 + S_{\omega,x})\omega_{r,x} \\ \omega_y &= b_{\omega,y} + (1 + S_{\omega,y})(-\omega_{r,x} \sin \theta_{yz} + \omega_{r,y} \cos \theta_{yz}) \\ \omega_z &= b_{\omega,z} + (1 + S_{\omega,z})(\omega_{r,x} \sin \theta_{zy} - \omega_{r,y} \sin \theta_{zx} \cos \theta_{zy} + \omega_{r,z} \cos \theta_{zx} \cos \theta_{zy}), \end{aligned} \quad (2.19)$$

where  $\omega_{r,x,y,z}$  are the components of a reference angular rate projected to sensor axes

The main drawback of this scheme is that the gyroscope scale factors and non-orthogonalities cannot be estimated reliably as the magnitude of the earth rotation rate is very small, leading to observability problems. The same concept was exploited by Skog and Händel in 2006 [41]. They proposed an ellipsoidal fitting approach to estimate gyroscope and accelerometer biases, scale factors and non-orthogonalities. The concept was similar to the one from Shin and El-Sheimy, but they proposed a Gauss-Newton's optimization method that estimates the set of calibration parameters  $\hat{\theta}$  as follows:

$$\hat{\theta} = \arg \min_{\theta} \{L(\theta)\}, \quad (2.20)$$

$$L(\theta) = \sum_{k=0}^{M-1} (\|\mathbf{u}_k\|^2 - \|\mathbf{h}(\mathbf{y}_k, \theta)\|^2). \quad (2.21)$$

Here,  $M$  is the number of samples taken,  $\mathbf{u}_k$  is the sensor input and the sensor model is given by  $\mathbf{h}(\mathbf{y}_k, \theta) = \mathbf{MS}^{-1}(\mathbf{y}_k - \mathbf{b})$ . For gyroscopes, the model has the same shortcomings as the ones from Shin and El-Sheimy, so they proposed to use a single axis turntable to provide a reference rotation rate, which makes it useless for in-field applications. Jurman et al. [42] proposed a similar methodology that does not need a turntable for gyroscope calibration since it takes the advantage of the time invariability of the error parameters. First, gyroscope biases are removed by averaging a static period, and then the gyroscope measurements are integrated and its result in the angle domain is compared to an accurately measured known angle. The method employs a constrained Newton optimization for all the nine parameters in the sensor model and a separate procedure to calculate inter-sensor misalignment. In 2008, Fong [43] proposed a similar method, but instead of comparing the integrated angle with a known reference it uses a rotation matrix given by integrating angular velocities and compares the estimated gravity direction with the one sensed by the accelerometer under static conditions. Gyroscope biases are removed a priori, by averaging the static parts. Other more recent publications exploited the same concept as Fong did: Tedaldi, [44] introduced an Allan-Variance analysis to characterize the bias drift together with a Runge-Kutta integration method applied to gyroscope strap-down integration (SDI), and Qureshia in 2017 [45] relaxed the conditions for gyroscope's required rotations. Later, in 2018, a very similar approach was followed by Ranjbaran [46]. Frosio et al. [47] proposed a similar scheme to calibrate accelerometer under the premise that the module of the measures should always be 1  $g$ . They used a Gauss-Newton algorithm with analytical expressions for the Gaussian and Hessian matrices, and they proved an improvement compared to classic six parameters methods.

In addition to the approaches discussed above, which exploit the fact that the norm of the specific force sensed by accelerometers should be equal to local gravity, other approaches were also discussed. Li, in 2012, introduced a method that exploited the invariability of the constant product of the magnetic and gravitational fields [48]. The Magnetometer and accelerometer error parameters were adjusted in the least-squares sense. However, this method required from a heading reference, such as GPS. Forsberg et al. [49], in 2013, proposed a non-iterative method for calibrating the accelerometer under a 9-position method. However, its accuracy highly depend on supplementary equipment. This work was extended by Cai [50] including non-orthogonalities and Particle Swarm Optimization.

### On-the-run based methods

Previous methods were restrictive in terms of motions or equipment needed. All of them needed some kind of specific motion to be performed, and strict sampling procedures. An user-friendlier approach is necessary to perform robust calibration with acceptable accuracies. In 2012, Cheuk et al. [51] defined a method to calibrate a 9-DOF MARG (IMU + magnetometer), that could be applied by performing random rotations by hand. However, the method needs from static pauses in between to compute the static gravity vector given by the accelerometer and to remove gyroscope biases. The method uses a quasi-static detector technique to find static periods and compare gyroscope integrated gravity with static gravity.

Recently, in 2018, Chow et al. proposed a tightly-coupled self-calibration method for a 9-DOF MARG [25]. Similar to the one from Cheuk, but it has the advantage of performing a simultaneous calibration of 42 parameters, assuming a piece-wise magnetic field rather than a homogeneous one. The method imposed a series of conditions that are meant to happen at a certain time of the motion. Similarly, they use a quasi-static detector to find moments of rest on the motion, and then apply a set of conditions that estimate the parameters: Levelling update (i.e. inclination angles are measured by the accelerometer in a static period); zero update velocity (ZUPT), which considers that the change of velocity remains null between two quasi-static periods; and a coordinate update, which assumes an approximately change of zero the position when the IMU is rotated with no translation.

Tereshkov, in 2013, presented an interesting approach to accelerometer and gyroscope bias calibration [52]. It relied on an external input from heading (i.e. an AHRS system), and estimate the sensor's biases by projecting "virtual" forces (due to sensor drift errors) onto sensor's axes. By the help of the AHRS system, biases are estimated and compensated to estimate attitude. Although simple, the applications of this method are limited since it only considers two biases of each sensor, as the IMU always remains almost vertical, one component is non-observable. Besides, the convergence of the method is rather slow.

Spielvogel et al. [53] proposed an adaptive estimator for a sensor biases in a 9-DOF IMU. The reported observer requires smaller angular movements than the rest of state-of-the-art methods, and does not require inputs of gravity vector or local magnetic field. The method was tested through simulations with random motions, but the convergence was up to 1000 seconds in some of the cases.

Girrbach et al. [54] proposed a novel approach to sensor calibration using a GPS-aided moving horizon estimation. The proposed concept is based on an optimization-based estimator that fits the IMU trajectory to the one generated by GPS, and estimates the sensor pose, together with an unknown lever arm between the IMU and the sensor antenna. The method is complete and shows, although not very accurate, promising results, and it is a good candidate for embedded calibration algorithms if the computational load is reduced. However, it has the shortcoming of needing a GPS receiver, and the motion has to be properly defined to excite all errors properly since the IMU model used was very extensive.

### Kalman filter-based methods

More recently, several filter solutions for online calibration have been proposed which make use of the non-linear Kalman filter (KF) framework. These algorithms are less restrictive in terms of motion required due to the capabilities of the KF estimator. However a drawback from this framework is that initial estimates of the states and a very careful tuning of filter parameters (i.e. covariance matrices) are very sensitive and thus, prior to fail. Li et al. proposed a Kalman filter technique that estimates the bias and scale factor for the accelerometer and gyroscope, and introduces the concept of pseudo-measurements to remove the requirement for the quasi-static condition [55]. Pseudo-measurements consist on observations of the Kalman filter that assume zero-velocity, or zero-position variability, with a high noise variance:

$$\hat{\mathbf{r}}_k = \delta \mathbf{r}_k + \mathbf{n}_r, \quad (2.22)$$

or

$$\hat{\mathbf{v}}_k = \delta \mathbf{v}_k + \mathbf{n}_v, \quad (2.23)$$

were  $\hat{\mathbf{r}}_k$  and  $\hat{\mathbf{v}}_k$  are the position and velocity estimates, respectively;  $\delta\mathbf{r}_k$  and  $\delta\mathbf{v}_k$  are the position and velocity errors, respectively; and  $\mathbf{n}_r$  and  $\mathbf{n}_v$  the position and velocity noises. However, the algorithm requires at least a static period at the start to estimate gyroscope biases and the initial inclination angle, and requires to be rotated around the measurement vector.

Bancroft, in 2012, proposed a KF-based scheme to estimate gyroscope errors, including g-sensitivities, while the IMU is mounted on the feet of a pedestrian [56]. The method used zero velocity updates (ZUPT) and zero angular rate updates (ZARU) as measurements on the filter. However, ZARUs never occurred while testing on the walk. Moreover, the rest of parameters are not investigated in the paper. In 2014, Jørgensen et al. [57] proposed a Kalman filter-based framework to calibrate IMU in different settings: factory-based, on land and at-sea. They proposed an error state Kalman filter (ESKF) that estimates navigation parameters together with the sensor error states. However, their evaluation for the at-sea setting assumes periods of rest, since according to them, big vessels have negligible attitude motions. In addition to the Kalman filter approach, a batch/least squares method that follows a similar structure to the ones previously mentioned in this review was presented.

Some other less restrictive error parameter estimations were formulated by addition to EKF algorithms like the one from George et al. [58], which extended the INS/GNS navigation EKF framework with the accelerometer and gyroscope biases. Nevertheless, the accuracy of the estimation is not investigated in the paper. Rodrigo, in 2018, used an EKF framework made for vehicle motion estimation that estimated sensor's biases [59]. Their approach is based on the fact that under static conditions, the gyroscope biases can be removed and, if the inclination angles (i.e. pitch and roll angles) are small and the lateral and longitudinal accelerations are under a certain threshold, the vertical acceleration bias can be estimated. Then, the lateral and longitudinal biases of the accelerometer are estimated by the EKF on-the-run. The accuracy of the method is promising, although it relies on static periods to remove gyroscope biases.

A comprehensive overview of the most relevant and recent methods discussed above is shown in Table 2.2. It provides a concise and compiled glimpse of its structure, features and the restrictiveness of the methods. Restrictiveness is understood here, as the possibility of the algorithm for being carried in almost every setting, even in arbitrary motion conditions like for example, on a moving vessel at sea. Note that some multi-position-based methodologies have been omitted in the summary due to the similarities of the different approaches. The most recent works have been included here, since it is assumed that their work has been based on the assumptions and achievements of former authors, as cited in each of the correspondent papers.

## Summary

Table 2.2: Comprehensive summary of latest in-field IMU calibration techniques

Author	Sensors	Param. <sup>1</sup>	Features	Restrictive
Skog et al. [41]	A and G	$b$ , $S$ and $c$	Multi-position ellipsoidal parameter fitting. Requires a single-axis turntable	Very High
Fong et al. [43]	A and G	$b$ , $S$ and $c$	Multi-position ellipsoidal fitting. Uses strap-down integration between static periods to calibrate gyroscope	High
Cheuk et al. [51]	A, G and M	$b$ , $S$ and $c$	On-the run ellipsoidal fitting. Sensors calibrated consecutively by placing it in random positions	Medium
Li et al. [48]	A and M	$b$ , $S$ and $c$	Ellipsoidal fitting method based on the invariance of the dot-product of local gravity and magnetic field. Does not consider gyroscope.	High
Li et al. [55]	A and G	$b$ , $S$ and $c$	KF-based. Introduces pseudo-observations of velocity as updates and can be applied on any random motion. Relies on short resting periods to estimate gyroscope biases.	Low
Tereshkov et. al [52]	A and G	$b$ (partially)	Simple and intuitive method based on projecting forces. Does only consider 2 components of sensors' biases.	Low
Tedaldi et al. [44]	A and G	$b$ , $S$ and $c$	Multi-position based. Introduces static detector, Allan-Variance and Runge Kutta integration to other ellipsoidal fitting methods. <sup>2</sup>	High
Jørgensen et al. [57]	A and G	$b$ , $S$ and $c$	KF-based. Provides different scenarios for filter updates based on ZUPT and GPS-aided position. Assumptions for at-sea setting are not realistic.	Medium
Chow et al. [25]	A, G and M	$b$ , $S$ , $c$ , $a$ and $G$	Simultaneous self-optimization of 42 parameters, even in disturbed magnetic fields. Complex algorithm that relies on static periods.	Medium
Rodrigo et al. [59]	A and G	$b$	EKF framework that estimates biases on the run. Relies on static periods to remove gyroscope biases.	Medium
Girrbach et al. [54]	A and G	$b$ and $S$	GPS-aided moving-horizon optimization. Depends on GNSS receiver and accuracy is only tested on simulations.	Low
Spielvogel et al. [53]	A, G and M	$b$	Adaptive estimator for 9-DOF IMU biases. Tested only in simulations. Convergence time of up to 1000 seconds.	Very Low

<sup>1</sup>Note:  $b$  - bias;  $S$  - Scale factors;  $c$  - Axes misalignments;  $G$  - Gyro g-sensitivities;  $a$  - Cross-axis sensitivities; A - Accelerometer; G - Gyroscope; M - Magnetometer.

<sup>2</sup>This method was also extended by other authors a posteriori in [45, 46]

### 2.3.3 Discussion

An overview over a wide range of different methodologies for inertial sensor calibration have been presented in this chapter. However, this review does not exhaustively reflect every technique used by researchers and manufacturers. Factory calibration schemes that use high-end equipment have been introduced, as a preamble to in-field calibration methodologies. Factory calibration is undoubtedly effective, but it has several drawbacks as well: It is time consuming, costly, and the overall accuracy depends on the alignment accuracy of the equipment. Therefore, these schemes seem to be reserved for higher-quality IMUs, usually more than 2000 €. From the literature surveyed, some gaps and limitations can be extracted:

- The IMU calibration schemes are complicated for the end-user. Most of them require expert knowledge and precise motions; accuracy and observability of certain parameters highly depends on the motion performed.
- Kalman filter-based techniques are, although very effective, too complicated for an end-user, due to the non-intuitive tuning parameters: covariance and propagation matrices.
- Most of the in-field calibration methods rely on static or quasi-static conditions that might not be existent in many scenarios, and a false detection of them might lead to large calibration errors. For instance, most of the methods rely in some static periods to estimate gyroscope biases.
- MEMS sensors present higher run-to-run biases and thermal drifts, that cannot be calibrated in a laboratory setting. Thus, special emphasis in applicable and useful calibration techniques for different scenarios must be given. Calibration errors are dynamic, and several re-calibrations of the sensor might be needed to keep the usefulness of the INS.

In-field calibration techniques are mostly based on the observability of the local gravity vector as a reference signal to calibrate sensors. Static periods are a crucial part for all the schemes presented above. Constant biases are one of the most crucial parameters to be compensated, since its effect when integrating the signals becomes very large on a short period of time. However, none of the schemes is realistic to be applied on a moving platform, where no static periods are guaranteed. It is necessary, for the application in the SAR drone studied in this Master thesis, to develop a new method that does not consider this constraint and provides a tool to compensate the run-to-run biases on almost every condition. The description and mathematical derivation of this algorithm is provided on the next chapter.

# 3 | Methods for IMU bias estimation

Comprehensive literature overview has shown the gaps on research regarding non-restrictive methodologies for inertial measurement unit calibration. Among the IMU errors, constant biases are the most influential parameter in integration error, and also the most prone to vary with ageing and environmental conditions. Prior research relies on static periods that might be unrealistic to occur on several scenarios. In this chapter, a novel scheme to calibrate accelerometer and gyroscope biases in a non-restrictive setting is presented. The method considers a simplified IMU model that accounts for constant biases as the only source of error present. Two different optimization-based schemes are presented: one for gyroscope, and one for accelerometer (or simultaneous) bias estimation. Unlike other schemes, none of the methods assumes the observability of a signal as references, which makes it possible to be applied in scenarios with constant motion. The proposed methods leverage the fact that accelerations and velocities are bounded when biases are accurately compensated. Taking this into account, an optimization problem can be defined and solved using a Gauss-Newton approach, so that the best approximation of IMU biases can be accurately estimated.

## 3.1 Model

The mathematical model used to derive the problem is defined by two coordinate frames, one attached to the sensor, and one fixed, inertial reference frame. The sensor model defines the relation between the true and the observed quantities.

**IMU:** The body coordinate system by the IMU sensor axes is referred by body-frame and the index  $\mathcal{S}$ . It is assumed that the body frame has negligible misalignment.

**Reference frame:** The inertial reference frame  $\mathcal{R}$  is assumed to be fixed and stationary and defined by the initial arbitrary position of the IMU at the beginning of the motion. Then we assume  $\mathcal{S}_0 = \mathcal{R}$ .

**Sensor model:** A simplified sensor model is used. The model accounts for gyro and accelerometer biases as the only errors that disturb the measurements. This can be written as follows:

$${}^{\mathcal{S}}\hat{\boldsymbol{\omega}}_{t_k} = {}^{\mathcal{S}}\boldsymbol{\omega}_{t_k} + {}^{\mathcal{S}}\mathbf{b}_{\omega} + \boldsymbol{\eta}_{t_k, \omega}, \quad (3.1)$$

$${}^{\mathcal{S}}\hat{\mathbf{a}}_{t_k} = {}^{\mathcal{S}}\mathbf{a}_{t_k} + {}^{\mathcal{S}}\mathbf{b}_a + \boldsymbol{\eta}_{t_k, a}, \quad (3.2)$$

where  ${}^{\mathcal{S}}\hat{\boldsymbol{\omega}}_{t_k}$  and  ${}^{\mathcal{S}}\hat{\mathbf{a}}_{t_k}$  represent the raw measurements of the gyroscope and accelerometer, respectively at the instant of time of the  $k^{\text{th}}$  sample represented in sensor coordinates. The angular velocity and acceleration in the sensor frame  $\mathcal{S}$  are represented by  ${}^{\mathcal{S}}\boldsymbol{\omega}_{t_k}$  and  ${}^{\mathcal{S}}\mathbf{a}_{t_k}$ , respectively. Accelerometer and gyroscope constant biases are represented by  ${}^{\mathcal{S}}\mathbf{b}_a$  and  ${}^{\mathcal{S}}\mathbf{b}_{\omega}$ , respectively. Gaussian white noise processes are represented by  $\boldsymbol{\eta}_{t_k, a}$  and  $\boldsymbol{\eta}_{t_k, \omega}$ , for accelerometer and gyroscope, respectively.

## 3.2 Optimization-based methods for bias estimation

In this presented framework, and with the aim of tackling the problem of bias calibration in a non-restrictive setting, we now derive methods for bias estimation. In this section, two optimization-based methods for bias compensation are described. Gyroscope and accelerometer biases can be estimated either consecutively or simultaneously. First, a method for gyroscope bias estimation is derived. The gyroscope bias is estimated by minimizing the largest distance between any pair of samples of the integrated acceleration in the strap-down frame. Then, a second scheme to estimate accelerometer biases is presented. The accelerometer biases are estimated by minimizing the maximum distances of any pair of samples of the velocity vector over sufficiently small windows in the inertial frame. Then, improvements on computational load are presented. Finally, we present an extension of the accelerometer bias estimation method to simultaneously estimate both sensor biases, converting it into a six dimensional problem.

### 3.2.1 Quaternions and unit rotations

Prior to the derivation of the methods, we briefly review concepts and important notation aspects regarding mathematical representation of rotations in euclidean space. We make use of quaternions, as they represent a useful and comfortable alternative to Euler angles. Quaternions are a four-dimensional extension of the set of complex numbers. The most remarkable feature of the set of quaternions is that the multiplication is not commutative. The complex form of quaternions is usually written as:

$$q = a + bi + cj + dk, \quad (3.3)$$

where  $a, b, c$  and  $d$  are real numbers and  $i, j$  and  $k$  are the quaternion units.

In 3-dimensional space, according to Euler's rotation theorem, any rotation or sequence of rotations of a rigid body or coordinate system about a fixed point is equivalent to a single rotation by a given angle  $\alpha_{\text{rot}}$  about a fixed axis  $\mathbf{u}_{\text{rot}}$ , with unitary norm  $\|\mathbf{u}_{\text{rot}}\|_2 = 1$  that runs through the fixed point. This rotations can be represented by unitary quaternions in a vectorial notation such as in (3.4). Quaternions provide a more compact and stable representation of three-dimensional rotations. Compared to Euler angles, they are simpler to compose and avoid the problem of the gimbal lock.

$$\mathbf{q} = \begin{bmatrix} \cos(\frac{\alpha_{\text{rot}}}{2}) \\ \mathbf{u}_{\text{rot}} \sin(\frac{\alpha_{\text{rot}}}{2}) \end{bmatrix} \in \mathbb{R}^4, \quad \|\mathbf{q}\|_2 = 1. \quad (3.4)$$

A common notation system of leading super-scripts and sub-scripts adopted is used to denote the relative frames of orientations and vectors. A leading super-script denotes the frame being described and a leading sub-script denotes the frame this is with reference to. Thus,  ${}^{\mathcal{B}}_{\mathcal{A}}\mathbf{q}$  denotes the rotation of the frame  $\mathcal{B}$  relative to the frame  $\mathcal{A}$ . The quaternion denoted by  $(\bar{\cdot})$  is the conjugated quaternion. Then,  ${}^{\mathcal{B}}_{\mathcal{A}}\bar{\mathbf{q}}$  is the conjugate of  ${}^{\mathcal{B}}_{\mathcal{A}}\mathbf{q}$  and describes the orientation of  $\mathcal{A}$  relative to  $\mathcal{B}$ , defined by (3.5).

$${}^{\mathcal{A}}_{\mathcal{B}}\mathbf{q} = {}^{\mathcal{B}}_{\mathcal{A}}\bar{\mathbf{q}} = \begin{bmatrix} q_1 & -q_2 & -q_3 & -q_4 \end{bmatrix}^{\top}, \quad (3.5)$$

where  $q_1, q_2, q_3$  and  $q_4$  are the components of  ${}^{\mathcal{B}}_{\mathcal{A}}\mathbf{q}$ .

The quaternion multiplication, which is denoted by  $\otimes$  can be used to compound orientations. For example the orientation of a the frame  $\mathcal{C}$  relative to  $\mathcal{A}$  can be represented by the composition of the rotation between  $\mathcal{A}$  and  $\mathcal{B}$  and the rotation between  $\mathcal{B}$  and  $\mathcal{C}$  such as in (3.6).

$${}^{\mathcal{C}}_{\mathcal{A}}\mathbf{q} = {}^{\mathcal{B}}_{\mathcal{A}}\mathbf{q} \otimes {}^{\mathcal{C}}_{\mathcal{B}}\mathbf{q} \quad (3.6)$$



Let  ${}^{\mathcal{A}}\mathbf{x}$  be a three-dimensional vector expressed in the frame  $\mathcal{A}$ , and a quaternion  ${}^{\mathcal{B}}_{\mathcal{A}}\mathbf{q}$  that describes the rotation of coordinate system  $\mathcal{B}$  with respect to  $\mathcal{A}$ . Then, interpret the vector  ${}^{\mathcal{A}}\mathbf{x}$  as the last three entries of an unit quaternion, with first entry zero. The quaternion product allows to transform the vector from one reference system to another according to (3.7).

$$\begin{bmatrix} 0 \\ {}^{\mathcal{B}}\mathbf{x} \end{bmatrix} = {}^{\mathcal{A}}_{\mathcal{B}}\mathbf{q} \otimes \begin{bmatrix} 0 \\ {}^{\mathcal{A}}\mathbf{x} \end{bmatrix} \otimes {}^{\mathcal{B}}_{\mathcal{A}}\mathbf{q} = {}^{\mathcal{B}}_{\mathcal{A}}\bar{\mathbf{q}} \otimes \begin{bmatrix} 0 \\ {}^{\mathcal{A}}\mathbf{x} \end{bmatrix} \otimes {}^{\mathcal{B}}_{\mathcal{A}}\mathbf{q} \quad (3.7)$$

From now on, and for simplicity on the notation, we implicitly regard three-dimensional vectors as their corresponding pure quaternions with zero real part.

### 3.2.2 Gyroscope bias estimation

For a moving object, the specific forces on the body frame can be represented by the sum of gravity vector and a vector representing the free acceleration, which causes the changes of velocity on the object. We understand change of velocity as the time derivative of the velocity vector, which, in essence is an acceleration. If one logs data from a tri-axial accelerometer during sufficiently smooth, arbitrary motions, it can be seen that the resultant acceleration at any instant of time will be composed on its majority of the gravity vector plus a small difference, i.e. the change of velocity. In other words, the norm of the measured acceleration is the local gravity plus a small change of velocity. If one performs a more aggressive motion, then these differences will become bigger, and the bigger they get the more they shadow the gravity part of the accelerations.

Consider now a simplified case that illustrates idea behind the method. An accelerometer is measuring the data on a smooth, arbitrary motion. The sensor is moving arbitrarily without changing its orientation, i.e., its body-axes are always ideally aligned with the orientation at the initial instant of time  $t_0$ . As stated, all the measured accelerations on a certain time window would be contained on a pseudo-sphere with its diameter axes equal to the maximum change of velocity, as illustrated in Figure 3.1. If one calculates the distances between any pair of samples at any instant of time, it would never be bigger than the diameter of this sphere. In other words, the pair-wise distances of the acceleration in the strap-down frame are bounded.

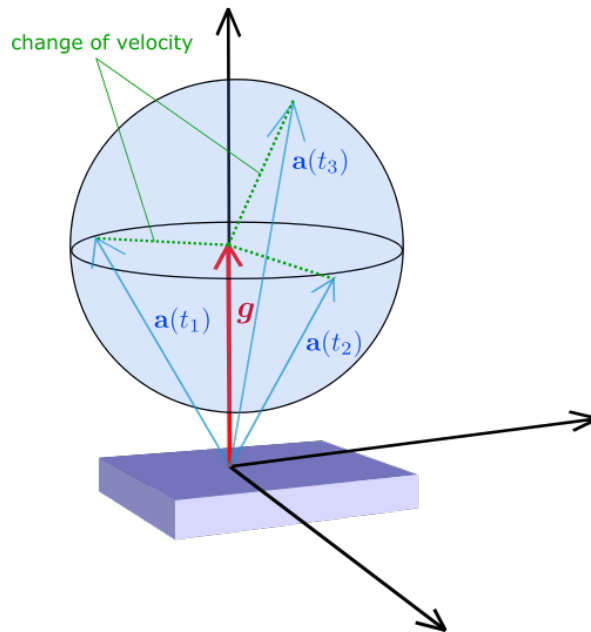


Figure 3.1: Specific forces on a moving object: change of velocity and gravity

Consider now a sensor that performs a motion that involves both accelerations and rotations between its body-frame and the inertial frame. The measured accelerations can be transformed to the inertial/strap-down frame after applying pure gyroscope strap-down integration (SDI). This is, by integrating the measurements of the gyroscope to get the absolute angle rotated by the IMU. In ideal conditions, this would always yield the above mentioned vector contained on the sphere with radius equal to the maximum change of velocity, but when gyroscope bias errors are present, this vector would rotate over the inertial frame due to the integration of a constant offset on the measurements. The sphere that contains the trajectory of the measured acceleration grows over time and, likewise, the maximum distance between any pair of samples gets bigger, due to gyroscope biases. This effect is illustrated in Figure 3.2, which shows the path over the sphere of unitary radius of the normalized acceleration after performing SDI, before and after removing gyroscope biases. At the initial time, it is aligned with the gravity vector in the Z-axis, and then it is slowly rotating due to the drift provoked by the bias offsets.

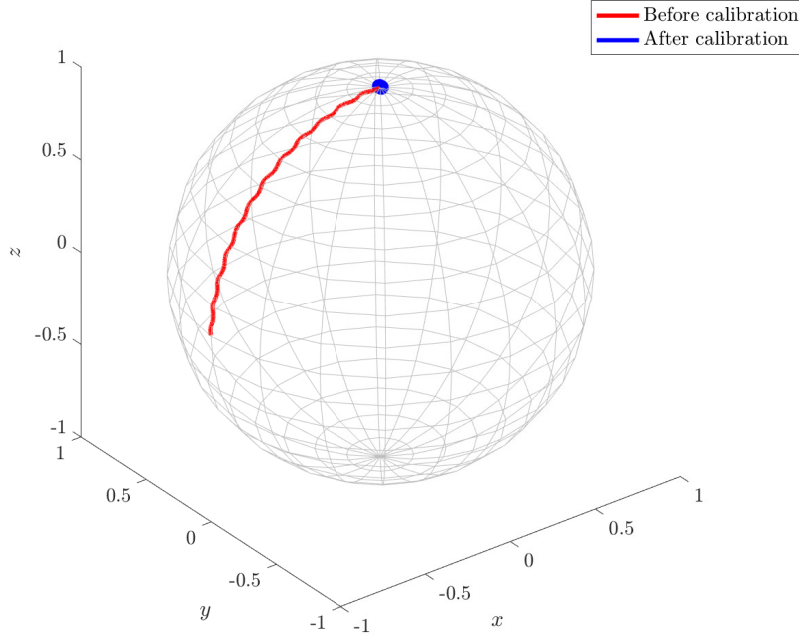


Figure 3.2: Normalized acceleration on the strap-down frame

It is important to note that other source of errors are present too, and will affect the accuracy of the SDI. Those errors can be both deterministic or indeterministic: axes misalignments, missed samples, clipping due to very fast motions, or an angle random walk (ARW) due to the integration of Gaussian Noise are some examples of this. In the following sections, the influence of the different source of errors will be illustrated in following sections.

Consider now a typical SDI problem. The integration can be performed by using the quaternion of the gyroscope measurements at every sample. The quaternion that defines the change of orientation over one sample is given by:

$$\Delta \mathbf{q}_{[t_k, t_k + T_s]} = \begin{bmatrix} \cos(\frac{\alpha_{t_k}}{2}) \\ \mathbf{u}_{t_k} \sin(\frac{\alpha_{t_k}}{2}) \end{bmatrix}, \quad (3.8)$$

where  $T_s$  is the sampling time interval,  $\mathbf{u}_{t_k}$  represents the unitary vector of the rotation axis at a

determined instant  $t_k$ , such as  $\boldsymbol{\omega}_{t_k}/\|\boldsymbol{\omega}_{t_k}\|$ , and  $\alpha_{t_k}$  represents the rotated angle at the instant  $t_k$ , which can be approximated by:

$$\alpha_{t_k} = \frac{\|\boldsymbol{\omega}_{t_k}\|}{T_s}, \quad (3.9)$$

where  $T_s$  is the sampling interval.

