DM

Development of MEMS-based IMU for Position Estimation
Comparison of Sensor Fusion solutions

MASTER DISSERTATION

Cláudio Duarte Rodrigues
MASTER IN INFORMATICS ENGINEERING

UNIVERSIDADE da MADEIRA
A Nossa Universidade
www.uma.pt

December | 2021

Development of MEMS-based IMU for Position Estimation
Comparison of Sensor Fusion solutions

MASTER DISSERTATION

Cláudio Duarte Rodrigues
MASTER IN INFORMATICS ENGINEERING

ORIENTATION
Marko Radeta

CO-ORIENTATION
Filipe Magno Gouveia Quintal

UNIVERSIDADE da MADEIRA

A Nossa Universidade

www.uma.pt

FCEE

Mestrado em Engenharia Informática

Development of MEMS-based IMU for Position Estimation: Comparison of Sensor Fusion solutions

Cláudio Rodrigues

supervisionado por

Marko Radeta and Filipe Quintal

19 de abril de 2022

Abstract

With the surge of inexpensive, widely accessible, and precise Micro-Electro-Mechanical Systems (MEMS) in recent years, inertial systems tracking movement have become ubiquitous nowadays. Contrary to Global Positioning System (GPS)-based positioning, Inertial Navigation System (INS) are intrinsically unaffected by signal jamming, blockage susceptibilities, and spoofing. Measurements from inertial sensors are also acquired at elevated sampling rates and may be numerically integrated to estimate position and orientation knowledge. These measurements are precise on a small-time scale but gradually accumulate errors over extended periods. Combining multiple inertial sensors in a method known as sensor fusion makes it possible to produce a more consistent and dependable understanding of the system, decreasing accumulative errors. Several sensor fusion algorithms occur in literature aimed at estimating the Attitude and Heading Reference System (AHRS) of a rigid body with respect to a reference frame. This work describes the development and implementation of a low-cost, multi-purpose INS for position and orientation estimation. Additionally, it presents an experimental comparison of a series of sensor fusion solutions and benchmarking their performance on estimating the position of a moving object. Results show a correlation between what sensors are trusted by the algorithm and how well it performed at estimating position. Mahony, SAAM and Tilt algorithms had best general position estimate performance.

Keywords: Sensor Fusion · Inertial Navigation System (INS) · Micro-Electro-Mechanical Systems (MEMS) · Inertial Measurement Unit (IMU) · Attitude and Heading Reference System (AHRS)

Resumo

Com o recente surgimento de sistemas micro-eletromecânico amplamente acessíveis e precisos nos últimos anos, o rastreio de movimento através de sistemas de inerciais tornou-se omnipresente nos dias de hoje. Contrariamente à localização baseada no Sistema de Posicionamento Global (GPS), os Sistemas de Navegação Inercial (SNI) não são afetados intrinsecamente pela interferência de sinal, suscetibilidades de bloqueio e falsificação. As medições dos sensores inerciais também são adquiridas a elevadas taxas de amostragem e podem ser integradas numericamente para estimar os conhecimentos de posição e orientação. Estas medições são precisas numa escala de pequena dimensão, mas acumulam gradualmente erros durante longos períodos. Combinar múltiplos sensores inerciais num método conhecido como fusão de sensores permite produzir uma mais consistente e confiável compreensão do sistema, diminuindo erros acumulativos. Vários algoritmos de fusão de sensores ocorrem na literatura com o objetivo de estimar os Sistemas de Referência de Atitude e Rumo (SRAR) de um corpo rígido no que diz respeito a uma estrutura de referência. Este trabalho descreve o desenvolvimento e implementação de um sistema multiusos de baixo custo para estimativa de posição e orientação. Além disso, apresenta uma comparação experimental de uma série de soluções de fusão de sensores e compara o seu desempenho na estimativa da posição de um objeto em movimento. Os resultados mostram uma correlação entre os sensores que são confiados pelo algoritmo e o quão bem ele desempenhou na posição estimada. Os algoritmos Mahony, SAAM e Tilt tiveram o melhor desempenho da estimativa da posição geral.

Keywords: Fusão de Sensores · Sistemas de navegação Inercial · Sistemas Microeletromecânicos · Unidade de Medição Inercial · Sistemas de Referência de Atutude e Rumo

Acknowledgements

I would like first to send a special feeling of gratitude to my loving parents, Ivo and Manuela Rodrigues, to my brother and best friend André, who have been a constant source of support and understanding, teaching me the value of hard work and to aspire for great things. Secondly, I want to manifest sincere gratitude to my supervisor, and above all my friend, Professor Marko Radeta, for all the assistance and inspiration, passing me his vast knowledge and experience. I would also like to thank all my friends, family and loved ones who have been a vital part of my life over the past year, this would not be possible without you. Lastly, I would like to dedicate this dissertation to my deceased grandmother Maria Conceição de Sousa who passed away in the making of this work and had been a great source of happiness and comfort in my life. This is in your homage.

Cláudio Rodrigues

Table of Contents

List of Figures **8**

List of Tables **12**

1 Introduction 1

1.1 Problem Statement 2

1.1.1 A. GNSS only positioning is prone to signal jamming and impacts battery autonomy 2

1.1.2 B. Dead reckoning is susceptible to cumulative errors and suffers from gimbal lock 2

1.1.3 C. Gravity acceleration greatly impacts sensor readings 3

1.2 Research Questions 4

1.3 Thesis Contribution 5

1.4 Thesis Outline 5

2 Background 6

2.1 Frames of coordinates 6

2.1.1 ECEF and ENU frame 6

2.1.2 Body frame 8

2.2 Orientation 10

2.2.1 Rotation Matrix 10

2.2.2 Direct Cosine Matrix 11

2.2.3 Euler angles 12

2.2.4 Quaternions 13

2.3 Inertial Measurement Units 17

2.3.1 Accelerometer 18

2.3.2 Gyroscope 20

2.3.3 Magnetometer 22

2.4 Sensor Fusion 23

2.4.1 Kalman Filter 24

2.4.2 Complementary Filter 25

2.4.3 Optimization Algorithm 27

2.4.4 Other Algorithms 28

3 Related Work 31
3.1 Position estimation using inertial sensor systems 31
3.1.1 Pedestrian Dead Reckoning 31
3.1.2 Strapdown Inertial Integration 32
3.2 Sensor fusion in position and orientation estimation . . . 33
3.2.1 Kalman filter 33
3.2.2 Complementary filter 34
3.2.3 Sensor fusion algorithms comparison 35
4 Methodology 37
4.0.1 Estimating Orientation 37
4.0.2 Estimating Displacement 40
4.0.3 Estimating Position 42
4.0.4 Applying methodology in 2D 45
4.0.4.1 Measuring error 45
4.0.5 Applying methodology in 3D 47
4.0.5.1 Measuring error 47
5 Experimental Setup 50
5.1 Hardware 50
5.1.1 Accelerometer 52
5.1.2 Gyroscope 52
5.1.3 Magnetometer 53
5.2 Software 54
5.3 Calibration 55
5.3.1 Accelerometer Calibration 55
5.3.2 Gyroscope Calibration 57
5.3.3 Magnetometer Calibration 59
5.4 Experiments 62
5.5 Indoor Experiments 62
5.6 Outdoor Experiments 64
5.7 Underwater Experiments 67
6 Results 69
6.1 Line 69
6.1.1 4 meter 69
6.1.2 16 meter 71
6.1.3 28 meter 72
6.2 Triangle 73

6.2.1 4 meter 73
6.2.2 16 meter 74
6.2.3 28 meter 75
6.3 Square 76
6.3.1 4 meter 76
6.3.2 16 meter 77
6.3.3 28 meter 78
6.4 Spiral 79
6.5 Squares 80
7 Discussion 81
7.1 Discussion of Results 81
7.2 Comparing with Related Work 83
7.3 Challenges Faced 84
7.4 Future Work 85
8 Conclusion 86

References **87**

List of Figures

1 An illustrative diagram for the WGS84, ECEF, and ENU coordinate systems for the Earth and their transformation correlation (PM line is the Prime Meridian; λ and ϕ are latitude and longitude in WGS84; X, Y, Z for ECEF; and E, N, U for ENU). 7

2 Navigation and body frame coordinate systems. 9

3 Frame axis example. 10

4 3D rotation around vector $\hat{r}$ of angle θ between frame A and B. 11

5 Representation of the Gimbal Lock problematic [1] - The exterior blue gimbal characterizes the x-axis, the middle, red-colored gimbal the y-axis, and the inner green gimbal the z-axis. In the initial arrangement a), every axis is perpendicular to one another. Following a rotation of 90○ across the red arrow (y-axis), the blue and the green gimbals occupy the same rotation axis. This condition inhibits the clear determination of the rotation axes when subsequently rotating around the x or z-axis. 13

6 90-degree rotation in a 2-dimensional coordinate frame with a real x-axis and an imaginary y-axis. 14

7 Three main types of sensor error. 18

8 Schematic demonstrating the working principle of an accelerometer. [2] 19

9 Representation of a conventional gyroscope accommodating a spinning wheel mounted on two gimbals. [3] 20

10 System where two masses are oscillating in opposite directions. When a rotation is applied, the masses are affected by the Coriolis force and the displacement is measured by a change in capacitance. [4] 21

11 Representation of a magnetometer's different types of disturbances. [5] 24

12 Kalman-filter-based multi-sensor state-vector fusion. [6] 25

13 Kalman-filter-based multi-sensor measurement fusion. [6] 25

14 Basic complementary filter [7] - Two different measurement sources for estimating one variable. The noise properties of the two measurements are such that one source gives good information only in low frequency region while the other is good only in high frequency region. 26
15 Raw sensor data on each axis obtained by the accelerometer, gyroscope, and magnetometer. 37
16 Madgwick's algorithm sensor fusion output of the raw measurement data of figure 15. 38
17 Madgwick's algorithm sensor fusion output of the raw measurement data of 15 converted into Euler angles. 39
18 Comparison between raw acceleration (blue) and measurements with gravity compensation (red). When tilting the sensor, the accelerometer will measure gravity's acceleration (around 9.8 ms^{-2}). The gravity compensation filter can counteract virtually all the gravitational influence on the sensor readings. 41
19 Overview of the position estimation methodology. 42
20 Plot showing how acceleration is integrated into velocity and displacement. 43
21 Illustration of a unit circle where θ is the attitude angle. 43
22 Illustration of the dead reckoning approach. The object's position at a given moment is given by the last known position, the heading angle $\Delta\theta$, and the travelled distance Δx that occur during the last sample. 44
23 Result after applying the position estimation methodology to the raw sensor data. 45
24 Illustration of the real-world test ground truth (represented by the $4m \times 4m$ blue line square) and how the estimated path compares. Orange dots represent the estimated corners of the RDP Algorithm. . 46
26 Three-dimensional view of the experiment. 47
27 The same methodology applies to three-dimensional estimation. The real-world test ground truth (represented by the $4m \times 4m$ blue line square) and how the estimated path compares. Orange dots represent the estimated corners of the RDP Algorithm. 48
29 Pin connection between the microcontroller (left) and the IMU (right). Modules are linked through SCL, CLK, VDD and GND pins. . 51

30 Coordinate system of an MPU-6050 (3-axis accelerometer and 3-axis gyroscope combination) and AK8963 3-axis magnetometer. [8] [9] 51
31 I^2C protocol. 54
32 Accelerometer calibration offset correction. 57
33 Gyroscope calibration offset correction. 58
34 Raw magnetometer measurements. 60
35 Raw and calibrated magnetometer measurements. 61
36 4-meter side ground square used for baseline of accuracy of inertial system. 62
37 Skateboard being dragged along the guideline. 63
38 Platform where outdoors experiments were conducted. 64
39 Bird's-eye view representation of the 7x7 grid platform where outdoor experiments were conducted. 64
40 Photos illustrating how outdoor experiments were performed. 65
41 Bird's-eye view representation of the more complex path shapes taken. 66
42 View of the tests made to assure the underwater housing was waterproof. 67
43 Photos of the underwater tests conducted at Madeira Carlton hotel divepoint. 68
44 Position estimation by the best performing algorithms in the 4-meter line experiment. 70
45 Position estimation by the best performing algorithms in the 16-meter line experiment. 71
46 Position estimation by the best performing algorithms in the 28-meter line experiment. 72
47 Position estimation by the best performing algorithms in the 4-meter side triangle experiment. 73
48 Position estimation by the best performing algorithms in the 16-meter side triangle experiment. 74
49 Position estimation by the best performing algorithms in the 28-meter side triangle experiment. 75
50 Position estimation by the best performing algorithms in the 4-meter side square experiment. 76
51 Position estimation by the best performing algorithms in the 16-meter side square experiment. 77

52 Position estimation by the best performing algorithms in the 28-meter side square experiment. 78
53 Position estimation by the best performing algorithms in the spiral experiment. 79
54 Position estimation by the best performing algorithms in the squares experiment. 80

List of Tables

2 WGS 84 needed parameters to convert ECEF coordinates into ENU. [10] 8
3 Quaternion multiplication of basis elements. 15
4 Accelerometer Specifications. 52
5 Gyroscope Specifications. 53
6 Magnetometer specification. 53
7 Available sensor fusion algorithms in AHRS library and what sensors they use. [11] 56
8 4 meter line position estimation error (displacement and turn) of the sensor fusion algorithms. 69
9 16 meter line position estimation error (displacement and turn) of the sensor fusion algorithms. 71
10 28 meter line position estimation error (displacement and turn) of the sensor fusion algorithms. 72
11 4 meter triangle position estimation error (displacement and turn) of the sensor fusion algorithms. 73
12 16 meter triangle position estimation error (displacement and turn) of the sensor fusion algorithms. 74
13 28 meter square position estimation error (displacement and turn) of the sensor fusion algorithms. 75
14 4 meter square position estimation error (displacement and turn) of the sensor fusion algorithms. 76
15 16 meter square position estimation error (displacement and turn) of the sensor fusion algorithms. 77
16 28 meter square position estimation error (displacement and turn) of the sensor fusion algorithms. 78
17 Spiral position estimation error (displacement and turn) of the sensor fusion algorithms. 79
18 Squares position estimation error (displacement and turn) of the sensor fusion algorithms. 80
19 Overall ranking of how every algorithm performed in the given test. . 81

20 Algorithms ranked according to of how well performed at displacement error. 82
21 Algorithms ranked according to of how well performed at turn error. . 83

Abbreviations

I^2C	Inter-Integrated Circuit.
AHRS	Attitude and Heading Reference System.
AMC	Accelerometer-Magnetometer combination.
AOA	Angle of Arrival.
AQUA	Algebraic Quaternion Algorithm.
CPU	Central Processing Unit.
DCM	Direct Cosine Matrix.
ECEF	Earth Centered, Earth Fixed.
ECF	Extended Complementary Filter.
EKF	Extended Kalman Filter.
ENU	East North Up.
FAMC	Fast Accelerometer-Magnetometer Combination.
FLAE	Fast Linear Attitude Estimator.
GLONASS	Globalnaya Navigatsionnaya Sputnikovaya Sistema.
GNSS	Global Navigation Satellite System.
GPS	Global Positioning System.
IMU	Inertial Measurement Unit.
INS	Inertial Navigation System.
IoT	Internet of Things.
LMA	Levenberg Marquardt Algorithm.
LoRa	Long Range Communication.
MAE	Mean Average Error.
MARG	Magnetic, Angular Rate, and Gravity.
MEMS	Micro-Electro-Mechanical Systems.
NED	North East Down.
OLEQ	Optimal Linear Estimator of Quaternion.
PDR	Pedestrian dead reckoning.
QUEST	QUaternion ESTimator.
RDP	Ramer-Douglas-Peucke.

ROLEQ	Recursive Optimal Linear Estimator of Quaternion.
ROV	Remotely Operated Underwater Vehicle.
RSS	Received Signal Strength.
SAAM	Super-fast Attitude from Accelerometer and Magnetometer.
SCL	Serial Clock.
SDA	Serial Data.
SIP	System-in-package.
SNIS	Strapdown Inertial Navigation System.
SNR	Signal to Noise Ratio.
TOA	Time of Arrival.
UAV	Unmanned Aerial Vehicle.
WGS84	World Geodetic System.
ZVU	Zero-Velocity Update.

1 Introduction

Navigation systems have become a popular subject in the designing of unmanned and autonomous systems in recent years. Still, most navigation systems highly depend on the Global Navigation Satellite System (GNSS) receivers as the central resource of navigation data. Although the satellite system can deliver accurate and long-term positioning in open spaces, when relying on GNSS only localization, there are often circumstances when the satellite signal is obstructed or weakened, resulting in degradation or even loss of position estimate precision [12]. Such is especially threatening in high urbanized centers where satellite signals can suffer multipath propagation from tall glass-covered buildings [13]. Satellite positioning is especially impactful in battery autonomy, whereas GNSS receivers remain draining a substantial amount of electric current. Inevitably, there is a need to become less dependent on GNSS-based localization, particularly in autonomous settings. Substantial research has been conducted to enhance the localization precision of an object devoid of satellite signals [14] [15] [16] [17] [18]. Increased accuracy is a precursor to establishing autonomous agents for a diversity of functions. Inertial Measurement Unit (IMU) have become standard in embedded inertial systems due to their low cost, lightweight, and low power consumption. They can provide short-term position and orientation changes. Furthermore, inertial systems have been employed in wearable applications with uses in Unmanned Aerial Vehicle (UAV) [19] [20] [21], telemedicine [22], and robotics [23]. Despite these accomplishments, inertial systems suffer from rapid drift owing to the existence of disturbances and noise in the measurements. Practically all present commercial applications are restricted to minimal motion recognition. Position and orientation in real-world applications are rarely employed due to difficulties in precise integration. Recently, innovative fusion algorithms have emerged, which can diminish the impacts of noise and disturbances, broadening these devices' capabilities. There is an increasing need, which remains unmet, for inertial orientation and position applications since they are key to automation and the Internet of Things (IoT). We propose a low-cost, multipurpose inertial solution with modules for an Inertial Measurement Unit (IMU) that may support maintaining high levels of orientation and location exactness even when satellite-based location is not possible. Focusing on how different sensor fusion algorithms perform in assessing a body's position in space.

1.1 Problem Statement

1.1.1 A. GNSS only positioning is prone to signal jamming and impacts battery autonomy

Modern navigation relies greatly on the Global Navigation Satellite System (GNSS) constellations (such as Global Positioning System (GPS), Galileo, GLONASS, etc.); being straightforward to operate, accurate and trustworthy, it is widely employed in navigation systems. Nevertheless, GNSS signals still encounter numerous vulnerabilities and can often be compromised by natural and human sources [12]. Attacks against GNSS-based localization are becoming more frequent and of intensifying damage [24]. Satellite signals can also be affected by abnormal activity due to solar winds, creating temporary gaps in coverage [25]. Moreover, while GNSS offers seamless navigation with inexpensive receivers, they are also prone to signal jamming, where satellites are unable to detect the objects. Repeatedly, position accuracy is reduced or even lost, such as in tunnels or underground sections [26] [13]. Furthermore, GNSS cannot accurately determine altitude with the necessary exactness, which is essential to accurately depict a body in three-dimensional space. Lastly, localization technologies demand high processing capacity and communication costs. This is particularly impactful in autonomous settings, where battery autonomy is crucial, while GNSS receivers continue to drain a large amount of electric current [27]. Consequently, power optimization is critical, and there is a necessity to become less reliant on GNSS-based localization.

1.1.2 B. Dead reckoning is susceptible to cumulative errors and suffers from gimbal lock

When understanding the alternatives to GNSS for estimating the position, the dead reckoning technique is often employed to resolve the location of a moving object. Using sensor data (gyroscope, accelerometer, magnetometer, etc.), it is possible to assess current position even when GNSS positioning is not possible [13]. It has been recognized as a low-power alternative to GNSS localization that can deliver high-resolution position data [15]. It is possible to estimate the current position from an obtained distance (which may be estimated from velocity), the known starting point, and estimated drift. However, the precision of the dead-reckoning approach is continuously worsening while measurement errors accumulate during the current position estimation [16]. The approach is also embedded sensorial fusion techniques (will be explored in Section 2.4 Sensor

Fusion), which mathematically merge a series of navigation solutions to obtain the best estimate of the navigator's current position, velocity, attitude angles, etc [28].

Still, precise tracking of a moving and rotating body in three-dimensional space implies a superior degree of complexity (compared to two-dimensional tracking). It can be accomplished in a range of approaches. Most commonly, the orientation of bodies that move in three dimensions may be explained by a combination of their angle of rotation across each of their three axes (e.g., such as in trigonometry, where Euler angles can approximate yaw, roll, and pitch). Fundamentally, a specific movement could be defined by multiple rotations. In such case, to perform a rotation over a particular axis, rotational matrices, vector operations, and trigonometric functions are required [29]. This involves numerous complex mathematical operations and several clock cycles in a microprocessor that could negatively impact computational performance [30].

Nevertheless, the rotation axes are not always independent, and results are not necessarily distinctive or unique. This further implies that the plane of two gimbals (rotational axes) to align, which causes the recognized gimbal lock phenomenon, where two out of three gimbals are parallel or very nearly parallel, reducing the output to two degrees of freedom [31]. When gimbal lock happens, it is not possible to reorientate the axes without an external reference (a more in-depth analysis will be given at 2.2 Orientation). Alternatives using quaternions may be used, while most studies apply them in microcomputers rather than on microcontrollers. Quaternions give the capability to define any three-dimensional rotation around an axis distinctively and are not susceptible to gimbal lock [32].

1.1.3 C. Gravity acceleration greatly impacts sensor readings

An accelerometer is generally utilized to estimate the velocity and position of a given body devoid of the usage of GNSS. These electromechanical devices can measure proper acceleration forces, which can be employed to determine a body's velocity and position relative to a starting point. In theory, this can be done by integrating the resultant of acceleration yielding velocity; double integrating will deliver the body's accumulative position [33]. In practice, these measurements are influenced by Earth's gravitational field and rotational components of acceleration, significantly magnifying numerical errors during the readings [34]. The gravity component will not be differentiated from the physical acceleration of the device and will eventually generate exceedingly elevated errors in the measured

acceleration. Double integrating these measurements will inherently amplify errors which will accumulate exponentially, translating into a yet greater offset in velocity and position estimates [35].

A potential solution to minimize this difficulty is to assume the body moves on a flat surface, thus greatly diminishing the influence of the gravity component on acceleration readings. Every inaccuracy that may arise from hiring such supposition can be regarded as noise and is smoothly filtered [34]. Understandably, such an assumption is not possible with a body moving through three-dimensional space since it is subject to many kinds of forces and movements. A numerical process is required for handling accelerometer measurements that utterly removes the effect of the gravity component and further undesirable acceleration vectors.

1.2 Research Questions

While research has been performed towards orientation estimation by the fusion of multiple sensors, few studies sought to assess position due to difficulties removing the measurement of normal forces that do not cause physical acceleration of the sensor. This investigation seeks to comprehend how different sensor fusion approaches perform in approximating the orientation of a moving system, along with how to overcome the positioning challenge through a technique that combines orientation estimation to filter non-physical acceleration.

- **[RQ1]. Estimate** - How to perform a low-cost orientation and position estimation of a moving object devoid of GNSS?
  The study will emphasize on the possibility of estimating the orientation and position of a moving object in three-dimensional space short of GNSS-based positioning by combining multiple low-cost sensors to provide an object's navigation information.

- **[RQ2]. Comparing** - How distinct fusion techniques perform in orientation and position estimation?
  An experimental comparison between various sensor fusion algorithms is used to evaluate how accurate each approach can approximate orientation and position.

1.3 Thesis Contribution

With this study, we aim to design and build a low-cost, multipurpose Inertial Navigation System, intending to estimate the orientation and position of a moving object in three-dimensional space. This research additionally proposes introducing an experimental comparison among several established AHRS sensor fusion algorithms such as the EKF, Madgwick, and Mahony algorithms. Furthermore, a quaternion-based gravity compensation filter will be presented, diminishing the influence of the gravity component on acceleration readings.

1.4 Thesis Outline

This section will describe this document's organization and structure. First, there will be an introductory section to some background knowledge regarding different concepts and notions that will be adopted throughout the dissertation. Building a foundation of understanding wherein the mathematics and arithmetic behind attitude representation. An introduction to inertial sensors and how they can be employed to estimate orientation using sensor fusion. This is followed by a related work section, which will narrow the existing literature within this research's scope, examining the methods each author used in their work and interpreting key insights and how each source contributes to the existing pool of knowledge on its subject. A position estimation methodology is then presented providing an overview of the process employed to assess the system's orientation and position. From the raw sensor measurements to final position output, and every step in the method. Next, a hardware and software experimental setup outline, describing and specifying development and implementation of the low-cost Inertial Navigation System (INS) aimed at position estimation in three-dimensional space describing the hardware and software solution. Later in the section, the experimental tests are depicted, describing how they were conducted to assess the position performance of the INS under different conditions. The results from the experiments are then presented followed by a discussion of results drawing conclusions and comparing with the existent work. This section will also illustrate some challenges faced and future work. Lastly, a conclusion chapter will finalize restating the work done, summarize the key supporting ideas discussed throughout the work, and offering some final impressions wrapping final thoughts and main points.

2 Background

The subsequent section presents background knowledge regarding different concepts and notions that will be adopted throughout the dissertation. First, a definition of orientation frames and coordinate systems will be presented, followed by an introduction to rotation matrices, Euler angles, and quaternions, building a foundation of understanding wherein the mathematics and arithmetic behind attitude representation. An introduction to inertial sensors and how they can be employed to estimate orientation is then exhibited. The chapter concludes with an analysis and summary of different sensor fusion algorithms utilized to determine orientation.

2.1 Frames of coordinates

This section will focus on defining and distinguishing the different concepts of frame coordinate systems. An emphasis will be given to East North Up (ENU), Earth Centered, Earth Fixed (ECEF), and the World Geodetic System (WGS84). The notion of body frame will also be defined. The ECEF and WGS84 can be considered supplementary frame systems applied to describe the ENU frame, which can be understood as the world frame.

