FCEE

Mestrado em Engenharia Informática

Exploiting IoT-enabled Inertial Sensors for Position Estimation

Cláudio Rodrigues

supervisionado por

Marko Radeta and Filipe Quintal

1 de novembro de 2021

Abstract

With the surge of inexpensive, widely accessible, and precise microelectromechanical systems (MEMS) in recent years, inertial systems tracking movement have become ubiquitous nowadays. Contrary to GPS-based positioning, Inertial Navigation Systems (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, multipurpose 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.

Keywords: Sensor Fusion · Inertial Navigation System (INS) · Microelectromechanical system (MEMS) · Inertial Measurement Unit (IMU)

Resumo

Keywords: Ubiquitous Computing · Aerial Assessments · Wildlife Monitoring · Machine Learning · Application Development · Low-Altitude Balloons.

Acknowledgements

Cláudio Rodrigues

Table of Contents

List of Figures

List of Tables

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 [15]. Such is especially threatening in high urbanized centers where satellite signals can suffer multipath propagation from tall glass-covered buildings [33]. 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 [30] [7] [18] [6] [44]. Increased accuracy is a precursor to establishing autonomous agents for a diversity of functions. Inertial measurement units (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 vehicles (UAV) [13] [25] [37], telemedicine [27], and robotics [43]. 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 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

This dissertation explores three core challenges that remain with significant interest in literature.

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 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 [15]. Attacks against GNSS-based localization are becoming more frequent and of intensifying damage [34]. Satellite signals can also be affected by abnormal activity due to solar winds, creating temporary gaps in coverage [2]. 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 [35] [33]. 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 [22]. 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 [33]. It has been recognized as a low-power alternative to GNSS localization that can deliver high-resolution position data [7]. 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 [18]. 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 [20].

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 [4]. This involves numerous complex mathematical operations and several clock cycles in a microprocessor that could negatively impact computational performance [17].

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 [12]. 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 [1].

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 [45]. In practice, these measurements are influenced by Earth's gravitational field and rotational components of acceleration, significantly magnifying numerical errors during the readings [31]. 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 [40].

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 [31]. 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.

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. Mathematically, this translates to the integral of the position vector times the density over the Earth being zero (equation 1).

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

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. The right-hand rule explains its orientation.

The East North Up (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.

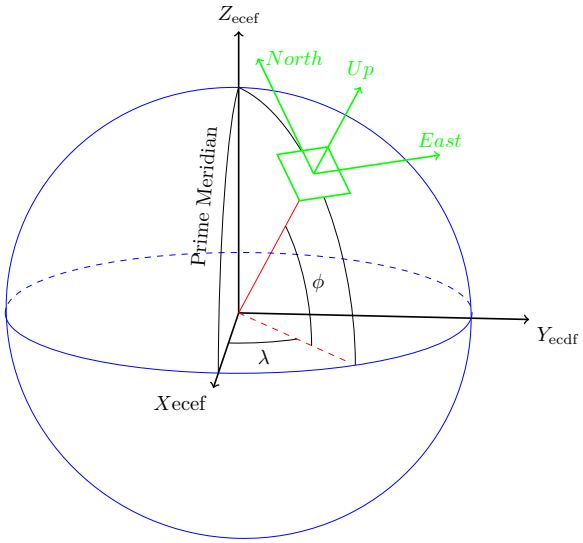


Fig. 1: An illustrative diagram for the WGS84, ECEF, and ENU coordinate systems for the Earth and their transformation correlation (PM line is the Prime Meridian; ϕ and λ are latitude and longitude in WGS84; X, Y, Z for ECEF; and E, N, U for ENU).

Considering a point of reference in the ECEF frame, it is required to discover the equivalent latitude and longitude of the reference point (Xr, Yr, Zr). The parameters of WGS84 required to perform such transformation are presented in table 1.

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 1: WGS 84 needed parameters to convert ECEF coordinates into ENU.