The orientation of the sensor  $\mathcal{S}$  at any instant of time with respect of the reference frame  $\mathcal{R} = \mathcal{S}(t_0)$  can be obtained by successively multiplying the differential quaternion at each sample, as:

$${}^{\mathcal{S}}_{\mathcal{R}}\mathbf{q}_{t_k} = \prod_{j=0}^k \Delta\mathbf{q}_{[t_j, t_j+T_s]}, \quad (3.10)$$

where we assume the  $\prod$  operator to implicitly represent the quaternion multiplication.

The measured acceleration in sensor coordinates can be transformed to the reference frame by (3.7). Thus, by using the orientation quaternion from (3.10), it yields:

$$\begin{bmatrix} 0 \\ \mathcal{R}\mathbf{a}_{t_k} \end{bmatrix} = {}^{\mathcal{S}}_{\mathcal{R}}\mathbf{q}_{t_k} \otimes \begin{bmatrix} 0 \\ \mathcal{S}\mathbf{a}_{t_k} \end{bmatrix} \otimes {}^{\mathcal{S}}_{\mathcal{R}}\bar{\mathbf{q}}_{t_k} \quad (3.11)$$

As stated, the measured acceleration in the reference frame  $\mathcal{R}$  will be composed in its majority by gravity, and will be slowly drifting due to gyroscope bias errors. The longer we perform the SDI, the longer it will drift and so the bias errors will manifest. However, the measurements will also be disturbed by high-frequency Gaussian noises as well as other faster motions that can hide the bias presence. Since the bias effect is visible in relatively long periods of time it is important to low-pass filter the accelerations in the reference frame to get rid of the other effects. Let the  $LPF()$  be the low pass filter operator, so that the filtered acceleration on the reference frame can be written like:

$$\mathcal{R}\mathbf{a}_{F,t_k} = LPF(\mathcal{R}\mathbf{a}_{t_k}). \quad (3.12)$$

There are a several ways to implement the low-pass filter. We choose the forward-backward moving average with zero-phase filter [60], since it properly smooths the data and removes the delay between the original and the filtered signals. For a moving average filter, the relation between the input,  $\mathbf{x}$ , and the output,  $\mathbf{y}$ , is given by:

$$\mathbf{y}_{t_k} = \frac{1}{N_\tau} \sum_{j=0}^{N_\tau-1} \mathbf{x}_{t_k+t_j}, \quad (3.13)$$

where  $N_\tau$  the time window size, which is function of the the time constant  $\tau$  and the sampling frequency  $F_s$  such as:

$$N_\tau = F_s \tau. \quad (3.14)$$

Without gyroscope bias errors, the distance between all samples of  $\mathcal{R}\mathbf{a}_F$  will be bounded by the maximum change of velocity on the motion, but due to bias errors this will drift over time and get bigger as time goes on. A cost function that computes the maximum of the pair-wise distances between all the samples of the recording can be defined, and the best approximation of gyroscope bias errors  ${}^{\mathcal{S}}\mathbf{b}_\omega$  is the solution of the following optimization problem:

$$\mathbf{s}_{\mathbf{b}_\omega}^* = \arg \min_{\mathbf{s}_{\mathbf{b}_\omega}} \phi(\mathbf{s}_{\mathbf{b}_\omega}), \quad \phi(\mathbf{s}_{\mathbf{b}_\omega}) = \max(\|\mathcal{R}\mathbf{a}_{F,t_j} - \mathcal{R}\mathbf{a}_{F,t_i}\|) \quad \forall t_i \neq t_j \in [t_0, T] \quad (3.15)$$

Since the size of the  $\mathbf{a}_F$  vector is assumed to be large when sampling at a sufficiently fast frequency, the computational cost of the cost function can get way too expensive. It is important for the gyroscope strap-down integration to be done at a sufficiently big sampling rate so it does not fail due to the miss of higher-frequency motions. Nevertheless, the pairwise distances matrix can get way too big when considering a long recording, therefore, down-sampling can be applied to the acceleration prior to calculate the pair-wise distances. The down-sampling of  $n$ -order can be done as simple as taking every  $n^{th}$  sample of the recording. The down-sampled acceleration can be written then as follows:

$$\mathcal{R}\mathbf{a}_{F,t'_k} = \mathcal{R}\mathbf{a}_{F,nt_k}, \quad (3.16)$$

and the number of samples of the down-sampled vector is:

$$N' = \frac{N}{n}. \quad (3.17)$$

The tuning parameters of the algorithm are, therefore, the filter time constant  $\tau$  and the down-sampling factor  $n$ .

The described method described an optimization problem to be applied *offline* over the entire recording. Alternatively, the optimization can be done in an overlapping time-windows approach. We consider the recording divided in  $N_w$  windows that overlap with the previous half, as illustrated in Figure 3.3. The window number is indicated by the  $w^{th}$  index. Let  $d_{[t_i,t_j]}^w$  be the distance between any pair of samples of the transformed acceleration of a small time window of size  $T_w$ :

$$d_{[t_i,t_j]}^w = \|\mathcal{R}\mathbf{a}_{F,t_j} - \mathcal{R}\mathbf{a}_{F,t_i}\| \quad \forall t_i, t_j \in [t_{k,w}, t_{k,w} + T_w], \quad (3.18)$$

where  $t_{k,w}$  indicates the index of the first sample of the  $w^{th}$  window. From now onwards, we use the  $\|\cdot\|$  operator to represent the 2-norm in 3-dimensional euclidean space  $\mathbb{R}^3$ , such as:

$$\|\mathbf{x}\| = \|\mathbf{x}\|_2 = \sqrt{x_1^2 + x_2^2 + x_3^2}. \quad (3.19)$$

Following the derivations from before, and according to the description made at the beginning of this section, the best approximation of gyroscope biases is the one that minimize the norm of the maximum between any pair of distances on every time-window:

$$\mathbf{s}_{\mathbf{b}_\omega}^* = \arg \min_{\mathbf{s}_{\mathbf{b}_\omega}} \phi(\mathbf{s}_{\mathbf{b}_\omega}), \quad \phi(\mathbf{s}_{\mathbf{b}_\omega}) = \sqrt{\sum_{w=0}^{N_w} \|d_{[t_i,t_j]}^w\|_\infty^2}. \quad (3.20)$$

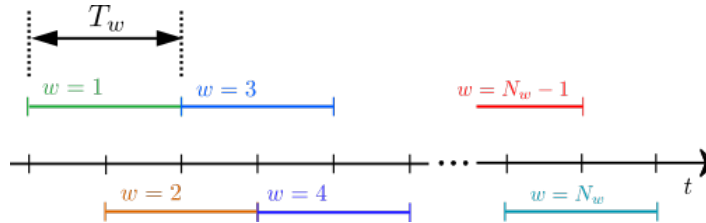


Figure 3.3: Representation of the time-windows for the estimation.

It is important to note that the cost function relies on the gyroscope strap-down integration to minimize the distance between samples. In order to be able to identify the biases on all components, the acceleration in the strap-down frame has to drift. Note that wrong biases cause the acceleration

to rotate away on the strap-down frame, and the cost function value will get higher as the distances between samples increase. However, if one axis of the sensor frame remains almost aligned with gravity at any time, that specific bias component becomes unobservable since the strap-down acceleration is rotating along its own axis and the distances between samples of time do not increase. Therefore, it is important to change the orientation of the IMU sufficient times to observe all bias components.

### 3.2.3 Accelerometer bias estimation

The accelerometer biases can be estimated once the gyroscope measurements are corrected after the calibration. Consider now the same problem as before: A sensor moving at low speeds, and performing pure gyroscope SDI. If we consider sufficiently small time-windows where the velocity remains as constant as possible, the changes of velocity (i.e. free acceleration) are usually small. Therefore, if we obtain the velocity by integrating the accelerometer measures on the reference frame, this will quickly drift and get large due to the sensor's biases. Without the presence of any sensor biases, the change of velocity would be small and close to zero, but if accelerometer bias errors exist (and assuming that gyroscope biases have been compensated in advance), velocity will quickly drift due to the integration of a constant.

Following the same procedure as for the gyroscope, the acceleration in the inertial frame is given by (3.11), and then filtered by (3.12). From now on, we assume that the gyroscope biases have been correctly estimated and compensated as stated in (3.20). If the recording is large enough and the forces involved are sufficiently small, the local gravity vector can be estimated by the average of the filtered acceleration in the strap-down frame over the whole recording, as in (3.21).

$$\hat{\mathbf{g}} = \frac{\sum_{k=0}^N \mathcal{R} \mathbf{a}_{F,t_k}}{N}, \quad (3.21)$$

and the change of velocity, is then estimated by:

$$\mathcal{R} \dot{\mathbf{v}}_{t_k} = \mathcal{R} \mathbf{a}_{F,t_k} - \hat{\mathbf{g}}. \quad (3.22)$$

The difference of velocity (or relative velocity) between a number  $N$  of samples can be defined by:

$$\mathcal{R} \mathbf{v}_{[t_0, N]} = \sum_{k=0}^N \mathcal{R} \dot{\mathbf{v}}_{t_k} T_s, \quad (3.23)$$

We consider now a sufficiently small window length  $T_w$ , where the velocity differences do not get large due to the performed motion. Then in an analogous way to the previous section, we can calculate the pair-wise distances of the velocity samples over the window, which should remain upper-bounded when no biases are present.

$$d_{[t_i, t_j]}^w = \|\mathcal{R} \mathbf{v}_{[t_k, t_i]} - \mathcal{R} \mathbf{v}_{[t_k, t_j]}\| \quad \forall t_i, t_j \in [t_{k,w}, t_{k,w} + T_w]. \quad (3.24)$$

For every window, the maximum of the distance is stored in a vector of length equal to the number of windows,  $N_w$ . Then, a cost function that computes the norm of this vector can be defined, so that the correct accelerometer biases  $^S \mathbf{b}_a$  are the solution of the following optimization problem:

$$^S \mathbf{b}_a^* = \arg \min_{^S \mathbf{b}_a} \psi(^S \mathbf{b}_a), \quad \psi(^S \mathbf{b}_a) = \sqrt{\sum_{w=0}^{N_w} \|d_{[t_i, t_j]}^w\|_\infty^2}. \quad (3.25)$$

The window length,  $T_w$ , together with the low-pass filter time constant,  $\tau$ , act as a tuning parameter of the algorithm. Note that the cost function assumes that the gyroscope measurements have been

corrected by subtracting the correct biases after calibration. If this is not the case, the cost function value will also get bigger due to the error on the strap-down integration step, see (3.10). A large error on the gyroscope bias estimation will affect the cost function and thus, the solution of the optimization problem. To avoid this, several iterations should be conducted.

### Simultaneous gyroscope and accelerometer bias estimation

The accelerometer bias estimation method presented above is based in a pure-gyroscope SDI method to transform the IMU measurements to the inertial frame. The velocity differences over small time-windows will get too large due to accelerometer biases when the gyroscope biases have been compensated, so the strap-down frame does not drift much from the reference frame  $\mathcal{R}$ . However, if gyroscope measurements are biased, the velocity over these small time windows will drift even more due since the strap-down frame will be rotating around due to gyroscope biases. Therefore, the differences of velocity will drift not only because of accelerometer biases but also due gyroscope biases. Therefore, the optimization problem defined in (3.25) can be extended to define a 6-dimensional problem which solutions are the accelerometer and gyroscope biases.

Analogously to (3.11), we integrate the raw gyroscope measurements to get the absolute orientation quaternion between the sensor frame and the reference frame at any instant of time,  ${}^{\mathcal{S}}\mathbf{q}_{t_k}(\omega_{t_k}, {}^{\mathcal{S}}\mathbf{b}_\omega)$ . Then, analogously as in section 3.2.2, we rotate the accelerometer measurements to the strap-down (reference) frame and low-pass filter, following (3.26) and (3.27).

$$\begin{bmatrix} 0 \\ \mathcal{R}\mathbf{a}_{t_k} \end{bmatrix} = {}^{\mathcal{S}}\mathbf{q}_{t_k} \otimes \begin{bmatrix} 0 \\ {}^{\mathcal{S}}\mathbf{a}_{t_k} \end{bmatrix} \otimes {}^{\mathcal{S}}\bar{\mathbf{q}}_{t_k}. \quad (3.26)$$

$$\mathcal{R}\mathbf{a}_{F,t_k} = LPF(\mathcal{R}\mathbf{a}_{t_k}). \quad (3.27)$$

Then, we integrate accelerometer measurements, using (3.22), and get the velocity in the reference, strap-down frame  $\mathcal{R}\dot{\mathbf{v}}_{t_k}({}^{\mathcal{S}}\mathbf{b}_a, {}^{\mathcal{S}}\mathbf{b}_\omega)$ .

Analogously to section 3.2.3, we can calculate the pair-wise distances of the velocity samples over small time windows  $T_w$ , which should remain upper-bounded when no biases are present.

$$d_{[t_i, t_j]}^w = \|\mathcal{R}\mathbf{v}_{[t_k, t_i]} - \mathcal{R}\mathbf{v}_{[t_k, t_j]}\| \quad \forall t_i, t_j \in [t_{k,w}, t_{k,w} + T_w]. \quad (3.28)$$

If none of the bias has been compensated, the velocity will drift and become large for two reasons: a) Due to gyroscope biases, as the strap-down frame is drifting away from the reference frame, and b) due to accelerometer biases, as the velocities over small time windows will get too large. Then, analogously to (3.25), we can derive a cost function that estimates both sensor errors, as:

$$[{}^{\mathcal{S}}\mathbf{b}_a^*, {}^{\mathcal{S}}\mathbf{b}_\omega^*] = \arg \min_{{}^{\mathcal{S}}\mathbf{b}_a, {}^{\mathcal{S}}\mathbf{b}_\omega} \psi({}^{\mathcal{S}}\mathbf{b}_a, {}^{\mathcal{S}}\mathbf{b}_\omega), \quad \psi({}^{\mathcal{S}}\mathbf{b}_a, {}^{\mathcal{S}}\mathbf{b}_\omega) = \sqrt{\sum_{w=0}^{N_w} \|d_{[t_i, t_j]}^w\|_\infty^2}. \quad (3.29)$$

Estimating the 6 parameters at once increases the calibration robustness and this way avoids the algorithm to get 'stuck' in local minimum during the estimation process, having as well an overall faster computational performance. Note that, analogously to the gyroscope optimization-based method, the convergence of simultaneous estimation depends on exciting all IMU axes so the bias components are observable. Besides, it is important that no axis of the sensor frame is aligned with gravity all the time, since the bias components of that axis would become unobservable.

### 3.2.4 Computational complexity and tuning parameters

The calibration algorithm discussed above is intended to be implemented in an *offline* setting, and requiring a considerable computational capacity and memory. Using a gradient-based optimization algorithm, the number of iterations needed for the algorithm to converge is typically 30 to 40 (gradient evaluations), which is usually more than 300 evaluations of the cost function. For a 2 minutes recording sampled at 100 Hz, the number of samples would be 12,000 and for each function evaluation this has to be integrated such as in (3.10), and then calculate pairwise distances of every time window for the accelerometer case and over the down-sampled data for the gyroscope bias estimation.

In a 2.5 GHz Dual-Core processor with a dataset of around 12,000 samples, which is 2 minutes at 100 Hz sampling frequency, the average optimization time rounds 45 seconds. A single cost function implementation in the same system is about 0.6 seconds. This procedure is computationally expensive and, if the device does not have the computational power needed, can be really time consuming. The MATLAB implementation of the code is shown in Listing 3.1. This section will discuss less computationally expensive ways to implement the method and its potential effects to the accuracy.

```

1 function x = simCostFcn(bias,fs,acc,gyr,T_filt,Tw)
2
3     gyr_bias = bias(1,:);
4     acc_bias = bias(2,:);
5
6     acc = acc - acc_bias;
7     gyr = gyr - gyr_bias;
8
9     q_gyro = strapdownIntegration(gyr, fs);
10    acc_inertial = quaternionRotate(q_gyro, acc);
11
12    %Low pass filter the acceleration
13    Nfilt = round(T_filt*fs);
14    acc_inertial_filt = filtfilt(ones(Nfilt,1)/Nfilt, 1, acc_inertial);
15    g_est_inertial = mean(acc_inertial_filt); % best guess of gravity
16
17    N = size(acc,1);
18    nw = round(Tw*fs); % samples per time window
19    delta_v = cumsum(free_acc)*1/fs;
20
21    Nw = 0;
22    x = 0;
23
24    % Overlapping windows
25    for k = 1:nw:N
26        if k + nw > N + 1
27            break
28        end
29
30        delta_v_k = delta_v(k:(k+nw-1),:);
31        x = x + max(pdist(delta_v_k'))^2;
32        Nw = Nw + 1;
33
34        if k + 1.5*nw > N + 1
35            break
36        end
37        % overlap in next window
38        delta_v_k_2 = delta_v(round(k+nw/2):round(k+3/2*nw-1),:);
39        x = x + max(pdist(delta_v_k_2'))^2;
40    end
41    x = sqrt(x);
42 end

```

Listing 3.1: MATLAB implementation

#### Substitution of the pair-wise distance

The computational cost of calculating the distances for every pair of samples at each window is quite expensive. Thus, a simpler and more computationally effective approach is to compare, component-

wise, the maximum value of the velocity with the average. The accelerometer biases will make this value to quickly drift and become very large, even in relatively small time-windows. Consider the relative velocity, as defined by (3.23), of the  $i^{\text{th}}$  sample in a time-window of length  $T_w$ . For simplicity, we can write its components:

$$\mathcal{R}_{\mathbf{v}_{[t_{k,w}, t_j]}} = \sum_{k=wT_w/T_s}^j \mathcal{R}_{\dot{\mathbf{v}}_{t_k} T_s} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \quad \forall t_j \in [t_{k,w}, t_{k,w} + T_w]. \quad (3.30)$$

Let  $\Delta \mathbf{v}_w^*$  be the velocity vector that stores the component-wise difference between the maximum and the average velocities over the time-window as in (3.31).

$$\Delta \mathbf{v}_w^* = \begin{bmatrix} \max(v_x) - \hat{v}_x & \max(v_y) - \hat{v}_y & \max(v_z) - \hat{v}_z \end{bmatrix}^T, \quad (3.31)$$

where:

$$\hat{v}_i = \frac{\sum v_i}{N_w} \quad i \in x, y, z. \quad (3.32)$$

and the cost function is then:

$$\psi(\mathbf{b}_a, \mathbf{b}_g) = \sqrt{\sum_{w=0}^{N_w} \Delta \mathbf{v}_w^{*2}} \quad (3.33)$$

The MATLAB implementation of this code is analogous to the one shown in Listing 3.1, with the substitution of the `pdist()` function with a less computationally expensive alternative. The code snippet is shown in Listing 3.2.

```

1 function x = simCostFcn(bias,fs,acc,gyr,T_filt,Tw)
2
3     gyr_bias = bias(1,:);
4     acc_bias = bias(2,:);
5
6     acc = acc - acc_bias;
7     gyr = gyr - gyr_bias;
8
9     q_gyro = strapdownIntegration(gyr, fs);
10    acc_inertial = quaternionRotate(q_gyro, acc);
11
12    %Low pass filter the acceleration
13    Nfilt = round(T_filt*fs);
14    acc_inertial_filt = filtfilt(ones(Nfilt,1)/Nfilt, 1, acc_inertial);
15    g_est_inertial = mean(acc_inertial_filt); % best guess of gravity
16
17    N = size(acc,1);
18    nw = round(Tw*fs); % samples per time window
19    delta_v = cumsum(free_acc)*1/fs;
20
21    Nw = 0;
22    x = 0;
23
24    % Overlapping windows
25    for k = 1:nw:N
26        if k + nw > N
27            break
28        end
29        delta_v_k = delta_v(k:(k+nw),:);
30        x = x + norm((max(delta_v_k)-mean(delta_v_k)))^2;
31        Nw = Nw + 1;
32
33        if k + 1.5*nw > N
34            break
35        end
36
37        delta_v_k_2 = delta_v(round(k+nw/2):round(k+3/2*nw),:); %overlap

```



```

38     x = x + norm((max(delta_v_k_2)-mean(delta_v_k_2)))^2;
39     Nw = Nw + 1;
40     end
41 x = sqrt(x);
42 end

```

Listing 3.2: Simultaneous estimation cost function without pairwise distance calculation

### Low pass filter implementation

The above presented method includes a low-pass filter implementation for both gyroscope and accelerometer calibration. The proposed filter is a zero-phase forward-backward moving average, which time constant is represented by  $\tau$ . The advantage of using this filter is that avoids phase delays and its widely available in current scientific software packages (SciPy or MATLAB). Nevertheless, a time-shift may not be an inconvenient when processing the data *offline* as the case of this calibration algorithm, so other two possibilities are discussed hereafter.

**a. Butterworth filter.** A  $n$ -order butterworth filter is defined by the following transference function:

$$H(s) = \frac{1}{\prod_{k=1}^n (s - s_k)/\omega_c}, \quad (3.34)$$

where  $\omega_c$  is the cut-off frequency and  $s_k$  the poles of the filter defined by:

$$s_k = e^{\frac{j(2k+n-1)\pi}{2n}}. \quad (3.35)$$

For a order-2 butterworth filter the cost function is 10 % faster than the zero-phase implementation. In MATLAB, the implementation of the butterworth filter substitutes de `filtfilt()` function from Listing 3.1.

```

1 [b,a] = butter(1,1/fs/T_filt);
2 acc_inertial_filt = filter(b,a, acc_inertial, acc_inertial(1,:));

```

Listing 3.3: Acceleration Butterworth filter implementation in MATLAB

**b. Exponential smoothing.** Exponential smoothing is a rule of thumb for removing high-frequency noises that uses the Poisson window function. A simple implementation of this technique on the time series  $x_t$  is given by the following expression:

$$y_t = \alpha x_t + (1 - \alpha)x_{t-1}, \quad (3.36)$$

where  $y_t$  is the smoothed signal, and  $\alpha$  is the smoothing factor, that satisfies  $0 < \alpha < 1$  and which is function of the time by:

$$\alpha = 1 - e^{-\frac{T_s}{\tau}} \approx \frac{T_s}{\tau} \quad (T_s \ll \tau) \quad (3.37)$$

```

1 function y = expSmoothing(x,tau,Fs)
2     alpha = tau/Fs;
3     y = zeros(size(x));
4     y(1,:) = x(1,:);
5     for k = 2:length(y)
6         y(k,:) = y(k-1,)*(1-alpha) + alpha*x(k,:);
7     end
8 end

```

Listing 3.4: Exponential Smoothing implementation in MATLAB

Implementing exponential smoothing on (3.12) has a increment in speed of 10 % when computing the cost function in (3.29).

The following table shows the results of testing the improvements in computational performance of every of the proposed modifications, in terms of time complexity. Discussions on results will be presented later on.

<sup>1</sup>With respect to the implementation in (3.29).

Table 3.1: Computational speed benchmarking of cost function modifications

Modification	Computational improvement <sup>1</sup>
Max-mean approach	48.72 %
Exp-smoothing	9.80%
Butterworth filter	10.12 %
Max-mean + Butterworth	53.91%
Max-mean + Exp-smoothing	53.75%

### Summary of tuning parameters

The following table summarizes the tuning parameters for both gyroscope and accelerometer methods, together with the values tested.

Table 3.2: List of tuning parameters

Parameter	Symbol	Description	Typical value
Downsampling factor	$n$	Decimation integer for the acceleration on the inertial frame on the gyroscope bias estimation	5 - 20
Filter time constant (Simultaneous) [s]	$\tau_a$	Length of the window size for the moving average filter on the accelerometer bias estimation	0-8 - 1.2
Filter time constant (Gyroscope) [s]	$\tau_g$	Length of the window size for the moving average filter on the gyroscope bias estimation	1 - 1.6
Time window length (Simultaneous) [s]	$T_w$	Length of the time windows that divide the data for the accelerometer bias estimation	10 - 20
Time window length (Gyroscope) [s]	$T_w$	Length of the time windows that divide the data for the accelerometer bias estimation	5 - 10

### 3.3 Extended Kalman filter with gyroscope bias estimation

Kalman filtering is among the most used orientation estimation techniques in sensor fusion. The nature of the filter allows to very diverse configurations regarding states and updates. In this section, two different approaches of extended Kalman filtering for orientation estimation are derived. The filter states are composed by the orientation quaternion between the reference and the sensor frames, and it is extended with gyroscope biases. The first approach is based on a common orientation estimation EKF, where the filter update is made by accelerometer measurement, that under static conditions should always measure gravity. The second approach is closer to the optimization-based methods and a simplified version from [55]. The filter states are in this case the orientation quaternion, the velocity in the reference frame and the velocity. The filter update is made by pseudo-observations of velocity: we assume that under a non-translational motion the velocity in the reference frame should always be ideally zero.

#### 3.3.1 EKF based on gravity observations

A simple orientation inclination EKF can be extended with the gyroscope biases three additional components of the state vector. Consider now a typical orientation estimation problem, where  $\mathcal{S}$  denotes the sensor frame and  $\mathcal{R}$ , a reference frame which vertical axis is aligned with gravity. Note that since yaw-observations by magnetometer or similar are not considered in this problem, we implicitly regard orientation as inclination estimation, and therefore, the heading part of the orientation might drift apart. Let  $\hat{\mathbf{x}}_k$  be the states of the filter:

$$\hat{\mathbf{x}}_{t_k} = \begin{bmatrix} {}^{\mathcal{S}}_{\mathcal{R}}\hat{\mathbf{q}}_{t_k} \\ {}^{\mathcal{S}}\hat{\mathbf{b}}_{\omega, t_k} \end{bmatrix}, \quad (3.38)$$

where  ${}^{\mathcal{S}}_{\mathcal{R}}\hat{\mathbf{q}}_{t_k}$  is the orientation quaternion mentioned above, and  ${}^{\mathcal{S}}\hat{\mathbf{b}}_{g, t_k}$  the gyroscope biases in sensor coordinates. The inputs of the filter are the gyroscope readings in the body frame,  $\mathbf{u}_{t_k} = {}^{\mathcal{S}}\boldsymbol{\omega}_{t_k}$ .

##### State update.

The state update is made by gyroscope strap-down integration, and the biases are predicted constant. Let  $\{\|\boldsymbol{\omega}_{t_k}\|T_s @ \frac{\boldsymbol{\omega}_k}{\|\boldsymbol{\omega}_{t_k}\|}\}$  be the orientation differential quaternion from gyroscope output at a time instant  $t_k$  and with a sample period  $T_s$ , the state update is given by:

$$\hat{\mathbf{x}}_{t_{k+1}} = \mathbf{f}(\hat{\mathbf{x}}_{t_k}, \boldsymbol{\omega}_{t_k}) \rightarrow \begin{cases} {}^{\mathcal{S}}_{\mathcal{R}}\hat{\mathbf{q}}_{t_{k+1}} &= {}^{\mathcal{S}}_{\mathcal{R}}\hat{\mathbf{q}}_{t_k} \otimes \{\|\boldsymbol{\omega}_{t_k} - {}^{\mathcal{S}}\hat{\mathbf{b}}_{\omega, t_k}\|T_s @ \frac{\boldsymbol{\omega}_{t_k} - {}^{\mathcal{S}}\hat{\mathbf{b}}_{\omega, t_k}}{\|\boldsymbol{\omega}_{t_k} - {}^{\mathcal{S}}\hat{\mathbf{b}}_{\omega, t_k}\|}\} + \mathbf{v}_{q, t_k}, \\ {}^{\mathcal{S}}\hat{\mathbf{b}}_{g, t_{k+1}} &= {}^{\mathcal{S}}\hat{\mathbf{b}}_{g, t_k} + \mathbf{v}_{b, t_k}, \end{cases} \quad (3.39)$$

where  $\mathbf{v}_{q, t_k}$  and  $\mathbf{v}_{b, t_k}$  are the process random variables of the orientation quaternion and the gyroscope bias, which are assumed to be Gaussian.