2.1.1 ECEF and ENU frame

ECEF coordinate system describes a referential axis where the origin of the coordinates is at the center of mass of the Earth, also known as barycenter [36]. Mathematically, this translates to the integral of the position vector times the density over the Earth being zero (equation 1) [37].

$$\int \overrightarrow{x} \rho \, dx^3 = 0 \tag{1}$$

The X-axis is described by the intersection of the zero-latitude line (Equator) plan and the zero-longitude line (prime meridian) plan. The orientation of the X-axis is deemed to be positive from the center towards the point defined by zero latitude and longitude. Z-axis is expressed by the line interconnecting North and South Poles, staying positive in the Earth's barycenter to the North Pole. Y-axis lies in the equatorial plane and is perpendicular to the plane described by the X and Z-axis, and it is in a positive direction (figure 1). The right-hand rule explains its orientation.

The ENU system is a geographic coordinate structure where the origin is placed at an empirical point in the ECEF coordinate system. In the ENU coordinate system, the X-axis points towards the East, and the Y-axis aims over the North Pole. The plane described by the X and Y-axis is tangential to the WGS84 frame on the origin of ENU. Z-axis designates the elevation from a defined geographical plane (figure 1). The ENU frame is considered in this work as the reference frame.

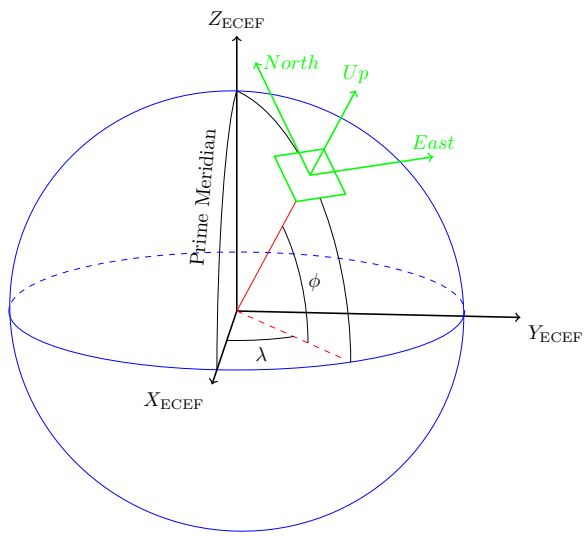


Fig. 1: An illustrative diagram for the WGS84, ECEF, and ENU coordinate systems for the Earth and their transformation correlation (PM line is the Prime Meridian; λ and ϕ are latitude and longitude in WGS84; X, Y, Z for ECEF; and E, N, U for ENU).

Considering a point of reference in the ECEF frame, it is required to discover the equivalent latitude and longitude (λ_r and ϕ_r respectively) of the reference point (X_r, Y_r, Z_r). The parameters of WGS84 required to perform such transformation are presented in table 2 [10].

Parameter	Notation	Value
Semi-major axis	a	6 378 137.0 m
Reciprocal of flattening	$1/f$	298.257 223 563
Semi-minor axis	b	6 356 752.3142 m
First eccentricity squared	e^2	6.694 379 990 14x10-3
Second eccentricity squared	e'^2	6.739 496 742 28x10-3

Table 2: WGS 84 needed parameters to convert ECEF coordinates into ENU. [10]

The conversion from ECEF to ENU coordinates is done with respect to the receiver reference point. Applying the set of equations (2) to approximate latitude (λ_r) and longitude (ϕ_r) for the reference coordinate point. The last transformation outcome in applying equation (3) to ECEF physical quantities with the vector pointing from the receiver reference point (X_r, Y_r, Z_r) to estimated position (X_p, Y_p, Z_p) in the ENU coordinate frame [38].

$$\begin{aligned} p &= \sqrt{(X_r)^2 + (Y_r)^2} \\ \theta &= \arctan(Z_r \frac{a}{pb}) \\ \lambda_r &= \arctan(Y_r, X_r) \\ \phi_r &= \arctan(\frac{Z_r + e^2 b \sin^3(\theta)}{p - e^2 a \cos^3(\theta)} \end{aligned} \tag{2}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{ENU} = \begin{bmatrix} -\sin(\lambda_r) & \cos(\phi_r) & 0 \\ -\sin(\phi_r)\cos(\lambda_r) & -\sin(\phi_r)\sin(\lambda_r) & \cos(\phi_r) \\ \cos(\phi_r)\cos(\lambda_r) & \cos(\phi_r)\sin(\lambda_r) & \sin(\phi_r) \end{bmatrix} \begin{bmatrix} X_p - X_r \\ Y_p - Y_r \\ Z_p - Z_r \end{bmatrix}_{ECEF} \tag{3}$$

2.1.2 Body frame

As the name indicates, the body reference frame is attached to, and moves with, the platform body itself. It is an orthogonal cartesian system with an origin specified at a point on the platform. Axes can be defined both relative to the platform or to the navigation frame. The body frame being an entity which moves and rotates freely inside the navigation frame coordinate system (figure 2). This dissertation uses the term body frame to explain the IMU's frame of

reference relative to the navigation frame coordinate system. TheX_b, Y_b, and Z_b are defined by the MPU-6050 coordinate system (explored in chapter 5.1 Hardware).

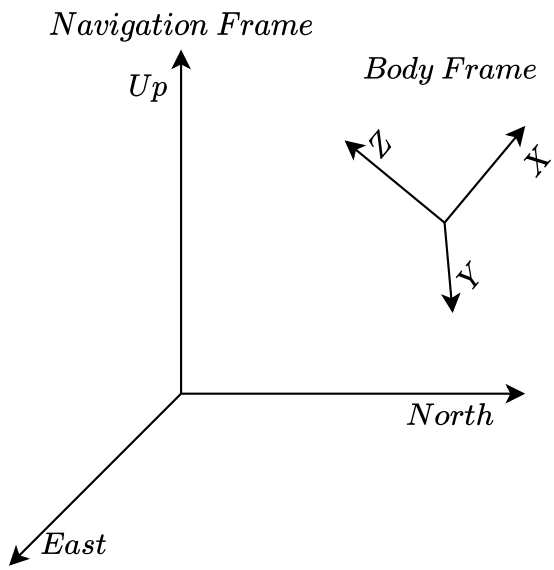


Fig. 2: Navigation and body frame coordinate systems.

2.2 Orientation

The attitude orientation of a body in space is a critical aspect in position estimation. In a low-cost AHRS, accuracy and low complexity are important in essential the attitude of a body. There are various ways to represent attitude, including rotational matrices, Euler angles, and quaternions.

2.2.1 Rotation Matrix

The rotation matrix is a concept employed to express the transformation of coordinates from one frame to another. In addition, it can also convey the orientation of one frame relative to another. Any orientation can be attained by composing three elemental rotations, beginning from a known standard orientation [39]. Analogously, any rotation matrix R can be decomposed as a product of three elemental rotation matrices (equation 4).

$$R = X(\alpha)Y(\beta)Z(\gamma) \tag{4}$$

In equation 4, R is a rotation matrix that can be applied to represent a composition of intrinsic rotations about axes X, Y, Z (in that order), or a composition of extrinsic rotations about axes Z, Y, X (in that order) [40].

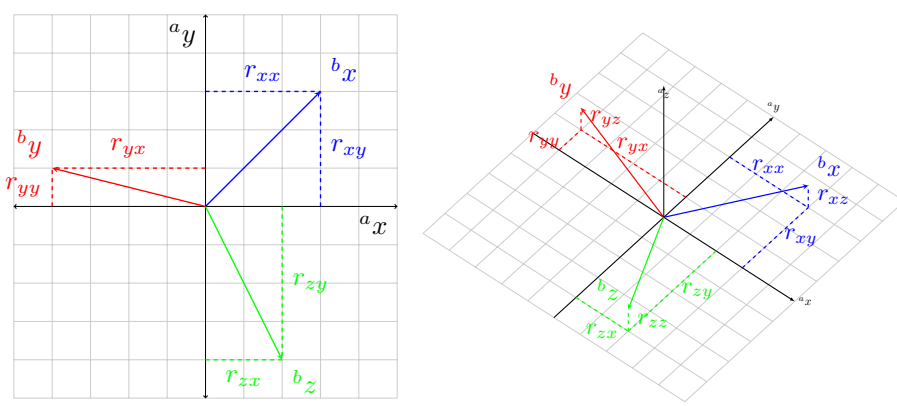


Fig. 3: Frame axis example.

The convention used in this work is represented by equation 5 and maps quantities described in frame b to frame a. Comparing the structure of equation

5 with figure 3 it is seen that columns of ${}^a_bR$ represent each unity vector defining all axes of frame b.

$$ {}^a_bR = \begin{bmatrix} {}^ar_{xx} & {}^ar_{yx} & {}^ar_{zx} \\ {}^ar_{xy} & {}^ar_{yy} & {}^ar_{zy} \\ {}^ar_{xz} & {}^ar_{yz} & {}^ar_{zz} \end{bmatrix} \tag{5} $$

Rotation matrices fit in to the orthonormal class and one crucial property is that ${}^a_bR^a_bR^T = I$ indicating ${}^a_bR^T = {}^a_b R^{-1}$. Likewise, a ${}^a_bR^T$ is equivalent to ${}^b_aR$ [41].

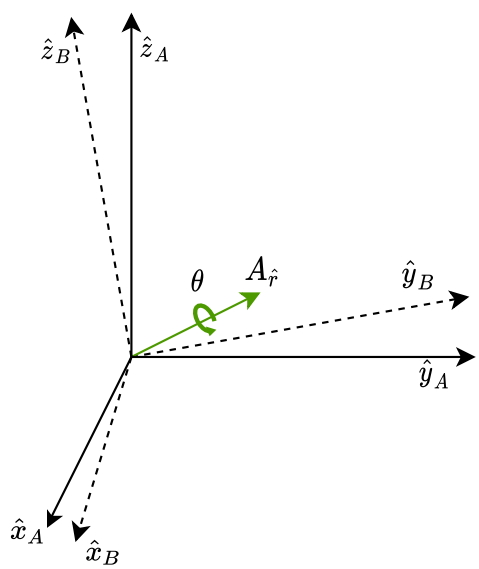


Fig. 4: 3D rotation around vector $\hat{r}$ of angle θ between frame A and B.

2.2.2 Direct Cosine Matrix

The Direct Cosine Matrix (DCM) is related to the rotational matrix, however it is a 3x3 matrix that comprises the cosines of the 9 possible sets of axes of two distinct Cartesian coordinate systems. The DCM is normally employed to

convert from the body frame into the earth frame. The DCM methodology has achieved considerable recognition for its improvement of linear measurement equations in Kalman filters (more in-depth review at chapter 2.4 Sensor Fusion) [42].

2.2.3 Euler angles

Euler angles are a well-known form to represent attitude with respect to a fixed coordinate system. Mathematically, Euler angles represent three composed sequential rotations that stir a reference frame to a given referred frame [43]. It expresses a rotation in three angles, often referred to as pitch, roll, and yaw, denoted by ϕ, θ, and ψ (The roll will have the range of $\pm 180°$, this allows the pitch to have the range of $\pm 90°$. The yaw is represented using the range of $\pm 180°$). These angles depict three successive rotations about the axes of the coordinate frame, for instance, $x-y-z$: First rotate ϕ radians about the x-axis, then rotate θ radians about the y-axis and finally rotate ψ radians about the z-axis [44]. As stated by the differential equations of Euler angles given at (equation 6), ϕ, θ, and ψ can be found based on angular rate measurements.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \frac{1}{\cos(\theta)} \begin{bmatrix} \sin(\phi) & 0 & -\cos(\phi) \\ \cos(\theta)\cos(\phi) & 0 & \cos(\theta)\sin(\phi) \\ \sin(\theta)\sin(\phi) & 1 & \sin(\theta)\cos(\phi) \end{bmatrix} \begin{bmatrix} \omega^b_{nbx} \\ \omega^b_{nby} \\ \omega^b_{nbz} \end{bmatrix} \tag{6}$$

Merely three differential equations require solving prior to the attitude gets distinctly approximated [45]. Each individual rotation is expressed by its own rotation matrix using the respective Euler angle associated, and the final orientation is the result of successive matrices multiplications, as shown in equation 7.

$${}^b_aR = {}^b_2R_x(\phi){}^2_1R_y(\theta){}^1_aR_z(\psi) = \begin{bmatrix} \sin(\phi) & 0 & -\cos(\phi) \\ \cos(\theta)\cos(\phi) & 0 & \cos(\theta)\sin(\phi) \\ \sin(\theta)\sin(\phi) & 1 & \sin(\theta)\cos(\phi) \end{bmatrix} \tag{7}$$

Nevertheless, there is a significant flaw with this approach. As $\cos(\theta)$ approaches zero, the differential equations degrade rapidly, and the output solution becomes vastly imprecise, which indicates that these equations cannot provide effective attitude results at unique points in space [46]. This is also known as

Gimbal lock. Euler angles can be represented as a gimballed system, where the three axes can be thought of as three distinct gimbals attached together [47]. Gimbal lock happens when two axes line up, such as when the pitch and the yaw axis are aligned, and as the roll gimbal is rotated, the pitch and the yaw angle are both affected simultaneously, consequently losing orientation (figure 5) [31]. Euler angles commonly involve a large number of complex mathematical operations, such as matrix manipulations, which typically take up several clock cycles in a Central Processing Unit (CPU), such can negatively impact computational performance.

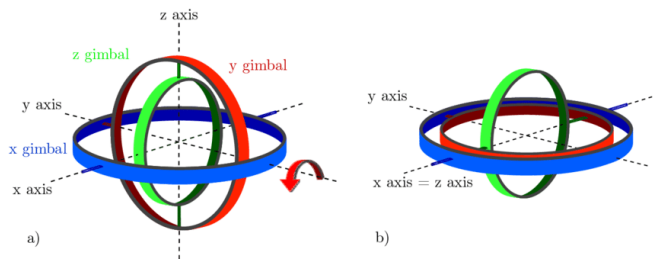


Fig. 5: Representation of the Gimbal Lock problematic [1] - The exterior blue gimbal characterizes the x-axis, the middle, red-colored gimbal the y-axis, and the inner green gimbal the z-axis. In the initial arrangement a), every axis is perpendicular to one another. Following a rotation of 90◦ across the red arrow (y-axis), the blue and the green gimbals occupy the same rotation axis. This condition inhibits the clear determination of the rotation axes when subsequently rotating around the x or z-axis.

2.2.4 Quaternions

Another commonly used attitude representation is the quaternion. To comprehend quaternions, one must first grasp the complex number relationship. A complex number can depict a rotation in a 2-dimensional coordinate frame with a real x-axis and an imaginary y-axis (or vice versa) (figure 6) [48]. A quaternion builds upon this concept, but rather than one imaginary axis, it makes use of three imaginary axes: similar to merging three complex numbers into one. Four components are required to progress from a two-dimensional definition to a three-dimensional plane: one real component q_0 and three imaginary q_1, q_2 and q_3. Quaternions are mathematically denoted as equation 8, and are commonly

represented as a vector (equation 9), where q_0 is the norm, q_1, q_2, and q_3 are complex coordinates with i, j, k being the axis versors. [49].

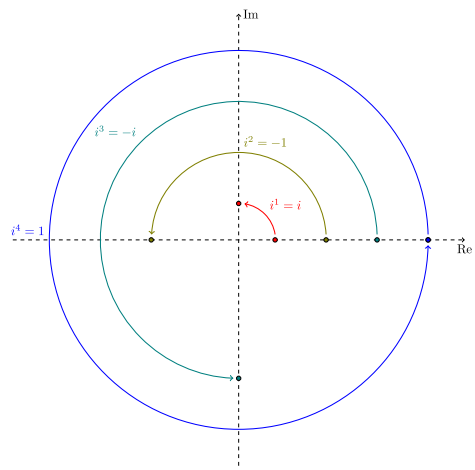


Fig. 6: 90-degree rotation in a 2-dimensional coordinate frame with a real x-axis and an imaginary y-axis.

$$q = q_0 + q_1 i + q_2 j + q_3 k \tag{8}$$

$$q = \begin{bmatrix} q_0 \ q_1 \ q_2 \ q_3 \end{bmatrix} \tag{9}$$

Specifically, when used for attitude description the quaternions should have a norm equal to 1 [50]:

$$\|q\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1 \tag{10}$$

Normalizing a quaternion is identical to normalizing a vector [45]. The quaternion q is divided by the norm of the quaternion $\|q\|$. When a quaternion is normalized, it is known as unit quaternion (also mentioned as versor of q) and denoted with a circumflex accent ($\hat{q}$):

$$\hat{q} = \frac{q}{\|q\|} \tag{11}$$

A quaternion's conjugate (q^*) is expressed by:

$$q^* = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \end{bmatrix} \tag{12}$$

The inverse of a quaternion (q^{-1}) is therefore given by:

$$q^{-1} = \frac{q^*}{\|q\|} \tag{13}$$

Applying the notation as declared previously, ${}^{w}_{b}q$ constitutes the orientation of body frame (b) with respect to world frame (w). From equation 8, the subsequent list of properties/operations is derived:

Multiplication of basis elements Every quaternion multiplication of basis elements $\boldsymbol{i}$, $\boldsymbol{j}$ and $\boldsymbol{k}$ ought to follow the succeeding list of properties (also visible in table 3 containing multiplication of basis elements):

$$\begin{gathered} i^2 = j^2 = k^2 = ijk = -1 \\ ij = -ji = k \\ jk = -kj = i \\ ki = -ik = j \end{gathered} \tag{14}$$

$\times$	1	i	j	k
1	1	i	j	k
i	i	-1	k	$-j$
j	j	$-k$	-1	i
k	k	j	$-i$	-1

Table 3: Quaternion multiplication of basis elements.

Multiplication of quaternions The product of two quaternions, a and b, is denoted by $\otimes$ and it is defined by the Hamilton product. It can be calculated by the product of the basis elements and the distributive law. The product can then be expanded by the distributive law into a a sum of products of basis elements. Returning the next expression:

$$a \otimes b = [a_0 a_1 a_2 a_3] \otimes [b_0 b_1 b_2 b_3] = \begin{bmatrix} a_0 b_0 + a_1 b_0 \boldsymbol{i} + a_2 b_0 \boldsymbol{j} + a_3 b_0 \boldsymbol{k} \\ a_0 b_1 \boldsymbol{i} + a_1 b_1 \boldsymbol{i^2} + a_2 b_1 \boldsymbol{ij} + a_3 b_1 \boldsymbol{ik} \\ a_0 b_2 \boldsymbol{j} + a_1 b_2 \boldsymbol{ji} + a_2 b_2 \boldsymbol{j^2} + a_3 b_2 \boldsymbol{jk} \\ a_0 b_3 \boldsymbol{k} + a_1 b_3 \boldsymbol{ki} + a_2 b_3 \boldsymbol{kj} + a_3 b_3 \boldsymbol{k^2} \end{bmatrix} \tag{15}$$

As a result, the rules defined in equation 14 and table 3 can be applied at this step producing:

$$a \otimes b = \begin{bmatrix} a_0 b_0 - a_1 b_1 - a_2 b_2 - a_3 b_3 \\ (a_0 b_1 + a_1 b_0 + a_2 b_3 - a_3 b_2)\boldsymbol{i} \\ (a_0 b_2 - a_1 b_3 + a_2 b_0 + a_3 b_1)\boldsymbol{j} \\ (a_0 b_3 + a_1 b_2 - a_2 b_1 + a_3 b_0)\boldsymbol{k} \end{bmatrix} \tag{16}$$

Quaternion conjugate Conjugation is an involution (a function that is its own inverse), conjugating an element twice yields the original constituent. The conjugate of a quaternion relates to an inverse rotation, in this case constitutes the orientation of world frame with respect to body frame:

$${}^{w}_{b}q^{*} = {}^{b}_{w}q = [q_0 - q_1 - q_2 - q_3] \tag{17}$$

Vector rotation Defining the subsequent quaternion representation of the same vector but in each referential by using is pure quaternion as

${}^{w}v = \left[0 \; {}^{w}x \; {}^{w}y \; {}^{w}z\right]$ and ${}^{b}v = \left[0 \; {}^{b}x \; {}^{b}y \; {}^{b}z\right]$. The rotation of vector v from one frame to other, using a quaternion is given by:

$${}^{b}v = {}^{w}_{b}\hat{q} \otimes {}^{w}v \otimes {}^{w}_{b}\hat{q}^{*} \tag{18}$$

Composed rotations The composition of rotations can be defined in quaternions as the product between two quaternions. For instance, the sequence a ${}^{a}_{b}\hat{q} \rightarrow {}^{b}_{c}\hat{q}$ is equal to ${}^{a}_{c}\hat{q}$ and is described by:

$${}^{a}_{c}\hat{q} = {}^{b}_{c}\hat{q} \otimes {}^{a}_{b}\hat{q} \tag{19}$$

Quaternion representation permits a more truthful and efficient attitude estimation approach. This is because operations Euler angles entail matrix multiplication which involves additional computational effort as it implicates more mathematical numerical calculations. An additional benefit of employing quaternions is that any transformation is intrinsically orthonormal. Normalizing a quaternion is significantly computationally more efficient than normalizing a matrix. Furthermore, since every rotation needs normalization when completed, the computational costs accumulate [51]. Owing to these benefits, the quaternions are used to determine attitude in this project, while Euler angles are applied merely as a representation of the orientation defined by the quaternions (Euler angles are simpler in human physical interpretation) [52].

2.3 Inertial Measurement Units

An Inertial Measurement Unit (IMU) is an electronic tool that quantifies and describes specific force, angular rate, and occasionally the orientation of a body. It comprises an amalgamation of accelerometers, gyroscopes, and optionally magnetometers. They have also become standard in embedded inertial systems due to their low cost, lightweight, and low power consumption. IMUs are normally employed in aircraft maneuvering (via an Attitude and Heading Reference System), such as spacecrafts [53] [54] [55], satellites [56], and Unmanned Aerial Vehicle (UAV) [57] [58], to name a few. IMUs have been employed in wearable applications with uses in telemedicine [22] [59] [60] [61], and robotics [23]. Newly developed IMUs integrate satellite localization capabilities permitting these devices to operate even when satellites signals are unobtainable, such as in tunnels, indoors, or in the presence of electronic interference. These types of electronic devices are highly susceptible to the occurrence of errors. There are several different types of sensor errors. The main errors that cause unreliable readings are high frequent noise, offset error and drift error.

The first diagram in figure 7 corresponds to the signal error categorized as noise. Noise is made of high frequencies and might be reduced by applying a low pass filter [62]. The primary sensor noise sources can be environmental noise, such as temperature oscillations, gravity, etc. From movements, such as vibrations, shocks. Or even hardware noise, such as electrical and mechanical thermal noise. The second type is the offset error, also known as bias error (second graph in the figure). This offset originates from a shift from the real value and can be resolved by deducting the offset component with a high pass filter. The last type

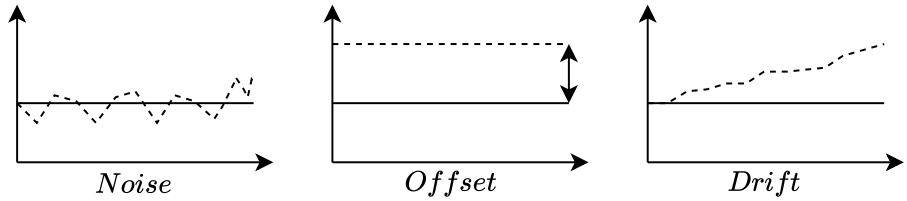


Fig. 7: Three main types of sensor error.

of error is the drift components which can arise from vibrations, shocks, and temperature variations. The primary drift source in this study will be produced by numerical integration through the position calculation. Temperature fluctuations can trigger instabilities in the bias of the sensor signal. By utilizing a supplementary sensor to determine the ambient temperature, the bias generated error can be counteracted by applying adjustments to the output signals. The correlation between bias error and temperature depends on the sensor mechanism. For a gyroscope, the temperature shift will produce an error in orientation which increases linearly with time. For an accelerometer, the temperature will produce an error in position, therefore developing quadratically with time. For the magnetometer the increasing temperature will lead to softening in the moving structure [63]. As mentioned before Inertial Measurement Unit (IMU) are generally comprised of an amalgamation of accelerometers, gyroscopes, magnetometers. The following sub section will describe each one of these electronic utensils.

2.3.1 Accelerometer

An accelerometer is a device capable of measuring proper acceleration. Proper acceleration is the physical acceleration experienced by an object. It is thus acceleration relative to an inertial observer who is momentarily at rest relative to the object being measured [64]. As an example, if an accelerometer would be placed at rest on the surface of the Earth, it will measure an upwards acceleration due to Earth's gravity of g $\simeq$ 9.81 m/s^2. This is due to the Earth's surface exerting a normal force upwards relative to the local inertial frame. On the other hand, a free-falling accelerometer (moving in the center of the Earth's direction at around 9.81 m/s^2) would quantify no acceleration. Acceleration is quantified in the SI unit meters per second (m/s^2), or standard gravity, denoted

by g_n, being the nominal gravitational acceleration of an object in a vacuum near the surface of the Earth. It is defined by standard as 9.80665 m/s^2.

The working principle of an accelerometer can be expounded by a single mass (m) fixed to a stiffness spring (k) that in turn is fastened to outer frame, as illustrated in figure 8 [2].