Applying the set of equations (2) to approximate latitude (ψ_r) and longitude (ϕ_r) for the reference coordinate point. The last transformation outcome in applying equation (3) to ECEF physical quantities.

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

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

2.1.2 Body frame

Each sensor present in the razor board is aligned to match the sensor axes printed in the board as seen in figure 3.1. For simplicity, the razor sensors are placed as possible near the middle point of the bisector segment between the two wheel axes in such a away that YY axis as marked in the figure 3.1 is pointing towards the front of vehicle, XX axis is pointing to the right side of the car and ZZ axis is pointing to the top. This way, if the Euler angles describing the orientation of body frame related to world frame are all equal to

zero, it means the axes in each frame are coincident apart from an offset in origin. Rotation angles are considered positive following the right hand rule in each axis. The origin of body frame is equal to intersection of rear wheel axis with the bisector defined above.

2.2 Orientation

The attitude orientation of a body in space a critical aspect in position estimation. In a low-cost AHRS, accuracy and low complexity are important in calculating 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 orientation of one frame relative to another. Any orientation can be attained by composing three elemental rotations, beginning from a known standard orientation. Analogously, any rotation matrix R can be decomposed as a product of three elemental rotation matrices (equation 4).

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

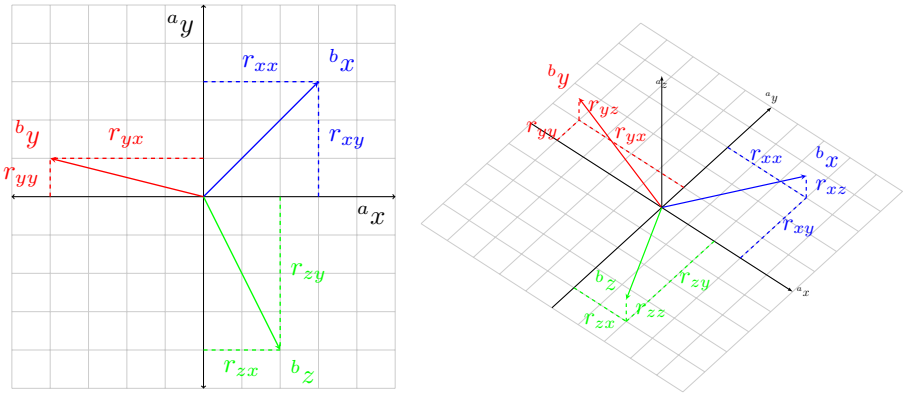


Fig. 2: frame axis example.

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). 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 (2.3) with figure 2

it is seen that columns of ${}^{a}_{b}R$ represent each unity vector defining all axes of frame b.

$$ {}^{a}_{b}R = \begin{bmatrix} {}^{a}r_{xx} & {}^{a}r_{yx} & {}^{a}r_{zx} \\ {}^{a}r_{xy} & {}^{a}r_{yy} & {}^{a}r_{zy} \\ {}^{a}r_{xz} & {}^{a}r_{yz} & {}^{a}r_{zz} \end{bmatrix} \tag{5} $$

Rotation matrices belong to the orthonormal group and one important property is that bRa bRT = I meaning a bRT = a bR-1 . Also a bRT is equal to b aR.

2.2.2 Direct Cosine Matrix

The direct cosine matrix (DCM) is similar to the rotational matrix, however it is a 3x3 matrix that contains the cosines of the 9 possible pairs of axes of two different Cartesian coordinate systems. The DCM is typically used to translate from the body frame into the earth frame. A great introduction to the Direct Cosine Matrix theory is written by Premerlani, W. and Bizard, P. [32], providing information on how to use the DCM method integrated with IMUs. The DCM approach has gained much popularity for its advantage of linear measurement equations in Kalman filters[33, 34, 35, 36]. An analysis of various error sources using low-cost IMUs and optimizing fusion algorithms using the DCM method is presented in [37].