**Measurement:** The filter updates are made by the accelerometer measurements. Under static conditions, the measured acceleration should equal gravity, and the gravity direction by the orientation quaternion is given by:

$$\hat{\mathbf{z}}_{t_k} = \mathbf{h}(\mathbf{x}_{t_k}) = {}^{\mathcal{R}}_{\mathcal{S}}\hat{\mathbf{q}}_{t_k} \otimes \begin{bmatrix} 0 \\ \mathbf{g} \end{bmatrix} \otimes {}^{\mathcal{S}}_{\mathcal{R}}\hat{\mathbf{q}}_{t_k} + \mathbf{w}_{t_k}, \quad (3.40)$$

where  $\mathbf{w}_{t_k}$  is the measurement Gaussian noise, which is assumed to be very large, in order to account

for non-zero changes of velocity. The EKF algorithm is summarized below:

**Input** :  $\hat{\mathbf{x}}_{t_{k-1}}, \mathbf{P}_{t_{k-1}}, \mathbf{u}_{t_k}, z_{t_k}$

1.  $\hat{\mathbf{x}}_{t_k}^- = \mathbf{f}(\hat{\mathbf{x}}_{t_{k-1}}, \mathbf{u}_{t_k})$
2.  $\mathbf{P}_{t_k}^- = \mathbf{F}_{t_k} \mathbf{P}_{t_{k-1}} \mathbf{F}_{t_k}^\top + \mathbf{V}$
3.  $\mathbf{K} = \mathbf{P}_{t_k}^- \mathbf{H}_{t_k}^\top (\mathbf{H}_{t_k} \mathbf{P}_{t_k}^- \mathbf{H}_{t_k}^\top + \mathbf{W})^{-1}$
4.  $\hat{\mathbf{x}}_{t_k} = \hat{\mathbf{x}}_{t_k}^- + \mathbf{K}(z_{t_k} - \mathbf{H}_{t_k} \hat{\mathbf{x}}_{t_k}^-)$
5.  $\mathbf{P}_{t_k} = (\mathbf{I} - \mathbf{K} \mathbf{H}_{t_k}) \mathbf{P}_{t_k}^- (\mathbf{I} - \mathbf{K} \mathbf{H}_{t_k})^\top + \mathbf{K} \mathbf{W} \mathbf{K}^\top$

**Output** :  $\hat{\mathbf{x}}_{t_k}, \mathbf{P}_{t_k}$

**Algorithm 1:** Extended Kalman Filter

Here,  $\mathbf{V}$  and  $\mathbf{W}$  are the state update and measurement covariance matrices, respectively, which are assumed to remain constant;  $\mathbf{P}_{t_k}$  is the process covariance matrix; and  $\mathbf{F}_{t_k}$  and  $\mathbf{H}_{t_k}$  the Jacobian matrices which are obtained through linearization of the system as:

$$\mathbf{F}_{t_k} = \frac{\partial \mathbf{f}(\mathbf{x}_{t_k}, \mathbf{u}_{t_k})}{\partial \mathbf{x}_{t_k}}, \quad \mathbf{H}_{t_k} = \frac{\partial \mathbf{f}(\mathbf{x}_{t_k}, \mathbf{u}_{t_k})}{\partial \mathbf{x}_{t_k}}, \quad (3.41)$$

and which analytic expressions can be easily derived by using a symbolic calculation package.

**Noise covariance matrices:** The accelerometer measurements are more noisy than the ones from the gyroscope, and there is always a change of velocity, so the chosen value should be big and proportional to the norm of the change of velocity. The gyroscope bias deviation depends on the bias instability of the sensor, which is typically on the order of  $10^{-5}$  rad/s. The gyroscope noise is around  $7 \cdot 10^{-4}$  rad/s. Thus, we chose the matrices as:

$$\mathbf{W} = \sigma_v^2 a \mathbf{I}_3 \quad \mathbf{V} = \begin{bmatrix} b \cdot \mathbf{I}_4 & 0_{4,3} \\ 0_{3,4} & c \cdot \mathbf{I}_3 \end{bmatrix}, \quad (3.42)$$

where  $a, b$  and  $c$  are constants which reasonable parametrization values lay on the following grids:  $a \in [10^6, 10^9]$ ,  $b \in [0.01, 1]$ ,  $c \in [0.001, 0.1]$ . Note that this values represent only the acceptable values for which the filter correct performance was tested, and do not indicate optimization by any means.

The filter is initialized in an approximate orientation that is aligned with the reference frame, so that  ${}^{\mathcal{S}}_{\mathcal{R}}\hat{\mathbf{q}}_{t_0} = [1 \ 0 \ 0 \ 0]^\top$  and  ${}^{\mathcal{S}}\mathbf{b}_{t_0} = [0 \ 0 \ 0]^\top$ . This is not completely true for the orientation but if the IMU is placed with its face on top it represents a good approximation. The initial biases, on the other hand are arbitrary, so we consider a high variance on the bias states on the initial process covariance. Therefore, we choose:

$$\mathbf{P}_0 = \begin{bmatrix} 0.1 \cdot \mathbf{I}_4 & 0_{4,3} \\ 0_{3,4} & 100 \cdot \mathbf{I}_3 \end{bmatrix}. \quad (3.43)$$

The election of the initial process covariance is determinant for the initial convergence of the filter. Hence, a high initial covariance will make the filter to 'trust' more the measurements from the accelerometer than the state prediction. Note that since the filter does not consider a yaw measurement, such as a magnetometer, the filter will only estimate correctly the inclination part of the orientation quaternion whereas the heading might drift. The same occurs with gyroscope biases. If the sensor remains with its vertical axis approximately aligned with the gravity axis, then vertical component of the bias is not observable for the filter and cannot be estimated unless a rotation is provided.

### 3.3.2 EKF based on velocity observations

In addition to the other methods, a second approach of an EKF for gyroscope bias estimation is derived in this section. The filter consists of 10 states and the pseudo-measurements of velocity. If

the sensor only performs rotational motions with no translation, the pseudo-measurement of velocity should be zero, in addition to a large noise so that allows non-zero residual velocities. The states vector is given by:

$$\hat{\mathbf{x}}_{t_k} = \begin{bmatrix} \mathcal{S}_{\mathcal{R}} \hat{\mathbf{q}}_k \\ \mathcal{R} \hat{\mathbf{v}}_{t_k} \\ \mathcal{S} \hat{\mathbf{b}}_{g,k} \end{bmatrix}, \quad (3.44)$$

which is identical to the states from the the previous EKF, but with the addition of the velocity on the reference frame  $\mathcal{R} \hat{\mathbf{v}}_{t_k}$ .

#### State update.

The state update is made in a similar way to the previous EKF: strap-down integration of gyroscope measurements, constant biases and velocity is rotated by the orientation quaternion as in (3.7). The inputs of the system are the angular velocity and acceleration,  $\mathbf{u}_{t_k} = [\mathcal{S} \boldsymbol{\omega}_{t_k} \quad \mathcal{S} \mathbf{a}_{t_k}]^\top$ . The state update is expressed in (3.45).

$$\hat{\mathbf{x}}_{t_{k+1}} = \mathbf{f}(\hat{\mathbf{x}}_{t_k}, \mathbf{u}_{t_k}) \rightarrow \begin{cases} \mathcal{S}_{\mathcal{R}} \hat{\mathbf{q}}_{t_{k+1}} &= \mathcal{S}_{\mathcal{R}} \hat{\mathbf{q}}_{t_k} \otimes \{ \|\boldsymbol{\omega}_{t_k}\| T_s @ \frac{\boldsymbol{\omega}_{t_k}}{\|\boldsymbol{\omega}_{t_k}\|} \} + \mathbf{v}_{q,t_k}, \\ \mathcal{R} \hat{\mathbf{v}}_{t_{k+1}} &= \mathcal{R} \hat{\mathbf{v}}_{t_k} + (\mathcal{S}_{\mathcal{R}} \hat{\mathbf{q}}_{t_k} \otimes \mathcal{S} \hat{\mathbf{a}}_{t_k} \otimes \mathcal{S}_{\mathcal{R}} \hat{\mathbf{q}}_{t_k}) T_s + \mathbf{v}_{v,t_k}, \\ \mathcal{S} \hat{\mathbf{b}}_{g,t_{k+1}} &= \mathcal{S} \hat{\mathbf{b}}_{g,t_k} + \mathbf{v}_{b,t_k}, \end{cases} \quad (3.45)$$

where  $\mathbf{v}_{q,t_k}$ ,  $\mathbf{v}_{v,t_k}$  and  $\mathbf{v}_{b,t_k}$  are the process random variables of the orientation quaternion, velocity and the gyroscope bias, which are assumed to be Gaussian.

**Measurement:** The filter update is based on the pseudo-measurement of velocity which, under no translational motions, should be zero.

$$\mathbf{z}_{t_k} = \mathbf{v}_{t_k} = 0 + \mathbf{w}_{t_k}. \quad (3.46)$$

Here  $\mathbf{w}_{t_k}$  is the measurement Gaussian noise, which is assumed to be very large, in order to account for non-zero velocities. Thus, the innovation or measurement residual is given by:

$$\hat{\mathbf{y}}_{t_k} = h(\hat{\mathbf{x}}_{t_k}) - \mathbf{z}_{t_k} = \mathcal{R} \hat{\mathbf{v}}_{t_k} - 0. \quad (3.47)$$

The Jacobian matrices are obtained in the same way than in (3.41) and the filter steps are the ones expressed in Algorithm 1.

**Noise covariance matrices:** The same deductions as in the previous section apply to this 10 states filter. A large covariance that symbolizes non-zero residual velocities must be chosen. The gyroscope bias covariance should be lower than the velocity and orientation quaternion covariances, as the IMU noises are higher than the bias instability. Let  $\mathbf{V}$  and  $\mathbf{W}$  be the process and measurement covariance matrices, respectively, represented in (3.48).

$$\mathbf{W} = \sigma_v^2 I_3 \quad \mathbf{V} = \begin{bmatrix} \sigma_q^2 \cdot I_4 & 0_{4,3} & 0_{4,3} \\ 0_{3,4} & \sigma_v^2 \cdot I_3 & 0_{3,3} \\ 0_{3,4} & 0_{3,3} & \sigma_b^2 \cdot I_3 \end{bmatrix}, \quad (3.48)$$

where the reasonable parametrization values are:  $\sigma_v^2 \in [10^4, 10^8]$ ,  $\sigma_q^2 \in [0.01, 1]$ ,  $\sigma_b^2 \in [0.1, 100]$  and  $\sigma_b \in [0.001, 0.1]$ . Note that this values represent only the acceptable values for which the filter correct performance was tested, and do not indicate optimization by any means.

Following the same reasoning than in the previous section, the filter is initialized in an approximate orientation that is aligned with the reference frame, with zero velocity and a zero bias. The velocity and quaternion initial estimates are more accurate than the bias, therefore, we choose:

$$\mathbf{P}_0 = \begin{bmatrix} 0.1 \cdot I_4 & 0_{4,3} & 0_{4,3} \\ 0_{3,4} & 0.1 \cdot I_3 & 0_{3,4} \\ 0_{3,4} & 0_{3,3} & 100 \cdot I_3 \end{bmatrix}. \quad (3.49)$$

Note that, like in the first presented EKF, the pseudo-velocity based filter might also diverge in the heading component of the orientation quaternion, since no measurement corrects the prediction in that component. Likewise, if one axis of the sensor frame is aligned with gravity, the bias in this component is not observable by the filter, as the update measurement will not identify the drift on the angle of the heading component.

# 4 | On-ship calibration scenario

In the previous chapter, two novel optimization-based schemes to estimate gyroscope and accelerometer biases have been presented. These methods are powerful in the sense that they do not require any static reference nor input to estimate the parameters, which make them applicable under a wide range of settings. Additionally, two different extended Kalman filter-based approaches to estimate gyroscope biases have been introduced, which represent state-of-the-art methodologies for IMU bias compensation. In this chapter, a moving-vessel scenario is considered, in order to verify and evaluate the applicability of both the novel and the state-of-the-art methods for the Searchwing calibration use-case. First, a framework to simulate IMU measurements on a moving vessel is described by deriving the simplified equations of motion of a moving vessel and using the sensor model defined in chapter 2. Then, the methods are applied on simulated data and the influence of motion and sensor error parameters on the accuracy is investigated. Last, the performance of the methods is evaluated with real IMU data, where three different scenarios are considered.

## 4.1 Simulated-vessel IMU data generation

This section describes a framework to simulate 6-DOF IMU measurements on a boat under general conditions. The IMU is supposed to be still in 6 different orientations while standing on the moving deck of the boat. This motion pretends to simulate the user input over the IMU when collecting data for calibration. A simplified boat motion model is used in order to describe the deck's motion. We assume the boat to move only due to the impact of waves on the hull, and the waves to come from single direction, so that the boat only moves vertically (thus, in the direction of local gravity vector) and around its roll axis. This is illustrated in Figure 4.1.

At first, the equations of motion of a point on the vessel's deck are derived from the simplified ship-model. Then, using an Euler angle extrinsic representation of the IMU orientation, these equations are translated to the body frame by using quaternions, so that the sensed acceleration of the accelerometer is calculated. Likewise, the gyroscope measurements are extracted by differentiating the orientation quaternion, following the reverse procedure as the one presented in (3.10). Finally, an IMU error model is exploited in order to simulate the IMU readings under different assumptions.

The goal of the simulation-framework is double: First, to analyse potential ship-motion conditions that influence the accuracy of the bias estimation method and, secondly, to determine the influence of other deterministic and non-deterministic errors that could also have an impact on the accuracy of the method.

### 4.1.1 Motion

Consider now three different coordinate systems: The sensor frame  $\mathcal{S}$ , the vessel frame  $\mathcal{V}$ , whose vertical axis coincides with the vertical line to the vessel's deck plane, and the reference frame  $\mathcal{R}$

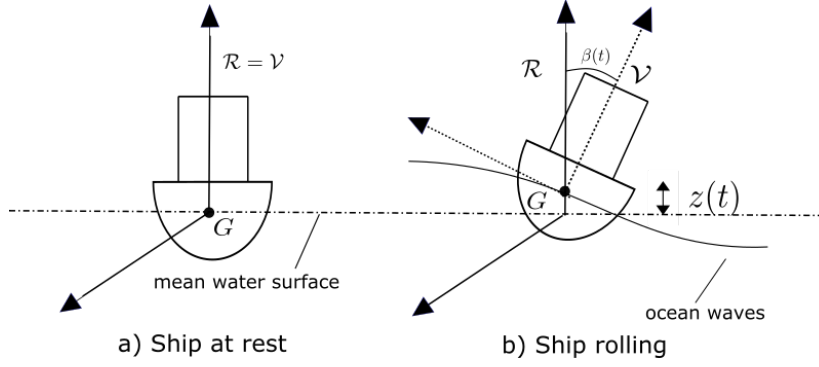


Figure 4.1: Ship in a) rest and b) rolling

whose vertical  $z$ -axis is aligned with the local gravity vector. Two coordinates describe the motion, such as  $z(t)$ , which is the elevation of the ship's deck from the mean sea level, and  $\beta(t)$ , which represents the roll angle of the ship, defining roll as the angle around the longitudinal axis. The waves are supposed to impact the hull parallel to its longitudinal axis of the ship at a period  $T_c$ . The waves are assumed to have a constant height,  $h_w$ . The elevation of the vessel's deck due to the impact of the waves can be approximated as a sinusoidal function with amplitude equal to half of the size of the waves and the same frequency such as:

$$z(t) = \frac{h_w}{2} \sin(\omega_c t) = \frac{h_w}{2} \sin(2\pi f_c t), \quad (4.1)$$

where  $f_c = 1/T_c$ . The vertical velocity of the ship can be written as:

$$v_z(t) = \dot{z}(t) = \pi f_c h_w \cos(2\pi f_c t), \quad (4.2)$$

and the vertical acceleration expressed on the reference frame as:

$${}^{\mathcal{R}}a_z(t) = \dot{v}_z(t) = -2(\pi f_c)^2 h_w \sin(2\pi f_c t). \quad (4.3)$$

The ship rolling angle is assumed to move sinusoidally together with the vertical motion, at the same frequency, and for simplicity, will be always equal or lower than a certain value, say  $\beta_{s,max}$ . Therefore, the time law of the rolling angle is:

$$\beta_s(t) = \beta_{s,max} \sin(2\pi f_c t). \quad (4.4)$$

The IMU stands in 6 different orientations, so that the Euler angles: pitch, roll and yaw; i.e.  $(\gamma, \phi, \alpha)$  are varying in the range  $[-180^\circ, 180^\circ]$ . Note that an extrinsic convention is used here, so the axes where the angles are measured remain fixed and do not move with the physical IMU body axes. The IMU remains still in each of the faces and in every side rotation a yaw rotation is performed too, so that an excitation of all axis of gyroscope is simulated properly. A summary of the value of the angles in each orientation is defined in Table 4.1.

The rotation from one orientation to another can be approximated linearly in a time,  $T_r$ , so that, for example, the pitch angle between two orientations is:

$$\gamma(t) = \frac{\gamma_2 - \gamma_1}{T_r} t. \quad (4.5)$$

The quaternion that defines the orientation between the vessel  $\mathcal{V}$  and the sensor  $\mathcal{S}$  frames at sample,  $t_k$ , is defined by the Euler angles as:

$${}^{\mathcal{S}}_{\mathcal{V}}\mathbf{q}_{t_k} = \mathbf{q}_{\alpha, t_k} \otimes \mathbf{q}_{\gamma, t_k} \otimes \mathbf{q}_{\phi, t_k}, \quad (4.6)$$



Table 4.1: Euler angles in the simulated orientations

Orientation	Pitch, $\gamma$ [deg]	Roll, $\phi$ [deg]	Yaw, $\alpha$ [deg]
#1	0	0	0
#2	90	0	90
#3	-90	0	0
#4	0	-90	0
#5	0	-90	-90
#6	0	-180	-180

where  $\mathbf{q}_\alpha$ ,  $\mathbf{q}_\gamma$  and  $\mathbf{q}_\phi$  are the quaternions representing the rotations of the yaw, pitch and roll angles, respectively, obtained by using the definition of quaternion in (3.8). Analogously, the quaternion that defines the orientation between the ship  $\mathcal{V}$  and the inertial frame,  $\mathcal{R}$ , is:

$${}^{\mathcal{V}}_R \mathbf{q}_{t_k} = \mathbf{q}_{\beta, t_k}, \quad (4.7)$$

where  $\mathbf{q}_\beta$  is the quaternion representing the roll angle of the ship. Thus, the quaternion representing the absolute orientation between the IMU,  $\mathcal{S}$ , and the reference,  $\mathcal{R}$ , frames can be defined such as:

$${}^{\mathcal{S}}_{\mathcal{R}} \mathbf{q}_{t_k} = {}^{\mathcal{S}}_{\mathcal{V}} \mathbf{q}_{t_k} \otimes {}^{\mathcal{V}}_{\mathcal{R}} \mathbf{q}_{t_k}. \quad (4.8)$$

The simulated gyro measurements, can be obtained from the absolute quaternion and the sample frequency by applying the inverse procedure described in (3.8). The change of orientation between two samples is:

$$\Delta \mathbf{q}_{t_k, t_k + T_s} = {}^{\mathcal{S}}_{\mathcal{R}} \bar{\mathbf{q}}_{t_k} \otimes {}^{\mathcal{S}}_{\mathcal{R}} \mathbf{q}_{t_k + T_s}, \quad (4.9)$$

and the rotation axis,  $\mathbf{u}_{t_k}$  and angle,  $\alpha_{t_k}$  can be calculated as in (3.8). From (3.9), the angular velocity measured in body frame can be expressed as:

$$\boldsymbol{\omega}_{t_k} = \alpha_{t_k} \cdot \mathbf{u}_{t_k} \cdot F_s. \quad (4.10)$$

The acceleration at the deck of the vessel is composed by the vertical acceleration in (4.3) and a lateral acceleration on the  $y$  axis of the ship-frame due to the rotation movement. Let  $h$  be the height of the deck from the mean sea level, and neglecting the change of this radius, the lateral acceleration can be approximated as uniform circular motion such as:

$$s_{a_y}(t) = \frac{d(h\beta_s)}{dt} = \beta_{s, max} 2\pi f_c h \cos(2\pi f_c t). \quad (4.11)$$

Assuming that the rolling angle will be always sufficiently small to apply the small angle approximation, the 3 components of the acceleration expressed on the inertial frame and in discrete time are:

$$\begin{cases} \mathcal{R} a_{x, t_k} & \approx 0, \\ \mathcal{R} a_{y, t_k} & \approx s_{a_y, t_k}, \\ \mathcal{R} a_{z, t_k} & \approx s_{a_y, t_k} \beta_{t_k} - \dot{v}_{z, t_k} - g \end{cases}. \quad (4.12)$$

This can be transformed to body frame analogously as in (3.11), applying the inverse quaternion than in (4.8),  ${}^{\mathcal{S}}_{\mathcal{R}} \bar{\mathbf{q}}_{t_k}$ , such as:

$$\begin{bmatrix} 0 \\ s\mathbf{a}_{t_k} \end{bmatrix} = {}^S_{\mathcal{R}}\bar{\mathbf{q}}_{t_k} \otimes \begin{bmatrix} 0 \\ \mathcal{R}\mathbf{a}_{t_k} \end{bmatrix} \otimes {}^S_{\mathcal{R}}\mathbf{q}_{t_k}. \quad (4.13)$$

Note that the IMU is placed on the deck with its body-axes aligned with the ship-axes initially, then the measured acceleration in body frame is opposed to the vessel (gravity is measured positive when standing still on the deck and with its frame aligned).

## 4.2 Validation on simulated data

The expressions in (4.13) and (4.10) describe the acceleration and angular velocity of the IMU on a simulated-ship motion of an IMU placed on the deck while standing in different orientations as indicated in Table 4.1. Figure 4.2 shows the generated accelerometer and gyroscope measurements. This framework is used to validate the optimization-based algorithms presented on the previous chapter. The goal of the simulations is both to test the accuracy and limitations of the algorithms, due to motion constraints or other sources of errors. The experiments showed that conducting consecutive estimation (i.e. first gyroscope bias optimization and then accelerometer bias optimization) performed with lower accuracy than following the simultaneous estimation approach presented in section 3.2.3. Several tests proved that the optimization would easily get stuck in local minima due to the low steepness of the cost function. Therefore, only the simultaneous optimization method is considered from now on.

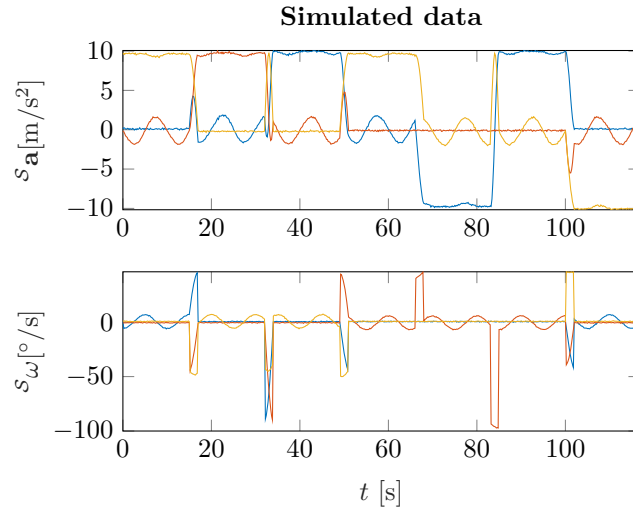


Figure 4.2: Synthetic IMU data on a vessel

Several experiments to determine the best performance of the tuning parameters of both algorithms were conducted. For each experiment synthetic data was generated by adding up only random noise to the signal and gyroscope and accelerometer biases of magnitudes of  $[0.7 - 1.5]^\circ/\text{s}$  and  $[0.1 - 0.4] \text{ m/s}^2$ . The sampling frequency used is  $F_s = 100 \text{ Hz}$ , and Gaussian noise density of  $ND_g = 0.0057^\circ/\text{s}$  and  $ND_a = 150 \mu\text{g}$  for gyroscope and accelerometer measurements, respectively, as indicated in commercially available MEMS sensors datasheet<sup>1</sup>. The length of the synthetic recordings is 12,000 samples (2 min).

A number of 25 trials for each parameter combination was conducted for each of the algorithms with time window length  $T_w \in [0.5 \text{ s}, 30 \text{ s}]$  and the moving average low-pass filter window size  $\tau \in [0.1 \text{ s}, 3.1 \text{ s}]$ , varying with intervals of 5 s and 0.5 s respectively. The discussion over these experiments is excluded of this section for brevity, but the results can be consulted in Appendix A. On simulated data, the gyroscope bias estimation algorithm performed better when using smaller window sizes

<sup>1</sup><https://www.bosch-sensortec.com/products/motion-sensors/imus/bmi055.html>

(around 5 s length) and moving average window size lower than 2 s to smooth the data. Besides, three down-sampling factors were tested,  $n = 2, 5, 10$ . A small reduction of computational time was achieved when using higher downsampling factors, at a cost of slightly reducing the accuracy of the method. On the other hand, smaller downsampling factors did not improve the accuracy significantly, so a convenient value of  $n = 5$  was chosen, representing a balance between accuracy and computational expense. Thus, the gyroscope SDI was performed at a rate of 100 Hz, whereas the distance calculations were performed at 20 Hz. Regarding simultaneous optimization, bigger window sizes presented better accuracies, with values of around 25 s, and filter window size of less than 1 s. Results are shown in Tables A.2 and A.3 of Appendix A. The error,  $\xi$  of the estimation is quantified as the norm of the difference between the ground truth bias  $\mathbf{b}_{GT}$  and the estimated value  $\hat{\mathbf{b}}$ , so that:

$$\xi = \|\hat{\mathbf{b}} - \mathbf{b}_{GT}\|, \quad (4.14)$$

for both gyroscope and accelerometer bias, with the corresponding units in each case.

### 4.2.1 Limitations

The accuracy of the method can be influenced from different factors: other sources of deterministic and non-deterministic errors, and the type of motion. The influence of the different parameters of the on-ship calibration scheme is studied by using the simulated data framework. The parameters under study include the sea motion parameters described in the previous section (i.e. the height  $h_w$  and the frequency  $f_c$  of the waves), high-frequency Gaussian noise, low frequency noises such as bias instabilities and accelerometer scale factors and misalignments. Gyroscope scaling is out of the scope of this study since its precise identification usually requires complex calibration procedures and/or highly specialized equipment, and the Searchwing sensor model does not account for these errors either.

#### Effect of motion parameters

Results on motion bias simulations appear in Table 4.2. Four different states of the sea were investigated as labelled in the first column of the table. On the calmer setting, small waves of half a meter with small period of 20 s between waves are considered, which cause a small maximum roll angle of  $5^\circ$ . The magnitudes are consecutively increasing as the waves' size and frequency become higher and faster.

Table 4.2: Average (standard deviation) bias estimation error (4.14) under different motion settings

Sea state	$T_c$ [s]	$\beta_{\max}$ [m]	$h_w$ [m]	Sim opt.		Gyro opt.	Avg. gyro
				Gyro [ $^\circ$ /s]	Acc [m/s $^2$ ]	Gyro [ $^\circ$ /s]	Gyro [ $^\circ$ /s]
Very calm	20	5	0.5	0.004 (0.002)	0.006 (0.003)	0.043 (0.015)	0.023
Calm	15	10	1	0.005 (0.003)	0.009 (0.005)	0.043 (0.026)	0.024
Rough	10	15	1.5	0.020 (0.027)	0.028 (0.022)	0.037 (0.019)	0.028
Very Rough	5	25	2.5	0.018 (0.005)	0.052 (0.013)	0.059 (0.022)	0.038
<b>Average</b>				<b>0.011</b>	<b>0.024</b>	<b>0.045</b>	<b>0.029</b>

The average errors are represented in a bar plot in Figures 4.3 and 4.4, for gyroscope and accelerometer bias, respectively. It is visible from the plot that accelerometer biases are less identifiable with motion than gyroscope biases. Accuracies of gyroscope and accelerometer bias estimation by using both methods are compared over the different settings. The results promising For the calm-sea scenario, simultaneous estimation error is less than 0.01 m/s $^2$  and 0.005  $^\circ$ /s for accelerometer and gyroscope biases, respectively; whereas using pure-gyro optimization the biases can be estimated with an accuracy of less than 0.05  $^\circ$ /s. Simultaneous estimation shows significantly better accuracy for gyroscope bias estimation than the pure-gyro method. This suggest that accelerometer biases also influence the estimation of the gyroscope biases. Overall, the accuracy of the method is promising, considering the

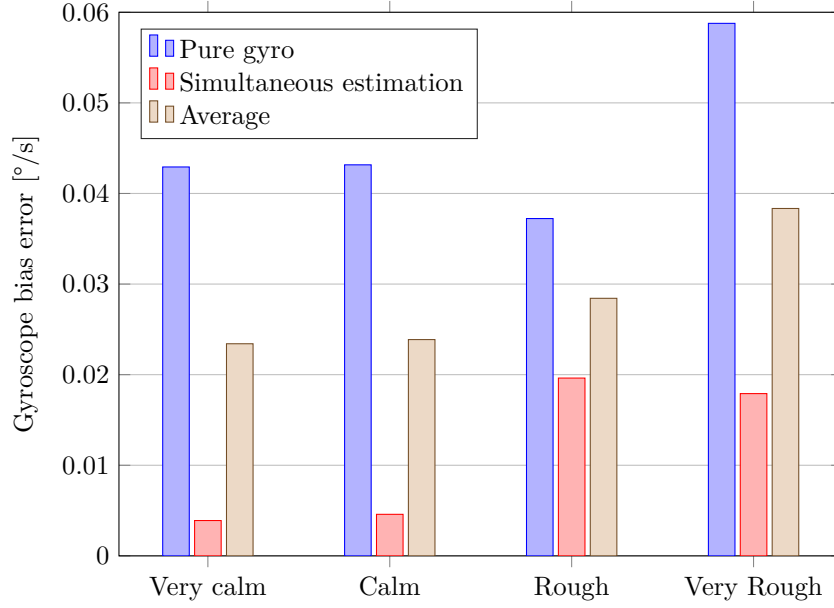


Figure 4.3: Accuracy of gyroscope bias estimation under different simulated motion parameters

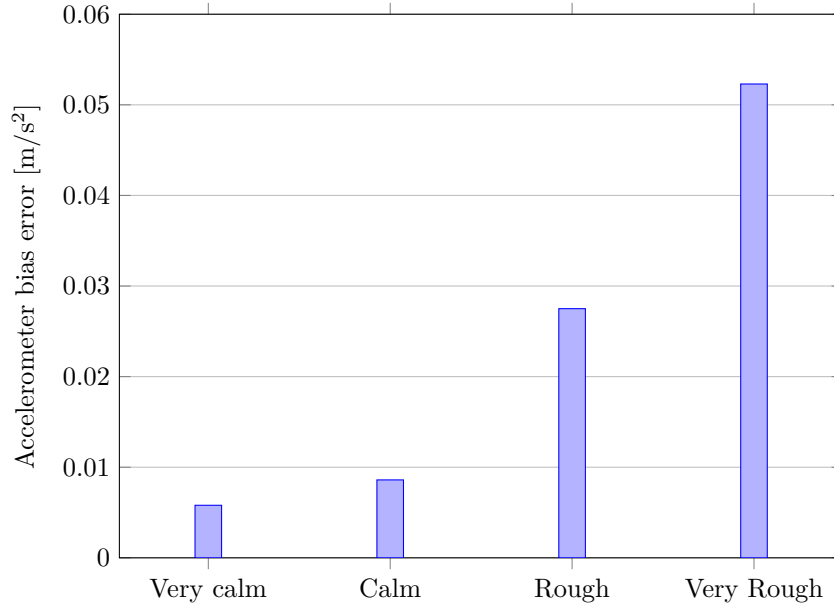


Figure 4.4: Accuracy of accelerometer bias estimation under different simulated motion parameters

large biases usually present in MEMS sensors. From the table, it can be deduced that aggressive motions are very determinant for the accuracy, and the faster the IMU moves, the less observables the biases become. Note that the tuning parameters used for these experiments were invariable and probably optimization for more aggressive motions would yield better results on the rough and very rough sea settings (for example, by using shorter time window lengths).