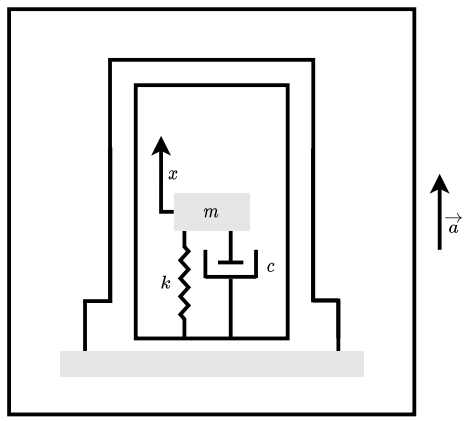


Fig. 8: Schematic demonstrating the working principle of an accelerometer. [2]

Often, the structure incorporates a dashpot (a mechanical device which resists motion via viscous friction). The dashpot has a resistive coefficient (c) and is tied to the mass parallel to the spring. When the system is affected by a linear acceleration, a force equivalent to mass times the acceleration acts upon the mass, causing it to deflect. This deflection is then converted into an analogous electrical signal. The dashpot mechanism causes the system to quickly stabilize following the acceleration. To derive the motion equation of the system, Newton's second law is used, where all real forces acting on the proof-mass are equal to the inertia force on the mass. With x being the displacement of the mass m relative to the outer system. When the system is subject to an acceleration a, the equation of motion for the mass is:

$$mx + kx + cx = \overrightarrow{F} \Leftrightarrow mx + kx + cx = m\overrightarrow{a} \tag{20}$$

Where c and k are the dashpot resistive coefficient and spring stiffness constant, respectively. Consequently, the acceleration can be computed by measuring x, compression of the spring.

2.3.2 Gyroscope

A gyroscope is a device utilized to quantify angular velocity and orientation based on the principles of conservation of angular momentum [65]. A conventional gyroscope accommodates a spinning wheel mounted on two gimbals letting it spin in all three axes, as shown in figure 9. The rotating wheel will resist changes in orientation as an effect of the angular momentum. As a result, when a rotation is exerted on the mechanical gyroscope, the wheel will persist in its global orientation meanwhile the angles between neighboring gimbals will change [3].

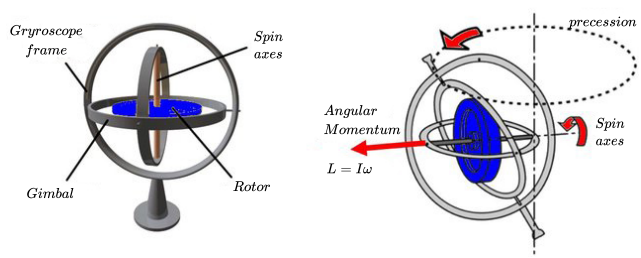


Fig. 9: Representation of a conventional gyroscope accommodating a spinning wheel mounted on two gimbals. [3]

A traditional mechanical gyroscope can merely evaluate orientation, and since they contain moving parts, they can cause the output to drift over time. Modern gyroscopes (such as optical and MEMS gyroscopes) are rate-gyros, meaning they can detect angular velocity [66]. Angular velocity (ω) is quantified in the SI unit radians per second (rad/s), or degrees per second ($^\circ/s$). MEMS sensors constructed using silicon electrical methods have a smaller number of parts and are inexpensive to produce [67]. MEMS gyroscopes based on the Coriolis effect, which states that in a frame of reference rotating at angular velocity (ω), a mass (m) moving with linear velocity (v) experiences a force (figure 10) [68]:

$$\overrightarrow{F}_{Coriolis} = -2m(\omega \times \overrightarrow{v}) \tag{21}$$

Above, the cross product between angular velocity (ω) and linear velocity (v) multiplies solely by orthogonal vector components. The outcome of the cross product is orthogonal to both v and ω. Its direction can be ascertained by the right-hand rule.

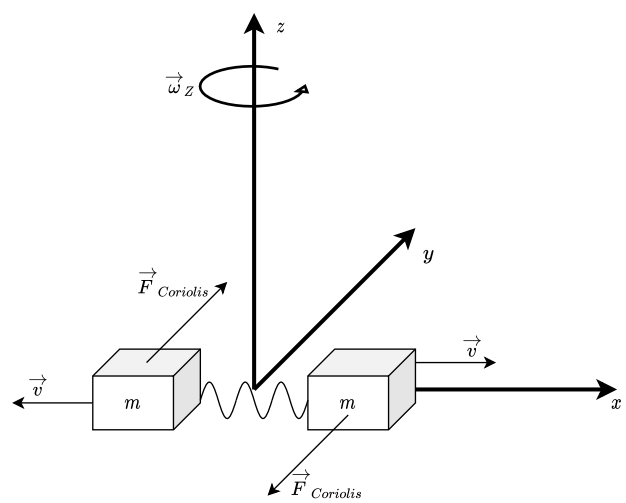


Fig. 10: System where two masses are oscillating in opposite directions. When a rotation is applied, the masses are affected by the Coriolis force and the displacement is measured by a change in capacitance. [4]

The gyroscope sensor measurements supply a body's angular rate, a measurement of the change in angle over time which is represented by $\omega(x)$ [68]:

$$\omega(x) = \frac{dx}{dt} \tag{22}$$

The angular rate supplied by the gyroscope sensor can be integrated over an interval of time (t) and a sampling period (T_s). The sum of every measurement will return an absolute angle θ. The integration of the angular rate is shown in equation 23 [68]:

$$\theta = \int_0^t \omega(x)dx = \sum_0^t \omega(x) \cdot T_s \tag{23}$$

Given a set of angular rates $w = [w_x, w_y, w_z]$ measured by the gyroscope, by equation 16 it is possible to convert the set into a matrix of quaternion rates (Ω) [68]:

$$\Omega = \begin{bmatrix} 0 & -w_x & -w_y & -w_z \\ w_x & 0 & w_z & -w_y \\ w_y & -w_z & 0 & w_x \\ w_z & w_y & -w_x & 0 \end{bmatrix} \tag{24}$$

From a quaternion rate at instant $t-1$, given a current angular speed it is possible to estimate the quaternion at instant t. Quaternion q_t revolved by the gyroscope, the antecedent attitude quaternion q_{t-1} is multiplied by Ω, then half of the period Ts [68].

$$q_t = q_{t-1} * \Omega * \frac{1}{2} * T_s \tag{25}$$

2.3.3 Magnetometer

A magnetometer is an electronic utensil that evaluates existent magnetic flux (a magnetic field's intensity). Magnetic flux is quantified in the SI unit microtesla (μT). Certain magnetometers can detect the orientation, magnitude, or relative change of a magnetic field [69]. A magnetometer accompanied with an accelerometer can successfully reckon an orientation angle. The magnetometer readings uniquely are not able to perceive a heading angle due to the presence of all sorts of distortions such as of the earth's own magnetic fields or even local disturbances of nearby metallic bodies on the sensor [70] [69]. There are two categories of magnetic distortions: Hard iron and soft iron [71] (figure 11). Hard iron distortions are generated by materials that exhibit a continuous additive field, and as a result, produce a persistent additive value to the measurements of the magnetometer [72]. A piece of magnetized iron or a speaker, for example, will induce a hard iron distortion in the magnetometer. If the orientation and position of the magnetic disturbance relative to the sensor is unchanging, the magnetic offset will remain constant. It is possible to compensate for hard

iron disturbances by discerning the maximum and minimum x and y offsets. The x-axis and y-axis offset (α and β, respectively) are given by the average of the maximum (x_{max} and y_{max}) and minimum (x_{min} and y_{min}) values for each axis [5]:

$$\begin{aligned} \alpha &= \frac{x_{max} + x_{min}}{2} \\ \beta &= \frac{y_{max} + y_{min}}{2} \end{aligned} \tag{26}$$

By subtracting these offsets from the magnetometer readings, it is possible to substantially remove any present hard iron disturbances.

Soft iron distortion is the consequence of a material's influence that can warp a magnetic field but does not necessarily produce a magnetic field itself, and as a result is not additive [73]. Unlike hard iron distortion which has a continuous disturbance, a soft iron distortion is determined by the direction of the material in respect to the sensor and the magnetic field of the same material. Calculating the soft iron distortion is computationally more expensive than the hard iron elimination. There are two axes to consider, the major and the minor axes [74].

As seen in equation 27 and 28, the major axis is the axis that runs along the x-axis. This will be used to find the angle θ by calculating the magnitude of the line segment. Essentially by computing r, it is the same as calculating the magnitude of each point on the ellipse and finding the maximum point [5].

$$r = \sqrt{(x_1)^2 + (y_1)^2} \tag{27}$$

$$\theta = \arcsin(\frac{y_1}{r}) \tag{28}$$

2.4 Sensor Fusion

Sensor fusion defines the blending of sensory information from two or more sources in a way that generates a more consistent and dependable understanding of the system. One that would otherwise not be possible when these sources were used individually [75]. Ideally, a gyroscope could deduce a body's attitude independently. As mentioned previously, integrating the gyroscope's readings can ascertain the attitude solution of a body. Notwithstanding, with less cost comes

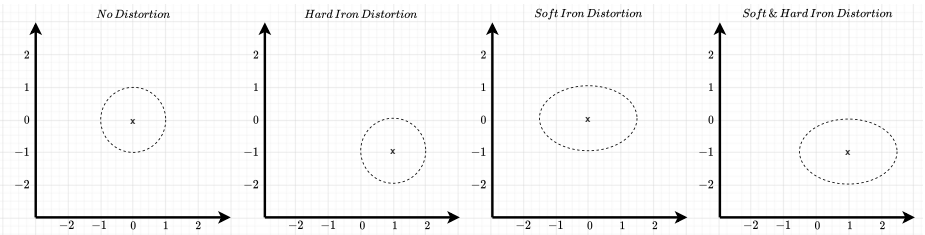


Fig. 11: Representation of a magnetometer's different types of disturbances. [5]

lower accuracy and precision. This originates from a drift build up on the gyro's readings, where error accumulates exponentially on the integrated output [76]. The accelerometer and magnetometer can work together to evaluate attitude straight from measurements of acceleration and present magnetic fields. This combination is often corrupted by disturbances such as vibrations and local magnetic interferences [70]. Hence, both solutions have their respective benefits and require a rectifying term from an additional sensor. This conducts to a demand for a sensor fusion algorithm that combines both solutions in a way that emphasizes their strengths and minimizes their weaknesses [77]. A sensor fusion algorithm will decide on the optimal trust combination of each sensor to assess attitude. A diversity of suitable algorithms exist that can fuse the output of different sensors to create a more reliable and consistent signal. This chapter will highlight some of the most prominent sensor fusion algorithms in existence:

2.4.1 Kalman Filter

The Kalman filter algorithm is a set of mathematical equations that provides a computationally efficient approach to estimate some unknown variables by the detected measurements [78]. Kalman filters operate recursive functions to predict the present state of a linear problem by monitoring the current input data, the previous input data, and the previous state prediction. Two generally assigned methods for Kalman filter-based sensor fusion are state-vector fusion and measurement fusion. The state-vector fusion method (figure 12) applies a group of Kalman filters to acquire individual sensor-based state estimates, which are subsequently fused to obtain an enhanced combined state estimate.

Measurement fusion (figure 13) approach directly combines the sensor data to achieve a joint measurement and later uses a single Kalman filter to get hold of the final state estimate centered on the fused measurement [6].

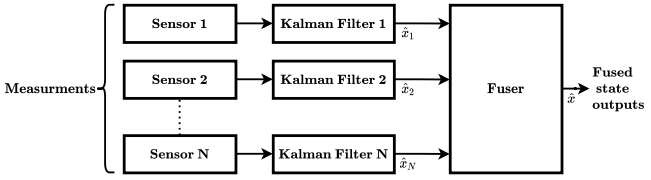


Fig. 12: Kalman-filter-based multi-sensor state-vector fusion. [6]

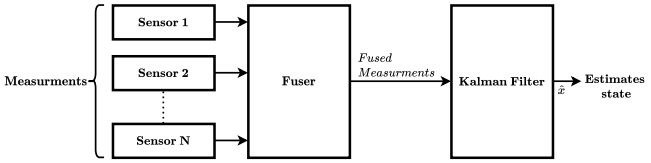


Fig. 13: Kalman-filter-based multi-sensor measurement fusion. [6]

When applied appropriately, Kalman filters offer highly precise orientation, even with the existence of substantial noise. Nevertheless, Kalman filters are computationally expensive rising hardware cost and latency. They are also of complex implementation, which, shared with computational overhead, can make the algorithm unfeasible for computationally restricted applications. They are regularly useful in a wide-ranging variety of applications and have become a standard method in sensor fusion. Several studies examine the possibility of using Kalman filters to predict a body's orientation and position by combining multiple sensors. The Kalman filter is founded on recursive Bayesian filtering. Consequently, the system's noise is assumed to be Gaussian. Therefore, the Kalman filter is generally suggested for linear systems. For this reason, an extension of the classic Kalman Filter designed for non-linear systems has emerged, recognized as Extended Kalman Filter (EKF) [23].

2.4.2 Complementary Filter

The complementary filter is considered a simpler approach relatively to the Kalman filter since it is a computationally lightweight solution and straightforward to implement [7]. This filter takes as input two noisy sensor measurements and assumes one input is mainly formed by high-frequency signals whereas the other is mostly by low-frequency signals. Through a low pass filter, the high-frequency noise of the first input is filtered out. An identical procedure occurs with the second signal, but this time with a high pass filter to remove low-frequency noises, as illustrated in figure 14.

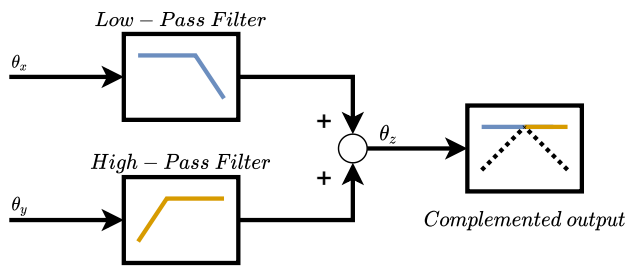


Fig. 14: Basic complementary filter [7] - Two different measurement sources for estimating one variable. The noise properties of the two measurements are such that one source gives good information only in low frequency region while the other is good only in high frequency region.

The accelerometer is especially vulnerable to vibrations and centripetal forces, these can be filtered by a low pass filter behaving as a moving average filter. The gyroscope is precise and exact in the short term, but quickly loses precision in integration due to drift. For that reason, a high pass filter that allows the short-term gyroscope data is desirable, thus, removing the long-term drift. At any given moment, the signal is subject to a low pass filter (equation 29) or a high pass filter (equation 30), where x_t is the measurements from the sensor, y_t is the filtered final output, and α is a filter coefficient:

$$y_t = (1 - \alpha)x_t + \alpha y_{t-1} \tag{29}$$

$$y_t = (1 - \alpha)y_{t-1} + (1 - \alpha)(x_t - x_{t-1}) \tag{30}$$

α filter coefficient describes the borderline where the gyroscope measurements end, and the accelerometer readings begin and vice versa. It rules how much the output relies on the current reading or a new value that arrives. α is typically above 0.5 by the definition above. Where the filter coefficient α is determined by:

$$\alpha = \frac{\tau}{\tau + \Delta t} \tag{31}$$

Where τ is the wanted time constant (time interval for the readings to respond) and Δt is the sampling period.

$$\theta = \frac{\tau}{\tau + s} a + \frac{s}{\tau + s} \frac{1}{s} \omega = \frac{a + \tau\omega}{\tau + s} \tag{32}$$

A mathematical model of the complementary filter can be represented as equation 33, where θ_t represents current orientation, α is the filter coefficient, ω the gyro's angular velocity, and a the acceleration obtained from the accelerometer:

$$\theta_t = (1 - \alpha)(\theta_{t-1} + \omega \Delta t) + \alpha a \tag{33}$$

Yet, the complementary filter is not especially robust to noisy or biased data since it simply uses currently available information, therefore, has no direct method of compensating for sensor noise [23]. A conventional application of the complementary filter is to bring together measurements of vertical acceleration and barometric readings to attain an approximation of vertical velocity. Similar to the Kalman filter, new versions built upon the principles of the classic complementary filter have emerged in recent times, such as the Extended Complementary Filter (ECF). They promise a high level of accuracy and enhanced robustness to noise while preserving computational efficiency.

2.4.3 Optimization Algorithm

Up until recently, there remained mainly two distinct AHRS fusion approaches. One category including the complementary filters, and the other is related to Kalman filtering. Some recent AHRS algorithms have emerged in the literature over the past years. Two of the most prominent are the Mahony and Madgwick algorithms, which have been categorized as optimization filters. Optimization filters obtain orientation by assessing a vector representative of the sensor output at the present orientation and lessening the disparity concerning predicted and observed outputs. Optimization filters are well established for linking accuracy with computational expense and simplicity of implementation [22]. Both methods make use of a quaternion representation, which is a four-dimensional complex number representing of an object orientation. Quaternions involve fewer computation time because of their minimal quantity of calculation parameters [79]. Additionally, vector rotations are easily executed by quaternion multiplications. Madgwick et al. [80] pioneered a gradient descent fusion algorithm, frequently recognized as 'Madgwick Algorithm.' This gradient descent fusion algorithm first

obtains a quaternion estimation of the gyroscope output integration and later corrects it with a quaternion from the accelerometer and magnetometer data. Madgwick's approach guarantees decent attitude estimation at a low computational cost. Further, it tackles the difficulty of the local magnetic disturbances that can influence all the orientation components. By reducing the constraint of the magnetic field vector rotation, it can limit the effect of the magnetic disturbances to only affect the yaw component of the orientation.

2.4.4 Other Algorithms

Angular Rate

Angular Rate updates a quaternion through incorporation of angular rate measurements of a gyroscope. The simplest approach to update the quaternions is by combining the differential equation for a local rotation rate [81]. In a kinematic system, the angular velocity ω of a rigid body at any instant time is defined with respect to a fixed frame coinciding directly with its body frame. Therefore, this angular velocity is in terms of the body frame [82].

AQUA

Roberto Valenti's Algebraic Quaternion Algorithm (AQUA) [50] approximates a quaternion with the algebraic solution of a system from inertial/magnetic observations. AQUA computes the "tilt" quaternion and the "heading" quaternion independently in two sub-parts. This prevents the effect of the magnetic disturbances on the roll and pitch components of the orientation. AQUA can be employed with a complementary filter to fuse the gyroscope data together with accelerometer and magnetic field readings. The correction part of the filter is based on the individually assessed quaternions and works for both IMU and Magnetic, Angular Rate, and Gravity (MARG) sensors [83].

FAMC

The most conventional low-cost sensor system is the Accelerometer-Magnetometer combination (AMC). It mixes the local gravity and the Earth's magnetic field simultaneously, establishing a full-attitude estimation system. The most famous attitude determination formulation is Wahba's Problem, which sprung several solutions using Euler Angles, Direction Cosine Matrices and Quaternions. The matrix operations in these solutions are the main focus of interest in this method. The operations are methodically simplified, where the accuracy is retained, while

the time expenditure is reduced, yielding the Fast Accelerometer-Magnetometer Combination (FAMC).

FLAE

The Fast Linear Attitude Estimator (FLAE) acquires the orientation quaternion with an eigenvalue-based solution as suggested by [84]. A figurative solution to the resultant characteristic polynomial is also derived for a higher computation performance.

Fourati

Attitude estimation set of rules as suggested by [85], whose method blends a quaternion-based nonlinear filter with the Levenberg Marquardt Algorithm (LMA). The estimation algorithm has a complementary construction that manages measurements from an accelerometer, a magnetometer, and a gyroscope, joint in a strap-down system, built on the time integral of the angular velocity, exploiting the Earth's magnetic field and gravity vector to compensate the attitude predicted by the gyroscope. The rigid body attitude in space is revealed when the body's orientation frame (X_B,Y_B,Z_B) is identified with respect to the navigation frame (X_N,Y_N,Z_N), where the navigation frame respects the convention North East Down (NED).

QUEST

QUaternion ESTimator (QUEST) as described by Shuster in [86] and [87].

ROLEQ

Recursive Optimal Linear Estimator of Quaternion (ROLEQ) is a revised OLEQ, where a recursive evaluation of the attitude is created with the computed angular velocity [88]. This estimation is set as the primary value for the OLEQ estimation, simplifying the rotational operations.

SAAM

Super-fast Attitude from Accelerometer and Magnetometer (SAAM) is a novel estimator suggested by [84], proposes an exceptionally simplified computation of Davenport's resolution to Wahba's problem, where the full solution is condensed to a couple of floating-point operations, not losing much accuracy, and keeping computational time low.

Tilt

Tilt does attitude estimation through measurements of gravity acceleration. The most straightforward approach to assess the attitude from the gravitational acceleration is utilizing 3D geometric quadrants. Even though some techniques use arctan to approximate the angles [89] [90], it is desired to utilize arctan2 to investigate all quadrants seeking the tilt angles.

3 Related Work

This section will narrow the existing literature within this research's scope, examining the methods each author used in their work and interpreting key insights and how each source contributes to the existing pool of knowledge on its subject. The literature review is divided into the subsequent sections: 1) Position estimation using inertial sensors systems. 2) Sensor fusion in position and orientation estimation.

3.1 Position estimation using inertial sensor systems

Inertial sensor systems have been thoroughly researched with the purpose of delivering position estimation of a moving body. Inertial systems are autonomous and independent and do not rely on external information, such as radio signals or electromagnetic waves. Their navigation data have short-term high accuracy, great constancy, and elevated data update rate. They may be applied in an array of distinct positioning methods.

3.1.1 Pedestrian Dead Reckoning

Pedestrian dead reckoning (PDR) is among the most explored. PDR combines step detection, step length estimation, and orientation approximations to calculate the absolute position and heading of a walking user. PDR can operate with a single accelerometer, although superior precision and robustness are obtained with more sensors. An IMU containing several accelerometers, gyroscopes, magnetometers, and even pressure sensors are commonly employed to recognize steps and orientation. Sensors might be body-mounted or shoe-mounted. Pedestrian navigation systems can aid the blind and visually impaired, locating and rescuing firefighters and other emergency workers, hiking, sports, and others. PDR is commonly reviewed in the literature, being subject to studies in various settings.

Ladetto et al. [91] applied PDR in urban and indoor areas seeking to assist blind people reaching unfamiliar locations along with aiming to facilitate emergency coordinators to track rescue workers. The study integrated a GPS receiver with a body mounted IMU applying pattern recognition to accelerometer signals, determining a user's step signature.

Stirling et al. [92] illustrate an experiment exploiting a shoe-mounted sensor prototype that calculates stride length with accelerometers and magnetometers.

Their system measures angular acceleration by manipulating pairs of accelerometers as an alternative to gyroscopes.

Several other studies investigate the prospect of using inertial sensor systems to estimate the absolute position and heading of a walking user for multiple purposes [93] [94] [95] [96]. The main drawback of PDR is its dependence on step prediction algorithms that must distinguish step direction and step lengths as the user changes pace.

3.1.2 Strapdown Inertial Integration

Strapdown inertial integration or Strapdown Inertial Navigation System (SNIS) is another prevalent position method. With SNIS, sensors are usually tightly strapped or attached to the axes of the moving body's structure, lowering costs, and enhancing the system's reliability. This technique integrates accelerometer and gyroscope measurements to distinguish the variation of position and heading. The strapdown system demands a high-level measurement rate, on average, beyond 2000 Hz. Typically, higher measurement rates translate into more accurate integration readings of position and attitude. Strapdown systems are currently employed in commercial and military applications (airplanes, vessels, Remotely Operated Underwater Vehicle (ROV), projectiles) and are a topic of study among scholars.

Jameian et al. [97] introduced SNIS to nautical environments, proposing a compensation method against disturbing forces affecting vessel motion caused by rough sea conditions. They aim to resolve attitude determination offset through self-alignment of SNIS by establishing vector observations. The implementation makes use of a quaternion estimator for attitude determination, significantly diminishing computational complexity.

An indoor strapdown inertial navigation with small foot-mounted and self-contained sensor systems was described by Bird et al. [98]. Similar to PDR, SNIS also has applications in pedestrian navigation systems, although operating in an utterly distinct fashion. Unlike PDR, the strapdown navigation algorithm traces the entire movement of the foot in between steps. Any movement like walking, running, climbing up or down, moving backward or sideways, sliding, and even jumping can be tracked. This is possible because of a Zero-Velocity Update (ZVU) which exploits the brief periods of zero velocity when the feet are stationary on the ground.

SNIS and PDR may also be used together, sharing the same inertial sensors. In this case, inertial navigation is incorporated within the multi-sensor integration architecture as the reference system and PDR as an aiding sensor.

With a focus on low-cost inertial motion sensors, Coyte et al. [17] applied PDR to sporting training and rehabilitation. They propose solutions to acceleration noise accumulation and gyroscope angle error problems. To improve the accuracy of displacement estimation with a low-grade IMU, they developed a ZVU algorithm.

3.2 Sensor fusion in position and orientation estimation

Sensor fusion defines the blending of sensory information from two or more sources in a way that generates a more consistent and dependable understanding of the system. One that would otherwise not be possible when these sources were used individually [99]. Fusing multiple inertial systems has raised significant interest and consideration in location and attitude performance improvement. Numerous methods arose in recent times that merge information from various systems such as inertial sensors, GNSS, radar, radio telescopes, signal of opportunity systems like Angle of Arrival (AOA), Time of Arrival (TOA), Received Signal Strength (RSS), and Signal to Noise Ratio (SNR). The combination of multiple sources can help reduce noise with two different sensor types. These separate systems are integrated by fusion filter algorithms which process each input and generate a more precise and reliable output [100]. A substantial sum of distinct solutions designed to assess the orientation of a rigid body regarding a reference frame exist in literature. Two main approaches aimed at sensor fusion exist, Kalman and complementary related filters. A comprehensive analysis of the literature will be conducted seeking to better understand the distinction between algorithms and how do they compare.

3.2.1 Kalman filter

Several research works have been conducted on Inertial Navigation System (INS) and Global Navigation Satellite System (GNSS) integration through data fusion, particularly using the Kalman filter. To overcome the shortcomings linked to the detached functioning of GNSS and INS, Wong et al. [18] Qi et al. [101], and Nassar et al. [14] combined both systems so that their disadvantages were lessened or eradicated, complementing one another. While GNSS was comparatively more

stable and consistent for long periods, INS had a more reliable and comprehensive short-term signal. Updating INS position and velocity with GNSS data corrected error expansion at the same time it delivered more precise estimates. The Kalman Filter attempted to adjust INS information based on the system error model whenever GNSS signals were interrupted or limited. These studies have demonstrated success in satisfying the accuracy requirements of low-precision applications. However, they could not deliver the high precision positioning some applications required. Hence, other studies attempted to achieve better performance of integrated INS/GNSS systems through the exploration of extended and adaptive Kalman filtering techniques. Mohamed and Schwarz [102] performed an analysis on INS/GNSS alternative integration through an adaptive Kalman filtering technique. Findings reveal that their adaptive Kalman filter outperformed by almost 50% the conventional filter.