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. 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. 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.

Merely three differential equations require solving prior to the attitude gets distinctly approximated.

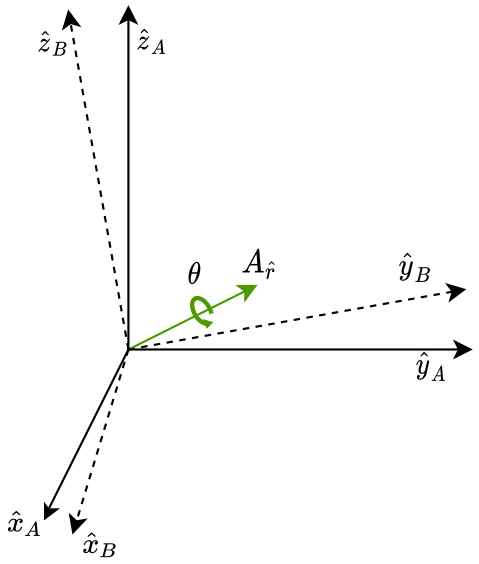


Fig. 3: 3D rotation around vector $\hat{r}$ between frame A and B.

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

$${}^b_a R = {}^b_2 R_x(\phi) {}^2_1 R_y(\theta) {}^1_a R_z(\psi) = \begin{bmatrix} \sin(\phi) & 0 & -\cos(\phi) \\ \cos(\theta)\cos(\phi) & 0 & \cos(\theta)\sin(\phi) \\ \sin(\theta)\sin(\phi) & 1 & \sin(\theta)\cos(\phi) \end{bmatrix} \tag{7}$$

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. 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. 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 4). Euler angles commonly involve a large number of complex mathematical operations, such as matrix manipulations, which typically take up several clock cycles in a CPU, such can negatively impact computational performance. In this dissertation, a different kind of orientation representation will be used quaternions.

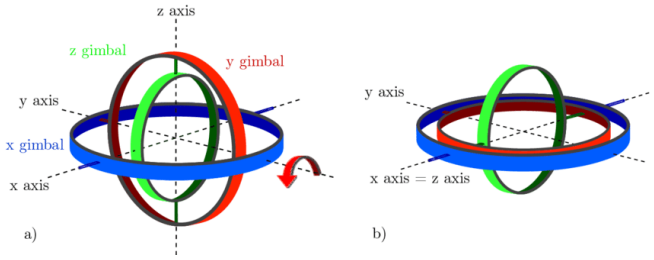


Fig. 4: Representation of the Gimbal Lock problematic [46] - The exterior blue gimbal characterizes the x-axis, the middle, red-colored gimbal the y axis, and the inner green gimbal the z-axis. In the initial arrangement a), every axis is perpendicular to one another. Following a rotation of 90ž across 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 5). 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. Equation (2.1) shows this relation with the complex number x having a real part a and imaginary part b; and the quaternion q containing a real scalar part s and an imaginary vector component v.