### Effect of scaling factors

Scaling factors and misalignments are one of the main sources of errors for accelerometers, and also easily identifiable since the gravity vector allow its observability without specialized equipment. We consider now the influence of these errors on the bias estimation problem. The same simulated setting as stated above is considered, with the addition of scaling factor  $S_i$  and cross-axis sensitivities

$c_{i,j}$ ,  $i, j \in [x, y, z]$  on accelerometer measurements. Figure 4.5 shows the influence of scaling on accelerometer measurements the sensor's biases estimation using the simultaneous optimization method.

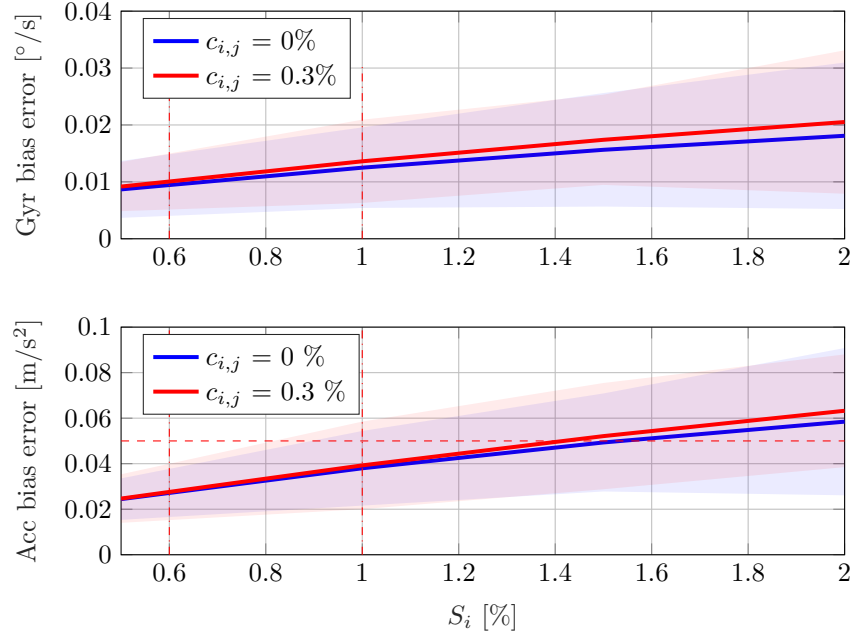


Figure 4.5: Scaling and misalignment effects on the accuracy of the simultaneous bias estimation

The figure suggest that the accuracy of both estimations highly depends on the scaling of the accelerometer measurements, with special relevance on the accelerometer bias estimation. A scaling of  $S_i = 2\%$  would produce errors of almost  $0.1 \text{ m/s}^2$ , which is unacceptable for calibration purposes. This effect becomes more evident when cross-axis sensitivities are higher. Therefore, this analysis is an indicator that scaling factors of accelerometer should be correctly identified by other state-of-the-art methods in order to obtain acceptable accuracies. Residual scaling factors up to  $1\%$  are acceptable, and realistic on MEMS-based inertial sensors.

### Effect of non-deterministic errors

Random errors are also relevant on the methods' effectiveness. If Gaussian noise is very high, it may lead to the gyroscope SDI to fail due to the big random-walk induced. Additionally, the low-frequency flicker noise on the sensor's offset (bias instability) needs to be studied. Figure 4.6 shows the effect of both Gaussian white noise and bias instability on gyroscope and accelerometer biases estimation accuracy using simultaneous optimization. The values of the errors were being successively increased, and the effect on both sensor's biases estimation accuracies was studied.

The first plot of the figure shows the influence of gyroscope Gaussian noise, which suggest that the methods accuracy and robustness decay when noise density is present. The typical noise density for a medium quality MEMS IMU is around  $0.015 \text{ }^\circ/\text{s}$ . The plot shows that the accuracy of the method is compromised only when reaching high levels of noise that are unlikely to appear in real applications. Likewise, accelerometer noise density reduces the method accuracy and robustness, although in less degree than gyroscope noise. The typical value considered is  $0.006 \text{ m/s}^2$ . In comparison with gyroscope noise, accelerometer noise influences the estimation in less degree, as when incrementing the noise density, the error band grows less than with gyroscope noise. This is due to the big influence of the strap down integration on the algorithm, since integrating pure gyroscope noisy measurements can lead to random walk. Nevertheless, typical IMU values are far from committing the method's reliability.

The 3<sup>rd</sup> plot and 14<sup>th</sup> plot of Figure 4.6 show the influence of gyroscope and accelerometer bias instability errors, respectively. Typical MEMS imu values are around  $35 \text{ }^\circ/\text{h}$  and  $250 \text{ } \mu\text{g}$  for gyroscope

and accelerometer, respectively. The results have shown that the methods' influence by these types of errors is not visible, as the error band remains approximately constant when growing from zero to values that double the typical magnitudes. Thus, it can be concluded that bias instability errors are not big limiting factors, since the standard deviation corridor is rather small as it can be seen in the plots.

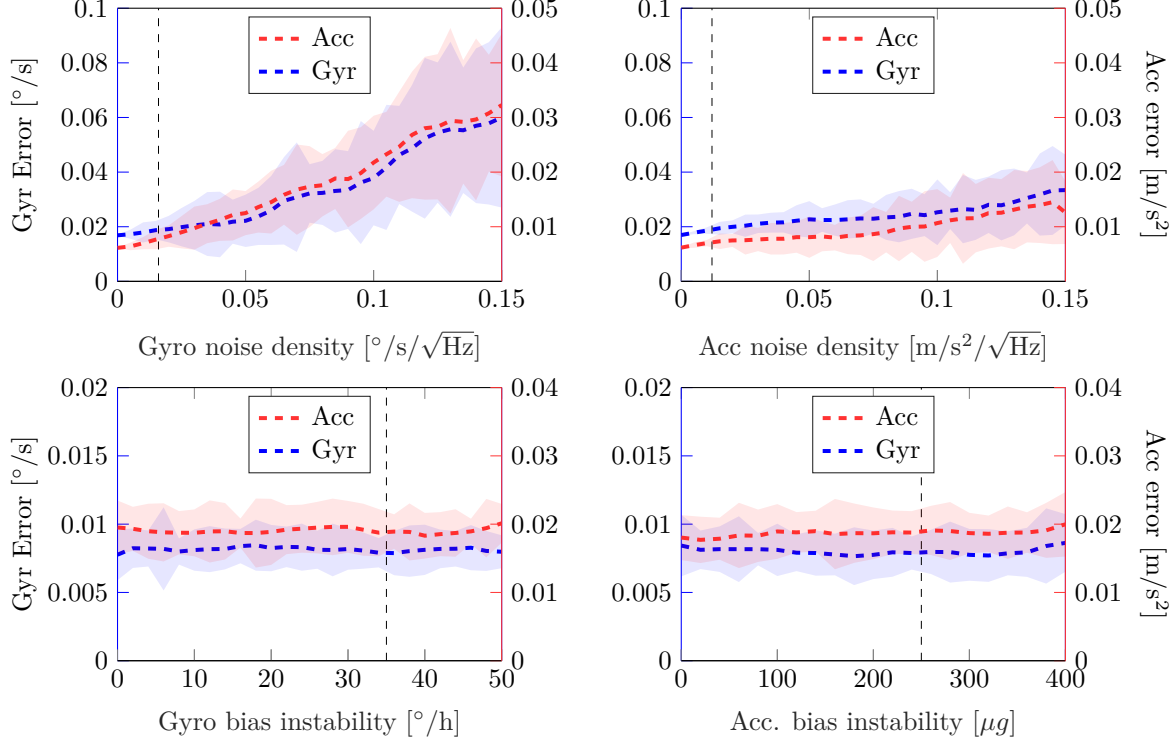


Figure 4.6: Noise effects on simultaneous accelerometer and gyroscope bias estimation. Gyroscope error is represented on the left, blue  $y$ -axis and accelerometer error on the right, red  $y$ -axis. On dashed black lines, the typical MEMS IMU values for each noise magnitude are represented.

### 4.3 Validation in ship-like simulated motion

The experiments on synthetic data have shown the accuracy and potential shortcomings of the novel optimization-based methods, regarding other limiting factors such as noises, motions or scaling. Simulated data is not very disturbed since in reality other errors intervene, such as clipping, missing samples, vibrations or irregular motions. Due to the lack of real data recorded on a moving vessel, a set of motions were recorded in order to evaluate the methods in the most realistic possible way. In this section, the results of the experiments on real IMU data are shown. The methods were tested in 3 different datasets that simulate the on-ship calibration procedure on a smartphone 6-DOF Bosch BMI055 IMU. The datasets represent an attempt to emulate the user inputs for calibration on a moving vessel, ranging from less to more challenging motions. In every of them the IMU stands for a period of approximately 20 seconds at each of its sides, following the methodology of the state-of-the-art calibration procedures. Thus, all the body axes are approximately aligned with the local gravity for some time in the measurement. Besides, an additional motion is provided to the IMU in two of the cases so that external disturbances prone to occur on a vessel at high-sea are simulated. The motion recorded in each of the cases is described below:

- **Static:** The IMU is placed in 6 different orientations, one at each face, while standing still. This dataset approximates the IMU motion on very calm waters.
- **Unimodal:** The IMU is moved vertically with an amplitude of approximately 2 meters while holding it over its 6 faces. This recording simulates the impact of the waves that makes the boat

to move along the vertical. Small lateral displacements and rotations were applied in order to simulate the worst case scenario.

- **Bimodal:** Going up- and down-stairs while holding the IMU over its 6 faces. In this motion two frequencies are mixed, the steps and the climbing period, and thus, it represents the most challenging scenario.

The datasets described motions that go from less to more challenging ways of simulating a calibration procedure on a vessel. Figure 4.7 shows the accelerometer and gyroscope outputs for each of the recordings. Short resting periods were recorded at the beginning and at the end of the recording in order to get ground truth values for gyroscope biases.

The evaluation methodology consisted in adding known artificial biases (ground truth values) and comparing the estimates with these values. First, accelerometer was calibrated in advance, following the 9-position method described by Frosio et al in [47], which accounts for a  $3 \times 3$  scaling matrix and bias offsets. Then, residual gyroscope biases were removed from the static part of the recording and the data was split, leaving only the parts with motion. Then, known random bias values were added to the recording. The magnitude of the biases considered are typical from MEMS sensors: Between  $1^\circ$  and  $2^\circ$  for gyro biases, and between  $0.2 \text{ m/s}^2$  and  $0.5 \text{ m/s}^2$  for accelerometer biases.

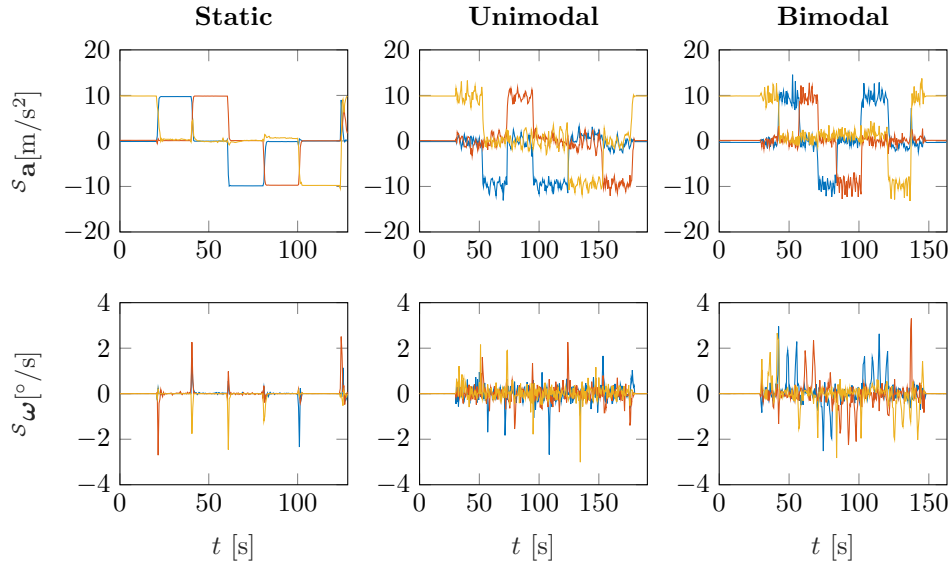


Figure 4.7: Accelerometer and gyroscope measurements in each of the motions

### 4.3.1 Gyroscope bias estimation

The proposed optimization-based method to estimate gyroscope biases was applied to the datasets described below. Using the parametrization obtained by simulations, the chosen tuning parameters for the algorithm were: A moving-average filter window of  $\tau = 1.6 \text{ s}$ , a downsampling factor of  $n = 5$  and a time-window length of  $T_w = 5 \text{ s}$ .

#### Cost function evaluation

To analyze the properties of the optimization problem, gyroscope cost function in (3.15) is evaluated for different values of the bias with  $b_{g,y}, b_{g,z} \in [-2^\circ/\text{s}, 2^\circ/\text{s}]$ . Figure 4.8 shows the cost function representation as a projection on the bias YZ plane, where the first component of the bias is fixed in the estimated value  $b_{g,x} = \hat{b}_{g,x}$ . The estimated and ground truth values of the two other components are marked as green and red dots, respectively. It can be seen that the cost function has a single minimum value and it gets flatter while getting closer to the minimum valley. This suggests that a

small tolerance in the stopping criteria of the optimization solver has to be chosen, so reaching the minimum is ensured.

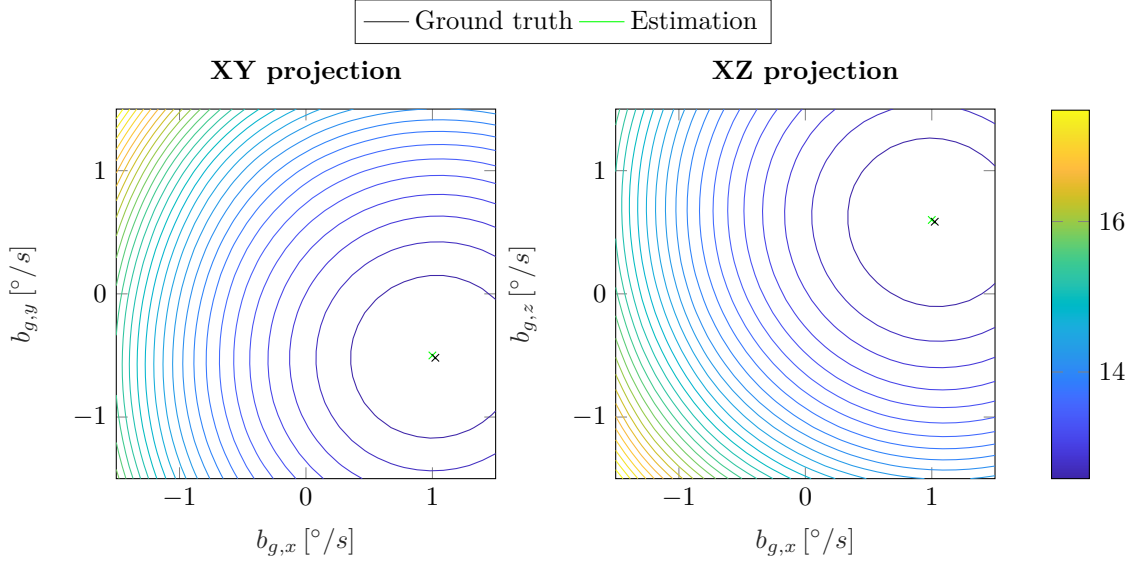


Figure 4.8: Gyro cost function projections on the  $XY$  and  $XZ$  planes with  $b_{g,z} = \hat{b}_{g,z}$  and  $b_{g,y} = \hat{b}_{g,y}$ , respectively.

### Optimization results

A total of 20 experiments were conducted in each of the recordings in order to evaluate the performance of the estimation. The optimization was conducted successively on the data in the time span  $t \in [0, t_f]$  with  $t_f$  varying from 10 s to the length of each recording, and with increments of  $T_w/2$ . The tolerance of the gradient-based solver was set as  $\epsilon = 10^{-9}$ , so that we ensure the convergence of the method. At each step of the optimization, the initial conditions of the problem were set as the ones from the estimation of the previous step. The overall accuracies and standard deviation of these experiments is shown in Table 4.3, compared to the rest of the methods. Figure 4.9 shows a single experiment where the ground truth value of the bias was set as  $\mathbf{b}_{GT} = [1^\circ/\text{s}, -0.5^\circ/\text{s}, -0.7^\circ/\text{s}]$ . The method converged to the ground truth values within 60 seconds on the static and the unimodal motions. However, it took more time to converge on the case of the bimodal data set. The accuracy of the method is around  $0.2^\circ/\text{s}$  on the unimodal motion,  $0.04^\circ/\text{s}$  on the static motion and  $0.75^\circ/\text{s}$  on the bimodal case.



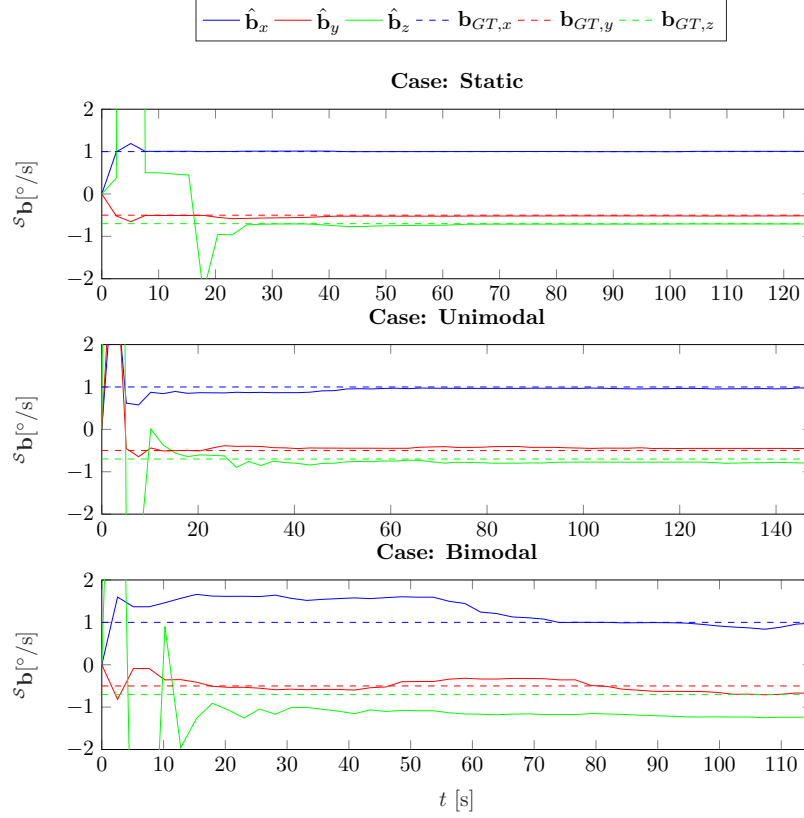


Figure 4.9: Estimated and ground truth values for gyroscope biases with the gyroscope optimization-based method on the three datasets

#### 4.3.2 Simultaneous gyroscope and accelerometer bias estimation

Analogously to the gyroscope optimization, the optimization-based simultaneous estimation in (3.33) was evaluated through 20 experiments on each of the datasets. The tuning parameters were set according to the results obtained by simulations, available in Appendix A: A moving-average filter window of  $\tau = 0.8$  s, and a time-window length of  $T_w = 15$  s. The optimization was carried out in a similar way to the pure-gyroscope optimization presented before: with  $t \in [0, t_f]$  and  $t_f = 15$  s, 20 s, 25 s... The tolerance of the gradient-based solver was set as  $\epsilon = 10^{-9}$ .

Figures 4.10 and 4.11 show the gyroscope and accelerometer bias estimation in each of the datasets, respectively. It can be seen that the gyroscope bias estimation converges in less than 40 seconds, whereas the accelerometer bias takes longer, around 70 seconds for every case. This suggests that the observability of accelerometer biases is smaller than gyroscope biases. The relative errors after 100 seconds of motion are for gyroscope and accelerometer, respectively : 2.8 % and 4.9 % in the bimodal dataset; 2.5 % and 17.6 % in the static case; and 2.7 % and 0.9 % in the unimodal data. These results also suggest that for the optimization needs a proper excitement of the velocities in order to identify accelerometer biases, since the accuracy in the static case is much lower than in the others.

Figure 4.12 shows the average accuracy of accelerometer bias estimation in the three motions on 20 trials. For unimodal and bimodal motions, the error is less than  $0.02 \text{ m/s}^2$  after 90 s of motion. The accuracy in the static case is considerably lower, which suggests that the method needs some excitement to converge.

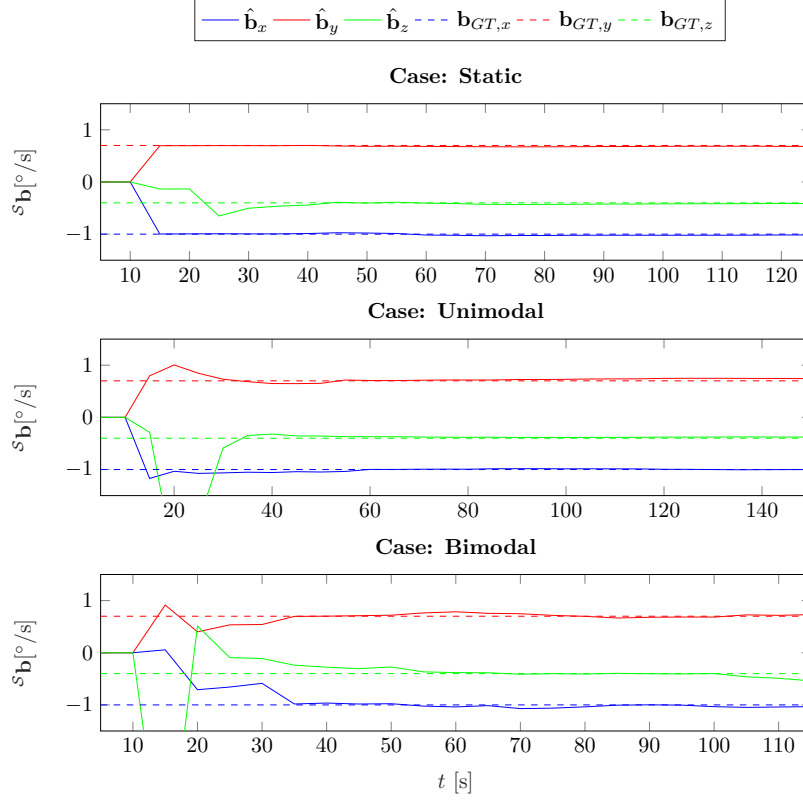


Figure 4.10: Estimated and ground truth values for gyroscope biases with the simultaneous optimization method on the three datasets

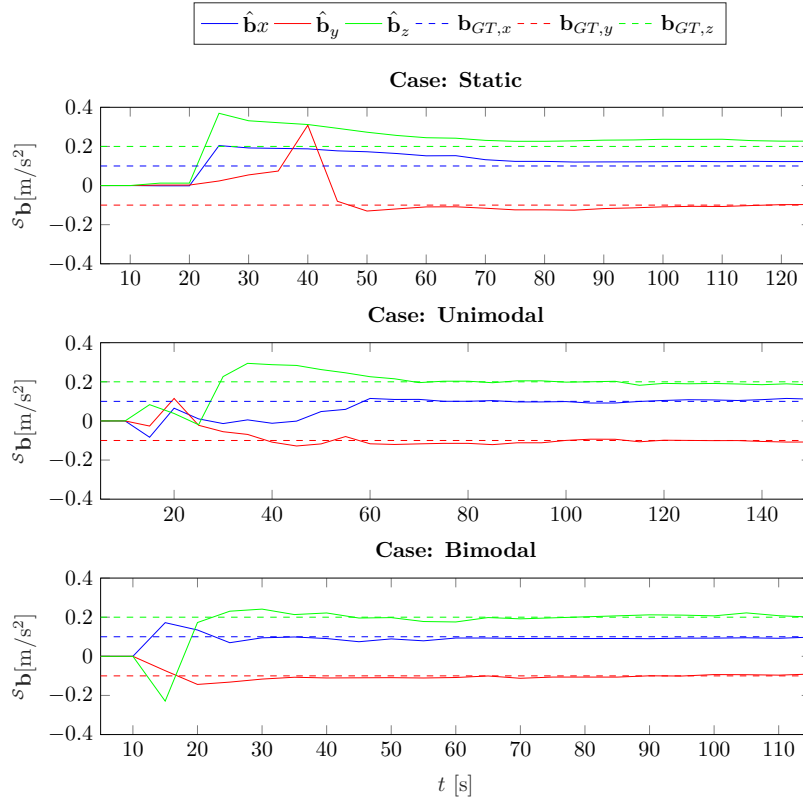


Figure 4.11: Estimated and ground truth values for accelerometer biases with the simultaneous optimization method on the three datasets

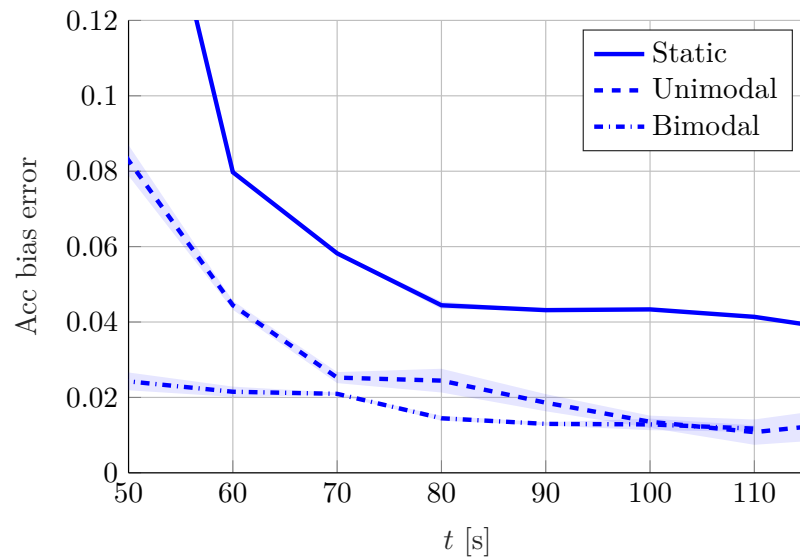


Figure 4.12: Average error and standard deviation of accelerometer bias error estimation on the three datasets (moving average of 20 seconds) on 20 experiments with randomized biases.

### 4.3.3 EKF-based gyroscope bias estimation results

Analogously to the optimization-based methods, the Kalman filter-based algorithms with bias estimations were evaluated in the above-mentioned datasets. The EKF estimates the bias for each sample of the recording. Likewise, both filters were applied in a total of 20 experiments, and the statistical results are shown in Table 4.3. The state-update covariance values defined in (3.42) and (3.48) were set as  $\sigma_q^2 = 0.1$ ,  $\sigma_b^2 = 0.01$  for both filters and  $\sigma_v^2 = 10$  for the 10-states EKF. The noise covariances were set  $\sigma_b^2 = 10^6$  and  $\sigma_w^2 = 10^8$ , for the 7-states and the 10-states EKF, respectively.

Figures 4.13 and 4.14 show the bias estimation of the gravity-based EKF and the pseudo-velocity-based EKF, respectively, on the same experiment as stated above. The gravity-based EKF showed fast convergence in all of the motions, with less than 40 seconds. The accuracy of this method was approximately 0.30 °/s on the bimodal motion, 0.04 °/s on the unimodal motion, and 0.01 °/s on the static case. The pseudo-velocity based EKF converged much slower and did not perform good in the most challenging data of the bimodal motion. Besides, this method presents a big overshoot in the first stages of the estimation. The accuracy of the 10-states EKF was around 1.00 °/s on the bimodal case, 0.17 °/s on the unimodal case, and 0.01 °/s on the static case.