The use of data fusion in autonomous flying units such as UAV has recently gained particular concern due to the dissemination of consumer-grade quadcopters. In autonomous aerial settings, an accurate altitude reading is crucial to control the position of the flying system. However, such measurements are repeatedly corrupted with signal noise produced by the vehicle's motors. Hetényi et al. [19] applied a Kalman Filter to fuse the sonar and accelerometer signals, obtaining a considerably improved altitude estimate with minimal error. Similarly, Luo et al. [20] combined the UAV sensor system and received signal strength (RSS) in a Kalman filter solution. The study sought to increase position and altitude estimation as well as collision avoidance precision by approximating the distance between the receiver and the transmitter via the use of radiofrequency signals reducing the noise component.

Sharma et al. [21] present an experiment of Kalman filter-based sensor fusion for extrapolation of a robot's orientation and depth to obstacle by fusing the inputs from three infrared sensors and an inertial sensor system. The combination of multiple sensor inputs allowed the robot to operate in fault-tolerant applications and enhanced its obstacle avoidance decision making, localization, and orientation estimations.

3.2.2 Complementary filter

Madgwick et al. [22] outlined the formulation of an Extended Complementary Filter (ECF) algorithm and exhibited its applicability as a human motion monitoring wearable. Their design fused magnetic, angular rate, and gravity sensor

data to remotely estimate limb orientation in stroke patients performing rehabilitation exercises. They analyzed performance under a range of circumstances and benchmarked alongside other frequently utilized sensor fusion algorithms. They claim an improved computational efficiency of over 30% when compared with standard alternative algorithms.

A complementary filter designed for sensor fusion in quadrotor UAV employing a low-cost inertial measurement system was proposed by Noordin et al. [103]. The complementary filter filtered high-frequency signals associated with the gyroscope and low-frequency signals linked to the accelerometer. Findings demonstrate that the complementary filter technique overcame the over drift conundrum related to gyroscopes and was capable of computing attitude angles efficiently. Euston et al. [104] conducted an analogous study with a non-linear complementary filter for attitude estimation in a UAV utilizing a low-cost IMU. They broadened the experiment to incorporate a model of the longitudinal angle-of-attack corresponding to the UAV's airframe acceleration using airspeed data. As a result, they could estimate the acceleration of the UAV during continuous turns based on gyroscope and airspeed data. They accomplished attitude filtering performance of similar quality as an EKF that fused GPS/INS at a far less computational cost.

3.2.3 Sensor fusion algorithms comparison

Some studies have conducted comparison experiments between the sensor fusion algorithms in distinct settings to assess their performance in that unique condition. Ludwig et al. [79] compared Madgwick and Mahony in a foot-mounted experiment. Their findings revealed that Madgwick achieved better heading orientation than Mahony when compared to the ground truth. Nonetheless, the performance of Mahony was superior to Madgwick. The same authors tested on [105] quadcopters the EKF, Madgwick, and Mahony filters. Results showed that Mahony delivered a more precise orientation estimation and faster execution time than Madgwick and EKF. Diaz et al. [106] present a comparison among Madgwick and Mahony, a basic AHRS estimation algorithm, and the recent algorithm proposed by the authors. The study centered around comparing the performance of Madgwick, Mahony, EKF, and their own sensor fusion algorithm, emphasizing the behavior under magnetic perturbations. Various examples of movement were analyzed, from carrying the sensor at separate places such as pocket, shoe, and hand. They concluded that their algorithm was slightly

less influenced by magnetic perturbations than the others, but overall, the algorithms performed similarly.

4 Methodology

This section provides an overview of the methodology employed to assess the system's orientation and position. Going through how the sensor's outputs are combined to obtain orientation by sensor fusion, and how displacement can be estimated using clever filter techniques to remove gravity's influence on accelerometer's readings. A position estimation method is then presented showing how combining orientation and displacement using dead reckoning technique can return a body's position over time. The chapter ends with a visualization of how the designed position methodology pipeline performs trying to estimate position of an experiment. It will also describe how well did the estimation perform by describing how to measure an experiment's estimation error.

4.0.1 Estimating Orientation

The first step in methodology is to obtain orientation (AHRS) from the fusion of the three sensor raw outputs (figure 15). The accelerometer provided the system's proper acceleration; the gyroscope supplied the body's angular rate, and the magnetometer presented the detected magnetic flux.

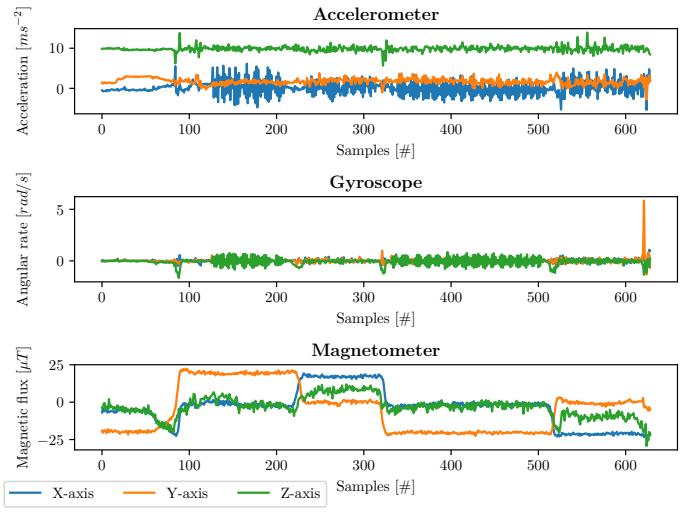


Fig. 15: Raw sensor data on each axis obtained by the accelerometer, gyroscope, and magnetometer.

Measurements were taken at periods of 50 ms (20 Hz). This frequency was chosen for being the lowest sample period that didn't compromise accuracy. To estimate orientation from the raw sensor data, two approaches were created: the first approach was to apply a sensor fusion algorithm directly in the inertial system's microcontroller, which could estimate orientation in real time while the measurements were being taken. The Madgwick algorithm was utilized for its well-recognized proficiency to merge accuracy with computational cost and simplicity of implementation. The second approach meant storing the raw sensor data in an SD card and process it later the experiment with a sensor fusion library, so a benchmark of how different sensor fusion algorithms perform under the same data. This chapter's data and plots will be based on the microcontroller's Madgwick real time sensor fusion, but this principle applies in the same manner to the post-experiment sensor fusion approach. The sensor fusion algorithm (in this case Madgwick's filter) takes as input the raw sensor measurements and outputs estimated orientation as a set of four quaternions (figure 16).

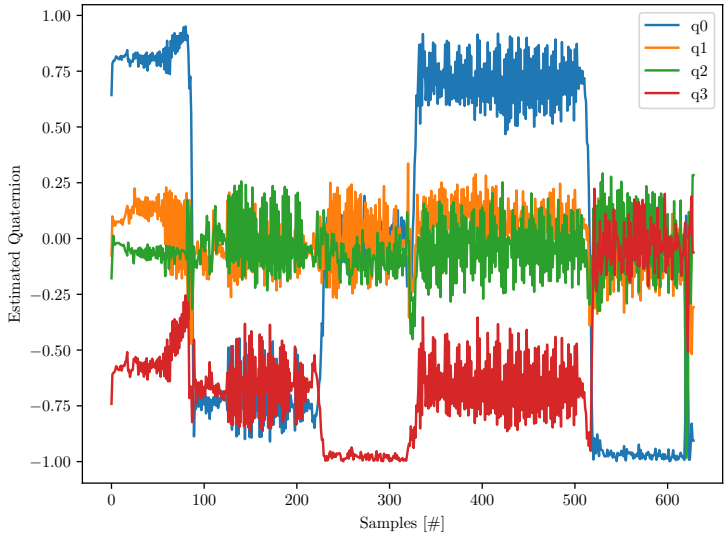


Fig. 16: Madgwick's algorithm sensor fusion output of the raw measurement data of figure 15.

As mentioned in chapter 2.2 Orientation, quaternions can be converted back into Euler angles by:

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \arctan \frac{2(q_0 q_1 + q_2 q_3)}{1-2(q_1^2+q_{2^2})} \\ \arcsin(2(q_0 q_2 - q_3 q_1)) \\ \arctan \frac{2(q_0 q_1 + q_1 q_2)}{1-2(q_{2^2}+q_{3^2})} \end{bmatrix} \tag{34}$$

Which Programmatically can be translated as the following script:

```
# roll (x-axis rotation)
sinr_cosp = 2 * (q0 * q1 + q2 * q3)
cosr_cosp = 1 - 2 * (q1 * q1 + q2 * q2)
angles.roll = math.atan2(sinr_cosp, cosr_cosp)
# pitch (y-axis rotation)
sinp = 2 * (q0 * q2 - q3 * q1)
angles.pitch = math.asin(sinp)
# yaw (z-axis rotation)
siny_cosp = 2 * (q0 * q3 + q1 * q2)
cosy_cosp = 1 - 2 * (q2 * q2 + q3 * q3)
angles.yaw = math.atan2(siny_cosp, cosy_cosp)
return angles
```

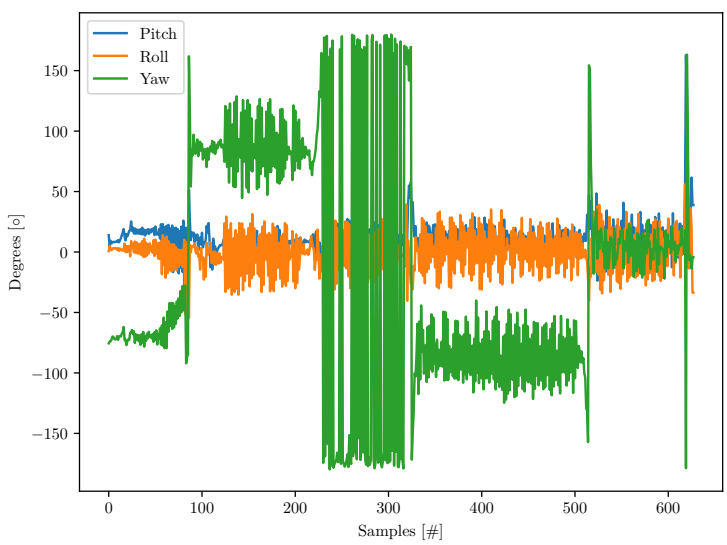


Fig. 17: Madgwick's algorithm sensor fusion output of the raw measurement data of 15 converted into Euler angles.

4.0.2 Estimating Displacement

Accelerometer readings can measure proper acceleration forces, which can be employed to determine a body's velocity and position relative to a starting point. Integrating the resultant of acceleration will yield velocity, and double integrating will provide the body's accumulative position.

Once the measured inertial-frame acceleration is attained, it can be integrated to obtain inertial frame velocity v_i and position x_i can be calculated:

$$v_i = v_0 + \int^t a_i \; dt \tag{35}$$

$$x_i = x_0 + \iint^t a_i \; dt \tag{36}$$

In practice, data is acquired at discrete time periods (Δt), so the approximate velocity and position are estimated by:

$$v_i[k+1] = v_i[k] + a_i[k]\Delta t \tag{37}$$

$$x_i[k+1] = x_i[k] + v_i[k]\Delta t \tag{38}$$

However, these measurements are affected by the Earth's gravitational field (as seen in figure 15 in the acceleration readings) and rotational components of acceleration, considerably amplifying numerical errors. The gravity component will not be differentiated from the physical acceleration of the device and will eventually generate exceedingly elevated errors in the measured acceleration. To overcome this challenge, a gravity compensation algorithm is crucial for subtracting the impact of the gravity component on acceleration readings.

```
# accelerometer reading
acceleration = [ax, ay, ay]

# quaternion corresponding to the orientation
q

# gravity on Earth in m/s^2
```

```
gravity = [0, 0, -9.81]

a_rotated = rotate(acceleration, q)
# rotate the measured acceleration into
# the Earth frame of reference

linear_acceleration = a_rotated - gravity
```

Rotating the accelerometer reading by the quaternion into the Earth frame of reference (into the Earth's coordinate system), then subtracting gravity. The remaining acceleration is the acceleration of the sensor in the Earth frame of reference often referred to as linear acceleration. Through orientation estimation determined previously, it is possible to find the orientation of the Earth frame with respect to the sensor frame. Therefore, compute the expected direction of gravity and then subtract that from the accelerometer readings (figure 18).

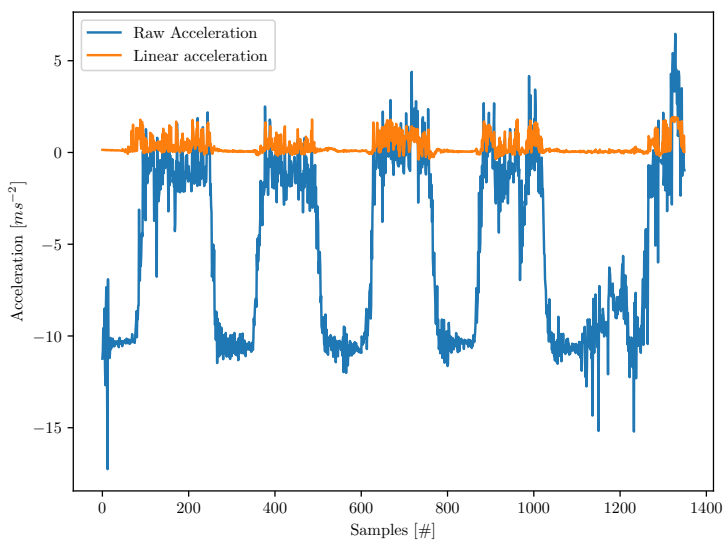


Fig. 18: Comparison between raw acceleration (blue) and measurements with gravity compensation (red). When tilting the sensor, the accelerometer will measure gravity's acceleration (around 9.8 ms^{-2}). The gravity compensation filter can counteract virtually all the gravitational influence on the sensor readings.

4.0.3 Estimating Position

Resulting in a linear acceleration that corresponds to the physical acceleration of the device. This linear acceleration can be numerically integrated returning velocity and double integrating delivering the position of the device (figure 19).

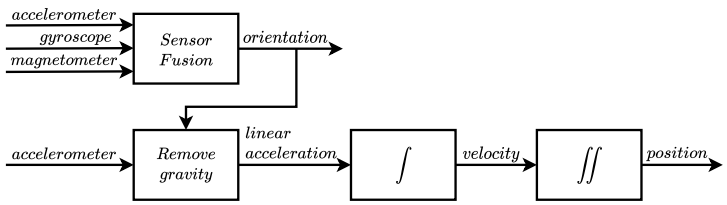


Fig. 19: Overview of the position estimation methodology.

The next step consisted of experimentally testing the inertial system seeking to achieve position estimation merely by integrating the accelerometer measurements with the gravity compensation filter. The tests involved walking on a straight line carrying a box containing the inertial system (with the inertial X-axis parallel to the path) for around 30 meters, evaluating the computed acceleration, velocity, and accumulative position estimation. Specific tests were executed at walking pace, others at running pace, and some with a combination of both.

First, one-dimensional experiments were conducted, pursuing to estimate accumulative position when moving on a straight line for 30 meters. These tests revealed an average error margin from 10% to 15% of the actual distance. Different movements and walking paces were tested, such as stopping at different time intervals. With the accomplishment of this first experiment, we decided to expand the testing to two dimensions. An experiment result is observed in figure 20, where the final observed accumulative position was 25 meters.

Combining the displacement over a period with the estimations of attitude, it is possible to implement a two-dimensional estimation of position using the dead reckoning technique. The new position x_i, is given by applying a direction vector to the previous determined position x_{i-1}. The norm of the direction vector is given by the displacement in that axis over the period since last sample (Δx) and its angle by the decomposing the attitude estimation (yaw θ) into two components, $\cos\theta$ for the x axis, and $\sin\theta$ for the y axis (as shown in figure 21):

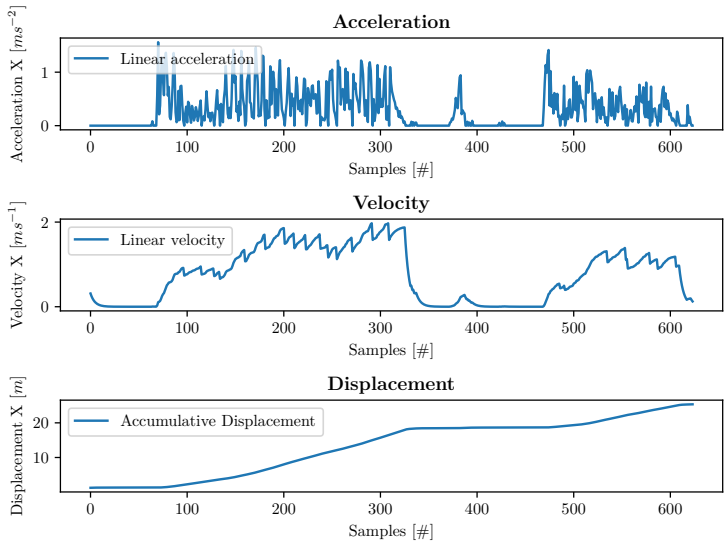


Fig. 20: Plot showing how acceleration is integrated into velocity and displacement.

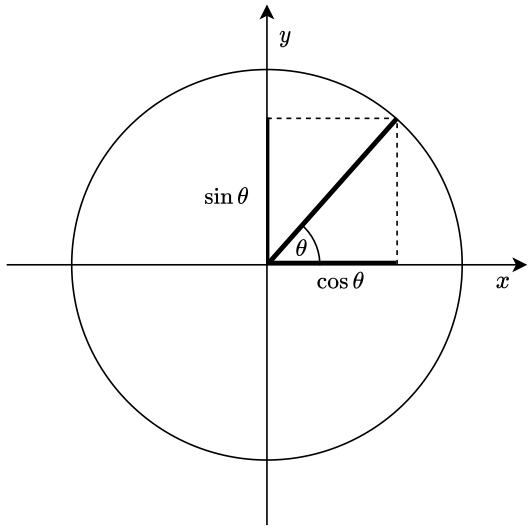


Fig. 21: Illustration of a unit circle where θ is the attitude angle.

A mathematical representation of this approach is seen at equation 39:

$$
\begin{aligned}
x_i &= \Delta x \cos(\theta) + x_{i-1} \\
y_i &= \Delta y \sin(\theta) + y_{i-1}
\end{aligned}
\tag{39}
$$

Programmatically this is equivalent to:

```
# Calculate X, Y, Z positions - Distance * heading angle
lastX, lastY, lastZ = getCoordinates()
newX = displacementX * cos(radians(yaw)) + lastX
newY = displacementY * sin(radians(yaw)) + lastY
newZ = displacementZ * sin(radians(pitch)) + lastZ

# Update the coordinate with each function
updateCoordinates(newX, newY, newZ)
```

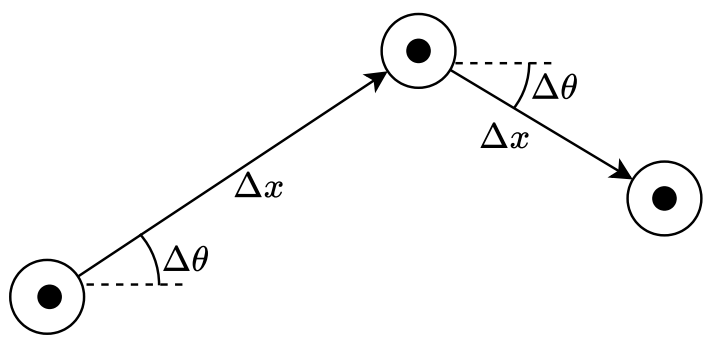


Fig. 22: Illustration of the dead reckoning approach. The object's position at a given moment is given by the last known position, the heading angle $\Delta\theta$, and the travelled distance Δx that occur during the last sample.

4.0.4 Applying methodology in 2D

In two-dimensional tests, the output from X-axis and Y-axis accelerometer measurements were integrated with orientation to acquire accumulative position on both axes. Integrated axis position was then plotted into a scatter plot that combines the accumulative position on the two axes. Figure 23 shows the output of applying the position methodology to an experiment of a $4m \times 4m$ square.

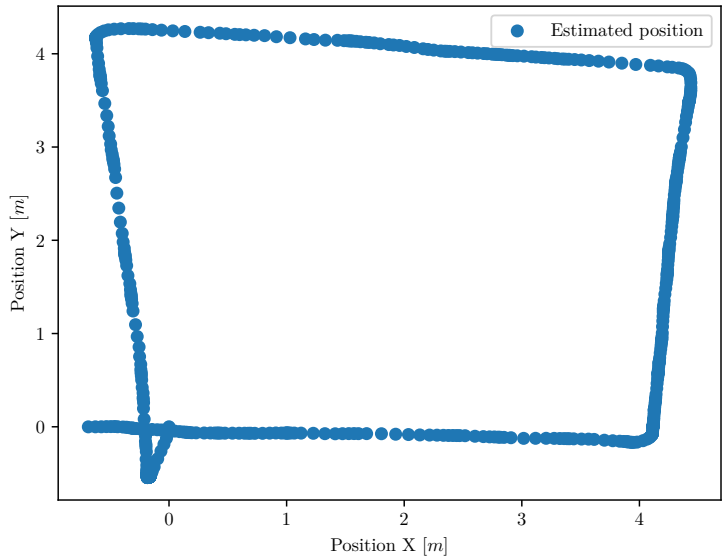


Fig. 23: Result after applying the position estimation methodology to the raw sensor data.

The figure shows a succession of position points, each calculated by the last known position, the heading angle, and the travelled distance. The position given is relative to the sensor's own body frame, not the world's frame.

4.0.4.1 Measuring error

In order to quantify how well a test performed in estimating position, it is necessary to compare it against a ground truth (explained in previous chapter). Error was quantified in two ways, the first being how far are the estimated shape corners from the ground truth's vertices. A turning detection algorithm, Ramer-Douglas-Peucke (RDP) Algorithm, was used to estimate when a vertice is present in the estimated path.

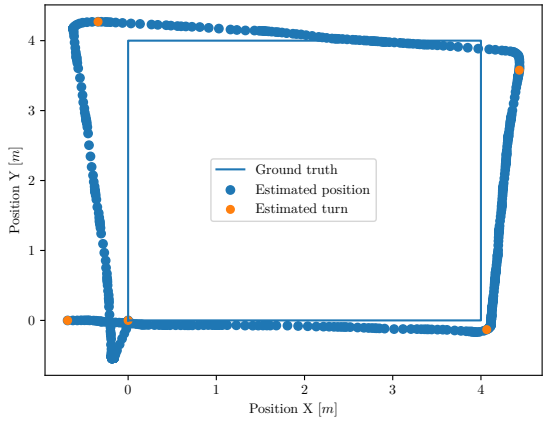
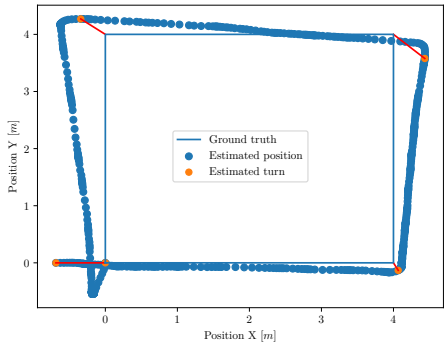
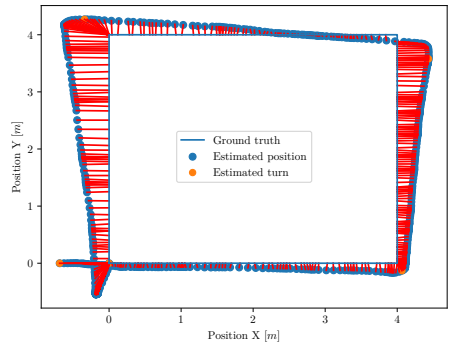


Fig. 24: Illustration of the real-world test ground truth (represented by the $4m \times 4m$ blue line square) and how the estimated path compares. Orange dots represent the estimated corners of the RDP Algorithm.

By measuring how far each estimated turn is from the closest real shapes vertices, it is possible to conclude how well that shape was estimated by the inertial solution. The second way to quantify error is to measure the distance from every estimated point to the closest neighbor in the ground truth's path.



(a) Estimated shape corners are connected to their closest ground truth vertice, averaging all these distances will give a better understanding of how the shape was estimated by the inertial system.



(b) The second error measurement technique where every point in the estimated position path is connected their closest neighbor in the ground truth's line. Averaging every distance here can give a better understanding of how precise the estimation was.

4.0.5 Applying methodology in 3D

Finally, by applying the method to the Z-axis, it is possible to obtain a three-dimensional estimation. The output from X-axis, Y-axis, and Z-axis accelerometer measurements were combined with orientation and integrated to obtain an accumulative position on all axes. A three-dimensional visualization of the previous experiment is visible at figure 26.

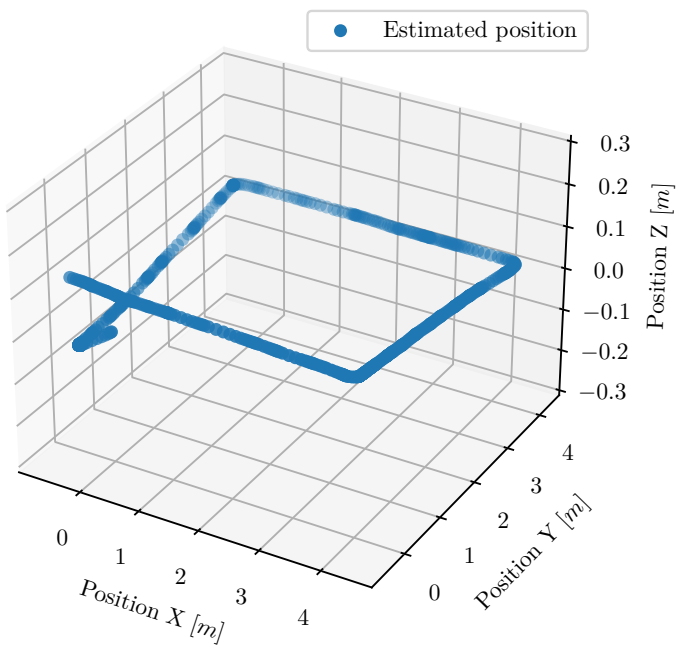


Fig. 26: Three-dimensional view of the experiment.

The same methodology mentioned in the previous chapter can be applied in three dimensions. The RDP algorithm was applied to detect turns in the estimated path. By measuring how far each estimated turn is from the closest real shapes vertices, it is possible to conclude how well that shape was estimated by the inertial solution.

4.0.5.1 Measuring error

To quantify how well a test performed in estimating position, it is necessary to compare it against a ground truth (explained in previous chapter). Error was

quantified in two ways, the first being how far are the estimated shape corners from the ground truth's vertices. A turning detection algorithm, RDP Algorithm, was used to estimate when a vertice is present in the estimated path.

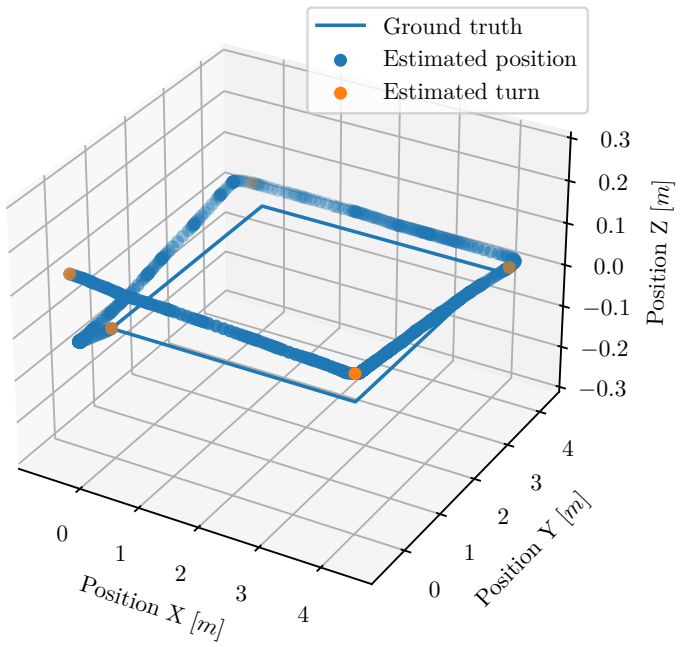
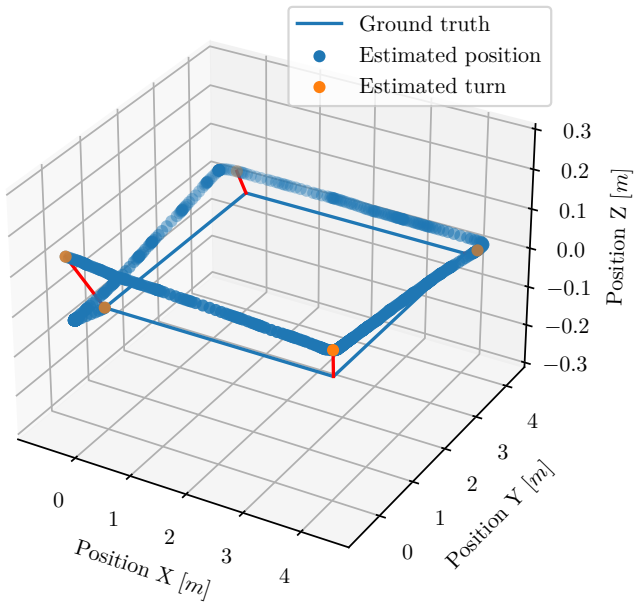
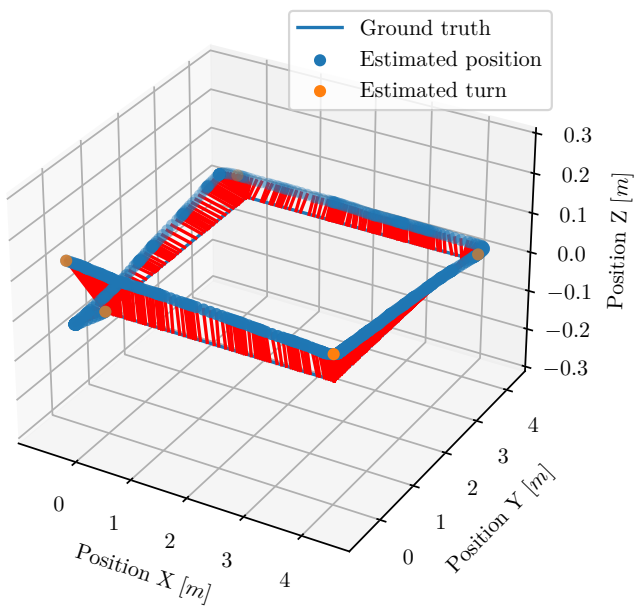


Fig. 27: The same methodology applies to three-dimensional estimation. The real-world test ground truth (represented by the $4m \times 4m$ blue line square) and how the estimated path compares. Orange dots represent the estimated corners of the RDP Algorithm.

The second way to quantify error is to measure the distance from every estimated point to the closest neighbor in the ground truth's path. By averaging every distance Mean Average Error (MAE), it is possible to have an overall understanding of how the estimation performed (equation 40).

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |x_i - x| \tag{40}$$


Ground truth
Estimated position
Estimated turn
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
Position Z [m]
0
1
2
3
4
Position X [m]
0
1
2
3
4
Position Y [m]


Ground truth
Estimated position
Estimated turn
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
Position Z [m]
0
1
2
3
4
Position X [m]
0
1
2
3
4
Position Y [m]

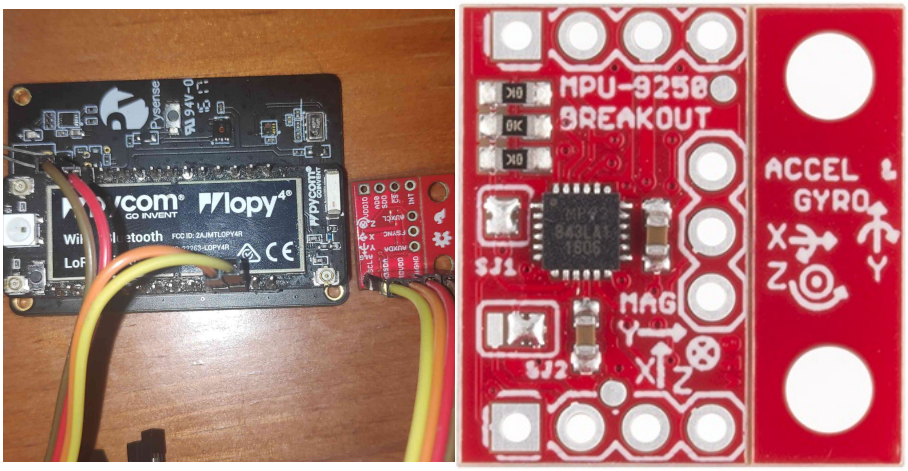
5 Experimental Setup

This section provides a hardware and software experimental setup outline, describing and specifying development and implementation of the low-cost Inertial Navigation System (INS) aimed at position estimation in three-dimensional space. This section will first describe what actual hardware was used and how it was assembled and going through hardware specifications. Followed by a software walkthrough describing what software and libraries were utilized and how hardware and communicate with each other.

5.1 Hardware

As our projected solution involved a low-cost navigation system, the hardware selection criteria were primarily founded on availability and cost. The cost reduction normally concedes in accurateness and reliability, although with the recent surge of inexpensive, widely accessible, and precise microelectromechanical systems (MEMS), that is no longer the case. Still, we aim to employ commercially available tools at the lowest possible cost without compromising the design of a robust and accurate Inertial Navigation System (INS). A LoPy microcontroller was selected as the navigational computing device of the inertial system, meeting the envisioned requirements for low power with flexible and diverse computational capabilities [107]. The LoPy development board interfaces with the external physical inertial sensor through connection pins (figure 29a) and communicates remotely by Long Range Communication (LoRa) protocol. A PySense expansion board connects with the LoPy module providing a programable interface for the microcontroller. The Inertial Navigation System (INS) encompasses a small MPU-9250 (figure 29b).

MPU-9250 IMU is one of the most widely available low cost commercial IMUs. It is also considered to have an exceptional quality price ratio. The MPU-9250 contains temperature and pressure sensors as well. This chip is extensively employed in wearable sensors for health [108] [109], fitness [110], and sports [111] [112] [113], motion-based game controllers [114], and portable gaming [115] [116]. This IMU is operated to estimate motion by identifying the presence of acceleration vectors, rotational rates, and local magnetic field direction. It features an embedded 9-axis MEMS sensor from InvenSense. The MPU-9250 is a combined System-in-package (SIP) comprising an MPU-6050 (3-axis accelerometer and 3-axis gyroscope combination) and an AK8963 3-axis



(a) INS Hardware

(b) MPU-9250 Breakout

Fig. 29: Pin connection between the microcontroller (left) and the IMU (right). Modules are linked through SCL, CLK, VDD and GND pins.

magnetometer [117]. The MPU-6050 coordinate system operates within a traditional cartesian coordinate system with a counter-clockwise rotation as the positive rotation direction [8]. The AK8963 shifts the x and y-axis directions, while reversing the direction of the z-axis [9]. These conventions will be utilized during calibration for each of the three sensors.

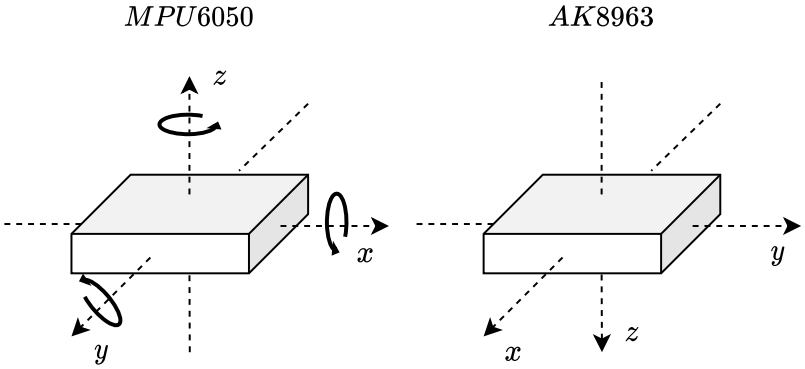


Fig. 30: Coordinate system of an MPU-6050 (3-axis accelerometer and 3-axis gyroscope combination) and AK8963 3-axis magnetometer. [8] [9]

5.1.1 Accelerometer

The triple-axis MEMS accelerometer in MPU-9250 the following features [118]:

- Digital-output triple-axis accelerometer with a programmable full-scale range of $\pm 2g$, ± 4, $\pm 8g$ and $\pm 16g$ and integrated 16-bit ADCs
- Accelerometer normal operating current: $450\mu A$
- The text in the entries may be of any length
- Low power accelerometer mode current: $8.4\mu A$ at $0.98Hz$, $19.8\mu A$ at $31.25Hz$
- Sleep mode current: $8\mu A$
- User-programmable interrupts
- Wake-on-motion interrupt for low power operation of applications processor
- Self-test

Parameter	Min.	Typ.	Max.	Units
Full-Scale range		± 16		g
Sensitivity Scale Factor		2,048		LSB/g
Nonlinearity		± 0.1		%
Rate Noise Spectral Density		300		$\mu g/\sqrt{Hz}$
Operating Current		3.2		mA
Startup Time		20		ms
Output Data Rate	4		4000	Hz

Table 4: Accelerometer Specifications.

5.1.2 Gyroscope

The triple-axis MEMS gyroscope in the MPU-9250 the following features [118]:

- Digital-output X-Axis, Y-Axis, and Z-Axis angular rate sensors (gyroscopes) with a user-programmable fullscale range of ± 250, ± 500, ± 1000, and $\pm 2000°/s$ and integrated 16-bit ADCs
- Digitally programmable low-pass filter
- Gyroscope operating current: $3.2mA$
- Sleep mode current: $8\mu A$

- Factory calibrated sensitivity scale factor
- Self-test

Parameter	Min.	Typ.	Max.	Units
Full-Scale range		± 2000		$^\circ/s$
Sensitivity Scale Factor		131		$LSB/(^\circ/s)$
Nonlinearity		± 0.5		%
Rate Noise Spectral Density		300		$^\circ/s/\sqrt{Hz}$
Operating Current		450		μA
Startup Time		35		ms
Output Data Rate	4		8000	Hz

Table 5: Gyroscope Specifications.

5.1.3 Magnetometer

The triple-axis MEMS accelerometer in MPU-9250 the following features [118]:

- 3-axis silicon monolithic Hall-effect magnetic sensor with magnetic concentrator
- Wide dynamic measurement range and high resolution with lower current consumption.
- Output data resolution of 14 bit ($0.6\mu T/LSB$) or 16 bits ($15\mu T/LSB$)
- Full scale measurement range is $\pm 4800\mu T$
- Magnetometer Normal operating current: $280\mu A$ at $8Hz$ repetition rate
- Self-test function with internal magnetic source to confirm magnetic sensor operation on end products

Parameter	Min.	Typ.	Max.	Units
Full-Scale range		± 4800		μT
Sensitivity Scale Factor		0.6		$\mu T/LSB$
Operating Current		280		μA
Initial Calibration Tolerance		± 500		LSB

Table 6: Magnetometer specification.

5.2 Software

The microcontroller operates MicroPython, a barebones and efficient implementation of Python 3, which incorporates a small subset of the Python standard library [119]. It is optimized to run on microcontrollers and in constrained environments [120]. The inertial module's raw measurements are interpreted by the microcontroller through Inter-Integrated Circuit (I^2C) MicroPython driver serial allowing to read the peripherals memory addresses synchronously. The I^2C communication makes use of the Serial Data (SDA) and Serial Clock (SCL) pins of the MPU9250. The SCL offers the serial clock, and the SDA gives the serial data. The memory address of the MPU9250 is at address is 0x68 when the pin is connected to ground and is set by pin AD0. The AK8963's address is by default at 0x0C. Prior to the communication begins the SCL and SDA are equally high, triggered by the pull-up resistors. These pull-up resistors are arranged to make specific pins turn out to be high when not being defined by the LoPy, this avoids communication difficulties when the LoPy is waiting for inbound serial data. When the transmission between the LoPy (master) and the MPU9250 (slave) initializes the start order is produced. This is accomplished by lowering SDA, see figure 31. Later the SCL commences to produce clock pulses and the initial byte is sent out by the master. This first byte holds the 7-bit address of the slave and a read/write bite. If the read/write bit is 0 the master writes to the slave. If the read/write bit is 1 the master reads from the slave. The bits can only be sent when the serial clock is high. Each byte sent should be followed by an acknowledge bit. When the master is done writing/reading then the stop condition is created.

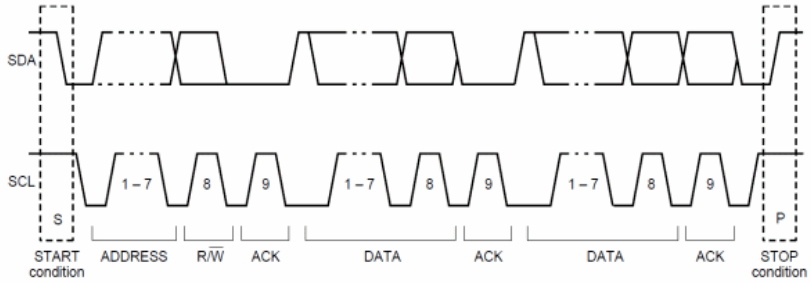


Fig. 31: I^2C protocol.

The readings of each sensor are later averaged and linearized to better detect and reduce the presence of outlier readings [121]. A fusion algorithm takes as input the averaged data of the accelerometer, gyroscope, and magnetometer. It returns the estimated inertial angles (pitch, roll, and yaw) as well as the projected linear acceleration with a gravity compensation numerical method that utterly removes the effect of the gravity component. Numerically integrating the resultant linear acceleration yields velocity, and double integrating will deliver the body's accumulative position (more in-depth explanation at chapter 4 Methodology). Merging the AHRS with an accumulative position allows tracking a moving body in three dimensions over time. To perform sensor fusion, two methodologies were used: the first relates our implementation of Madgwick's filter to conduct real-time sensor fusion in the microcontroller, which in a future study might be used to communicate in real-time a body's position. The Madgwick algorithm was utilized for its well-recognized proficiency to merge accuracy with computational cost and simplicity of implementation. But since the study aimed at comparing how different sensor fusion approaches perform under the same data, we needed a second approach which meant storing the raw sensor data in an SD card and process it later the experiment with a sensor fusion library, AHRS: Attitude and Heading Reference Systems [11], an open-source Python toolbox for attitude estimation using the most known algorithms, methods, and resources. To benchmark how different sensor fusion algorithms perform under the same data. Available sensor fusion algorithms in AHRS library and what sensors they use to perform AHRS estimation is present at table 7.

5.3 Calibration

A correct calibration of such sensors is essential for the compensation of their systematic errors, bias, and scale factor. Each time prior to an experiment, the inertial sensor is calibrated while the system is stationary and stabilized to compensate for static error that might corrupt the measurements.

5.3.1 Accelerometer Calibration

Calibration of the accelerometer requires taking advantage of the acceleration due to gravity, which we can use in the positive and negative orientation of the IMU [122]. Additionally, we can also position the IMU perpendicular to gravity in order to acquire a third calibration point. This results in three unique values

Algorithm	Accelerometer	Gyroscope	Magnetometer
Angular Rate	Yes	No	No
AQUA	Yes	Yes	Optional
Complementary	Yes	Yes	Optional
Davenport's	Yes	No	Yes
EKF	Yes	Yes	Yes
FAMC	Yes	No	Yes
FLAE	Yes	No	Yes
Fourati	Yes	Yes	Yes
Madgwick	Yes	Yes	Optional
Mahony	Yes	Yes	Optional
OLEQ	Yes	No	Yes
QUEST	Yes	No	Yes
ROLEQ	Yes	No	Yes
SAAM	Yes	No	Yes
Tilt	Yes	No	Optional

Table 7: Available sensor fusion algorithms in AHRS library and what sensors they use. [11]

that can be combined to formulate a linear fit between the three values and the values outputted by each axis of the accelerometer.

```
count=256
aox, aoy, aoz = (0.0, 0.0, 0.0)
self._accelerometer_offset = (0.0, 0.0, 0.0)
n = float(count)

while count:
    utime.sleep_ms(delay)
    # taking samples for a period of time
    ax, ay, az = self.acceleration
    # sum every sample on each axis
    aox += ax
    aoy += ay
    aoz += az
    count -= 1

# average the samples taken
self._accelerometer_offset = (aox / n, aoy / n, aoz / n)
return self._accelerometer_offset
```

The result of the accelerometer's calibration offset correction is present at figure 32 where the calibrated output clearly shows an offset correction with the a_x and a_y readings centered at 0 ms^{-2} while the a_z revolving at the 9.8 ms^{-2}.

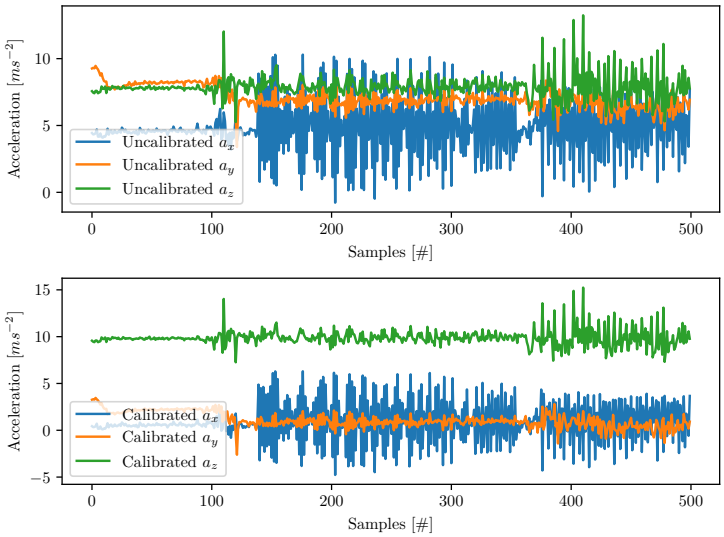


Fig. 32: Accelerometer calibration offset correction.

5.3.2 Gyroscope Calibration

The simplest calibration of an IMU consists of calculating the offset for each axis of the gyroscope. The gyroscope is the easiest calibration due to the expected readings outputted under steady conditions. Each of the three axes of the gyro should read 0 radians-per-second (rad/s) when the IMU is not moving. The offsets can be measured by first taking some readings while the IMU is not moving, then using those values as 'offsets' when reading the gyro values in the future. This is merely the simplest calibration method for the IMU and suffices for most casual uses of the gyroscope and IMU [123].

```
count=256
gox, goy, goz = (0.0, 0.0, 0.0)
self._gyro_offset = (0.0, 0.0, 0.0)

while count:
    gx, gy, gz = self.gyro
    # sum every sample on each axis
    gox += gx
    goy += gy
    goz += gz
    count -= 1

# average the samples taken
self._gyro_offset = (gox / count, goy / count, goz / count)
return self._gyro_offset
```

The result of the gyroscope's calibration offset correction is present at figure 33 where the calibrated output clearly shows an offset correction with the ω_x, ω_y and ω_z readings centered at 0 rad/s.

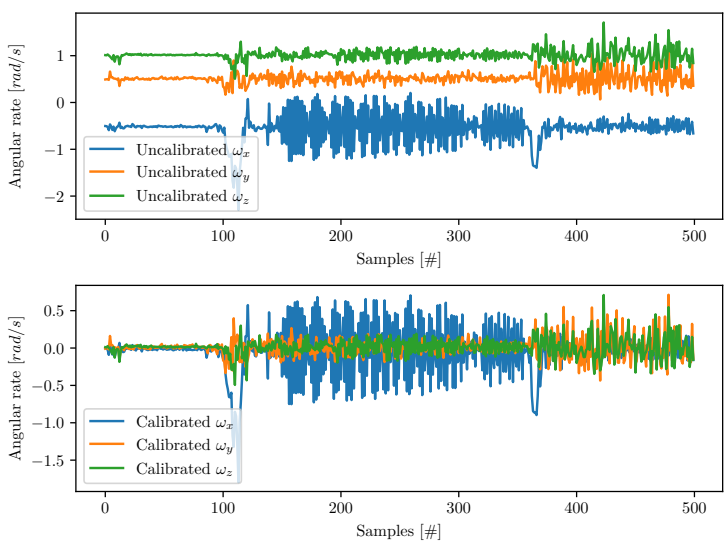


Fig. 33: Gyroscope calibration offset correction.

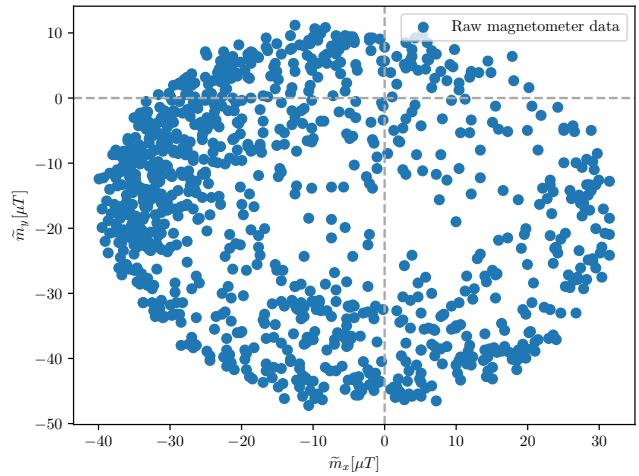
5.3.3 Magnetometer Calibration

To calibrate the magnetometer, a series of measurements of the nearby magnetic field are taken while holding the sensor in an eight-figure pattern through every orientation possible. Ideally the measurements should portray a perfect sphere centered at the origin and where the radius of the sphere is the magnetic field's strength [124]. Plotting the magnetometer's raw measurements (as seen in figure 34a and figure 34b), it is visible that these do not form a perfect sphere, nor are they centered at the origin. These are referred to as hard iron and soft iron errors or biases, respectively. If no magnetic anomaly algorithm is applied to perceive and avoid it, such distortions in the magnetic field can be interpreted as changes in orientation.

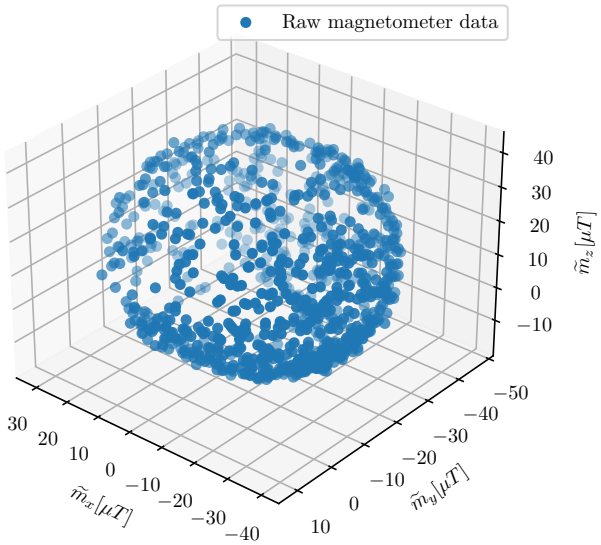
Hard iron biases are commonly the largest and the simplest errors to rectify. With the magnetic field measurements, it is straightforward to compensate for hard iron biases by keeping track of the minimum and maximum field measured in $\widetilde{m}_x$, $\widetilde{m}_y$ and $\widetilde{m}_z$. Once the minimum and maximum field measured in $\widetilde{m}_x$, $\widetilde{m}_y$ and $\widetilde{m}_z$ are known, the average can be subtracted from the following data which results in the re-centering of the response surface on the origin. The following script illustrates how a hard iron correction could be implemented:

```
# Hard iron correction

# get average x mag bias in counts
offset_x = (maxx + minx) / 2
# get average y mag bias in counts
offset_y = (maxy + miny) / 2
# get average z mag bias in counts
offset_z = (maxz + minz) / 2

self._offset = (offset_x, offset_y, offset_z)
return self._offset
```

It is possible now to filter any soft iron biases by taking the minimum and maximum field measured already computed and manipulate them to rescale the magnetometer data to normalize the output across $\widetilde{m}_x$, $\widetilde{m}_y$ and $\widetilde{m}_z$. It is possible to accomplish this by calculating the scale factor with the ratio of the average max - min through each axis and the numerical mean of all three axes. This means, for instance, if $x_{max} - x_{min}$ ratio is considerable, $\widetilde{m}_x$ has its magnetic



(a) Magnetometer measurements of magnetic field strength along orthogonal X and Y-axis.