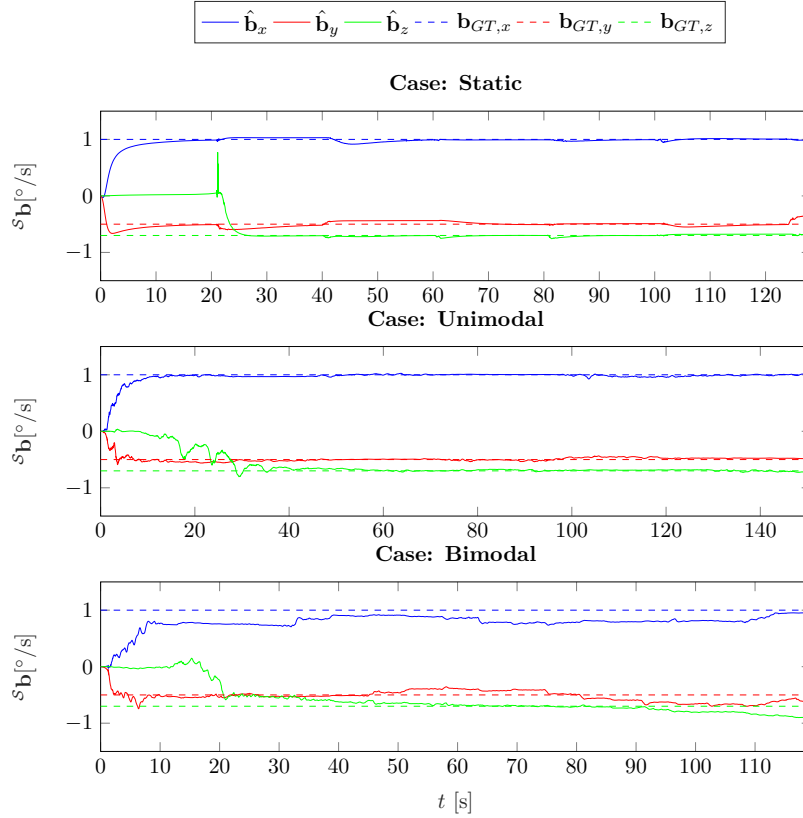


Figure 4.13: Estimated and ground truth values for gyroscope biases with the gravity measurements-based EKF on the three datasets

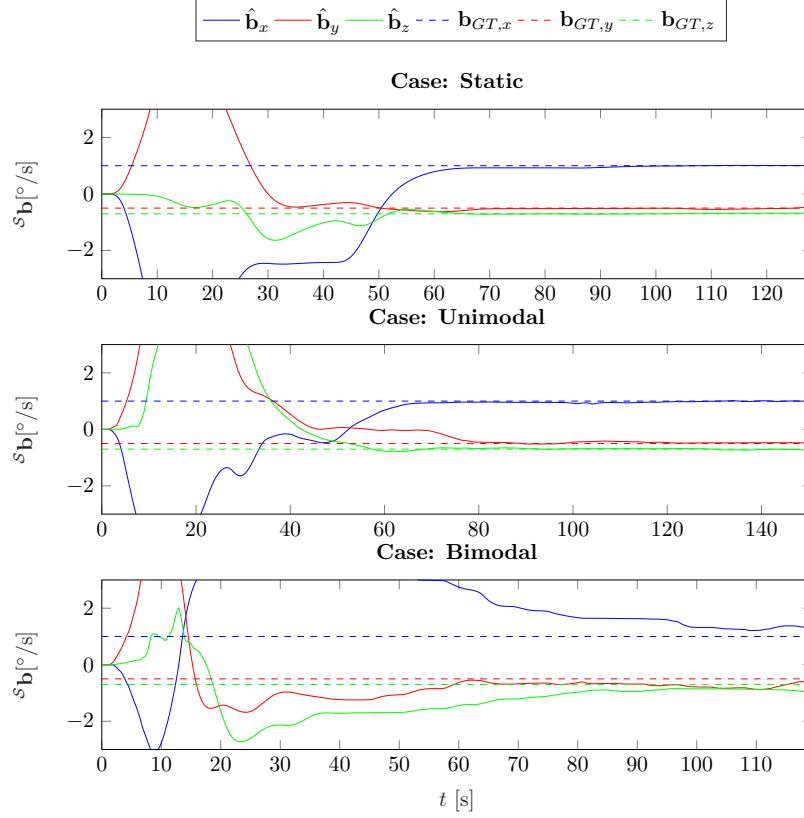


Figure 4.14: Estimated and ground truth values for gyroscope biases with the pseudo-velocity measurements-based EKF on the three datasets

#### 4.3.4 Comparison of the methods

Table 4.3 summarizes the results of the experiments with every method in the three motion cases. Figure 4.15 illustrates the results of the table in a compact bar-plot. The table shows the average error and the standard deviation of the set of the experiments performed. The color pallet indicates good or bad performance, according to the following threshold chosen values: Acceptable gyroscope values are below  $0.2^\circ/\text{s}$ . Values with errors bigger than this value are marked in dark red. The performance is considered as very good (green) when it is under  $0.1^\circ/\text{s}$ . For accelerometer, acceptable error values are below  $0.05 \text{ m/s}^2$  and are very good when they are under  $0.02 \text{ m/s}^2$ .

The gyroscope optimization-based method presents an average error of less than  $0.2^\circ/\text{s}$  in the waves case, which for a typical bias value of  $1^\circ/\text{s}$  yields a 80 % accuracy. On the static case, the accuracy for the same typical value is 97 %. However the method falls short in the most challenging motion of the bimodal motion, where the error is higher than  $0.7^\circ/\text{s}$ . On average, the simultaneous optimization method estimates gyroscope biases with an accuracy of  $0.6^\circ/\text{s}$ . Gyroscope optimization-based method average accuracy is considerably lower,  $0.35^\circ/\text{s}$ , due to the poor accuracy in the bimodal case. The overall accuracy of the 7-states EKF is  $0.14^\circ/\text{s}$ , and the accuracy of the 10-states EKF is  $0.31^\circ/\text{s}$ . Both EKF performed worse in the most challenging scenario with bimodal motion.

Simultaneous optimization-based method showed small errors in all the scenarios, with an overall accuracy of 96 % for gyroscope bias (considering a typical value of  $1^\circ/\text{s}$ ) and more than 93 % for accelerometer bias, considering a typical value of  $0.6 \text{ m/s}^2$ <sup>2</sup>. Results have also shown, that the overall accuracy is independent of the bias magnitude, when assuming MEMS typical bias values, since the standard deviation is small. The experiments showed, as well, that the method performs worse in static cases, since the accelerometer bias becomes unobservable.

<sup>2</sup>According to Pixracer IMU datasheet: <https://invensense.tdk.com/wp-content/uploads/2015/03/DS-000081-v1.01.pdf>

Both EKF-based methods have better accuracy in the static case than the two optimization-based algorithms. Nevertheless, the gyroscope bias estimations obtained by simultaneous optimization in the moving scenarios (i.e unimodal and bimodal) are more accurate than the EKF-based solutions. Pseudo-velocity based EKF have shown not to perform as accurately as the rest of methods, since it takes more time to converge, the accuracy is significantly lower for the motion cases and it has a big overshoot at the start. The gravity-based 7-states EKF presents an average error of approximately  $0.04^\circ/\text{s}$  in the unimodal case after 50 s. This suggest that the EKF convergence is faster than the optimization-based methods, as shown also in the plots of Figure 4.14.

The robustness (understood here as the repeatability of the results) of the methods can be evaluated by observing the standard deviation of the results. In general terms, it can be seen that simultaneous optimization is more robust than the rest of methods. On the other hand, the gravity-based EKF presented smaller standard deviation on the experiments than the pure gyroscope optimization-based method.

Figure 4.16 shows the accuracy of the methods over time. The 7-EKF states is considerably faster than the optimization-based methods. In less than 30 s, the EKF present an error of less than  $0.3^\circ/\text{s}$  for all motions. Note that the 10-states EKF has been discarded from the plot as its convergence time is much slower than the rest of the methods.

Table 4.4 and Figure 4.17 show the convergence time to acceptable values for gyroscope bias estimation. Note that the 7-states EKF is considerably faster than the optimization-based methods. Pure gyroscope optimization is almost as fast as the EKF in the static case, but its speed is reduced considerably in the two other cases. The 10-states EKF performs much slower than the rest of methods.

Table 4.3: Average (standard deviation) bias estimation error (4.14) of all the methods under different motion settings after a certain time  $t$  in 20 experiments with randomized biases between  $0.8^\circ/\text{s}$  and  $1.5^\circ/\text{s}$ .

$t$ [s]	Case	Simultaneous optimization		Gyro optimization	EKF-10 states	EKF-7 states	Avg . gyro
		Gyro $b_g$ [ $^\circ/\text{s}$ ]	Acc $b_a$ [ $\text{m}/\text{s}^2$ ]	Gyro $b_g$ [ $^\circ/\text{s}$ ]	Gyro $b_g$ [ $^\circ/\text{s}$ ]	Gyro $b_g$ [ $^\circ/\text{s}$ ]	Gyro $b_g$ [ $^\circ/\text{s}$ ]
60	Bimodal	0.11 (0.00)	0.03 (<0.01)	<b>0.81</b> (<0.01)	<b>3.23</b> (1.22)	0.17 (0.06)	<b>1.08</b>
60	Unimodal	<b>0.04</b> (<0.01)	0.03 (<0.01)	0.19 (0.08)	<b>1.31</b> (1.23)	<b>0.01</b> (<0.01)	<b>0.39</b>
60	Static	<b>0.02</b> (<0.01)	<b>0.08</b> (<0.01)	<b>0.06</b> (<0.01)	<b>0.34</b> (0.15)	<b>0.06</b> (<0.01)	<b>0.12</b>
60	All	<b>0.06</b>	<b>0.05</b>	<b>0.35</b>	<b>1.63</b>	<b>0.08</b>	<b>0.53</b>
110	Bimodal	<b>0.10</b> (<0.01)	<b>&lt;0.01</b> (<0.01)	<b>0.82</b> (<0.01)	<b>0.78</b> (0.40)	<b>0.31</b> (0.04)	<b>0.50</b>
110	Unimodal	<b>0.04</b> (<0.01)	<b>&lt;0.01</b> (<0.01)	0.20 (0.06)	<b>0.09</b> (0.03)	<b>0.06</b> (<0.01)	<b>0.10</b>
110	Static	<b>0.03</b> (<0.01)	0.04 (<0.01)	<b>0.05</b> (<0.01)	<b>0.05</b> (<0.01)	<b>0.05</b> (<0.01)	<b>0.04</b>
110	All	<b>0.06</b>	<b>0.02</b>	<b>0.35</b>	<b>0.31</b>	<b>0.14</b>	<b>0.39</b>

Table 4.4: Convergence time [s] to acceptable values of all methods for the three datasets

Case	Sim. Opt. (Acc)	Sim. Opt. (Gyro)	Gyro Opt.	EKF-7 states	EKF-10 states
Static	40	31	27	23	65
Unimodal	60	40	71	22	85
Bimodal	80	40	-	47	-

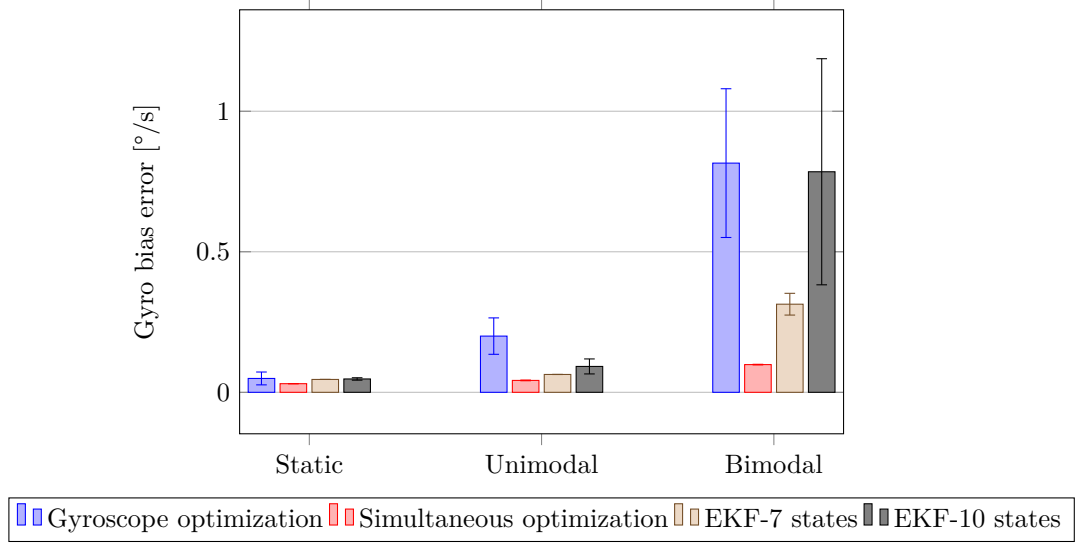


Figure 4.15: Average error and standard deviation of gyroscope bias error (4.14) estimation for all methods on the three datasets after  $t = 110$  s.

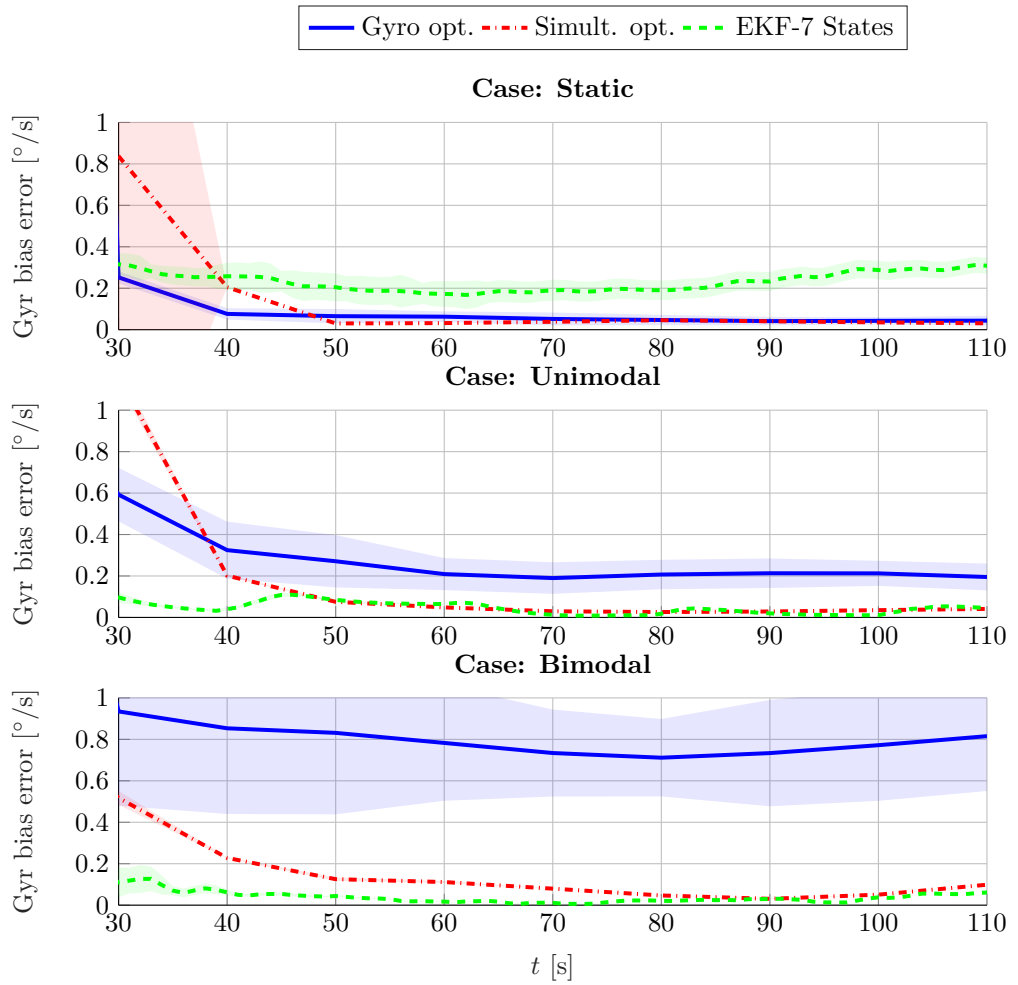


Figure 4.16: Average and standard deviation (shadowed area) of gyroscope bias estimation error (4.14) for the optimization based methods and the 7-states EKF over time.

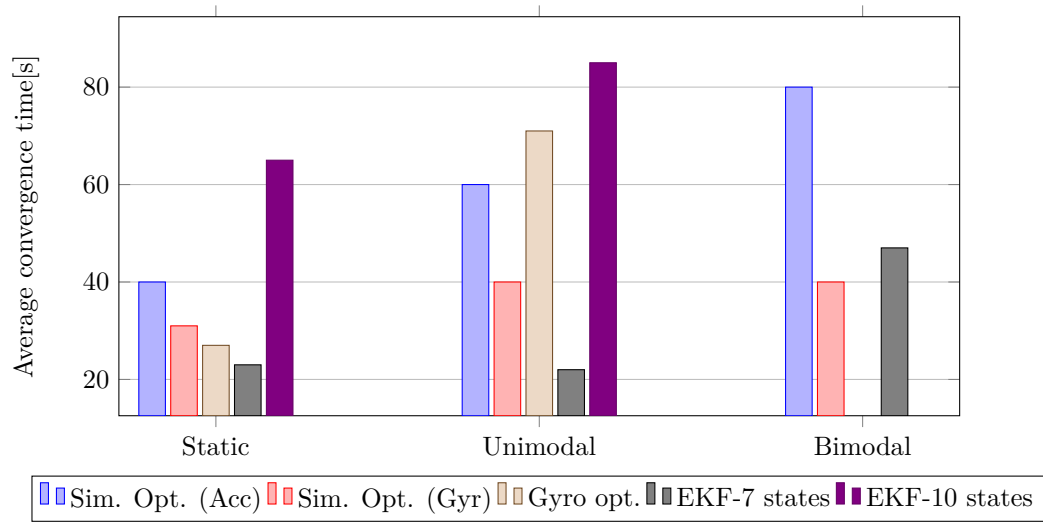


Figure 4.17: Average convergence time to acceptable values



# 5 | On-the-run bias compensation

The previous chapter studied the performance of the proposed optimization-based methods for gyroscope and accelerometer bias estimation on the specific scenario of the Searchwing drone calibration on a boat. The performance of the algorithms was compared with two state-of-the-art Kalman filter approaches. The results of the on-ship calibration scenario suggest the applicability of the optimization-based methods within other kinds of applications, in arbitrary motions. This chapter provides a second scenario for the validation of presented optimization-based methods, and it is supported by the application of an orientation estimation filter. The optimization-based and extended Kalman filter methods are assessed in a set of diverse experiments with disturbed and undisturbed IMU measurements. The methods are evaluated regarding the accuracy of gyroscope bias estimation, and orientation estimation. The chapter is structured as follows: First, a dataset comprising a series of diverse measurements with different motions is presented. In the data, several types of patterns at different speeds were recorded, including rotational motions, translational or arbitrary ones. Secondly, the performance of the optimization-based and the EKF-based methods for gyroscope bias estimation is evaluated in the measurements. Accelerometer bias estimation is not considered for these data-sets due to the lack of reliable accelerometer calibration data to compare as a baseline. Finally, we discuss the benefits of using the optimization-based methods for bias estimation in an online scenario, as an extension to an orientation estimation filter. We evaluate the accuracy of the orientation estimation provided by the filter and the optimization-based methods in all motions, and compare with the EKF.

## 5.1 Data-set description

For the purpose of evaluating the applicability of the method in non-restrictive contexts, the data-set recorded by Brands in [61] are chosen. The data-set includes IMU (including magnetometer) and optical tracking device measurements in an office environment, with several patterns of motion. The datasets include translational, rotational and arbitrary motions recorded at three different speeds. The following short-hands are used to represent with brevity each of the measurements:

- **Linear:** Moving the IMU in approximately translational movements, without big rotations. In these motions one axis remains approximately aligned with the gravity vector.
- **Rotational:** Rotating the IMU approximately in the same position.
- **Arbitrarily:** Moving the IMU arbitrarily, combining both sequences presented above.

Three different speeds are considered: Fast, slow and medium motions. Besides, the following disturbances are considered in the data-set:

- **Disturbance vibration:** Generated with a smartphone placed on top of the IMU with a vibration app. Magnetometer measures are disturbed too due to the phone's magnetic field, but it is out of the scope of this analysis. It is labelled in the data by the "-v"
- **Rotational clipping:** Caused by rotating the IMU faster than the maximum range of the gyroscope. It is labelled by "-rc".
- **Disturbance shock:** Generated by repetitively tap with the fingers on the IMU during the motion. It is labelled with the suffix "-s".
- **Magnetic disturbances:** Several kinds of magnetic disturbances are applied to the data. Magnetic disturbances are labelled in the data by "-md", "-mo" and "-ms". The magnetic disturbances are followed by  $N1 \times N2$  which indicate the number of phases where the data is disturbed, and the total time of disturbances. Nevertheless, since magnetometer data is not used for the considered use-case, these parameters are not relevant and are indicated here just for information purposes.

The number of rest phases applied in each of the recordings is denoted by " $bn$ ", where  $n$  is the number of phases. For example the name "rotational\_medium\_s" indicates a rotational motion at medium speed, with a rest phase at the beginning ("b0") and applied disturbance shock. For simplicity and algorithm's evaluation purposes, we regard only those measurements where there was only one rest at the beginning ("b0"). Figure 5.1 shows three different pairs of gyroscope and accelerometer measurement extracted from the dataset.

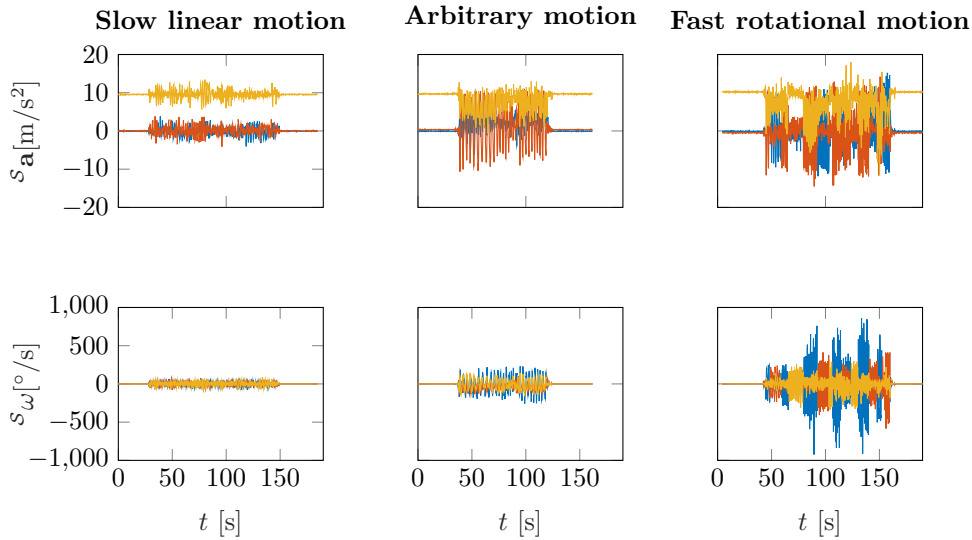


Figure 5.1: Accelerometer and gyro measurements on three recordings of the dataset

Table 5.5 lists all the experiments that have been analysed in this chapter. The norm of the average angular rate and the average change of velocity for each recording is indicated in the table, for each of the recordings. These values are indicators of the how challenging some of the motions are. For fast recordings, an average change of velocity of more than  $4 \text{ m/s}^2$  is applied. Also, in fast rotational files, average rate is more than  $200 \text{ }^\circ/\text{s}$ . The file "all\_movements" indicates a combination of all the motions described above. Note also, that files where disturbance shock was applied have a greater change of velocity due to the effect of tapping the sensor in the accelerations.

Table 5.1: List of measurements of the dataset and averages of gyroscope rate and change of velocity

Experiment	Avg. ang. rate [ $^{\circ}/s$ ]	Avg. change of vel. [ $m/s^2$ ]
1_rotational_slow_b0	54	0.239
1_rotational_fast_b0	153	0.512
1_linear_fast_b0	76	3.464
1_linear_slow_b0	27	0.515
2_rotational_medium_b0_v	103	0.498
2_linear_medium_b0	18	0.045
2_linear_medium_b0_s	26	0.262
2_rotational_medium_b0_s	94	0.184
2_linear_medium_b0_v	21	0.361
2_rotational_medium_b0	66	0.007
3_linear_fast_b0	90	4.384
3_arbitrarily_ms_2x_30	182	0.827
3_arbitrarily_ms_6x_10	219	1.327
3_rotational_fast_b0	248	0.271
3_all_movements	89	1.281
3_rotational_fast_b0_rc	234	0.29
3_rotational_slow_b0	49	0.144
3_linear_slow_b0	22	0.187
3_arbitrarily	240	1.859
4_arbitrarily_mo_2x60	92	0.111
4_arbitrarily_md_neo_1x60_4cm	77	0.018
4_arbitrarily_240s_slow	73	0.001
4_arbitrarily_360s_slow	70	0.016
4_arbitrarily_mo_2x30_2	90	0.07
4_arbitrarily_md_neo_1x60_1cm	63	0.003
4_arbitrarily_md_neo_1x60_5cm	109	0.101
4_arbitrarily_360s_medium	207	0.808
4_arbitrarily_240s_medium	234	0.79
4_arbitrarily_md_neo_1x60_3cm	79	0.024
4_arbitrarily_md_neo_1x60_2cm	69	0.006

## 5.2 Gyroscope bias estimation

In this section, the performance of the gyroscope bias estimation on the different types of motion considered in the data-set is evaluated. The two novel optimization-based methods are compared, together with the gravity-based extended Kalman filter as a state-of-the-art method. The pseudo-velocity based, 10-states EKF have been discarded in this analysis due to the poor performance on most of the measurements. The analysis procedure is similar to the followed in the on-ship calibration scenario: the residual gyroscope biases are removed by averaging the static parts of the recordings and known, random, biases between  $0.8^{\circ}/s$  and  $1.5^{\circ}/s$  are added to the gyroscope measurements. A number of 40 experiments for each file were performed. In the simultaneous estimation, the accuracy of accelerometer bias estimation is not evaluated due to the lack of reliable accelerometer calibration parameters. However, the estimated biases were rather small for all the experiments. The analysis is structured by the three different types of motion considered: linear, rotational and arbitrary motions.

### 5.2.1 Linear motion

As described below, linear motions consisted in moving the IMU with mostly translational, arbitrary, movements without applying big orientation changes. Therefore, one of the axes of the IMU remained vertical almost the whole motion (see Figure 5.1) and thus, the bias of that component is not observable by any of the methods. Thus, we evaluate the accuracy of the method with the two observable gyroscope bias components in the  $xy$  plane.

The results of the bias estimation experiments in all the measurements with linear motion are summarized in Table 5.2. Simultaneous estimation got higher accuracies than the gyroscope-based optimization and the EKF. In fast motions, where the average change of velocity is higher than  $3.5 \text{ m/s}^2$ , the accuracy drops considerably. All methods show a decrease in accuracy when vibrations or disturbance shocks are applied (files labelled with "\_s" and "\_v"), being more relevant in the case of the gyroscope optimization.

Without taking into account the unobservable bias component, the simultaneous estimation average error is below  $0.05 \text{ }^\circ/\text{s}$ , even in the most challenging cases. The EKF presented smaller average errors than the optimization-based counterparts in some of the experiments, but also less resistance to disturbances and vibrations than the optimization-based simultaneous bias estimation. Figures 5.2 and 5.3 show a single run of the results in a medium-speed linear motion data, for optimization-based and the EKF methods, respectively. The estimated bias component  $\hat{b}_z$  drifts away after a certain time since the  $z$  axis of the sensor frame is almost aligned with gravity during the whole motion.

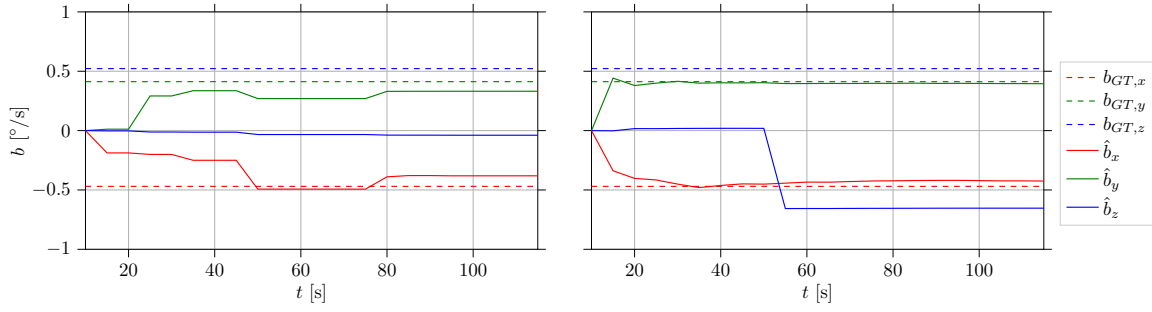


Figure 5.2: Optimization results on a single experiment in medium linear motion (2\_linear\_medium\_b0) of gyroscope bias optimization (left) and simultaneous optimization (right). The vertical bias components is not identifiable.