(b) Magnetometer measurements of magnetic field strength along orthogonal X, Y and Z-axis.

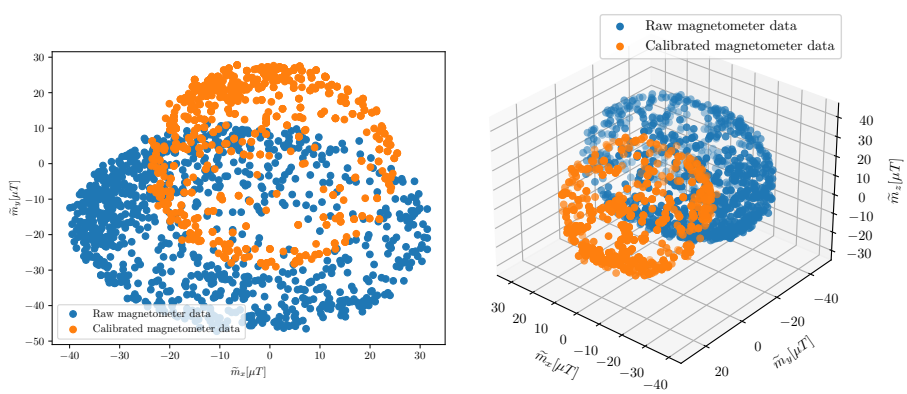
Fig. 34: Raw magnetometer measurements.

field scale reduced, or if the $y_{max} - y_{max}$ ratio is smaller compared to the other axes, it has its magnetic field values increased. This is known as orthogonal

rescaling, identical to a diagonalized 3 x 3 calibration matrix while enabling further scale bias correction.

```
# Soft iron correction
# get average axis max chord length in counts
avg_delta_x = (maxx - minx) / 2
avg_delta_y = (maxy - miny) / 2
avg_delta_z = (maxz - minz) / 2

avg_delta = (avg_delta_x + avg_delta_y + avg_delta_z)
avg_delta /= 3

scale_x = avg_delta / avg_delta_x
scale_y = avg_delta / avg_delta_y
scale_z = avg_delta / avg_delta_z

self._scale = (scale_x, scale_y, scale_z)
return self._scale
```

The result of rescaling the MPU9250 magnetometer data is shown at figure 35:



(a) Raw (blue) and calibrated (red) magnetometer measurements of magnetic field strength along orthogonal X, Y-axis.

(b) Raw (blue) and calibrated (red) magnetometer measurements of magnetic field strength along orthogonal X, Y and Z-axis.

Fig. 35: Raw and calibrated magnetometer measurements.

5.4 Experiments

Several experimental tests were conducted to assess the position performance of the INS under different conditions. Experiments were performed indoors, outdoors, and underwater. These tests typically consisted of carrying the INS in a movable platform, typically a skateboard or an electric scooter forming a geometric shape such as a square or a triangle following a precise path on the floor. Geometric shapes were used to better assess the estimating performance of the system against a ground truth baseline. This section will describe how each set of experiments were carried out and what challenges were faced.

5.5 Indoor Experiments

Indoor experiments were executed at Madeira Tecnopolo building, floor -1, in a wide-open interior. A guiding rope was layout out in the floor forming a geometric figure acting as baseline for the experiment. The inertial system was placed on an electrical skateboard powered by a laptop computer, which could have been replaced by a portable battery pack. Three kinds of geometric shapes were tested: line, square, triangle. To better understand how distance affects estimation of position, for every geometric figure, tests were made under two different distance magnitudes: 4 and 16 meters.

Fig. 36: 4-meter side ground square used for baseline of accuracy of inertial system.

Once the experiment was set up, it was time to start testing the INS. The board was dragged at constant speed through the guideline transporting the inertial system. The INS was continuously estimating orientation and position several times a second and storing data to an SD card. The board was dragged at walking speed and stopped at corners when direction changed.

Fig. 37: Skateboard being dragged along the guideline.

5.6 Outdoor Experiments

Outdoor experiments were conducted at praça do povo's square (figure 38), in a sport's platform with a guiding rope also acting as baseline for the experiment in the forming of geometric figures on the ground.

Fig. 38: Platform where outdoors experiments were conducted.

Here, the sports platform consisted of one 28mx28m square, composed of 4 meter squares, forming a 7x7 grid (figure 39). This was helpful since this knowledge could also be used to benchmark the position estimation of the system and easily calculate error margins.

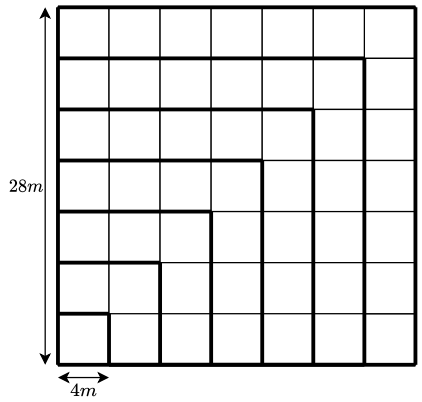
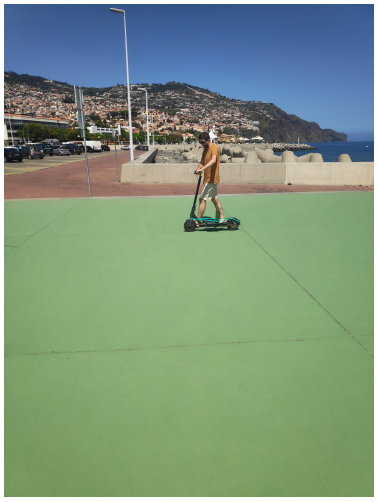


Fig. 39: Bird's-eye view representation of the 7x7 grid platform where outdoor experiments were conducted.

The inertial system was placed on an electrical scooter (figure 40a) in this case, being powered by a laptop computer (figure 40b). The scooter was dragged at walking pace (figure 40c) along the planned path marked by the rope.

(a) Spiral experiment path outlined by the white rope.

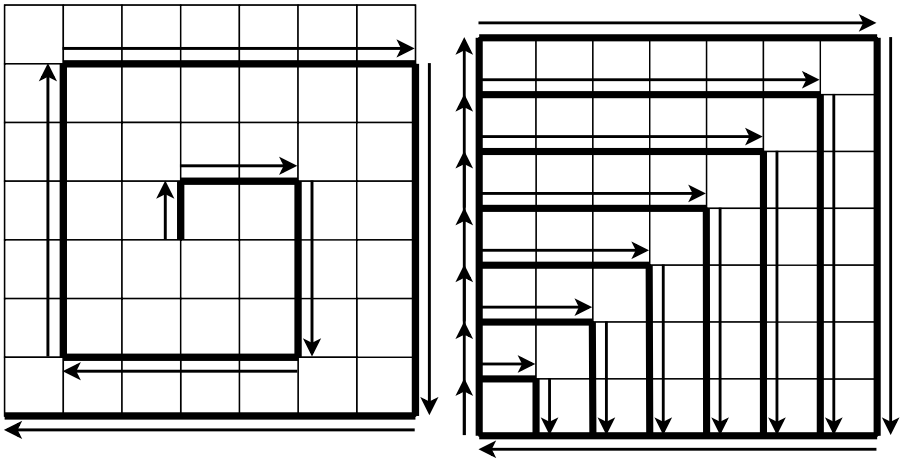
(b) Close-up view of the inertial system (white box) attach with an elastic band to the scooter's body frame.



(c) Pushing the electrical scooter containing the laptop and inertial system along the platform's edges forming geometrical shapes.

Fig. 40: Photos illustrating how outdoor experiments were performed.

Here several shapes were tested, from simple lines, squares, and triangles to more complex figures such as the ones illustrated at figure 41b and figure 41a.



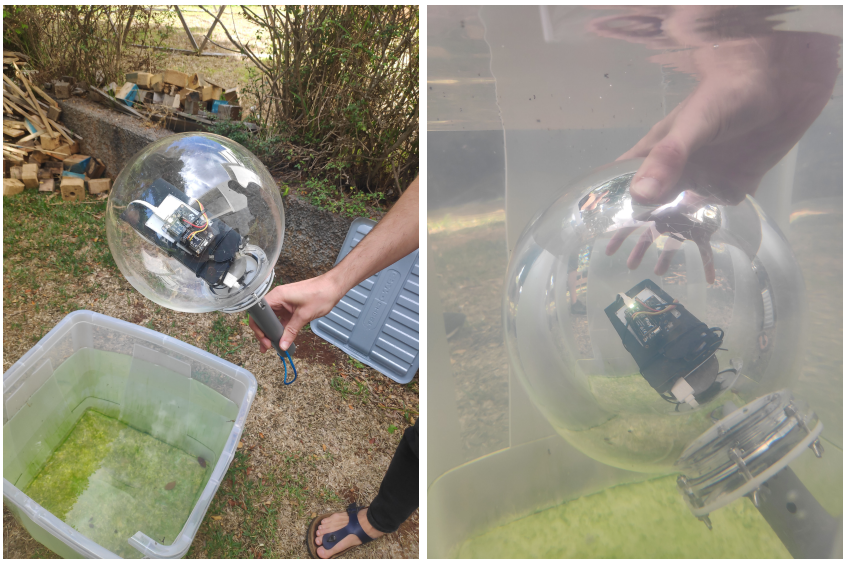
(a) Spiral-based shape with increasing length sides.

(b) Squares of increasing side length.

Fig. 41: Bird's-eye view representation of the more complex path shapes taken.

5.7 Underwater Experiments

Underwater experiments were carried out at Madeira Carlton hotel divepoint, where the INS was placed in an underwater spherical housing built for 360∘cameras with a portable battery pack (figure 42a). Since the hardware was inaccessible inside the housing, the software had to be adjusted to support remote triggering of the INS via a wireless hotspot using a mobile phone via an HTTP request signal. Tests of the housing were performed prior to the experiment verifying the presence of any water leaks and the well-functioning of the INS (figure 42b).



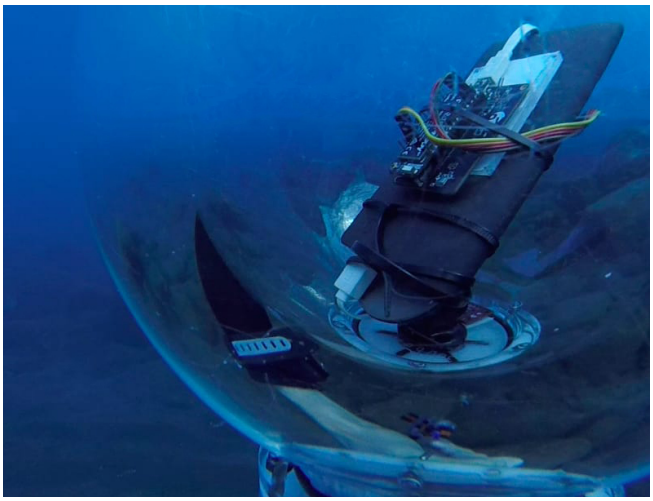
(a) Spherical housing encompassing the inertial system and portable battery pack.

(b) The capsule was submerged in a water tank to verify the presence of any water leaks.

Fig. 42: View of the tests made to assure the underwater housing was waterproof.

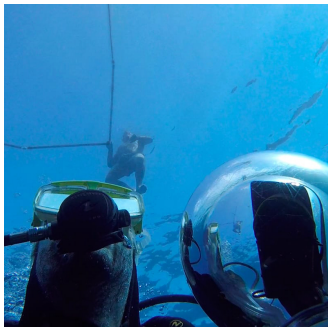
Once safety checks were completed, guaranteeing the feasibility of the study, the team went to Madeira Carlton hotel divepoint, where tests were conducted to evaluate the estimation quality of the proposed solution under water by a professional diver (figure 43a). A 5-meter square net being held by two people above water served as guideline for the diver performing the experiment under-

water. The diver held the spherical housing while swimming along the guideline maintaining a constant speed while slowly increasing depth (figure 43c).



(a) The housing being held underwater by the diver.

(b) The capsule housing the inertial system being handed out to a professional diver.



(c) View of the diver following the net underwater holding the spherical housing.

Fig. 43: Photos of the underwater tests conducted at Madeira Carlton hotel divepoint.

6 Results

This section presents results from the different set of experiments mentioned in chapter 5.4 Experiments. For every experiment, there is a table comparing every algorithm's displacement and turn error, in absolute and relative measurements. And lastly, a 3D visualization of the best performing algorithm (the one with the lowest displacement error and turn error respectively) for that experiment is present for every test. This chapter is divided by geometric shapes: lines, triangles, squares, each with their respective distances.

6.1 Line

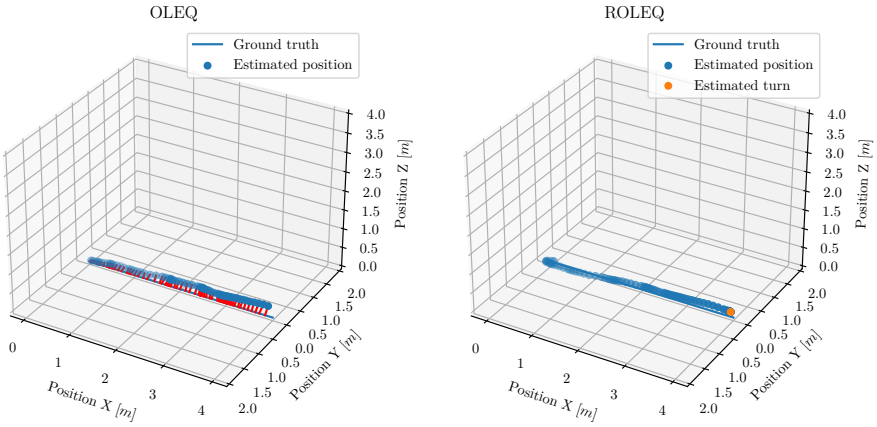
The line shape consisted of moving the inertial system in a straight line for a determined distance. Three line distances were tested: 4, 16, and 28 meter. The results are shown below:

6.1.1 4 meter

For the 4-meter line experiment, the OLEQ algorithm which had the lowest displacement error with an average of 0.13 meters (3.24% of error margin), and ROLEQ with an average of 0.24 meters of turn error (6.06% of error margin).

Algorithm	Displacement Error[m]	Displacement Error[%]	Turn Error[m]	Turn Error[%]
AngularRate	0.16	3.90	0.50	12.62
AQUA	0.90	22.59	1.46	36.50
Complementary	0.34	8.45	0.55	13.66
Davenport	0.49	12.28	0.64	16.02
EKF	1.01	25.28	1.35	33.73
FAMC	0.17	4.25	0.50	12.57
FLAE	0.49	12.29	0.64	16.04
Fourati	0.32	8.01	0.54	13.39
Madgwick	0.74	18.57	0.74	18.51
Mahony	0.25	6.21	0.55	13.80
OLEQ	0.13	3.24	0.41	10.28
QUEST	2.14	53.59	2.07	51.69
ROLEQ	0.16	4.06	0.24	6.06
SAAM	0.34	8.53	0.55	13.82
Tilt	0.34	8.53	0.55	13.82
Average	0.53	13.32	0.75	18.84

Table 8: 4 meter line position estimation error (displacement and turn) of the sensor fusion algorithms.



(a) OLEQ's 3D position estimation had the lowest displacement error for the 4-meter line experiment.

(b) ROLEQ's 3D position estimation had the lowest turn error for the 4-meter line experiment.

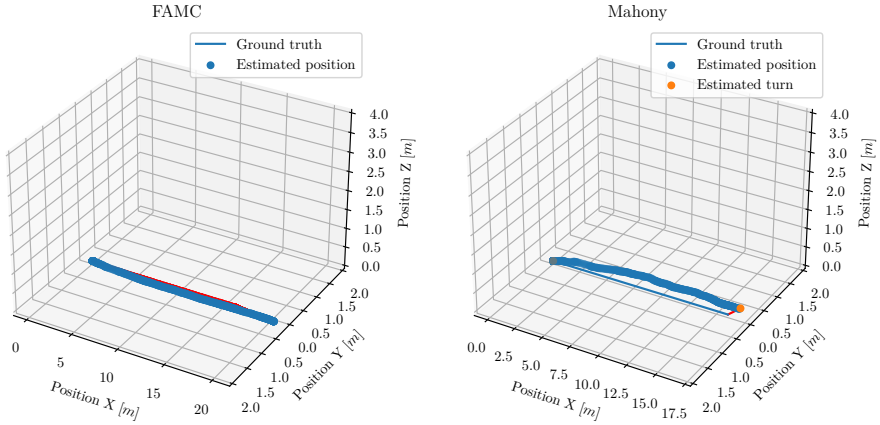
Fig. 44: Position estimation by the best performing algorithms in the 4-meter line experiment.

6.1.2 16 meter

For the 16-meter line experiment, the FAMC algorithm which had the lowest displacement error with an average of 0.43 meters (2.69% of error margin), and Mahony with an average of 2.11 meters of turn error (13.21% of error margin).

Algorithm	Displacement Error[m]	Displacement Error[%]	Turn Error[m]	Turn Error[%]
AngularRate	0.89	5.59	3.97	24.78
AQUA	6.41	40.04	9.27	57.92
Complementary	0.49	3.08	2.16	13.47
Davenport	0.51	3.17	2.16	13.52
EKF	0.67	4.17	2.22	13.86
FAMC	0.43	2.69	2.16	13.49
FLAE	0.51	3.17	2.16	13.52
Fourati	1.09	6.83	4.80	29.98
Madgwick	0.49	3.05	2.15	13.47
Mahony	0.48	3.01	2.11	13.21
OLEQ	0.68	4.28	3.54	22.14
QUEST	2.61	16.33	3.79	23.72
ROLEQ	0.72	4.51	3.56	22.26
SAAM	0.48	3.02	2.14	13.35
Tilt	0.48	3.02	2.14	13.35
Average	1.13	7.06	3.22	20.14

Table 9: 16 meter line position estimation error (displacement and turn) of the sensor fusion algorithms.



(a) FAMC's 3D position estimation had the lowest displacement error for the 16-meter line experiment.

(b) Mahony's 3D position estimation had the lowest turn error for the 16-meter line experiment.

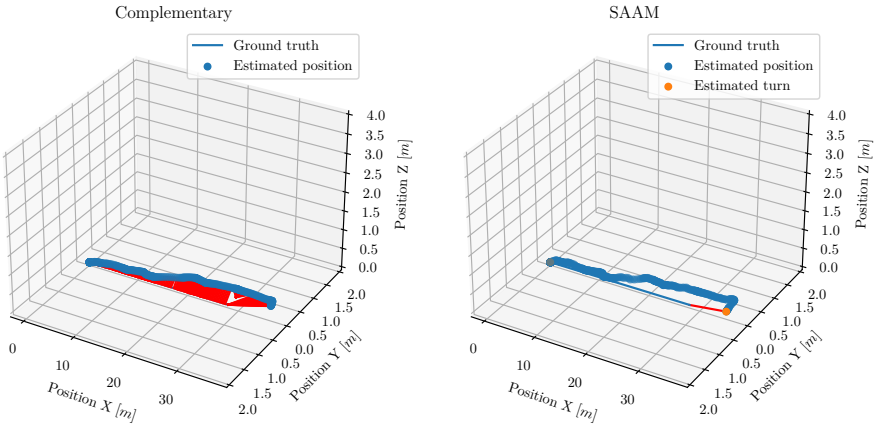
Fig. 45: Position estimation by the best performing algorithms in the 16-meter line experiment.

6.1.3 28 meter

For the 28-meter line experiment, the Complementary algorithm which had the lowest displacement error with an average of 0.52 meters (1.85% of error margin), and SAAM with an average of 4.23 meters of turn error (15.09% of error margin).

Algorithm	Displacement Error[m]	Displacement Error[%]	Turn Error[m]	Turn Error[%]
AngularRate	1.29	4.61	6.08	21.73
AQUA	8.66	30.93	14.77	52.75
Complementary	0.52	1.85	4.31	15.41
Davenport	0.73	2.60	5.47	19.53
EKF	0.76	2.73	4.35	15.55
FAMC	0.50	1.80	4.33	15.45
FLAE	0.73	2.59	5.47	19.53
Fourati	1.07	3.81	6.21	22.18
Madgwick	0.55	1.96	4.29	15.32
Mahony	0.53	1.88	4.19	14.98
OLEQ	0.69	2.47	5.35	19.11
QUEST	3.08	11.01	11.72	41.87
ROLEQ	0.83	2.95	5.39	19.24
SAAM	0.51	1.81	4.23	15.09
Tilt	0.51	1.81	4.23	15.09
Average	1.40	4.99	6.03	21.52

Table 10: 28 meter line position estimation error (displacement and turn) of the sensor fusion algorithms.



(a) Complementary's 3D position estimation had the lowest displacement error for the 28-meter line experiment.

(b) SAAM's 3D position estimation had the lowest turn error for the 4-meter line experiment.

Fig. 46: Position estimation by the best performing algorithms in the 28-meter line experiment.

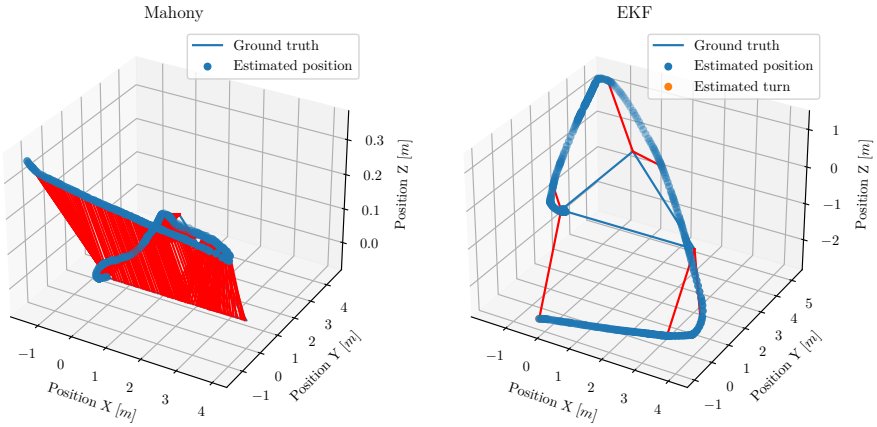
6.2 Triangle

6.2.1 4 meter

For the 4-meter triangle experiment, the Mahony algorithm which had the lowest displacement error with an average of 0.90 meters (7.50% of error margin), and EKF with an average of 0.49 meters of turn error (4.09% of error margin).

Algorithm	Displacement Error[m]	Displacement Error[%]	Turn Error[m]	Turn Error[%]
AngularRate	4.53	37.72	7.35	61.28
AQUA	3.10	25.82	4.27	35.61
Complementary	3.85	32.08	3.43	28.59
Davenport	1.12	9.30	1.39	11.57
EKF	1.24	10.31	0.49	4.09
FAMC	4.43	36.90	7.12	59.37
FLAE	1.09	9.08	1.32	11.00
Fourati	7.36	61.33	8.47	70.60
Madgwick	1.94	16.15	1.02	8.51
Mahony	0.90	7.50	0.99	8.28
OLEQ	1.18	9.81	1.03	8.58
QUEST	2.84	23.65	2.58	21.50
ROLEQ	1.17	9.71	0.85	7.06
SAAM	1.05	8.71	1.08	9.01
Tilt	1.05	8.71	1.08	9.01
Average	2.45	20.45	2.83	23.60

Table 11: 4 meter triangle position estimation error (displacement and turn) of the sensor fusion algorithms.



(a) Mahony's 3D position estimation had the lowest turn displacement for the 4-meter side triangle experiment.

(b) EFK's 3D position estimation had the lowest turn error for the 4-meter side triangle experiment.

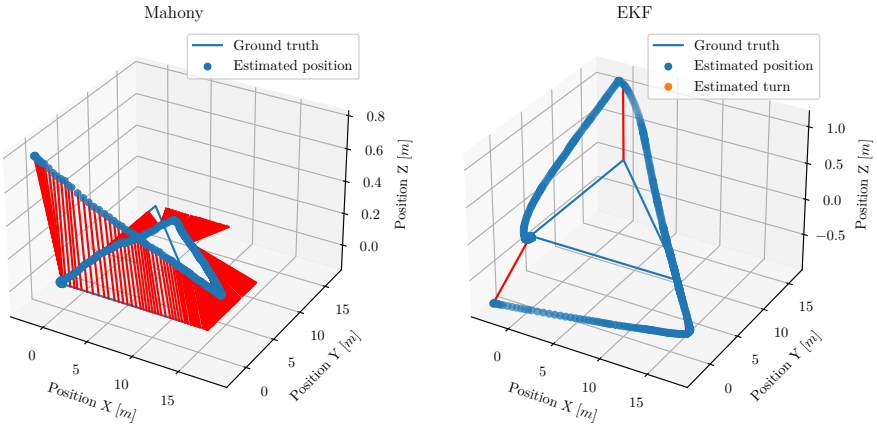
Fig. 47: Position estimation by the best performing algorithms in the 4-meter side triangle experiment.

6.2.2 16 meter

For the 16-meter triangle experiment, the Mahony algorithm which had the lowest displacement error with an average of 1.79 meters (3.74% of error margin), and EKF with an average of 1.59 meters of turn error (3.32% of error margin).