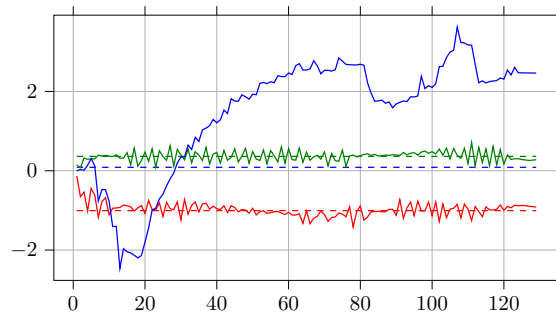
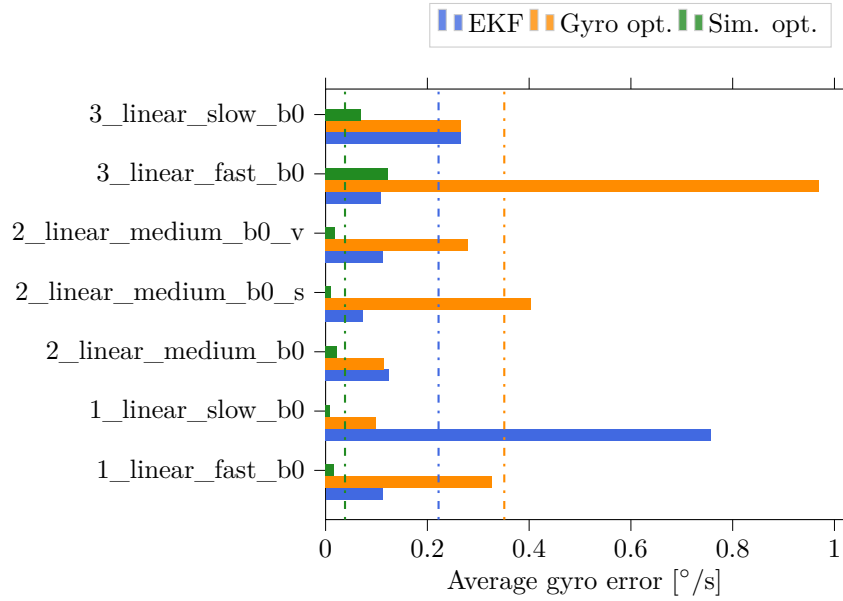


Figure 5.3: EKF bias estimation output on a single experiment in medium linear motion (2\_linear\_medium\_b0). The vertical bias component drifts away as it is non-identifiable.

Table 5.2: Gyroscope bias estimation error ( $x$ - $y$  components) for linear motions after  $t = 90$  s.

Measure	EKF		Gyro		Simult.		All methods
	Avg. [ $^{\circ}$ /s]	Std. [ $^{\circ}$ /s]	Avg. [ $^{\circ}$ /s]	Std. [ $^{\circ}$ /s]	Avg. [ $^{\circ}$ /s]	Std. [ $^{\circ}$ /s]	Avg. [ $^{\circ}$ /s]
1_linear_fast_b0	0.11	0.04	0.33	0.25	0.02	< 0.01	0.15
1_linear_slow_b0	0.76	0.04	0.10	0.05	0.01	< 0.01	0.29
2_linear_medium_b0	0.12	< 0.01	0.11	0.03	0.02	0.01	0.09
2_linear_medium_b0_s	0.07	0.02	0.40	0.07	0.01	< 0.01	0.16
2_linear_medium_b0_v	0.11	0.07	0.28	0.02	0.02	0.02	0.14
3_linear_fast_b0	0.11	0.08	0.97	0.21	0.12	0.07	0.40
3_linear_slow_b0	0.26	< 0.01	0.26	0.22	0.07	< 0.01	0.20
All	0.22	-	0.35	-	0.04	-	0.20

Figure 5.4: Average gyroscope bias estimation error ( $x$ - $y$  components) on linear motions for the three methods

### 5.2.2 Rotational motion

The next group of motions to be analysed consist of pure rotations along the three sensor axes. In rotational motions, the IMU approximately stays in the same place, i.e. no translation is applied. Unlike linear measurements from the previous section, for rotational motions all bias components are observable by any of the methods. Thus, let the bias estimation error be, like in the previous chapter, the norm of the difference between the estimated and the ground truth values:

$$\xi = \|\hat{\mathbf{b}} - \mathbf{b}_{GT}\|. \quad (5.1)$$

The estimation error for each of the files is summarized in Table 5.3.

Experiments showed that the three methods performed with acceptable accuracy in more or less all the measurements. Fast motions present larger errors than slower ones, as expected. The three methods converged to the ground truth values with an error of less than  $0.3^{\circ}$ /s on the most challenging cases of fast rotations, even with gyroscope clipping. Gyroscope optimization-based method performed better than the two other methods when vibrations and disturbance shocks were applied to the motion. Overall, the gyroscope optimization-based method performed with an error of less than  $0.2^{\circ}$ /s, approximately, in most of the motions. Regarding robustness, all the results show that Kalman filtering is slightly more robust than the optimization-based methods, as suggested on the standard deviation of the error values on the table.

To illustrate the convergence over time of the methods, Figures 5.5 and 5.6 show the bias estimation average error for all the conducted experiments in a fast and slow rotational motion, respectively. The plots suggest that, in fast motions, the gyro bias is less observable than when performing slow patterns of rotations. The EKF converged to a near-truth value in less than 20 seconds, whereas the optimization based methods were slightly slower. The plot also suggests that after a certain time the estimation might diverge. This could be due to unpredictable noise or other sources of errors that are more evident over time.

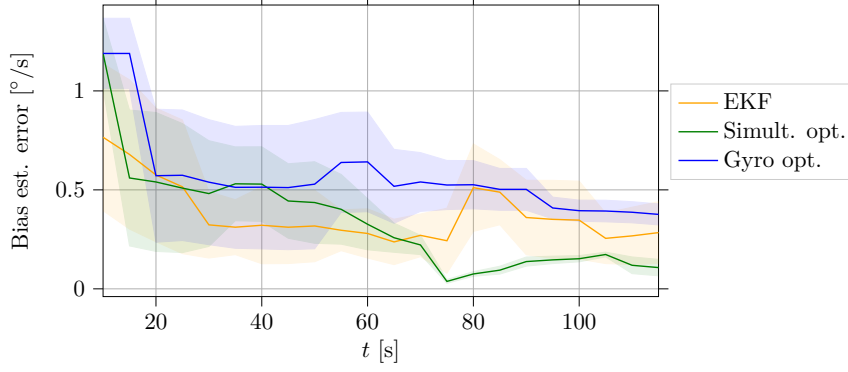


Figure 5.5: EKF bias estimation output on a single experiment in medium linear motion (experiment1\_rotational\_fast\_b0)

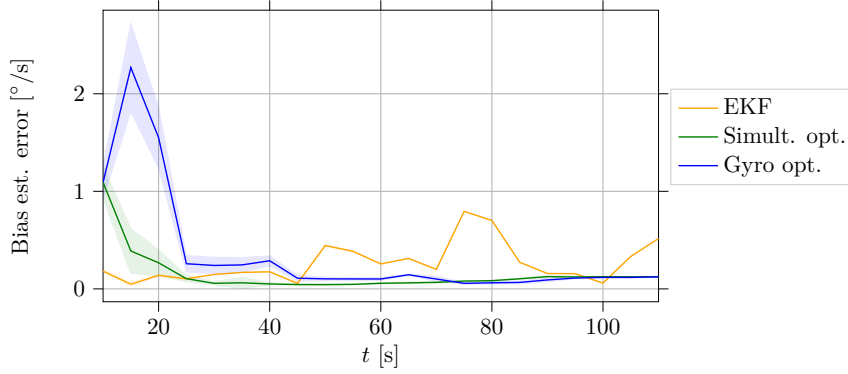


Figure 5.6: EKF bias estimation output on a single experiment in medium linear motion (experiment3\_rotational\_slow\_b0)

Table 5.3: Gyroscope bias estimation error for rotational motions after  $t = 90$  s.

Measure	EKF		Gyro		Simult.		All methods
	Avg. [°/s]	Std. [°/s]	Avg. [°/s]	Std. [°/s]	Avg. [°/s]	Std. [°/s]	Avg. [°/s]
1_rotational_fast_b0	<b>0.36</b>	0.19	<b>0.50</b>	0.11	0.14	0.02	<b>0.33</b>
1_rotational_slow_b0	<b>0.61</b>	< 0.01	<b>0.22</b>	0.01	<b>0.36</b>	< 0.01	<b>0.39</b>
2_rotational_medium_b0	<b>0.06</b>	< 0.01	0.19	0.01	0.13	0.02	0.13
2_rotational_medium_b0_s	0.14	0.03	0.14	0.05	<b>0.06</b>	< 0.01	0.11
2_rotational_medium_b0_v	0.11	0.01	0.15	0.09	0.14	0.01	0.13
3_rotational_fast_b0	<b>0.83</b>	< 0.01	<b>0.40</b>	0.03	0.12	0.01	<b>0.45</b>
3_rotational_fast_b0_rc	0.10	< 0.01	<b>0.29</b>	0.05	<b>0.23</b>	0.02	<b>0.21</b>
3_rotational_slow_b0	0.16	< 0.01	<b>0.09</b>	0.02	0.13	0.01	0.12
All	<b>0.30</b>	-	<b>0.25</b>	-	<b>0.16</b>	-	<b>0.24</b>

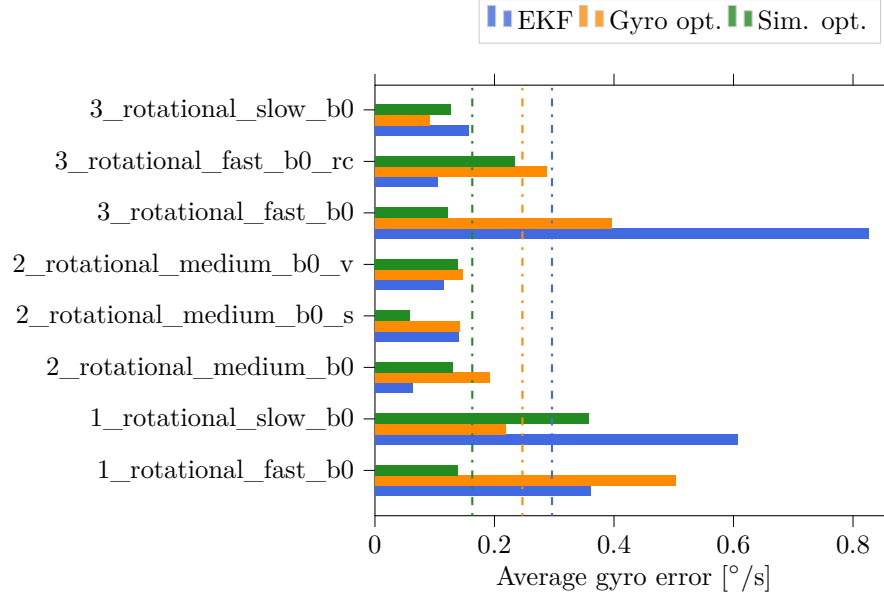


Figure 5.7: Average gyroscope bias estimation error on rotational motions for the three methods

### 5.2.3 Arbitrary motion

The last group of motions to be analysed correspond to the arbitrary measurements. These recordings are the most challenging ones in the sense that high angular rates and accelerations were provided, while simultaneously tapping the IMU or applying vibrations. Table 5.4 shows the results for all the recordings for each of the three methods, after  $t = 90$  s. None of the methods provided accurate results, due to the big challenge that the motions represented. Files where disturbances such as vibrations or clipping were applied, . Nevertheless, it is worth to note that simultaneous optimization-based method presented acceptable results with accuracy of less than  $0.2^\circ$  in some files. In most of the arbitrary files, the robustness of the methods decreased as well, in comparison with the rotational and linear motions, as the standard deviation values suggest. Figure 5.8 shows the estimation accuracies at different instants of time  $t$  for one of the files, with the three methods considered.

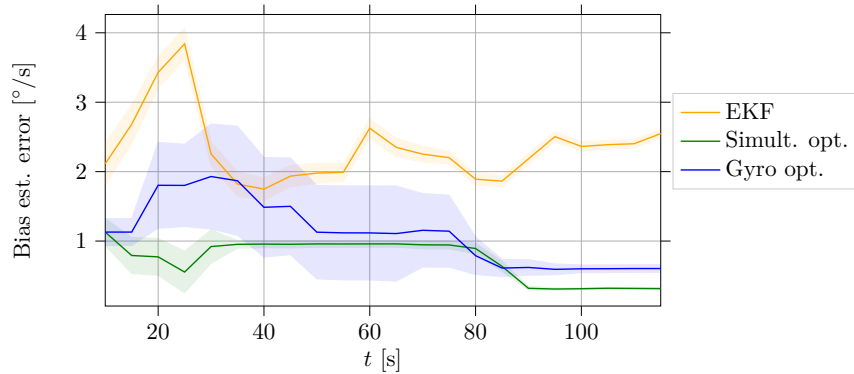


Figure 5.8: EKF bias estimation output on a single experiment in an arbitrary motion (experiment4\_arbitrarily\_mo\_2x60)

Table 5.4: Gyro bias estimation error for arbitrary motions after  $t = 90$ s.

Measure	EKF		Gyro		Simult.		All methods
	Avg. [°/s]	Std. [°/s]	Avg. [°/s]	Std. [°/s]	Avg. [°/s]	Std. [°/s]	
3_arbitrarily	2.22	0.13	2.31	0.10	2.56	0.02	2.36
3_arbitrarily_ms_2x_30	0.75	0.02	0.86	0.03	1.26	0.32	0.95
3_arbitrarily_ms_6x_10	4.04	0.36	0.87	0.38	0.66	0.30	1.86
4_arbitrarily_240s_medium	2.90	0.12	0.83	0.33	1.10	0.02	1.61
4_arbitrarily_240s_slow	1.48	< 0.01	1.45	< 0.01	0.26	0.03	1.06
4_arbitrarily_360s_medium	1.91	0.07	4.04	0.81	0.67	0.03	2.20
4_arbitrarily_360s_slow	1.83	< 0.01	1.46	0.08	0.21	0.01	1.17
4_arbitrarily_md_neo_1x60_1cm	5.11	0.10	0.82	0.34	0.17	0.06	2.03
4_arbitrarily_md_neo_1x60_2cm	5.35	0.12	1.90	0.38	0.66	0.08	2.64
4_arbitrarily_md_neo_1x60_3cm	1.74	0.15	0.78	0.14	0.16	0.09	0.89
4_arbitrarily_md_neo_1x60_5cm	5.42	0.36	3.42	0.89	0.83	0.03	3.22
4_arbitrarily_mo_2x30_2	1.51	0.14	0.60	0.43	0.74	0.06	0.95
4_arbitrarily_mo_2x60	2.18	0.08	0.62	0.11	0.32	0.03	1.04
All	2.80	-	1.53	-	0.74	-	1.69

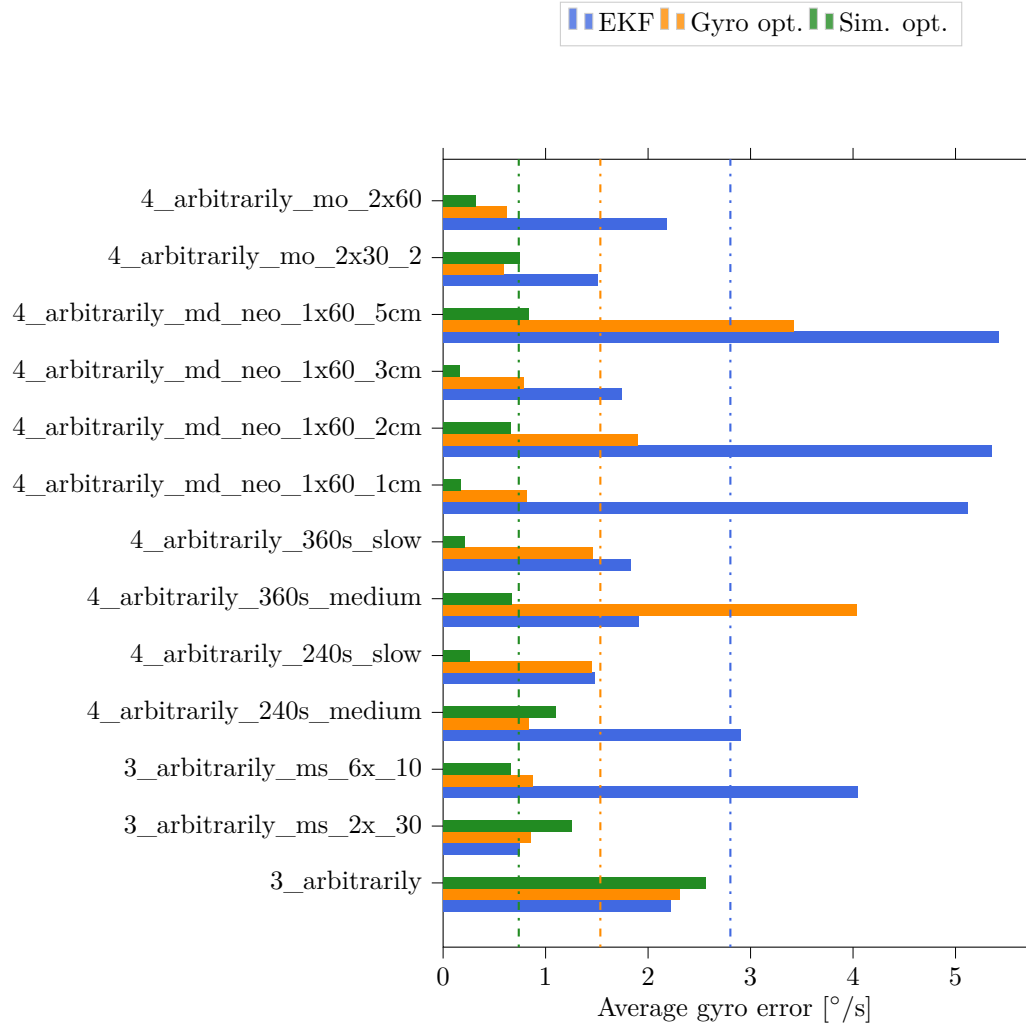


Figure 5.9: Average gyroscope bias estimation error on rotational motions with the three methods.



### 5.3 Application to an orientation estimation filter

Eliminating gyroscope and accelerometer biases has obvious beneficial results in sensor fusion applications, since deterministic constant errors are compensated and it yields a great improvement in the performance of the strap-down integration. Orientation estimation filters fuse the information of gyroscope, accelerometer and magnetometer to estimate the absolute orientation of the sensor from an absolute coordinate system. Quaternions are usually the most compact and precise form to represent the orientation estimation of an strap-down inertial navigation filter. The orientation, can naturally be visualized by the multiple conventions of Euler angles or Tait-Bryan angles. However, the object's orientation can also be defined as a combination of an inclination (attitude) angle and a heading angle, respect to a pair of orthogonal axes. We understand inclination as the shortest-path angle that aligns the object with the horizontal plane of the reference system, which in our case is perpendicular to the gravity vector. The heading, on the other hand, is the rotated angle around the gravity axis. Note that the heading angle does not necessarily correspond to the yaw angle of all Euler angles conventions. Consequently, and using the notation introduced in Chapter 3 to represent quaternions given an axis and an angle, the orientation quaternion can be expressed as:

$${}^S_{\mathcal{R}}\mathbf{q} = \mathbf{q}_\vartheta \otimes \mathbf{q}_\varphi, \quad (5.2)$$

with

$$\mathbf{q}_\vartheta = \left\{ \vartheta @ \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \right\} \quad \mathbf{q}_\varphi = \left\{ \varphi @ \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T \right\}, \quad (5.3)$$

where  $\vartheta$  and  $\varphi$  represent the heading and inclination angles, respectively. Following the definition of quaternion, the heading and inclination angles can be calculated from the quaternion components as:

$$\vartheta = 2 \arctan 2(q_4, q_1) \quad \varphi = 2 \arccos \left( \sqrt{q_4^2 + q_1^2} \right). \quad (5.4)$$

Accelerometers provide additional information to gyroscope strap-down integration on the inclination (attitude) angle, since gravity acceleration can be measured. On the other hand, the magnetometer provides a correction on the heading angle. When working in indoor environments, magnetometer measurements can easily be disrupted due to static (soft-iron) and dynamic (hard-iron) disturbances. This undesired effect makes magnetometer not very useful when working, for example, in an office setting. Some orientation estimation filters do not take into account these effects, and the disturbances influence the whole orientation rather than only the heading part.

In order to analyse the effect of bias compensation effects on the orientation estimation, the optimal quaternion-based filter from Seel et al. [17] is used (from now on, OriEstIMU filter). This filter has the advantage to remove magnetic disturbances effects by independently applying accelerometer and magnetometer corrections to the inclination and heading parts of the orientation. When operating in indoor environments where no absolute heading component is known, a heading offset can be defined. We apply the filter together with ground-truth orientation from an optical device, and evaluate the error by stages of  $T = 10$  s, 20 s, 30 s... For simplicity, and to avoid issues related with magnetic disturbances, we regard only the inclination part of the orientation. We define the error of the inclination estimation, at an instant  $t$ , as:

$$\epsilon_\varphi(t) = (\hat{\varphi}_t - \varphi_t), \quad (5.5)$$

and evaluate the Root Mean Square Error (RMSE) on a time period  $T$  as:

$$\text{RMSE} |_{T=} = \sqrt{\frac{\sum_{t=t_0}^T \epsilon_\varphi^2(t)}{T}}, \quad (5.6)$$

where  $\hat{\varphi}_t$  and  $\varphi_t$  are the estimated and ground truth inclination angles at an instant  $t$ , respectively.

In the following sections, the improvements on (inclination part of) the orientation estimation after on-the-run bias removal are presented. Thus, the applicability of the bias estimation optimization-based Analogously as the previous section, 10 experiments with random added gyroscope values were conducted for each of the measurements of the data-set. The orientation estimation quality was evaluated as follows:

1. **With OriEstIMU filter (OEI), before bias estimation**
2. **With 7-states EKF:** We apply the extended Kalman filter with accelerometer measurements for 6-DOF sensor fusion described in Section 3.3.1 to get inclination and gyroscope bias estimations.
3. **With OriEstIMU and gyroscope optimization (GO):** Using the gyroscope optimization-based method to remove gyroscope bias on moving windows of length  $T_w$ .
4. **With OriEstIMU and simultaneous optimization (SO):** Removing gyroscope and accelerometer biases using the simultaneous estimation method, on a moving windows of length  $T_w$ .

Table 5.5: Inclination RMSE (5.6) for each of the experiments with OriEstIMU filter. A bias of  $0.5^\circ/\text{s}$  around the  $(1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})$  axis was added to the gyroscope measurements. Parametrization of the filter was:  $\tau_{\text{acc}} = 4 \text{ s}$  and  $\text{acc}_{\text{rating}} = 0$ .

Measurement	RMSE( $\varphi$ ) [ $^\circ$ ]
1_linear_fast_b0	9.04
1_linear_slow_b0	2.51
1_rotational_fast_b0	3.01
1_rotational_slow_b0	1.08
2_linear_medium_b0	2.23
2_linear_medium_b0_s	2.48
2_linear_medium_b0_v	2.32
2_rotational_medium_b0	1.27
2_rotational_medium_b0_s	1.50
2_rotational_medium_b0_v	1.66
3_all_movements	6.71
3_arbitrarily	8.57
3_arbitrarily_ms_2x_30	4.15
3_arbitrarily_ms_6x_10	5.13
3_linear_fast_b0	25.60
3_linear_slow_b0	2.78
3_rotational_fast_b0	2.91
3_rotational_fast_b0_rc	3.40
3_rotational_slow_b0	1.61
4_arbitrarily_240s_medium	3.74
4_arbitrarily_240s_slow	1.31
4_arbitrarily_360s_medium	4.32
4_arbitrarily_360s_slow	1.33
4_arbitrarily_md_neo_1x60_1cm	1.28
4_arbitrarily_md_neo_1x60_2cm	1.35
4_arbitrarily_md_neo_1x60_3cm	1.65
4_arbitrarily_md_neo_1x60_4cm	1.51
4_arbitrarily_md_neo_1x60_5cm	1.92
4_arbitrarily_mo_2x30_2	1.82
4_arbitrarily_mo_2x60	1.84
<b>Average</b>	<b>3.67</b>

### 5.3.1 Linear motion

Bias estimation in linear motion measurements showed good accuracy, regarding the observable components. However, when the sensor remained mostly horizontal and one of the axes was almost aligned with gravity, that bias component remained hidden. The improvements in the orientation estimation are regarded hereafter. The vertical component of the bias does not influence the inclination when the sensor remains mostly in one of the faces.

The results of the experiments are summarized in Table 5.6. As it can be seen in the results, most challenging (fast) motions yielded poor orientation estimation accuracies of over  $6^\circ$  from the ground

truth. Table 5.7 shows the relative improvement on the inclination RMSE for each of the linear measurements. The results showed that optimization based methods yielded on average 64 % and 52 % orientation estimation improvements on linear motions. Fast motions were more challenging and thus, the improvement was less, or even made the estimation accuracy decrease. The simultaneous gyroscope and accelerometer optimization provided slightly larger improvements since the gyroscope bias estimation accuracy was notably better (see Table 5.2 ) and also by removing also accelerometer biases. Note that accelerometer bias in the available IMU was rather small. In Figure 5.10, a bar plot graphically illustrates these results.

Figures 5.11 and 5.12 show the calculated RMSE after several time-windows of 10 seconds, for medium-speed and slow-speed linear motions, respectively. In the second plot, it can be seen that both methods provided significant orientation estimation improvements after 20 seconds of motion. Note that the deviation window (shadowed area representing the standard deviation) is big when no bias correction is applied since the random generated biases varied in magnitude. The plots suggest that the simultaneous optimization performed more accurately and faster than the pure-gyroscope optimization. The EKF converged faster to near the ground-truth values than the optimization-based methods. The standard deviation windows in the first plot indicate that, if the remaining bias is small (less than  $0.3^\circ/\text{s}$ ) the optimization might yield to wrong biases and therefore, worsen the orientation estimation.

Figure 5.13 shows the inclination estimation error with OriEstIMU filter on a single experiment in a slow linear motion, before and after applying a) simultaneous accelerometer and gyroscope bias compensation and b) pure-gyroscope bias compensation. The estimation windows of the simultaneous estimation as represented as blue shadowed area at the beginning of the recording. It can be seen that the filter quickly converges once the biases have been identified and removed.

Table 5.6: Inclination estimation RMSE (5.6) for linear motions from  $t_0 = 60$  s to the end of the measurement. The parametrization of OriEstIMU filter was  $\tau = 4$  s and  $\text{acc}_{\text{rating}} = 0$ . The noise covariance of the EKF was set as  $\sigma_w = 10^7$ .

Measure	OEI before bias comp.		EKF		OEI and gyr. opt.		OEI and sim. opt.		Avg. [°]
	Avg. [°]	Std. [°]	Avg. [°]	Std. [°]	Avg. [°]	Std. [°]	Avg. [°]	Std. [°]	
2_linear_medium_b0_s	5.69	1.56	0.86	0.04	1.81	0.20	0.83	0.03	1.17
1_linear_slow_b0	5.35	1.74	1.96	0.02	1.20	0.08	0.83	0.07	1.33
3_linear_slow_b0	4.91	1.53	1.73	< 0.01	1.15	0.87	0.64	0.03	1.17
1_linear_fast_b0	8.74	2.42	1.91	0.02	6.58	0.94	5.79	0.05	4.76
2_linear_medium_b0	5.24	1.66	1.29	< 0.01	0.86	0.05	0.46	0.05	0.87
3_linear_fast_b0	17.61	6.71	1.45	0.06	21.82	2.04	18.05	0.43	13.77
2_linear_medium_b0_v	4.84	1.71	1.32	< 0.01	2.14	0.08	1.38	0.06	1.61
<b>All</b>	<b>7.48</b>	-	<b>1.5</b>	-	<b>5.08</b>	-	<b>4.0</b>	-	<b>3.53</b>

Table 5.7: Relative improvement on the RMSE (5.6) of the inclination in linear motions after applying bias compensation with the optimization-based methods.