Algorithm	Displacement Error[m]	Displacement Error[%]	Turn Error[m]	Turn Error[%]
AngularRate	12.84	26.74	24.84	51.76
AQUA	7.81	16.28	12.64	26.33
Complementary	2.60	5.41	3.30	6.87
Davenport	2.49	5.18	2.37	4.93
EKF	2.44	5.09	1.59	3.32
FAMC	11.93	24.86	23.55	49.06
FLAE	2.48	5.18	2.27	4.73
Fourati	23.17	48.26	29.76	62.01
Madgwick	6.88	14.32	8.29	17.28
Mahony	1.79	3.74	1.87	3.90
OLEQ	2.43	5.05	1.68	3.50
QUEST	9.51	19.82	11.48	23.92
ROLEQ	2.56	5.34	2.19	4.57
SAAM	1.88	3.91	1.81	3.77
Tilt	1.88	3.91	1.81	3.77
Average	6.18	12.87	8.63	17.98

Table 12: 16 meter triangle position estimation error (displacement and turn) of the sensor fusion algorithms.



(a) Mahony's 3D position estimation had the lowest displacement error for the 16-meter side triangle experiment.

(b) EKF's 3D position estimation had the lowest turn error for the 16-meter side triangle experiment.

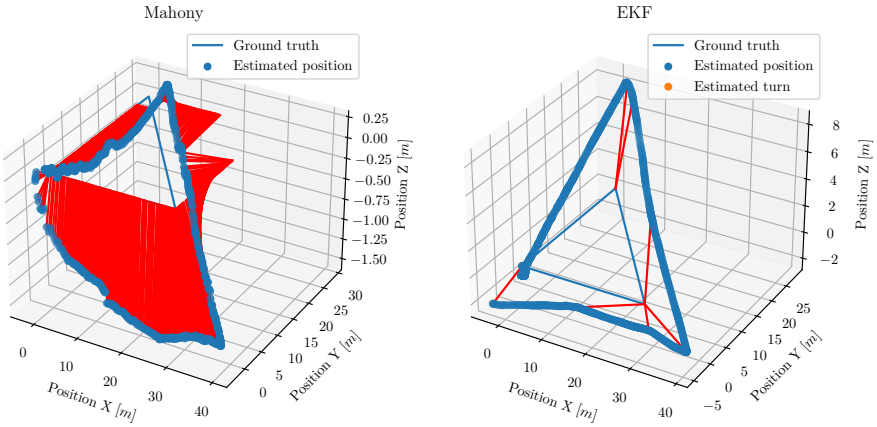
Fig. 48: Position estimation by the best performing algorithms in the 16-meter side triangle experiment.

6.2.3 28 meter

For the 28-meter triangle experiment, the Mahony algorithm which had the lowest displacement error with an average of 2.93 meters (3.49% of error margin), and EKF with an average of 5.04 meters of turn error (5.99% of error margin).

Algorithm	Displacement Error[m]	Displacement Error[%]	Turn Error[m]	Turn Error[%]
AngularRate	22.91	27.28	48.97	58.29
AQUA	6.77	8.06	15.53	18.49
Complementary	5.69	6.78	10.22	12.16
Davenport	3.93	4.68	5.06	6.02
EKF	3.52	4.19	5.04	5.99
FAMC	21.07	25.08	46.74	55.64
FLAE	3.88	4.62	4.98	5.93
Fourati	35.63	42.41	53.27	63.42
Madgwick	15.58	18.55	17.78	21.16
Mahony	2.93	3.49	4.69	5.59
OLEQ	4.69	5.58	5.05	6.01
QUEST	15.18	18.08	30.32	36.09
ROLEQ	4.38	5.22	5.40	6.43
SAAM	2.93	3.48	4.54	5.40
Tilt	2.93	3.48	4.54	5.40
Average	10.14	12.07	17.47	20.80

Table 13: 28 meter square position estimation error (displacement and turn) of the sensor fusion algorithms.



(a) Mahony's 3D position estimation had the lowest displacement error for the 28-meter side triangle experiment.

(b) EKF's 3D position estimation had the lowest turn error for the 28-meter side triangle experiment.

Fig. 49: Position estimation by the best performing algorithms in the 28-meter side triangle experiment.

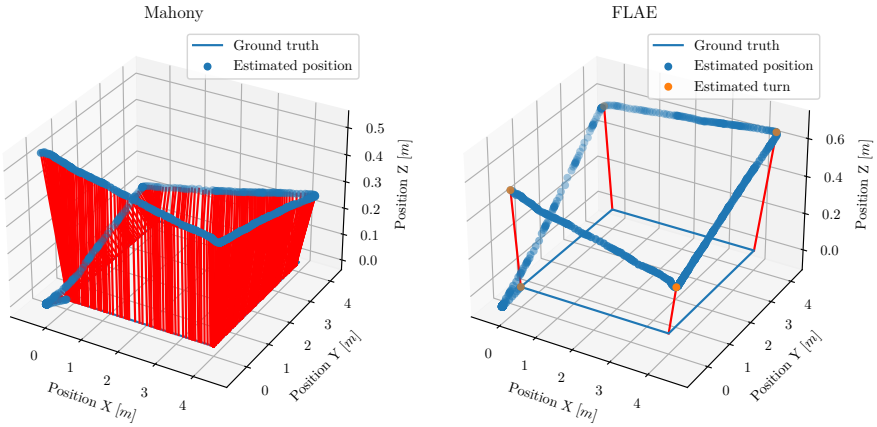
6.3 Square

6.3.1 4 meter

For the 4-meter square experiment, the Mahony algorithm which had the lowest displacement error with an average of 0.53 meters (3.34% of error margin), and FLAE with an average of 0.38 meters of turn error (2.37% of error margin).

Algorithm	Displacement Error[m]	Displacement Error[%]	Turn Error[m]	Turn Error[%]
AngularRate	5.97	37.29	7.29	45.58
AQUA	4.12	25.75	4.20	26.23
Complementary	1.20	7.50	1.46	9.13
Davenport	0.69	4.31	0.43	2.70
EKF	1.09	6.79	1.53	9.56
FAMC	6.01	37.57	6.98	43.61
FLAE	0.69	4.32	0.38	2.37
Fourati	9.08	56.78	9.74	60.87
Madgwick	1.02	6.40	1.18	7.36
Mahony	0.53	3.34	0.37	2.29
OLEQ	0.60	3.74	0.52	3.23
QUEST	5.07	31.71	5.57	34.79
ROLEQ	0.78	4.90	0.88	5.48
SAAM	0.59	3.69	0.37	2.33
Tilt	0.59	3.69	0.37	2.33
Average	2.54	15.85	2.75	17.19

Table 14: 4 meter square position estimation error (displacement and turn) of the sensor fusion algorithms.



(a) Mahony's 3D position estimation had the lowest displacement error for the 4-meter side square experiment.

(b) FLAE's 3D position estimation had the lowest turn error for the 4-meter side square experiment.

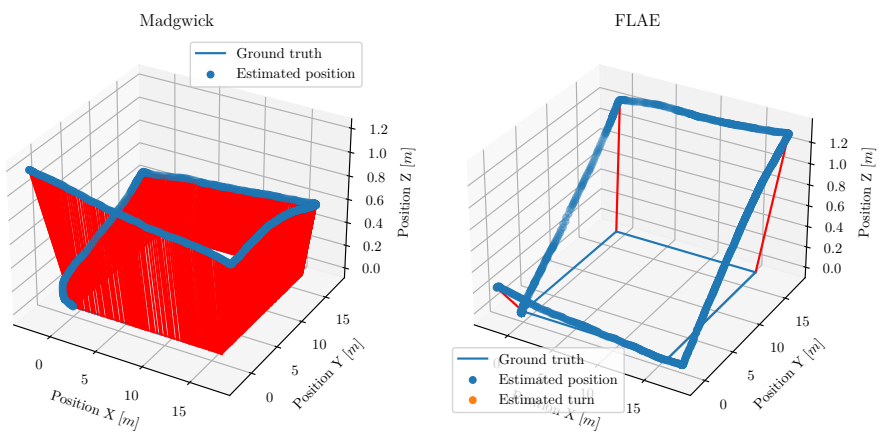
Fig. 50: Position estimation by the best performing algorithms in the 4-meter side square experiment.

6.3.2 16 meter

For the 16-meter square experiment, the Madgwick algorithm which had the lowest displacement error with an average of 1.93 meters (3.02% of error margin), and FLAE with an average of 1.82 meters of turn error (2.84% of error margin).

Algorithm	Displacement Error[m]	Displacement Error[%]	Turn Error[m]	Turn Error[%]
AngularRate	21.87	34.17	26.78	41.85
AQUA	10.13	15.83	10.25	16.02
Complementary	15.26	23.85	13.69	21.40
Davenport	2.33	3.65	2.15	3.36
EKF	2.32	3.63	2.35	3.67
FAMC	20.73	32.40	25.26	39.47
FLAE	2.09	3.27	1.82	2.84
Fourati	34.16	53.37	36.37	56.83
Madgwick	1.93	3.02	1.98	3.09
Mahony	1.94	3.02	2.13	3.33
OLEQ	3.52	5.50	2.64	4.12
QUEST	17.97	28.07	19.53	30.52
ROLEQ	3.64	5.69	2.76	4.32
SAAM	2.15	3.36	2.12	3.31
Tilt	2.15	3.36	2.12	3.31
Average	9.48	14.81	10.13	15.83

Table 15: 16 meter square position estimation error (displacement and turn) of the sensor fusion algorithms.



(a) Madgwick's 3D position estimation had the lowest displacement error for the 16-meter side square experiment.

(b) FLAE's 3D position estimation had the lowest turn error for the 16-meter side square experiment.

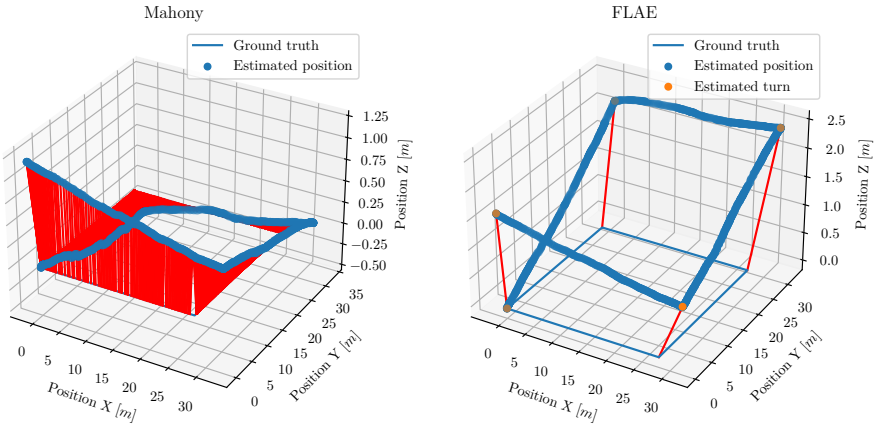
Fig. 51: Position estimation by the best performing algorithms in the 16-meter side square experiment.

6.3.3 28 meter

For the 28-meter square experiment, the Mahony algorithm which had the lowest displacement error with an average of 2.97 meters (2.65% of error margin), and FLAE with an average of 3.23 meters of turn error (2.89% of error margin).

Algorithm	Displacement Error[m]	Displacement Error[%]	Turn Error[m]	Turn Error[%]
AngularRate	35.00	31.25	46.96	41.92
AQUA	12.63	11.28	13.72	12.25
Complementary	15.95	14.24	16.54	14.77
Davenport	3.89	3.47	3.20	2.86
EKF	5.17	4.62	5.05	4.50
FAMC	32.67	29.17	45.35	40.49
FLAE	3.84	3.43	3.23	2.89
Fourati	57.83	51.63	66.25	59.15
Madgwick	4.03	3.60	3.58	3.20
Mahony	2.97	2.65	4.64	4.15
OLEQ	5.55	4.95	4.63	4.13
QUEST	26.64	23.78	32.23	28.78
ROLEQ	5.69	5.08	5.40	4.82
SAAM	3.22	2.87	3.37	3.01
Tilt	3.22	2.87	3.37	3.01
Average	14.55	12.99	17.17	15.33

Table 16: 28 meter square position estimation error (displacement and turn) of the sensor fusion algorithms.



(a) Mahony's 3D position estimation had the lowest displacement error for the 28-meter side square experiment.

(b) FLAE's 3D position estimation had the lowest turn error for the 28-meter side square experiment.

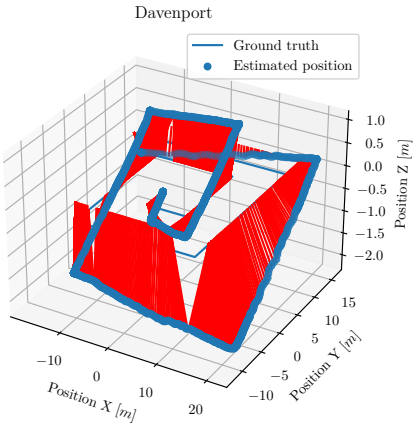
Fig. 52: Position estimation by the best performing algorithms in the 28-meter side square experiment.

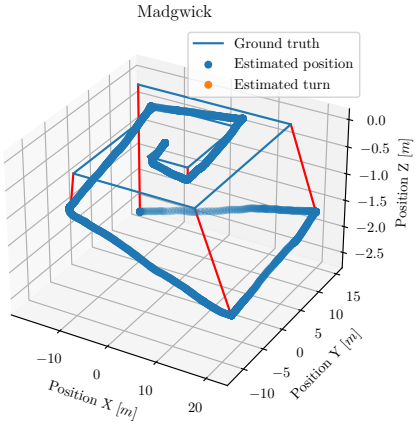
6.4 Spiral

For the spiral experiment, the Davenport algorithm which had the lowest displacement error with an average of 2.64 meters (1.78% of error margin), and Madgwick with an average of 2.46 meters of turn error (1.66% of error margin).

Algorithm	Displacement Error[m]	Displacement Error[%]	Turn Error[m]	Turn Error[%]
AngularRate	41.51	28.05	61.37	41.47
AQUA	3.34	2.26	6.23	4.21
Complementary	2.86	1.93	6.92	4.67
Davenport	2.64	1.78	3.02	2.04
EKF	3.01	2.03	3.03	2.05
FAMC	39.96	27.00	48.09	32.49
FLAE	3.18	2.15	3.71	2.51
Fourati	41.45	28.00	61.91	41.83
Madgwick	2.74	1.85	2.46	1.66
Mahony	2.81	1.90	3.47	2.34
OLEQ	3.26	2.20	3.06	2.07
QUEST	19.97	13.50	35.23	23.80
ROLEQ	3.39	2.29	3.17	2.14
SAAM	2.73	1.84	3.29	2.23
Tilt	2.73	1.84	3.29	2.23
Average	11.70	7.91	16.55	11.18

Table 17: Spiral position estimation error (displacement and turn) of the sensor fusion algorithms.





(a) Madgwick's 3D position estimation had the lowest turn error for the spiral experiment.

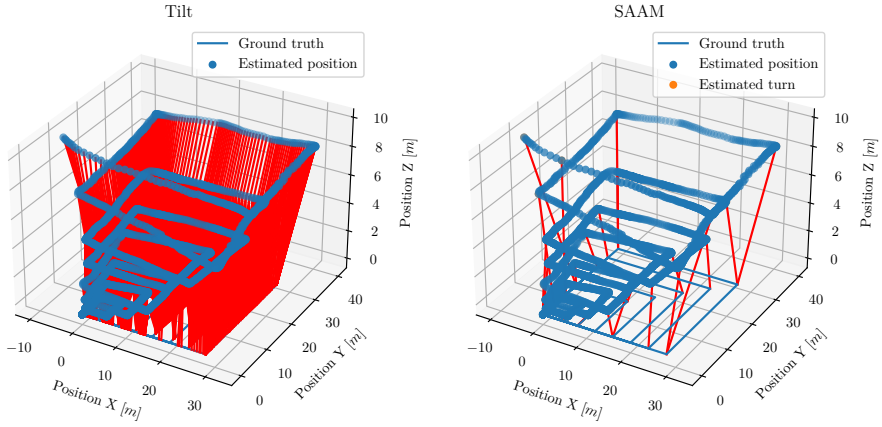
Fig. 53: Position estimation by the best performing algorithms in the spiral experiment.

6.5 Squares

For the 28-meter line experiment, the Tilt algorithm which had the lowest displacement error with an average of 6.11 meters (4.88% of error margin), and SAAM with an average of 7.23 meters of turn error (4.77% of error margin).

Algorithm	Displacement Error[m]	Displacement Error[%]	Turn Error[m]	Turn Error[%]
AngularRate	32.91	22.24	51.46	34.77
AQUA	7.08	4.78	8.32	5.62
Complementary	11.63	7.86	11.34	7.66
Davenport	6.17	4.17	7.21	4.87
EKF	7.29	4.93	8.33	5.63
FAMC	31.83	21.51	38.19	25.81
FLAE	7.61	5.14	9.16	6.19
Fourati	54.90	37.09	73.30	49.53
Madgwick	6.53	4.42	7.53	5.09
Mahony	6.18	4.17	7.18	4.85
OLEQ	6.43	4.35	7.00	4.73
QUEST	12.57	8.49	25.22	17.04
ROLEQ	6.53	4.41	7.06	4.77
SAAM	6.11	4.13	7.23	4.88
Tilt	6.11	4.13	7.23	4.88
Average	13.99	9.45	18.38	12.42

Table 18: Squares position estimation error (displacement and turn) of the sensor fusion algorithms.



(a) Tilt's 3D position estimation had the lowest displacement error for the squares experiment.

(b) SAAM's 3D position estimation had the lowest turn error for the spiral experiment.

Fig. 54: Position estimation by the best performing algorithms in the squares experiment.

7 Discussion

7.1 Discussion of Results

Facing the result from the previous chapter, a few conclusions can be drawn. Generally, the results showed a good estimation accuracy by most sensor fusion algorithms, with small difference between the best performing algorithms (table 19).

Algorithm	Displacement Error[m]	Displacement Error[%]	Turn Error[m]	Turn Error[%]
AngularRate	16.35	23.53	25.96	39.64
AQUA	6.45	18.51	9.15	26.54
Complementary	5.49	10.28	6.72	13.44
Davenport	2.27	4.96	3.01	7.95
EKF	2.59	6.71	3.21	9.27
FAMC	15.43	22.11	22.57	35.22
FLAE	2.42	5.02	3.19	7.96
Fourati	24.19	36.14	31.87	48.16
Madgwick	3.86	8.35	4.64	10.42
Mahony	1.94	3.72	2.93	6.97
OLEQ	2.65	4.65	3.17	7.99
QUEST	10.69	22.55	16.34	30.34
ROLEQ	2.71	4.92	3.35	7.92
SAAM	2	4.12	2.79	6.93
Tilt	2	4.12	2.79	6.93
Average	6.74	11.98	9.45	17.71

Table 19: Overall ranking of how every algorithm performed in the given test.

The results showed that the algorithms had smaller displacement error than turn error which indicates that algorithms were generally better at estimating position than necessarily the shape of the experiment. Another conclusion was that absolute error increases with distance, since longer experiments had bigger absolute error rates than shorter experiments. Relative error measurements, on the other side, decreased in longer tests. Some unexpected surprises from less known algorithms such as OLEQ, ROLEQ, SAAM, Tilt, FLAE and Davenport outperforming more well-established algorithms like the Complementary, EKF and Madgwick.

From every test made, Mahony ranks best at displacement and with an average of 1.94 meters of absolute displacement error and 3.72% relative error.

Closely followed by SAAM and Tilt with 2 meters of displacement error, and 4.12% of relative displacement error (table 20).

Algorithm	Displacement Error[m]	Displacement Error[%]
Mahony	1.94	3.72
SAAM	2	4.12
Tilt	2	4.12
Davenport	2.27	4.96
FLAE	2.42	5.02
EKF	2.59	6.71
OLEQ	2.65	4.65
ROLEQ	2.71	4.92
Madgwick	3.86	8.35
Complementary	5.49	10.28
AQUA	6.45	18.51
QUEST	10.69	22.55
FAMC	15.43	22.11
AngularRate	16.35	23.53
Fourati	24.19	36.14

Table 20: Algorithms ranked according to of how well performed at displacement error.

SAAM and Tilt ranked best at turn error both with an average of 2.79 meters of absolute turn error and 6.93% of relative turn error. Closely followed by Mahony with 6.97% meters of relative turn error (table 21).

On the other hand, Fourati had the worst general performance among all algorithms, with a 24.19 (36.14% relative displacement error) meter error average displacement and 31.87 (48.16% relative turn error) meter turn error average. Angular Rate, FAMC, QUEST and AQUA also had poor overall results generally performing under average of the other fusion algorithms.

From table 7, it is possible to draw a correlation between what sensor are used by each algorithm and how they performed at estimating position. Tables 20 and 21 show that the worst performing algorithms are generally the ones who don't use every sensor available, for instance FAMC, QUEST and AQUA don't fuse the gyro's readings to estimate orientation, while Angular Rate uses only the gyro's measurements. Fourati's poor performance might be explained by the fact that it depicts rigid body's attitude in space with respect to the navigation

frame (X_N,Y_N,Z_N), where the navigation frame follows the convention North East Down (NED), opposite of the convention used by the other algorithms, East North Up (ENU).

Algorithm	Turn Error[m]	Turn Error[%]
SAAM	2.79	6.93
Tilt	2.79	6.93
Mahony	2.93	6.97
Davenport	3.01	7.95
OLEQ	3.17	7.99
FLAE	3.19	7.96
EKF	3.21	9.27
ROLEQ	3.35	7.92
Madgwick	4.64	10.42
Complementary	6.72	13.44
AQUA	9.15	26.54
QUEST	16.34	30.34
FAMC	22.57	35.22
AngularRate	25.96	39.64
Fourati	31.87	48.16

Table 21: Algorithms ranked according to of how well performed at turn error.

7.2 Comparing with Related Work

Ladetto et al. [91] applied PDR in urban and indoor areas seeking to assist blind people reaching unfamiliar locations along with aiming to facilitate emergency coordinators to track rescue workers. The study integrated a GPS receiver with a body mounted IMU applying pattern recognition to accelerometer signals, determining a user's step signature. They verified an average position error of 5% of distance travelled which is significantly smaller than our results of 12% average travelled error. Nonetheless, Ladetto et al. [91] had GPS receivers and other sensors to help improve absolute position, which is not our case.

Stirling et al. [92] illustrate an experiment exploiting a shoe-mounted sensor prototype that calculates stride length with accelerometers and magnetometers. Their system measures angular acceleration by manipulating pairs of accelerometers as an alternative to gyroscopes. The foot pod stride length measurement was generally less than 10% different from the distances given by their control point surveyed path. With a total distance estimated by the foot pod of 978

meters, compared to 862 meters for the survey. Which goes into conformity with our own results 12% displacement error.

Simulation experiments conducted by Ludwig et al. [79] using quadcopter data and results show that Mahony provides better orientation estimation than both Madgwick and EKF when using optimum parameters. The same authors tested on [105] quadcopters the EKF, Madgwick, and Mahony filters. Results showed that Mahony delivered a more precise orientation estimation and faster execution time than Madgwick and EKF which also goes in compliance with our results.

7.3 Challenges Faced

Common difficulties encountered when attempting to employ an accurate position estimation system lied significantly in the sensor's output reliability and consistency since these types of electronic devices are highly susceptible to the existence of errors. Ever since the beginning of the development of this project we realized calibration would play an important role if this idea was to ever take off. The observations made throughout the examining of other positioning systems have been that calibration ensures accurate measurements, and accurate measurements are the foundation to the quality of data. These discoveries lead to the significance of calibration as a means of reducing inaccuracies and increase data quality significantly. As seen in chapter 5.3, several methods and algorithms have been established which evidently decreased the drift and noise components and have helped increase the accuracy of the positioning system.

To build a precise positioning system utilizing low-cost MEMS sensors was an exciting challenge that came with numerous obstacles. The system requirements were stringent owing to the environmental and economical requirements. To accomplish a superior precision, the electronic components ought to have a higher performance, higher resolution, and a higher sampling frequency.

Initially, this work planned to compare how estimation is affected by different settings (indoor, outdoor, and underwater), by creating the same setup for every environment and comparing how estimation was affected (underwater tests don't have dragging resistance which could influence displacement integration). This turned out to be more difficult than expected since we struggled to replicate the same experiments in every environment, couldn't find indoor locations that had enough space to replicate outdoor tests, the same happening with underwater

tests. In the future, there should be an effort to make environmental comparison possible to better understand how position estimation is affected by the environment.

7.4 Future Work

Future work can expand upon this by experimenting with different hardware by utilizing new inertial sensors and microcontrollers, testing new environments such as aerial and nautical, and applying innovative solutions to sensor fusion such as machine learning and deep learning algorithms to learn and better decide how to perform the fusion of sensors. A benchmark comparison of the sensor fusion algorithms on computational performance would also be an interesting topic of research since computing efficiency is also a major variable in performing real time position estimation.

8 Conclusion

Conceptualizing, building, and developing an Inertial Navigation System (INS) for position estimation was a challenging problem which involved the knowledge from distinct areas, from electronics to programming, mathematics, and physics. This work permitted to gain a deeper comprehension, understanding and appreciation of how Micro-Electro-Mechanical Systems (MEMS), and inertial sensors in general, operate. Grasping the multiple levels of complexity and abstraction of decades of knowledge from different fields that this work lies upon was a humbling experience.

Obtaining accurate position knowledge from low-cost inertial sensors is an arduous task that is often considered not possible to achieve without the aid of external sources of positioning or heading information. There was an effort to optimize these sensors to their maximum capacities and prove that a low-cost sensor-based positioning system is possible without the assistance of external sources of positioning. We believe that such was achieved with this work, knowing well our solution has its own limitations and there is further space for improvement.

An extensive comparison among different sensor fusion solutions was provided, demonstrating how distinct fusion techniques perform in orientation and position estimation. This experimental comparison of sensor fusion algorithms was the largest and most profound there was in literature. This work also implied a comprehensive experimental component which involved testing the Inertial Navigation System (INS) across different environments and conditions to better assess how this impacted positioning performance.