Measure	With simultaneous optimization [%]	With Gyro optimization [%]
2_linear_medium_b0_s	85.4	68.2
1_linear_slow_b0	84.5	77.6
3_linear_slow_b0	87.0	76.6
1_linear_fast_b0	33.8	24.7
2_linear_medium_b0	91.2	83.6
3_linear_fast_b0	-2.5	-23.9
2_linear_medium_b0_v	71.5	55.8
<b>All</b>	<b>64.4</b>	<b>51.8</b>

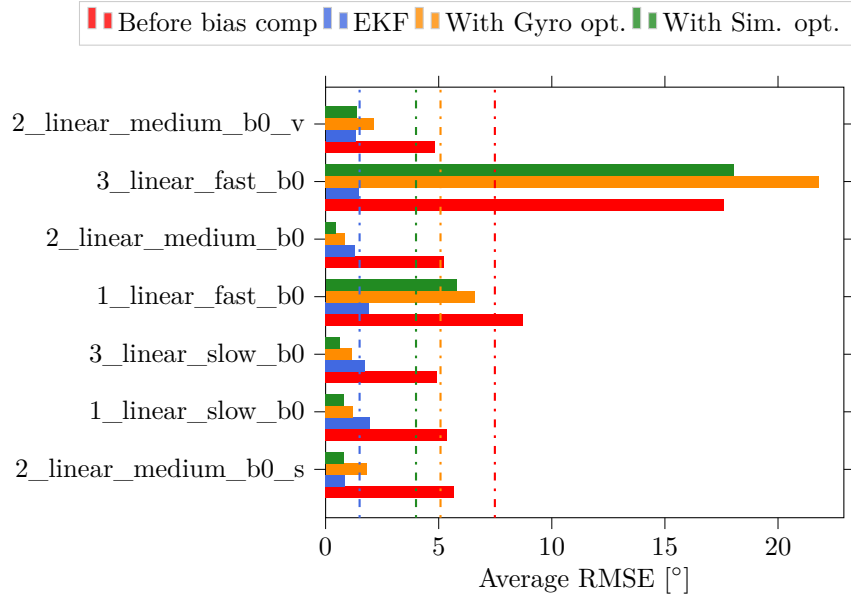


Figure 5.10: Average inclination RMSE on linear motions with EKF, and oriEstIMU filter before and after optimization-based bias compensation

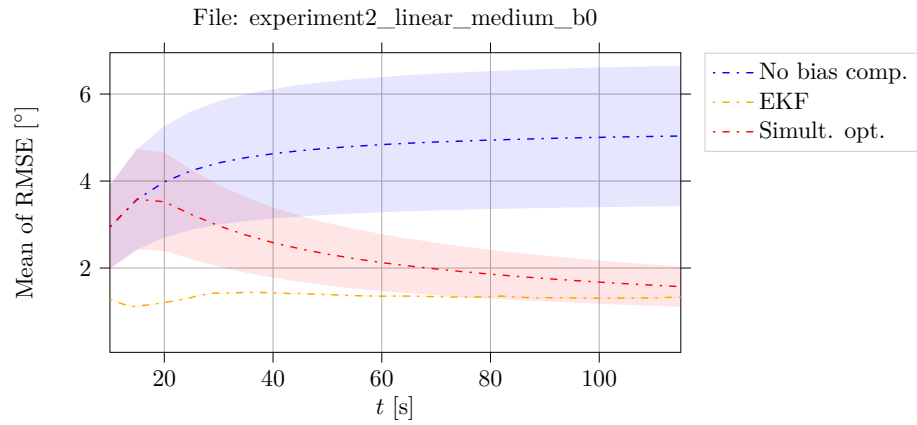


Figure 5.11: Average and standard deviation (shadowed area) of orientation estimation RMSE in a medium-speed linear measurement

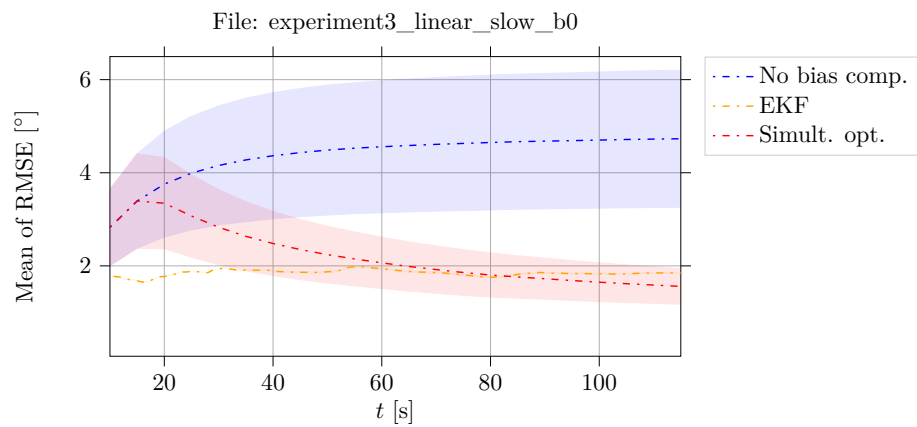


Figure 5.12: Average and standard deviation (shadowed area) of orientation estimation RMSE in a slow-speed linear measurement

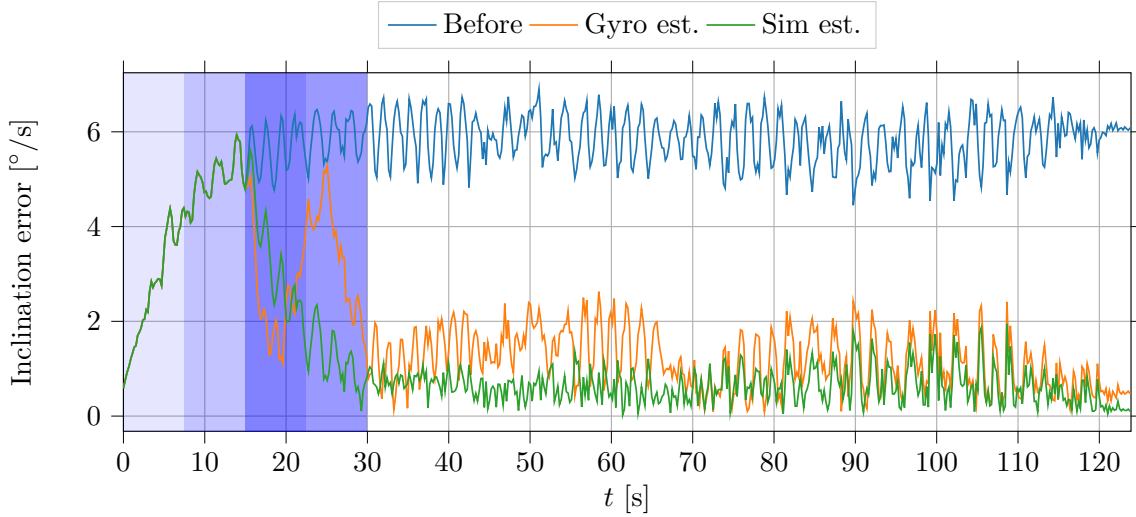


Figure 5.13: Inclination estimation error (5.5) with OriEstIMU on a slow linear motion, before and after applying optimization-based bias compensation. On blue coloured area, the first three overlapping windows of the estimation. The estimation quickly converges to near-ground-truth values after the biases have been identified.

### 5.3.2 Rotational motion

When performing rotational motions, all gyroscope and accelerometer axes are properly excited and the bias effect in all directions becomes observable. Experiments showed good estimation accuracies in almost all the measurements, even the most challenging ones with rotational clipping and vibrations. The inclination part of the orientation estimation yielded more accurate results than in linear motions, and in general the relative improvement of removing the gyroscope biases on-the-run is very noticeable.

The results of the experiments are summarized in Table 5.10. In Figure 5.14, a bar plot graphically illustrates these results. It is obvious that orientation estimation accuracy highly depends on the speed of the motion. For slow and medium speed measurements, the inclination RMSE is less than  $2^\circ$ , and it is worsen to up to  $3.7^\circ$  when vibrations and disturbances are applied. Table 5.9 shows the relative improvement on the inclination RMSE for each of the linear measurements. For fast files, the accuracy is approximately  $5^\circ$ . Optimization showed improvements in the orientation estimation of, at least, 32 % in the worst cases (fast motions) and up to 70 % in the best cases. Although no big differences between both methods were observed, the simultaneous optimization showed slightly higher improvement on most of the files.

Examples of RMSE after several time-windows of 10 seconds are plotted in Figures 5.15 and 5.16, for slow and fast rotational motions, respectively. Analogously to linear motions, the improvements in the orientation were observable after 20 seconds of motion. In the slow measurement, both methods provided significant improvements since the estimated bias error was under  $0.05^\circ/\text{s}$ , and the inclination estimation differs  $1^\circ$  from the ground truth in the whole recording. When moving the IMU faster, none of the methods was able to compensate 100% of the bias, which manifests in a lower orientation estimation improvement.

Figure 5.17 shows the inclination estimation error with OriEstIMU filter on a single experiment in a medium-speed rotational motion, before and after applying a) simultaneous accelerometer and gyroscope bias compensation and b) pure-gyroscope bias compensation. The estimation windows of the simultaneous estimation as represented as blue shadowed area at the beginning of the recording. The inclination estimation improves notably once the biases are compensated.

Table 5.8: Inclination estimation RMSE (5.6) for rotational motions from  $t_0 = 60$  s to the end of the measurement. The parametrization of OriEstIMU filter was  $\tau = 4$  s and  $\text{acc}_{\text{rating}} = 0$ . The noise covariance of the EKF was set as  $\sigma_w = 10^7$ .

Measure	OEI before bias comp.		EKF		OEI and gyr. opt.		OEI and sim. opt.		Avg. [°]
	Avg. [°]	Std. [°]	Avg. [°]	Std. [°]	Avg. [°]	Std. [°]	Avg. [°]	Std. [°]	
3_rotational_fast_b0_rc	4.68	0.63	3.17	< 0.01	3.29	0.04	3.17	0.01	3.21
2_rotational_medium_b0	3.24	0.85	1.05	< 0.01	1.01	0.01	0.96	< 0.01	1.00
2_rotational_medium_b0_s	3.56	0.85	1.52	< 0.01	1.61	0.02	1.69	0.02	1.61
1_rotational_slow_b0	2.39	0.82	1.57	< 0.01	0.90	0.01	0.83	0.02	1.10
3_rotational_slow_b0	2.74	0.98	1.25	< 0.01	0.96	< 0.01	0.97	0.01	1.06
3_rotational_fast_b0	4.75	1.04	2.39	< 0.01	2.54	0.02	2.50	0.01	2.48
2_rotational_medium_b0_v	3.16	1.08	1.44	< 0.01	1.57	0.05	1.64	< 0.01	1.55
1_rotational_fast_b0	5.22	1.16	2.74	0.01	3.00	0.02	2.99	0.02	2.91
<b>All</b>	<b>3.72</b>	-	<b>1.89</b>	-	<b>1.86</b>	-	<b>1.84</b>	-	<b>1.86</b>

Table 5.9: Relative improvement on the RMSE (5.6) of the inclination in rotational motions after applying bias compensation with the optimization-based methods.

Measure	With simultaneous optimization [%]	With gyro optimization [%]
3_rotational_fast_b0_rc	32.3	29.7
2_rotational_medium_b0	70.4	68.8
2_rotational_medium_b0_s	52.5	54.8
1_rotational_slow_b0	65.3	62.3
3_rotational_slow_b0	64.6	65.0
3_rotational_fast_b0	47.4	46.5
2_rotational_medium_b0_v	48.1	50.3
1_rotational_fast_b0	42.7	42.5
<b>All</b>	<b>52.9</b>	<b>52.5</b>

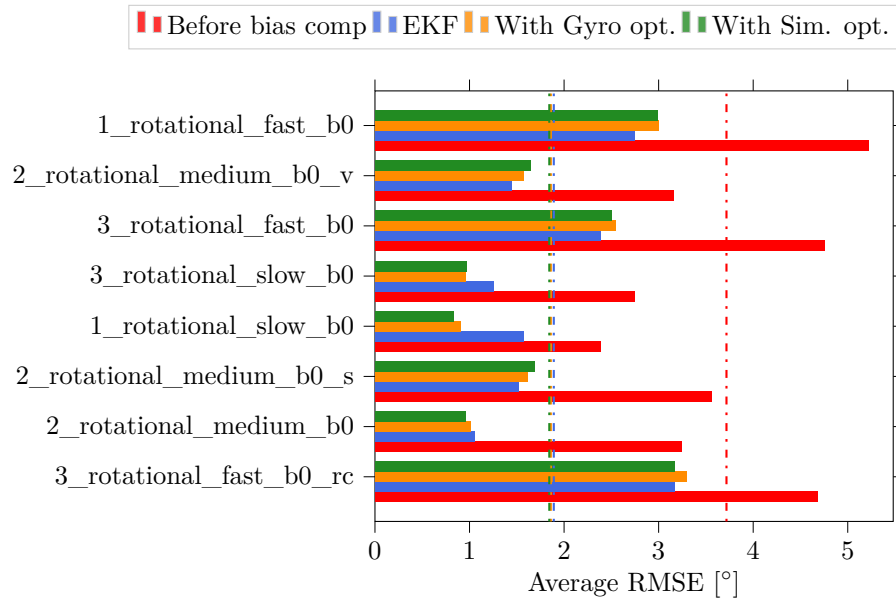


Figure 5.14: Average inclination RMSE (5.6) on rotational motions with EKF, and oriEstIMU filter before and after optimization-based bias compensation

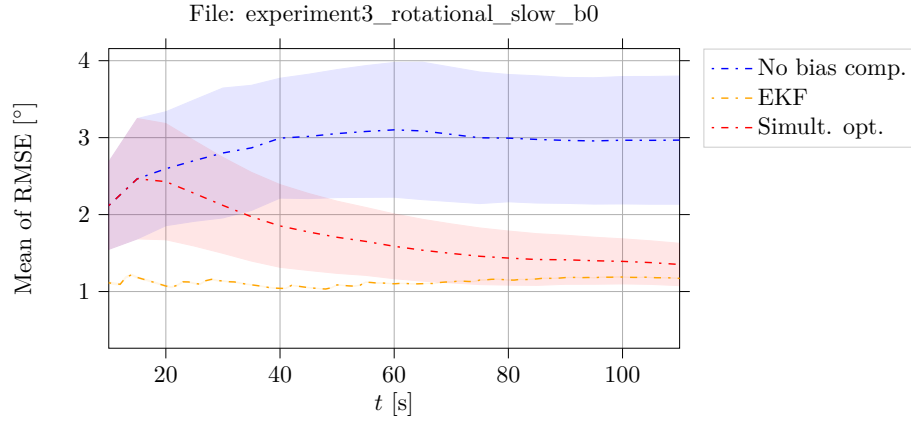


Figure 5.15: Average and standard deviation (shadowed area) of orientation estimation RMSE (5.6) in a slow rotational measurement

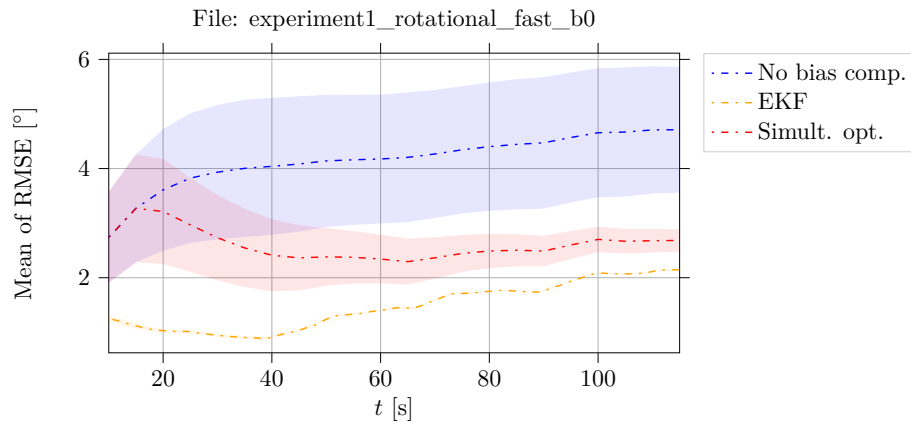


Figure 5.16: Average and standard deviation (shadowed area) of orientation estimation RMSE (5.6) in a fast rotational measurement



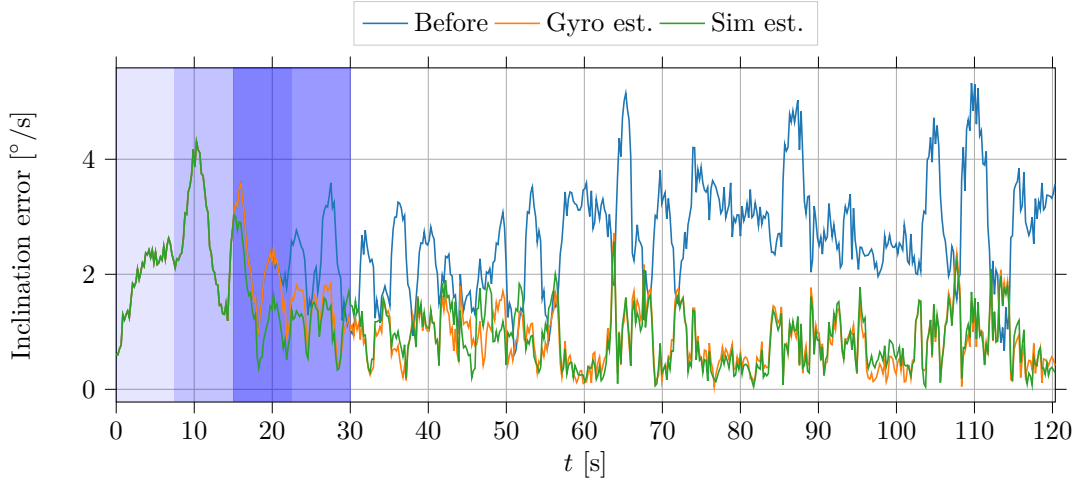


Figure 5.17: Inclination estimation error (5.5) with OriEstIMU on a medium-speed rotational motion, before and after applying optimization-based bias compensation. On blue coloured area, the first three overlapping windows of the simultaneous estimation method. The filter quickly converges to near-ground-truth values after the biases have been identified and compensated.

### 5.3.3 Arbitrary motion

The most challenging group of motions is the one formed by those where no specific pattern was followed. In this type of motions, the bias estimation showed significantly less accurate results than for rotational and linear files. Table 5.4 shows the results of all the experiments for the arbitrary motion files. Table 5.11 shows the relative improvement on the inclination estimation after applying the optimization-based bias estimation methods. In Figure 5.18, a bar plot graphically illustrates these results.

This set of measurements represent the most challenging scenario for the orientation estimation filter and bias compensation. The improvements in the inclination estimation after applying the optimization-based methods are notably lower than in the previous sets of motions, with 31 % for the simultaneous estimation, on average, and 16 % for the pure-gyroscope method. When moving the IMU slowly, a performance improvement of up to 60 % was achieved. Medium speed files and all the motions from "\_3" showed significant less orientation estimation improvement than the rest of measurements. Note that, in some of the motions the RMSE after applying the built-in bias estimation and the gyroscope optimization is higher than before, due to the high inaccuracy and non-observability of the bias in some of the motions. On average, the performance of the EKF is similar to the one provided by OriEstIMU filter with simultaneous bias compensation.

Two examples of RMSE at several steps of 10 seconds are plotted in Figures 5.19 and 5.20. The first figure refers to slow arbitrary motion where shock disturbances were applied, and the second one refers to an arbitrary motion with magnetic disturbances. In the first plot, it can be appreciated that the effects of the estimation are noticeable after 30 seconds, slightly later than in the rotational and linear files.

Figure 5.21 shows the inclination estimation error with OriEstIMU filter on a single experiment in a slow-speed arbitrary motion, before and after applying a) simultaneous accelerometer and gyroscope bias compensation and b) pure-gyroscope bias compensation. The estimation windows of the simultaneous estimation as represented as blue shadowed area at the beginning of the recording. The inclination estimation improves notably once the biases are compensated.

Table 5.10: Inclination estimation RMSE (5.6) for arbitrary motions with OriEstIMU filter, before and after applying optimization-based bias estimation, and 6DOF EKF.

Measure	OEI before bias comp.		EKF		OEI and gyr. opt.		OEI and sim. opt.		Avg. [°]
	Avg. [°]	Std. [°]	Avg. [°]	Std. [°]	Avg. [°]	Std. [°]	Avg. [°]	Std. [°]	
4_arbitrarily_360s_slow	2.84	0.94	1.76	< 0.01	2.30	0.06	1.03	0.03	1.70
3_arbitrarily	7.03	1.17	4.72	0.12	7.81	0.19	7.90	0.03	6.81
4_arbitrarily_md_neo_1x60_2cm	2.13	0.66	3.00	0.04	1.97	0.12	1.47	0.02	2.15
3_arbitrarily_ms_6x_10	5.54	0.92	2.92	0.07	5.12	0.56	5.02	0.11	4.35
4_arbitrarily_mo_2x60	2.93	0.63	2.11	0.04	2.40	0.50	1.94	0.04	2.15
4_arbitrarily_240s_medium	3.95	0.26	2.27	0.01	3.62	0.10	3.58	0.02	3.16
4_arbitrarily_md_neo_1x60_1cm	2.09	0.47	2.75	0.03	2.05	0.19	1.22	0.02	2.01
4_arbitrarily_md_neo_1x60_5cm	3.17	0.83	2.55	0.09	2.58	0.44	1.97	0.03	2.37
4_arbitrarily_mo_2x30_2	3.76	1.13	1.72	0.03	1.46	0.25	1.47	0.04	1.55
3_arbitrarily_ms_2x_30	5.02	1.04	4.19	0.01	4.57	0.05	4.68	0.05	4.48
4_arbitrarily_md_neo_1x60_4cm	2.68	0.74	2.28	0.06	2.67	0.07	1.35	0.02	2.10
4_arbitrarily_md_neo_1x60_3cm	2.79	0.65	2.09	0.03	1.71	0.04	1.78	0.02	1.86
4_arbitrarily_360s_medium	3.09	0.63	1.96	0.06	3.38	0.43	2.79	0.02	2.71
4_arbitrarily_240s_slow	2.97	1.14	1.58	< 0.01	1.37	0.01	1.18	0.03	1.38
<b>All</b>	<b>3.57</b>	-	<b>2.56</b>	-	<b>3.07</b>	-	<b>2.67</b>	-	<b>2.77</b>

Table 5.11: Relative improvement on the RMSE (5.6) of the inclination in arbitrary motions after applying bias compensation with the optimization-based methods.

Measure	With simultaneous optimization [%]	With gyro optimization [%]
4_arbitrarily_360s_slow	63.7	19.0
3_arbitrarily	-12.4	-11.1
4_arbitrarily_md_neo_1x60_2cm	31.0	7.5
3_arbitrarily_ms_6x_10	9.4	7.6
4_arbitrarily_mo_2x60	33.8	18.1
4_arbitrarily_240s_medium	9.4	8.4
4_arbitrarily_md_neo_1x60_1cm	41.6	1.9
4_arbitrarily_md_neo_1x60_5cm	37.9	18.6
4_arbitrarily_mo_2x30_2	60.9	61.2
3_arbitrarily_ms_2x_30	6.8	9.0
4_arbitrarily_md_neo_1x60_4cm	49.6	0.4
4_arbitrarily_md_neo_1x60_3cm	36.2	38.7
4_arbitrarily_360s_medium	9.7	-9.4
4_arbitrarily_240s_slow	60.3	53.9
<b>All</b>	<b>31.3</b>	<b>16.0</b>

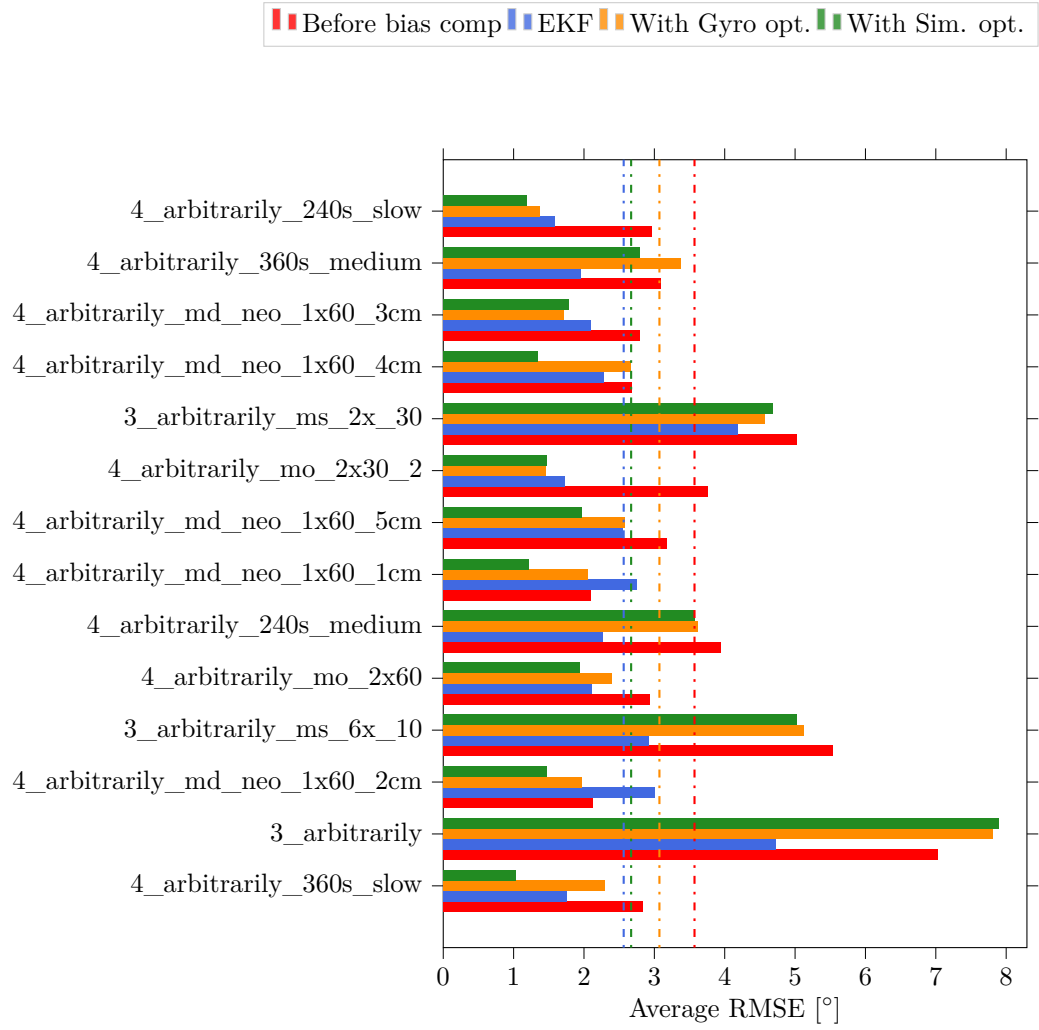


Figure 5.18: Average inclination RMSE (5.6) on arbitrary motions with EKF, and oriEstIMU filter before and after optimization-based bias compensation

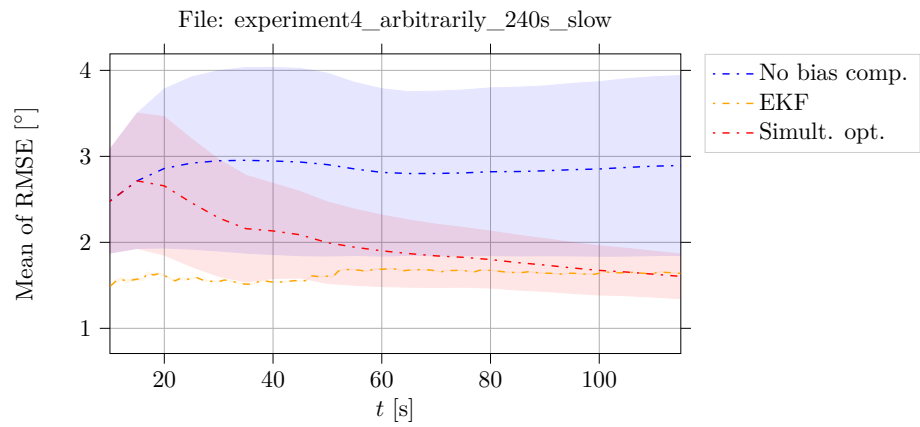


Figure 5.19: Average and standard deviation (shadowed area) of orientation estimation RMSE (5.6) in a slow arbitrary measurement (experiment4\_arbitrarily\_240s\_slow)

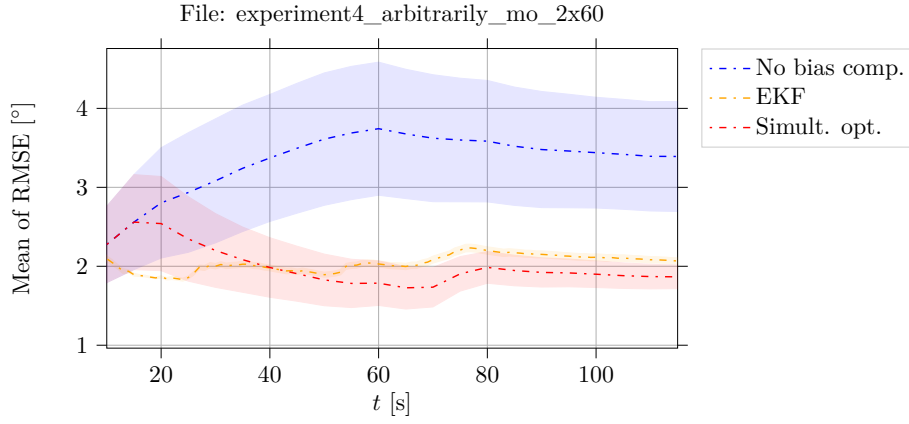


Figure 5.20: Average and standard deviation (shadowed area) of orientation estimation RMSE (5.6) in a fast arbitrary measurement(experiment4\_arbitrarily\_mo\_2x60)

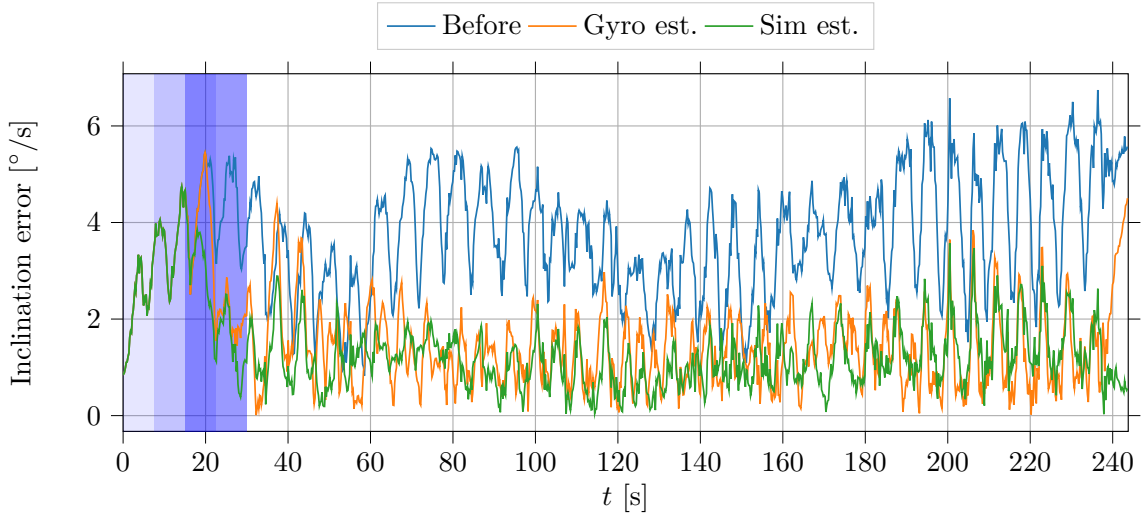


Figure 5.21: Inclination estimation error (5.5) with OriEstIMU on a slow arbitrary motion, before and after applying optimization-based bias compensation. On blue coloured area, the first three overlapping windows of the estimation. The estimation quickly converges to near-ground-truth values after the biases have been identified.

# 6

## Conclusions and recommendations

The goal of this Master's thesis was to develop an algorithm that ensures the calibration of the Searchwing inertial sensors while on a mission at sea, where assuming static conditions is not realistic. A comprehensive review of the state-of-the-art methods was carried, and the conceptual design of the algorithm and validation under several conditions was developed. This Chapter outlines the main conclusions of the study, the shortcomings and advantages of the proposed methods, as well as the research contributions of this work. First, in Section 6.1, an outlook over the work done is carried out, and the main results are interpreted. Then, in Section ??, the conclusions are summarized by addressing the research in a structured manner. Secondly a discussion on the applicability and shortcomings of this work is presented. Lastly, this Chapter ends with a set of recommendations for future work and research on the topic.

### 6.1 Outlook and discussion

In this section, the proposed methods and main results of this Master thesis research are summarized, in order to have a general view of the work done. We discuss the main points covered, and interpret the results.

In Chapter 1, an overview of the context and the application domain of this Master thesis was introduced. We concluded, that a new method for IMU bias identification on a high-sea scenario was necessary for the correct operation of the Searchwing drone. Since the drone is intended to spend long periods of time on the sea, the sensors will suffer drift that will negatively affect the accuracy of the built-in motion tracking algorithms.

In Chapter 2, a comprehensive overview of the calibration algorithms available in literature was provided. The advantages and disadvantages of the available methods was assessed. The research focused on in-field calibration techniques, making special emphasis on the motion-related restrictions that each of the methods assume. This overview divided in-field calibration techniques into two different and divided sub-categories: a) Multi-position optimization-based methods and b) Kalman filter-based solutions. Multi-position methods are usually effective and simple as it can be applied in-field by any user that follows the required pattern of rotations. Different approaches are considered, but most of the methodologies are capable of identifying at least a set of 18 parameters, including both accelerometer and gyroscope biases and two  $3 \times 3$  symmetric matrices that account for scale factors and non-orthogonalities. Kalman filter-based methods represent a different approach that does not restrict the required orientations to estimate calibration parameters as optimization-based methods do. However, tuning is usually a complicated task. Literature research has shown that among the methods reviewed, although effective, none of the proposed algorithms would be able to compensate sensor biases in a constant-motion setting. Therefore, none of them is applicable to the Searchwing drone calibration scenario.

In Chapter 3 two novel optimization-based methods were developed, in addition to two state-of-the-art extended Kalman filters for gyroscope and accelerometer bias estimation. The novel methods presented are based on the intuitive fact that acceleration and velocity are upper bounded within small time-windows in the absence of sensor's biases, and thus, a set of cost functions that penalize acceleration and velocity drifting was defined. The two methods have simple tuning parameters that only account for the moving-average low pass filter window-length, optimization time-window length and the down-sampling factor. By applying the two novel-methods, gyroscope and accelerometers biases can be estimated under any non-restrictive setting, even in a moving-vessel. Yet, Kalman filter approaches are a fast, well studied and reliable solution, they do not account for accelerometer biases. We highlight the main advantages of the proposed methods over Kalman filtering techniques as: simple, intuitive and with low set of tuning parameters.

Chapter 4 addressed the calibration procedure and was concerned with the validation of the algorithms under both synthetic and real IMU data that simulated the conditions met in a boat on the high-seas. First, in Section 4.1, a simplified mathematical framework to simulate the motions of the calibration procedure on the high-seas is presented. The purpose of this framework is to simulate IMU readings and validate the algorithms on synthetic data. In Section 4.2, the introduced novel-methods are validated on synthetic data, and the limitations of the accuracy of the algorithms is studied regarding different parameters that influence the estimation. Results on ship-like simulated data have shown that the accuracy of the novel methods is resilient to aggressive motions, and the calibration is still valid within the acceptable range of error, even in the case of very rough waters. The results on synthetic data have also shown that Gaussian noise and accelerometer scale factors are among other limiting factors of the algorithms, although values typical of MEMS sensors do not compromise the accuracy of the estimation. Nevertheless, it is important to note, that a good accelerometer scaling should be provided. In Section 4.3 the optimization-based methods were evaluated by using real IMU data that emulates the motions performed by an user on a vessel on the high-seas. Three different scenarios were considered, ranging from less to more challenging motions. The methods were evaluated by adding randomized biases to the data, over a total of 20 experiments. Its performance was compared with the EKF-based solutions presented in Chapter 3. The results of this experiments have lead to the following conclusions:

- Gravity-based 7-states EKF converges faster than the optimization-based methods for gyroscope bias calibration, but accuracy is limited. Simultaneous estimation performs more robustly and accurate than the EKF in all the studied motions. Thus, we conclude that simultaneous estimation is a less restrictive method than the EKF-based techniques.
- EKF-based methods are less computationally expensive solutions.
- Gyroscope bias optimization performs with an acceptable accuracy static and moderate motion (unimodal simulations) settings, although its accuracy is limited in more challenging cases or when two motion frequencies are mixed, like in the case of the experiments on the stairs. As the algorithm relies on the boundedness of the accelerations to identify gyroscope biases, the observability of such becomes limited in cases with more aggressive motions, where higher changes of velocity are applied.
- The 10-states EKF based in pseudo-velocity observations is not a good candidate to estimate gyroscope biases when heavy motions are performed. The results showed that the accuracy was very low in the two non-static scenarios analysed. The accuracy shown in less challenging cases is acceptable, but takes more time to converge.
- Simultaneous bias estimation showed very good accuracy in all the settings. However, in the static case it was considerably decrease. It can be concluded that the algorithm needs proper excitement in all the sensor axes, so that bias influence becomes relevant and thus, identifiable. Another conclusion that the results put in evidence, is that accelerometer biases are less observable than gyroscope biases, and therefore the estimation accuracy can be more compromised.

The presented novel-methods are good candidates to be applied in any motion setting and become an extension to orientation estimation filters. Chapter 5 explored this question and evaluated the accuracy and robustness of the methods in an extensive data-set that comprises a wide range of

measurements with different motions. The dataset comprised pure-translational, pure-rotational and arbitrary motions at different speeds and applying different types of disturbances (such as vibrations or rotational clipping). A total of 40 simulations by adding randomized gyroscope biases on the data were carried. In Section 5.2 the gyroscope bias estimation accuracy of the optimization-based methods and the 7-states EKF was evaluated. Then, in Section 5.3, optimization-based methods were applied on-the-run together with an orientation estimation filter. The inclination part of the orientation was compared with ground-truth values from an optical device, in order to explore beneficial effects in the estimation after bias compensation. The optimization-based methods and OriEstIMU modular set-up were compared with the EKF-based solution. The following conclusions are extracted from the analysis:

- Both optimization based methods and the EKF rely in some form to gravity effects that manifest bias drift. Therefore, if one of the sensor axes remains aligned with gravity the effect of this sensor. The results of the experiments showed that translational motions in where this occurred, one of the sensor biases became unobservable for any of the methods.
- When a proper excitement of all sensor axes occurs, sensor biases can be estimated with good accuracy, even during challenging motions, as the results in rotational motions suggested.
- The effects of bias compensation in orientation estimation become evident after a short period of time in any type of motion. Even if the accuracy is not very high for some of the motions, the algorithms can remove at least 50 % for most of the studied motions. However, in some cases the experiments have demonstrated that the algorithm could even worsen the estimation through big errors in the estimation.
- Extending the OriEstIMU with simultaneous estimation has beneficial results and the experiments showed that the accuracy is similar to the one provided by the EKF, with some exceptions: fast linear motions and some of the arbitrary motions, where the accuracy of the EKF was notably better.
- Optimization-based methods (especially simultaneous gyroscope and accelerometer estimation) provided better bias estimation accuracy, on average, than the EKF-based solution. However, when using OriEstIMU together with the optimization-based methods, the inclination estimation performed slightly worse than the EKF in linear and arbitrary motions. As the implementation with OriEstIMU was modular, the optimization-based methods for bias compensation could be easily applied with any other orientation estimation filter.

It is relevant to note that results for the on-the-run calibration scenario were obtained with different parametrizations of the algorithms, such that results on each of the motions were acceptable. However, this parametrization does not indicate an optimized way of using the algorithm for every file, and probably more accurate results could be exploited when performing finer tuning for specific sets of motions, as in the on-sea calibration scenario.

## 6.2 Conclusions and limitations

The following paragraphs tackle the main shortcomings and contributions of this work, in order to set a path of future work and recommendations.

### Conclusions

Overall, the main contribution of this research is outlined as an optimization-based framework to estimate inertial sensor biases in a wide variety of scenarios, without any input or known quantity required. The main research question of this Master thesis is outlined in Chapter 1, which aims to solve the problem of the Searchwing drone calibration on the high-seas. After carrying a literature research, deriving new methods and evaluating the experimental results, we summarize the main conclusions of this work hereafter:

- The two proposed methods for bias estimation comprise, up to date and to the author's knowledge, the first optimization-based algorithms that are non-restrictive in terms of motion, to estimate accelerometer and gyroscope biases. We highlight the applicability of these methods on any scenario, and more specifically on a high-sea calibration setting during a SAR mission of the Searchwing drone. Large sensor offsets compromise the accuracy of the whole system, and are of special relevance on the critical take-off and landing phases. The proposed methods can be applied prior to fly on a moving vessel, with good accuracy as shown in the results of Chapter 4. We highlight the contributions of this work as a significant improvement to the accuracy and safety of the whole system.
- We highlight, as well, the simplicity of the methods and the low-set of tuning parameters and inputs required. In opposition to state-of-the-art methods, as mentioned in Chapter 2, the proposed novel methods do not require complex tuning or local gravity field observations. In opposition to the Kalman filtering solutions, the flexibility (motion-wise) is larger and tuning is much more intuitive.
- The results of the experiments with orientation estimation have shown the benefits of applying optimization-based bias compensation on a time windows basis, while on-the-run. This results, although preliminary, have shown the potential of such methods on an *online* setting. The limitations regarding this implementation will be discussed hereafter.

## Limitations

The sensor model for calibration described in Chapter 3 represents a simplified useful model which is common on many applications. However, it is important to note that in MEMS sensors, accelerometer scale factors, non-orthogonalities and cross-axis sensitivities are important determinant errors that need to be identified due to its malicious effects in sensor fusion applications. Therefore, using the presented calibration method as a standalone method would not be sufficient for some applications. The results in Chapter 4 showed the importance of a good accelerometer scaling to identify the correct sensor biases. During the validation experiments, the accelerometer measures were correctly scaled and we assume that the algorithm is used this way. This assumption is plausible if the method is not used as standalone calibration. If the quality of the sensors used is limited and the scaling and misalignment effects are large, then the bias estimation would be inaccurate.

In Chapter 4, due to the lack of real boat data, the algorithm was validated by performing motions that simulated the boat motion. The assumptions made during this motion were that, the amplitude should not be more than 2 meters and the period of the motion is around 20 seconds. These assumptions are acceptable in moderate waters, although it is important to note that the case of an additional translational motion of the ship was not considered.

In Chapter 5, only the inclination part of the orientation was evaluated. This does not affect the main conclusions extracted in the previous section, since the orientation estimation filter relied on magnetometer inputs to correct the orientation, and the files were recorded in an indoor highly magnetically disturbed environment.

The method was developed to be applied in an *offline* setting. Thus, the measurements were recorded and the data was processed afterwards. Results in Chapter 5 were promising in the sense that showed that both algorithms would be nice candidates to be applied in an *online* setting as an extension to orientation estimation filters. However, the computational load of the algorithm should be then reduced and some changes on the cost functions should be performed.

## 6.3 Recommendations and further work

After carrying out this research, two main recommendations for future work are extracted. The first one, regards to a reduction in the computational load of the algorithm. The current implementation is made with Matlab and Python code, and is expensive in terms of computational time since it relies on



a numerical computation of the gradient for the optimization and pair-wise distance calculations (for the gyroscope optimization method). For *offline* implementation, the method also relies on loops that are usually slow in interpreted languages. Therefore, future lines of improvement are: a) Development of a fast efficient implementation that uses analytical expressions of the gradients for optimization, and b) implementation of the critical parts of the code in a faster language, like for example, C++.

The second future work proposed is related to an *online* implementation as an extension to an orientation estimation filter. To achieve this, a self-made optimization Gauss-Newton algorithm should be implemented so that the optimization is being carried in windows of a certain length  $T_w$ . Thus, the filter would recursively update sensor biases on a time-window basis through a gradient-based method, while the filter updates the orientation estimation. To achieve this, analytical gradient expressions should be obtained, and research on the optimization algorithm should be carried.

Last, an additional feature of the gyroscope based optimization would be to incorporate magnetometer measurements. Analogously to pair-wise distances of accelerations, the cost function could also use gyroscope strap-down integration to incorporate magnetometer disagreements, so that large drifts of the magnetic horizontal field component penalize the cost function. Thus, the effects of leaving the IMU over one single side discussed in the previous section, would be mitigated as the axis aligned with gravity would be observable by magnetometer disagreements. However, this would only be valid in outdoor environments where the magnetic field is not dynamically disturbed.



## Simulation results

Table A.1: Gyroscope optimization: Average (Maximum) error of gyroscope bias [ $^{\circ}$ /s]

$\tau$ [s]	$T_w$ [s]					
	5	10	15	20	25	30
0.1	0.07655 (0.1203)	0.08819 (0.1599)	0.07927 (0.1214)	0.1031 (0.1974)	0.06898 (0.1265)	0.1063 (0.207)
0.6	0.07712 (0.1109)	0.09074 (0.1752)	0.08946 (0.1218)	0.1164 (0.1846)	0.08688 (0.1487)	0.1099 (0.1805)
1.1	0.07116 (0.1104)	0.09739 (0.1734)	0.08876 (0.1493)	0.08488 (0.1543)	0.07959 (0.1546)	0.105 (0.1912)
1.6	0.06976 (0.109)	0.1074 (0.1528)	0.09107 (0.163)	0.1022 (0.1626)	0.0801 (0.1237)	0.1069 (0.1682)
2.1	0.09075 (0.1346)	0.1136 (0.1654)	0.0892 (0.1165)	0.1009 (0.1811)	0.08217 (0.1417)	0.1174 (0.2127)
2.6	0.08304 (0.1098)	0.1097 (0.1561)	0.09815 (0.1313)	0.1094 (0.1498)	0.09402 (0.1416)	0.1136 (0.1627)
3.1	0.09256 (0.1182)	0.112 (0.1825)	0.09816 (0.1616)	0.1089 (0.1689)	0.09632 (0.1383)	0.13 (0.2088)

Table A.2: Simultaneous optimization: Average (Maximum) error of gyroscope bias [ $^{\circ}$ /s]

$\tau$ [s]	$T_w$ [s]					
	5	10	15	20	25	30
0.1	0.0044 (0.008112)	0.003321 (0.005787)	0.003919 (0.006873)	0.00362 (0.006789)	0.003577 (0.00653)	0.003647 (0.007846)
0.6	0.004981 (0.007373)	0.004133 (0.008063)	0.003954 (0.007276)	0.003223 (0.005842)	0.003116 (0.006229)	0.003638 (0.007678)
1.1	0.005361 (0.008506)	0.003512 (0.006688)	0.003807 (0.00857)	0.004208 (0.00754)	0.003133 (0.005905)	0.004435 (0.009488)
1.6	0.005894 (0.009393)	0.003758 (0.006341)	0.004287 (0.005623)	0.003999 (0.009471)	0.003049 (0.005993)	0.004285 (0.007431)
2.1	0.006077 (0.01094)	0.00397 (0.006777)	0.004081 (0.00919)	0.003691 (0.006035)	0.003419 (0.005953)	0.003284 (0.005788)
2.6	0.006024 (0.009988)	0.003783 (0.008161)	0.004404 (0.007368)	0.00369 (0.007961)	0.004282 (0.007547)	0.00353 (0.007549)

Table A.3: Simultaneous optimization: Average (Maximum) error of accelerometer bias [m/s<sup>2</sup>]

$\tau$ [s]	$T_w$ (s)					
	5	10	15	20	25	30
0.1	0.008396 (0.0128)	0.005754 (0.008633)	0.005794 (0.007999)	0.006462 (0.009078)	0.006036 (0.009173)	0.007484 (0.01103)
0.6	0.01008 (0.01829)	0.005306 (0.01088)	0.005997 (0.008931)	0.005777 (0.00769)	0.005855 (0.01209)	0.007156 (0.01019)
1.1	0.01195 (0.01999)	0.006614 (0.01236)	0.006639 (0.009571)	0.00617 (0.01085)	0.00662 (0.0109)	0.008217 (0.01219)
1.6	0.0128 (0.02427)	0.006982 (0.01038)	0.006317 (0.009649)	0.006661 (0.008745)	0.006371 (0.01042)	0.007694 (0.009833)
2.1	0.013 (0.02496)	0.008425 (0.01336)	0.005899 (0.01315)	0.006202 (0.009307)	0.006502 (0.009052)	0.007246 (0.01037)
2.6	0.01331 (0.02182)	0.007699 (0.01474)	0.005912 (0.008547)	0.006492 (0.01165)	0.006591 (0.009212)	0.007933 (0.01013)

# References

- [1] F. Beckmann, “SearchWing - Search Area Calculations,” no. January, pp. 1–11, 2019.
- [2] P. Riseborough, “Inertial Navigation Library.”
- [3] UNHR, “Mediterranean situation,” 2019.
- [4] M. Bojadzic and S. Mezzadra, ““Refugee crisis” or crisis of European migration policies?,” *Focaal BLOG*, vol. 11, pp. 1–6, 2015.
- [5] N. Perkowski, “Deaths, Interventions, Humanitarianism and Human Rights in the Mediterranean ‘Migration Crisis’,” *Mediterranean Politics*, vol. 21, no. 2, pp. 331–335, 2016.
- [6] WatchThe Med - Alarm Phone, “From the sea to the city. 5 years of Alarm Phone,” tech. rep., 2019.
- [7] B. Bathke, “Space to the rescue: Non-profit Sea-Eye wants to detect migrant ships with satellite imagery - InfoMigrants.”
- [8] European Union Agency for Fundamental Rights, “Fundamental rights considerations : NGO ships involved in search and rescue in the Mediterranean and criminal investigations,” 2018.
- [9] IOM, *IAMSAR Manual*. 2001.
- [10] “Sea-Watch e.V. · Zivile Seenotrettung an Europas Grenzen.”
- [11] “Startseite - Sea-Eye.”
- [12] “RESQSHIP – Seenotrettung im Mittelmeer.”
- [13] “SearchWing Drone-Preliminary Datasheet,” tech. rep.
- [14] J. O. Nilsson and I. Skog, “Inertial sensor arrays - A literature review,” in *2016 European Navigation Conference, ENC 2016*, Institute of Electrical and Electronics Engineers Inc., aug 2016.
- [15] N. Ahmad, R. A. R. Ghazilla, N. M. Khairi, and V. Kasi, “Reviews on Various Inertial Measurement Unit (IMU) Sensor Applications,” *International Journal of Signal Processing Systems*, vol. 1, no. 2, pp. 256–262, 2013.
- [16] J. Luis Marins, X. Yun, E. R. Bachmann, R. B. McGhee, and M. J. Zyda, “An Extended Kalman Filter for Quaternion-Based Orientation Estimation Using MARG Sensors,” *International Conference on Intelligent Robots and Systems*, 2001.
- [17] T. Seel and S. Rupp, “Eliminating the Effect of Magnetic Disturbances on the Inclination Estimates of Inertial Sensors,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 8798–8803, 2017.
- [18] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, “Estimation of IMU and MARG orientation using a gradient descent algorithm,” 2011.
- [19] G. A. Pavlath, “Fiber optic gyros past, present, and future,” in *OFS2012 22nd International Conference on Optical Fiber Sensors* (Y. Liao, W. Jin, D. D. Sampson, R. Yamauchi, Y. Chung, K. Nakamura, and Y. Rao, eds.), vol. 8421, pp. 53–62, International Society for Optics and Photonics, SPIE, 2012.

- [20] Z. F. Syed, P. Aggarwal, C. Goodall, X. Niu, and N. El-Sheimy, "A new multi-position calibration method for MEMS inertial navigation systems," *Measurement Science and Technology*, vol. 18, no. 7, pp. 1897–1907, 2007.
- [21] Q. Cai, N. Song, G. Yang, and Y. Liu, "Accelerometer calibration with nonlinear scale factor based on multi-position observation," *Measurement Science and Technology*, vol. 24, no. 10, 2013.
- [22] S. Dhalwar, R. Kottath, V. Kumar, A. N. J. Raj, and S. Poddar, "Adaptive parameter based Particle Swarm Optimisation for accelerometer calibration," in *1st IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems, ICPEICES 2016*, Institute of Electrical and Electronics Engineers Inc., feb 2017.
- [23] C. S. Miranda and J. V. Ferreira, "Efficient Simultaneous Calibration of a Magnetometer and an Accelerometer," may 2015.
- [24] M. Kok and T. B. Schön, "Maximum likelihood calibration of a magnetometer using inertial sensors," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 19, pp. 92–97, 2014.
- [25] J. C. Chow, J. D. Hol, and H. Luinge, "Tightly-Coupled Joint User Self-Calibration of Accelerometers, Gyroscopes, and Magnetometers," *Drones*, vol. 2, no. 1, p. 20, 2018.
- [26] X. Cui, Y. Li, Q. Wang, M. Zhang, and J. Li, "Three-axis magnetometer calibration based on optimal ellipsoidal fitting under constraint condition for pedestrian positioning system using foot-mounted inertial sensor/magnetometer," *2018 IEEE/ION Position, Location and Navigation Symposium, PLANS 2018 - Proceedings*, pp. 166–174, 2018.
- [27] A. A. Hussen and I. N. Jleta, "Low-Cost Inertial Sensors Modeling Using Allan Variance," *International Scholarly and Scientific Research & Innovation*, vol. 9, no. 5, pp. 1069–1074, 2015.
- [28] X.-J. Niu and Z.-y. Gao, "Satellite TV Antenna Attitude Stabilization System Based on Micro-machined Inertial Sensors," *Journal of Chinese Inertial Technology*, 2002.
- [29] M. El-Diasty and S. Pagiatakis, "Calibration and Stochastic Modelling of Inertial Navigation Sensor Errors," *Journal of Global Positioning Systems*, vol. 7, no. 2, pp. 170–182, 2008.
- [30] E.-H. Shin and N. El-Sheimy, "New system calibration method for strapdown inertial navigation system," *Z. vermess*, vol. 38, no. 4, pp. 459–463, 2002.
- [31] J. K. Bekkeng, "Calibration of a novel MEMS inertial reference unit," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 6, pp. 1967–1974, 2009.
- [32] T. Nieminen, J. Kangas, S. Suuriniemi, and L. Kettunen, "An enhanced multi-position calibration method for consumer-grade inertial measurement units applied and tested," *Measurement Science and Technology*, vol. 21, no. 10, 2010.
- [33] L. Wang and F. Wang, "Intelligent Calibration Method of low cost MEMS Inertial Measurement Unit for an FPGA-based Navigation System," tech. rep.
- [34] A. Radi, G. Bakalli, N. E.-S. . . . M. of the . . . , and U. 2017, "An automatic calibration approach for the stochastic parameters of inertial sensors," *pennstate.pure.elsevier.com*.
- [35] A. Umeda, M. Onoe, K. Sakata, T. Fukushima, K. Kanari, H. Iioka, and T. Kobayashi, "Calibration of three-axis accelerometers using a three-dimensional vibration generator and three laser interferometers," *Sensors and Actuators, A: Physical*, vol. 114, pp. 93–101, aug 2004.
- [36] K.-Y. Choi, S.-A. Jang, and Y.-H. Kim, "Calibration of Inertial Measurement Units Using Pendulum Motion," *International Journal of Aeronautical and Space Sciences*, vol. 11, pp. 234–239, sep 2010.
- [37] E. L. Renk, W. Collins, M. Rizzo, F. Lee, and D. S. Bernstein, "Optimization-based calibration of a triaxial accelerometer-magnetometer," *Proceedings of the American Control Conference*, vol. 3, pp. 1957–1962, 2005.
- [38] F. Ferraris, I. Gorini, U. Grimaldi, and M. Parvis, "Calibration of three-axial rate gyros without angular velocity standards," *Sensors and Actuators, A: Physical*, vol. 42, no. 1-3, pp. 446–449, 1994.

- [39] E. R. Bachmann, X. Yun, D. McKinney, R. B. McGhee, and M. J. Zyda, "Design and implementation of MARG sensors for 3-DOF orientation measurement of rigid bodies," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 1, pp. 1171–1178, 2003.
- [40] J. C. Lötters, J. Schipper, P. H. Veltink, W. Olthuis, and P. Bergveld, "Procedure for in-use calibration of triaxial accelerometers in medical applications," *Sensors and Actuators, A: Physical*, vol. 68, no. 1-3, pp. 221–228, 1998.
- [41] I. Skog and P. Händel, "Calibration of an Inertial Measurement Unit," in *XVII IMEKO WORLD CONGRESS*, vol. 53, pp. 228–236, 2006.
- [42] D. Jurman, M. Jankovec, R. Kamnik, and M. Topič, "Calibration and data fusion solution for the miniature attitude and heading reference system," *Sensors and Actuators, A: Physical*, vol. 138, pp. 411–420, aug 2007.
- [43] W. T. Fong, S. K. Ong, and A. Y. Nee, "Methods for in-field user calibration of an inertial measurement unit without external equipment," *Measurement Science and Technology*, vol. 19, no. 8, 2008.
- [44] D. Tedaldi, A. Pretto, and E. Menegatti, "A robust and easy to implement method for IMU calibration without external equipments," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3042–3049, 2014.
- [45] U. Qureshi, F. G. I. S. Journal, and U. 2017, "An algorithm for the in-field calibration of a MEMS IMU," *ieeexplore.ieee.org*, 2017.
- [46] S. Ranjbaran, A. Roudbari, and S. Ebadollahi, "A Simple and Fast Method for Field Calibration of Triaxial Gyroscope by Using Accelerometer," vol. 6, no. 1, pp. 1–6, 2018.
- [47] M. Glueck, D. Oshinubi, P. Schopp, and Y. Manoli, "Autocalibration of MEMS Accelerometers," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 1, pp. 96–105, 2009.
- [48] X. Li and Z. Li, "A new calibration method for tri-axial field sensors in strap-down navigation systems," *Measurement Science and Technology*, vol. 23, no. 10, 2012.
- [49] T. Forsberg, N. Grip, and N. Sabourova, "Non-iterative calibration for accelerometers with three non-orthogonal axes, reliable measurement setups and simple supplementary equipment," *Measurement Science and Technology*, vol. 24, no. 3, 2013.
- [50] Q. Cai, N. Song, G. Yang, and Y. Liu, "Accelerometer calibration with nonlinear scale factor based on multi-position observation," *Measurement Science and Technology*, vol. 24, p. 105002, aug 2013.
- [51] C. M. Cheuk, T. K. Lau, K. W. Lin, and Y. Liu, "Automatic calibration for inertial measurement unit," *2012 12th International Conference on Control, Automation, Robotics and Vision, ICARCV 2012*, vol. 2012, no. December, pp. 1341–1346, 2012.
- [52] V. M. Tereshkov, "An intuitive approach to inertial sensor bias estimation," *International Journal of Navigation and Observation*, vol. 2013, 2013.
- [53] A. R. Spielvogel and L. L. Whitcomb, "Adaptive Sensor Bias Estimation in Nine Degree of Freedom Inertial Measurement Units: Theory and Preliminary Evaluation," *IEEE International Conference on Intelligent Robots and Systems*, pp. 5555–5561, 2018.
- [54] F. Girrbach, R. Zandbergen, M. Kok, T. Hageman, G. Bellusci, and M. Diehl, "Towards In-Field and Online Calibration of Inertial Navigation Systems using Moving Horizon Estimation," *2019 18th European Control Conference (ECC)*, pp. 4338–4343, 2019.
- [55] Y. Li, X. Niu, Q. Zhang, H. Zhang, and C. Shi, "An in situ hand calibration method using a pseudo-observation scheme for low-end inertial measurement units," *Measurement Science and Technology*, vol. 23, no. 10, 2012.
- [56] J. B. Bancroft and G. Lachapelle, "Estimating MEMS gyroscope g-sensitivity errors in foot mounted navigation," *2012 Ubiquitous Positioning, Indoor Navigation, and Location Based Service, UPINLBS 2012*, no. February, 2012.

- 
- [57] M. J. Jørgensen, D. Paccagnan, N. K. Poulsen, and M. B. Larsen, “IMU calibration and validation in a factory, remote on land and at sea,” *Record - IEEE PLANS, Position Location and Navigation Symposium*, pp. 1384–1391, 2014.
  - [58] M. George and S. Sukkarieh, “Tightly coupled INS/GPS with bias estimation for UAV applications,” *Proceedings of the 2005 Australasian Conference on Robotics and Automation, ACRA 2005*, 2005.
  - [59] V. R. Marco, J. Kalkkuhl, and J. Raisch, *EKF for simultaneous vehicle motion estimation and IMU bias calibration with observability-based adaptation*. 2018.
  - [60] F. Gustafsson, “Determining the initial states in forward-backward filtering,” *IEEE Transactions on Signal Processing*, vol. 44, no. 4, pp. 988–992, 1996.
  - [61] K. Brands, *Development and Validation of Long-Time Stable Magnetometer-free Inertial Lower-Body Motion Tracking*. 2020.