This research also permitted to gain some knowledge on why certain sensor fusion algorithms have superior position estimation performance when compared to others, hypothesizing that algorithms who trust the gyro measurements and slightly correct them with the absolute measurement of the magnetometer and accelerometer ended up having the best results.

The questions this work was set to solve in the begging were answered, not only is it possible to build a complete Inertial Navigation System (INS) out of low cost Micro-Electro-Mechanical Systems (MEMS), but it can deliver accurate and detailed position information as well.

References

[1] Julian Zeitlhöfler. Nominal and observation-based attitude realization for precise orbit determination of the jason sateilltes. 2019.

[2] Lynn Michelle Roylance and James B Angell. A batch-fabricated silicon accelerometer. *IEEE Transactions on Electron Devices*, 26(12):1911–1917, 1979.

[3] Valentina Marotto, Alberto Serra, D Carbouni, Mariella Sole, Tiziana Dessi, and Andrea Manchinu. Orientation analysis through a gyroscope sensor for indoor navigation systems. In *Proc. The Fourth International Conference on Sensor Device Technologies and Applications*, pages 85–90, 2013.

[4] Vittorio Passaro, Antonello Cuccovillo, Lorenzo Vaiani, Martino De Carlo, and Carlo Edoardo Campanella. Gyroscope technology and applications: A review in the industrial perspective. *Sensors*, 17(10):2284, 2017.

[5] Pasquale Daponte, Luca De Vito, Gianluca Mazzilli, Francesco Picariello, and Sergio Rapuano. Method for compensating the effect of disturbances on magnetometer measurements: experimental results. In *2017 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pages 1–6. IEEE, 2017.

[6] Mohsen Mosallaei, Karim Salahshoor, and Mohammad Reza Bayat. Process fault detection and diagnosis by synchronous and asynchronous decentralized kalman filtering using state-vector fusion technique. In *2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information*, pages 209–214. IEEE, 2007.

[7] Walter T Higgins. A comparison of complementary and kalman filtering. *IEEE Transactions on Aerospace and Electronic Systems*, (3):321–325, 1975.

[8] Online. Mpu-6050 six-axis (gyro + accelerometer) mems motiontracking devices. `https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/`. Accessed: 2021-04-29.

[9] Online. Ak8963 3-axis electronic compass. `https://download.mikroe.com/documents/datasheets/ak8963c-datasheet.pdf`. Accessed: 2021-04-29.

[10] Tomas Soler and Larry D Hothem. Coordinate systems used in geodesy: Basic definitions and concepts. *Journal of surveying engineering*, 114(2):84–97, 1988.

[11] Online. Ahrs: Attitude and heading reference systems. `https://ahrs.readthedocs.io/en/latest/`. Accessed: 2021-04-29.

[12] Rigas Themistoklis Ioannides, Thomas Pany, and Glen Gibbons. Known vulnerabilities of global navigation satellite systems, status, and potential mitigation techniques. *Proceedings of the IEEE*, 104(6):1174–1194, 2016.

[13] Halima Mansour Omar, Zhang Yanzhong, Zhang Bo, and Haris Ubaid Gul. Integration of gps and dead reckoning navigation system using moving horizon estimation. In *2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference*, pages 553–556. IEEE, 2016.

[14] S Nassar, A Noureldin, and N El-Sheimy. Improving positioning accuracy during kinematic dgps outage periods using sins/dgps integration and sins data de-noising. *Survey Review*, 37(292):426–438, 2004.

[15] Oliver P Dewhirst, Hannah K Evans, Kyle Roskilly, Richard J Harvey, Tatjana Y Hubel, and Alan M Wilson. Improving the accuracy of estimates of animal path and travel distance using gps drift-corrected dead reckoning. *Ecology and evolution*, 6(17):6210–6222, 2016.

[16] Wei-Wen Kao. Integration of gps and dead-reckoning navigation systems. In *Vehicle Navigation and Information Systems Conference, 1991*, volume 2, pages 635–643. IEEE, 1991.

[17] James L Coyte, David Stirling, Montserrat Ros, Haiping Du, and Andrew Gray. Displacement profile estimation using low cost inertial motion sensors with applications to sporting and rehabilitation exercises. In *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 1290–1295. IEEE, 2013.

[18] RVC Wong, KP Schwarz, and ME Cannon. High-accuracy kinematic positioning by gps-ins. *Navigation*, 35(2):275–287, 1988.

[19] Dániel Hetényi, Márton Gótzy, and László Blázovics. Sensor fusion with enhanced kalman filter for altitude control of quadrotors. In *2016 IEEE 11th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pages 413–418. IEEE, 2016.

[20] Chunbo Luo, Sally I McClean, Gerard Parr, Luke Teacy, and Renzo De Nardi. Uav position estimation and collision avoidance using the extended kalman filter. *IEEE Transactions on Vehicular Technology*, 62(6):2749–2762, 2013.

[21] K Rahul Sharma, Daniel Honc, and František Dušek. Sensor fusion for prediction of orientation and position from obstacle using multiple ir sensors an approach based on kalman filter. In *2014 International Conference on Applied Electronics*, pages 263–266. IEEE, 2014.

[22] Sebastian OH Madgwick, Samuel Wilson, Ruth Turk, Jane Burridge, Christos Kapatos, and Ravi Vaidyanathan. An extended complementary filter for full-body marg orientation estimation. *IEEE/ASME Transactions on Mechatronics*, 25(4):2054–2064, 2020.

[23] Samuel Wilson, Henry Eberle, Yoshikatsu Hayashi, Sebastian OH Madgwick, Alison McGregor, Xingjian Jing, and Ravi Vaidyanathan. Formulation of a new gradient descent marg orientation algorithm: Case study on robot teleoperation. *Mechanical Systems and Signal Processing*, 130:183–200, 2019.

[24] Panagiotis Papadimitratos and Aleksandar Jovanovic. Protection and fundamental vulnerability of gnss. In *2008 IEEE International Workshop on Satellite and Space Communications*, pages 167–171. IEEE, 2008.

[25] Moeness G Amin, Pau Closas, Ali Broumandan, and John L Volakis. Vulnerabilities, threats, and authentication in satellite-based navigation systems [scanning the issue]. *Proceedings of the IEEE*, 104(6):1169–1173, 2016.

[26] Aron Pinker and Charles Smith. Vulnerability of the gps signal to jamming. *GPS Solutions*, 3(2):19–27, 1999.

[27] A Tawalbeh Lo'ai, Anas Basalamah, Rashid Mehmood, and Hala Tawalbeh. Greener and smarter phones for future cities: Characterizing

the impact of gps signal strength on power consumption. *IEEE Access*, 4:858–868, 2016.

[28] Edward J Krakiwsky, Clyde B Harris, and Richard VC Wong. A kalman filter for integrating dead reckoning, map matching and gps positioning. In *IEEE PLANS'88., Position Location and Navigation Symposium, Record.'Navigation into the 21st Century'.*, pages 39–46. IEEE, 1988.

[29] Adam W Bojanczyk and Adam Lutoborski. Computation of the euler angles of a symmetric 3*3 matrix. *SIAM Journal on Matrix Analysis and Applications*, 12(1):41–48, 1991.

[30] Aleš Janota, Vojtech Šimák, Dušan Nemec, and Jozef Hrbček. Improving the precision and speed of euler angles computation from low-cost rotation sensor data. *Sensors*, 15(3):7016–7039, 2015.

[31] Evan G Hemingway and Oliver M O'Reilly. Perspectives on euler angle singularities, gimbal lock, and the orthogonality of applied forces and applied moments. *Multibody System Dynamics*, 44(1):31–56, 2018.

[32] Andrea Alaimo, V Artale, C Milazzo, and Angela Ricciardello. Comparison between euler and quaternion parametrization in uav dynamics. In *AIP Conference Proceedings*, volume 1558, pages 1228–1231. American Institute of Physics, 2013.

[33] J Yang, JB Li, and G Lin. A simple approach to integration of acceleration data for dynamic soil–structure interaction analysis. *Soil dynamics and earthquake engineering*, 26(8):725–734, 2006.

[34] Jonathan R Nistler and Majura F Selekwa. Gravity compensation in accelerometer measurements for robot navigation on inclined surfaces. *Procedia Computer Science*, 6:413–418, 2011.

[35] YK Thong, MS Woolfson, JA Crowe, BR Hayes-Gill, and DA Jones. Numerical double integration of acceleration measurements in noise. *Measurement*, 36(1):73–92, 2004.

[36] Guowei Cai, Ben M Chen, and Tong Heng Lee. Coordinate systems and transformations. In *Unmanned rotorcraft systems*, pages 23–34. Springer, 2011.

[37] Karl Magnus Laundal and Arthur D Richmond. Magnetic coordinate systems. *Space Science Reviews*, 206(1-4):27–59, 2017.

[38] Claude Boucher and Zuheir Altamimi. Itrs, pz-90 and wgs 84: current realizations and the related transformation parameters. *Journal of Geodesy*, 75(11):613–619, 2001.

[39] Du Q Huynh. Metrics for 3d rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164, 2009.

[40] Eric W Weisstein. Rotation matrix. *https://mathworld. wolfram. com/*, 2003.

[41] Diana Gruber. The mathematics of the 3d rotation matrix. In *Xtreme Game Developers Conference*, pages 1–14, 2000.

[42] William Premerlani and Paul Bizard. Direction cosine matrix imu: Theory. *Diy Drone: Usa*, pages 13–15, 2009.

[43] James Diebel. Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix*, 58(15-16):1–35, 2006.

[44] Gilbert R Grado. A solution to the euler angle transformation equations. *IRE Transactions on Electronic Computers*, (3):362–369, 1960.

[45] IY Bar-Itzhack and Yaakov Oshman. Attitude determination from vector observations: Quaternion estimation. *IEEE Transactions on Aerospace and Electronic Systems*, (1):128–136, 1985.

[46] Kerry W Spring. Euler parameters and the use of quaternion algebra in the manipulation of finite rotations: a review. *Mechanism and machine theory*, 21(5):365–373, 1986.

[47] KENNETH G MCCONNELL. Kinematics of a three-axis gimbal system. In *Developments in Theoretical and Applied Mechanics*, pages 515–541. Elsevier, 1967.

[48] Jehee Lee. Representing rotations and orientations in geometric computing. *IEEE Computer Graphics and Applications*, 28(2):75–83, 2008.

[49] Ramakrishnan Mukundan. Quaternions: From classical mechanics to computer graphics, and beyond. In *Proceedings of the 7th Asian Technology conference in Mathematics*, pages 97–105, 2002.

[50] Roberto G Valenti, Ivan Dryanovski, and Jizhong Xiao. Keeping a good attitude: A quaternion-based orientation filter for imus and margs. *Sensors*, 15(8):19302–19330, 2015.

[51] Janez Funda, Russell H Taylor, and Richard P Paul. On homogeneous transforms, quaternions, and computational efficiency. *IEEE transactions on Robotics and Automation*, 6(3):382–388, 1990.

[52] Mario Toso, Ettore Pennestrì, and Valerio Rossi. Esa multibody simulator for spacecrafts ascent and landing in a microgravity environment. *CEAS Space Journal*, 7(3):335–346, 2015.

[53] Bin Li and Qiang Zhang. Calibration and interrelated factors analysis of imu for spacecraft. *Computer Measurement & Control*, 19(8):1885–1888, 2011.

[54] Samuel J Haberberger. *An IMU-based spacecraft navigation architecture using a robust multi-sensor fault detection scheme*. Missouri University of Science and Technology, 2016.

[55] Ramsey Faragher. Understanding the basis of the kalman filter via a simple and intuitive derivation [lecture notes]. *IEEE Signal processing magazine*, 29(5):128–132, 2012.

[56] Jiazhen Lu, Maoqing Hu, Yanqiang Yang, and Minpeng Dai. On-orbit calibration method for redundant imu based on satellite navigation & star sensor information fusion. *IEEE Sensors Journal*, 20(9):4530–4543, 2020.

[57] L Sahawneh and MA Jarrah. Development and calibration of low cost mems imu for uav applications. In *2008 5th International Symposium on Mechatronics and Its Applications*, pages 1–9. IEEE, 2008.

[58] Christian Eling, Lasse Klingbeil, and Heiner Kuhlmann. Real-time single-frequency gps/mems-imu attitude determination of lightweight uavs. *Sensors*, 15(10):26212–26235, 2015.

[59] Brandon Allen, Robert Derveloy, Nancy Fell, Wade Gasior, George Yu, and Mina Sartipi. Telemedicine assessment of fall risk using wireless sensors. In *2013 IEEE International Conference on Sensing, Communications and Networking (SECON)*, pages 245–247. IEEE, 2013.

[60] Daniele Giansanti, Sandra Morelli, Giovanni Maccioni, and Giovanni Costantini. Toward the design of a wearable system for fall-risk detection in telerehabilitation. *Telemedicine and e-Health*, 15(3):296–299, 2009.

[61] Megan O'Grady, Tom O'Dwyer, James Connolly, Joan Condell, Karla Muñoz Esquivel, Finbar D O'Shea, Philip Gardiner, and Fiona Wilson. Measuring spinal mobility using an inertial measurement unit system: A reliability study in axial spondyloarthritis. *Diagnostics*, 11(3):490, 2021.

[62] JF Kaiser and WA Reed. Data smoothing using low-pass digital filters. *Review of Scientific Instruments*, 48(11):1447–1457, 1977.

[63] M El-Diasty, A El-Rabbany, and S Pagiatakis. Temperature variation effects on stochastic characteristics for low-cost mems-based inertial sensor error. *Measurement Science and Technology*, 18(11):3321, 2007.

[64] Josep M Pons and Ferran de Palol. Observers with constant proper acceleration, constant proper jerk, and beyond. *General Relativity and Gravitation*, 51(6):1–40, 2019.

[65] Xuye Zhuang, Pinghua Li, Dongxing Li, and Wentao Sui. Introductory chapter: Gyroscopes-principles and applications. *Gyroscopes: Principles and Applications*, page 1, 2020.

[66] Riccardo Antonello and Roberto Oboe. Mems gyroscopes for consumer and industrial applications. *Microsensors, Intech*, pages 253–280, 2011.

[67] Riccardo Antonello and Roberto Oboe. Exploring the potential of mems gyroscopes: Successfully using sensors in typical industrial motion control applications. *IEEE Industrial Electronics Magazine*, 6(1):14–24, 2012.

[68] Alexander A Trusov. Overview of mems gyroscopes: history, principles of operations, types of measurements. *University of California, Irvine, USA*, 2011.

[69] Michael J Caruso, Tamara Bratland, Carl H Smith, and Robert Schneider. A new perspective on magnetic field sensing. *SENSORS-PETERBOROUGH-*, 15:34–47, 1998.

[70] Felipe O Silva, Lucas PS Paiva, and Gustavo S Carvalho. Error analysis of accelerometer-and magnetometer-based stationary alignment. *Sensors*, 21(6):2040, 2021.

[71] Ahmed Wahdan, Jacques Georgy, Walid F Abdelfatah, and Aboelmagd Noureldin. Magnetometer calibration for portable navigation devices in vehicles using a fast and autonomous technique. *IEEE Transactions on Intelligent Transportation Systems*, 15(5):2347–2352, 2014.

[72] Pengfei Guo, Haitao Qiu, Yunchun Yang, and Zhang Ren. The soft iron and hard iron calibration method using extended kalman filter for attitude and heading reference system. In *2008 IEEE/ION Position, Location and Navigation Symposium*, pages 1167–1174. IEEE, 2008.

[73] Eric R Bachmann, Xiaoping Yun, and Christopher W Peterson. An investigation of the effects of magnetic variations on inertial/magnetic orientation sensors. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 2, pages 1115–1122. IEEE, 2004.

[74] Daniel Roetenberg, Henk J Luinge, Chris TM Baten, and Peter H Veltink. Compensation of magnetic disturbances improves inertial and magnetic sensing of human body segment orientation. *IEEE Transactions on neural systems and rehabilitation engineering*, 13(3):395–405, 2005.

[75] Henrik Boström, Sten F Andler, Marcus Brohede, Ronnie Johansson, Alexander Karlsson, Joeri Van Laere, Lars Niklasson, Marie Nilsson, Anne Persson, and Tom Ziemke. On the definition of information fusion as a field of research, 2007.

[76] Johann Borenstein and Lauro Ojeda. Heuristic reduction of gyro drift in gyro-based vehicle tracking. In *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense VIII*, volume 7305, page 730507. International Society for Optics and Photonics, 2009.

[77] Jeroen D Hol. *Sensor fusion and calibration of inertial sensors, vision, ultra-wideband and GPS*. PhD thesis, Linköping University Electronic Press, 2011.

[78] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter, 1995.

[79] Simone A Ludwig and Kaleb D Burnham. Comparison of euler estimate using extended kalman filter, madgwick and mahony on quadcopter flight data. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1236–1241. IEEE, 2018.

[80] Sebastian Madgwick. An efficient orientation filter for inertial and inertial/magnetic sensor arrays. *Report x-io and University of Bristol (UK)*, 25:113–118, 2010.

[81] Joan Sola. Quaternion kinematics for the error-state kalman filter. *arXiv preprint arXiv:1711.02508*, 2017.

[82] Yan-Bin Jia. Quaternions. *Course Com S*, 477:577, 2019.

[83] Roberto G Valenti, Ivan Dryanovski, and Jizhong Xiao. A linear kalman filter for marg orientation estimation using the algebraic quaternion algorithm. *IEEE Transactions on Instrumentation and Measurement*, 65(2):467–481, 2015.

[84] Jin Wu, Zebo Zhou, Hassen Fourati, and Yuhua Cheng. A super fast attitude determination algorithm for consumer-level accelerometer and magnetometer. *IEEE Transactions on Consumer Electronics*, 64(3):375–381, 2018.

[85] Hassen Fourati, Noureddine Manamanni, Lissan Afilal, and Yves Handrich. A nonlinear filtering approach for the attitude and dynamic body acceleration estimation based on inertial and magnetic sensors: Bio-logging application. *IEEE Sensors Journal*, 11(1):233–244, 2010.

[86] Malcolm David Shuster and S D_ Oh. Three-axis attitude determination from vector observations. *Journal of guidance and Control*, 4(1):70–77, 1981.

[87] M Shuster. Approximate algorithms for fast optimal attitude computation. In *Guidance and Control Conference*, page 1249, 1978.

[88] Zebo Zhou, Jin Wu, Jinling Wang, and Hassen Fourati. Optimal, recursive and sub-optimal linear solutions to attitude determination from

vector observations for gnss/accelerometer/magnetometer orientation measurement. *Remote sensing*, 10(3):377, 2018.

[89] Christopher J Fisher. Using an accelerometer for inclination sensing. *AN-1057, Application note, Analog Devices*, pages 1–8, 2010.

[90] ST Microelectronics. Tilt measurement using a low-g 3-axis accelerometer. *AN3182 Application Note, Microelectronics, ST*, 2010.

[91] Quentin Ladetto and Bertrand Merminod. In step with ins navigation for the blind, tracking emergency crews. *Gps World*, 13(ARTICLE):30–38, 2002.

[92] Ross Stirling, Jussi Collin, Ken Fyfe, and Gérard Lachapelle. An innovative shoe-mounted pedestrian navigation system. In *proceedings of European navigation conference GNSS*, volume 110, 2003.

[93] Ulrich Steinhoff and Bernt Schiele. Dead reckoning from the pocket-an experimental study. In *2010 IEEE international conference on pervasive computing and communications (PerCom)*, pages 162–170. IEEE, 2010.

[94] Harvey Weinberg. Using the adxl202 in pedometer and personal navigation applications. *Analog Devices AN-602 application note*, 2(2):1–6, 2002.

[95] Jeong Won Kim, Han Jin Jang, Dong-Hwan Hwang, and Chansik Park. A step, stride and heading determination for the pedestrian navigation system. *Journal of Global Positioning Systems*, 3(1-2):273–279, 2004.

[96] Jussi Collin, G Lachapelle, and Jani Käppi. Mems-imu for personal positioning in a vehicle–a gyro-free approach. In *Proceedings of ION GPS*, pages 24–27, 2002.

[97] Hassan Jameian, Behrouz Safarinejadian, and Mokhtar Shasadeghi. A robust and fast self-alignment method for strapdown inertial navigation system in rough sea conditions. *Ocean Engineering*, 187:106196, 2019.

[98] Jeff Bird and Dale Arden. Indoor navigation with foot-mounted strapdown inertial navigation and magnetic sensors [emerging opportunities for localization and tracking]. *IEEE Wireless Communications*, 18(2):28–35, 2011.

[99] David L Hall and James Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997.

[100] Wilfried Elmenreich. An introduction to sensor fusion. *Vienna University of Technology, Austria*, 502:1–28, 2002.

[101] Honghui Qi and John B Moore. Direct kalman filtering approach for gps/ins integration. *IEEE Transactions on Aerospace and Electronic Systems*, 38(2):687–693, 2002.

[102] AH Mohamed and KP Schwarz. Adaptive kalman filtering for ins/gps. *Journal of geodesy*, 73(4):193–203, 1999.

[103] A Noordin, MAM Basri, and Z Mohamed. Sensor fusion algorithm by complementary filter for attitude estimation of quadrotor with low-cost imu. *Telkomnika*, 16(2):868–875, 2018.

[104] Mark Euston, Paul Coote, Robert Mahony, Jonghyuk Kim, and Tarek Hamel. A complementary filter for attitude estimation of a fixed-wing uav. In *2008 IEEE/RSJ international conference on intelligent robots and systems*, pages 340–345. IEEE, 2008.

[105] Simone A Ludwig, Kaleb D Burnham, Antonio R Jiménez, and Pierre A Touma. Comparison of attitude and heading reference systems using foot mounted mimu sensor data: basic, madgwick, and mahony. In *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2018*, volume 10598, page 105982L. International Society for Optics and Photonics, 2018.

[106] Estefania Munoz Diaz, Fabian de Ponte Müller, Antonio R Jiménez, and Francisco Zampella. Evaluation of ahrs algorithms for inertial personal localization in industrial environments. In *2015 IEEE International Conference on Industrial Technology (ICIT)*, pages 3412–3417. IEEE, 2015.

[107] Online. Ahrs: Attitude and heading reference systems. `https://docs.pycom.io/datasheets/development/lopy/`. Accessed: 2021-04-29.

[108] Sumit Majumder, Tapas Mondal, and M Jamal Deen. Wearable sensors for remote health monitoring. *Sensors*, 17(1):130, 2017.

[109] Mohammed Al-Khafajiy, Thar Baker, Carl Chalmers, Muhammad Asim, Hoshang Kolivand, Muhammad Fahim, and Atif Waraich. Remote health monitoring of elderly through wearable sensors. *Multimedia Tools and Applications*, 78(17):24681–24706, 2019.

[110] Marco Altini, Pierluigi Casale, Julien Penders, and Oliver Amft. Cardiorespiratory fitness estimation in free-living using wearable sensors. *Artificial intelligence in medicine*, 68:37–46, 2016.

[111] Lorenzo Rum, Oscar Sten, Eleonora Vendrame, Valeria Belluscio, Valentina Camomilla, Giuseppe Vannozzi, Luigi Truppa, Marco Notarantonio, Tommaso Sciarra, Aldo Lazich, et al. Wearable sensors in sports for persons with disability: a systematic review. *Sensors*, 21(5):1858, 2021.

[112] Deirdre Morris, Benjamin Schazmann, Yangzhe Wu, Shirley Coyle, Sarah Brady, Jer Hayes, Conor Slater, Cormac Fay, King Tong Lau, Gordon Wallace, et al. Wearable sensors for monitoring sports performance and training. In *2008 5th International Summer School and Symposium on Medical Devices and Biosensors*, pages 121–124. IEEE, 2008.

[113] Miikka Ermes, Juha Pärkkä, Jani Mäntyjärvi, and Ilkka Korhonen. Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. *IEEE transactions on information technology in biomedicine*, 12(1):20–26, 2008.

[114] Ernst A Heinz, Kai S Kunze, Matthias Gruber, David Bannach, and Paul Lukowicz. Using wearable sensors for real-time recognition tasks in games of martial arts-an initial experiment. In *2006 IEEE Symposium on Computational Intelligence and Games*, pages 98–102. IEEE, 2006.

[115] Bobak Mortazavi, Suneil Nyamathi, Sunghoon Ivan Lee, Thomas Wilkerson, Hassan Ghasemzadeh, and Majid Sarrafzadeh. Near-realistic mobile exergames with wireless wearable sensors. *IEEE Journal of Biomedical and Health Informatics*, 18(2):449–456, 2013.

[116] Mingliang Cao, Tianhua Xie, and Zebin Chen. Wearable sensors and equipment in vr games: a review. *Transactions on Edutainment XV*, pages 3–12, 2019.

[117] Online. Mpu-9250 six-axis (gyro + accelerometer) mems motiontracking devices. https://invensense.tdk.com/products/motion-tracking/9-axis/mpu-9250/. Accessed: 2021-04-29.

[118] Online. Mpu-9250 product specification. https://cdn.sparkfun.com/assets/learn_tutorials/5/5/0/MPU9250REV1.0.pdf. Accessed: 2021-04-29.

[119] Online. Micropython documentation. http://docs.micropython.org/en/latest/. Accessed: 2021-04-29.

[120] Nicholas H Tollervey. *Programming with MicroPython: embedded programming with microcontrollers and Python.* " O'Reilly Media, Inc.", 2017.

[121] Matenat Khamphroo, Natavut Kwankeo, Kamol Kaemarungsi, and Kazuhiko Fukawa. Micropython-based educational mobile robot for computer coding learning. In *2017 8th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)*, pages 1–6. IEEE, 2017.

[122] Seong-hoon Peter Won and Farid Golnaraghi. A triaxial accelerometer calibration method using a mathematical model. *IEEE transactions on instrumentation and measurement*, 59(8):2144–2153, 2009.

[123] Alberto Olivares, Gonzalo Olivares, JM Gorriz, and J Ramirez. High-efficiency low-cost accelerometer-aided gyroscope calibration. In *2009 International Conference on Test and Measurement*, volume 1, pages 354–360. IEEE, 2009.

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