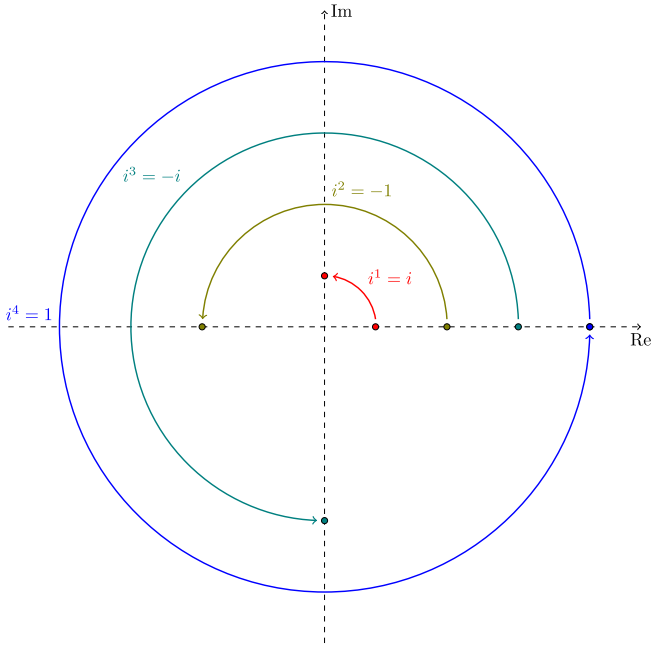


Fig. 5: 90-degree rotation in a 2-dimensional coordinate frame with a real x-axis and an imaginary y-axis.

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

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

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

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

Normalizing a quaternion is identical to normalizing a vector. 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 2 containing multiplication of basis elements):

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

$\times$	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 2: 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 2 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:

$$ {}^{w}_{b}q^{*} = {}^{b}_{w}q = [q_1 - q_2 - q_3 - q_4] \tag{17} $$

Vector rotation Let's define the following quaternion representation of the same vector but in each referential by using is pure quaternion as ${}^{w}v = \begin{bmatrix} 0 & {}^{w}x & {}^{w}y & {}^{w}z \end{bmatrix}$ and ${}^{b}v = \begin{bmatrix} 0 & {}^{b}x & {}^{b}y & {}^{b}z \end{bmatrix}$. The rotation of vector v from one frame to other, using a quaternion, is performed by equation 2.10.

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

Composed rotations The composition of rotations can be described in quaternions as the product between quaternions. For example the sequence a ${}^{a}_{b}\hat{q} \rightarrow {}^{b}_{c}\hat{q}$ is equal to ${}^{a}_{c}\hat{q}$ and is defined as equation 2.11.

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

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, satellites, and unmanned aerial vehicles (UAVs), to name a few. IMUs have been employed in wearable applications with uses in telemedicine [27], and robotics [43]. 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. There are several different types of sensor errors. These can be noise, offset, drift, scale factor error, non linearity and accuracy error. The main errors that causes unreliable readings are high frequent noise, offset error and drift error. Thus, these three will be considered and reduced using suitable filters.

The first graph in the picture above represents the signal error classified as noise. The noise is high frequent and can be eliminated by using a low pass

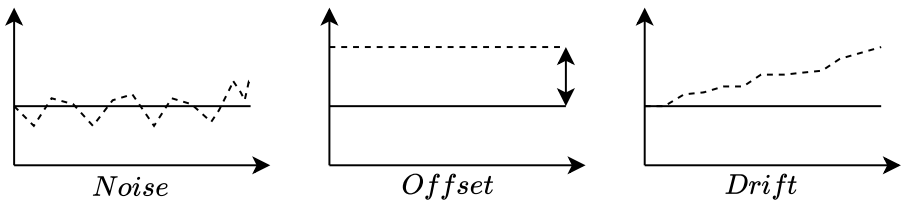


Fig. 6: Three main types of sensor error

filter. The noise sources in the sensor can be categorized into three types. 1. Environmental noise, such as temperature fluctuations, gravity , etc. 2. Movements, such as vibrations, shocks . 3. Hardware noise, such as electrical and mechanical thermal noise. The second graph is the offset error, called bias error. This error results in a shift from the zero value and can be solved by subtracting the DC component with a high pass filter. In the last graph the drift error is presented. The drift components occurs from vibration, shock, and temperature variations. The main drift component in this study will be caused by numerical integration during the position calculation. Temperature changes may cause fluctuations in the bias of the sensor signal. By using an additional sensor to measure the ambient temperature the bias induced error can be compensated by applying corrections to the output signals. The relationship between bias error and temperature depends on the sensor device. For a gyroscope the temperature change will cause an error in orientation which grows linearly with time. For an accelerometer the temperature will cause an error in position, thus grow quadratically with time. For the magnetometer the increasing temperature will lead to softening in the moving structure.

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. 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 7.

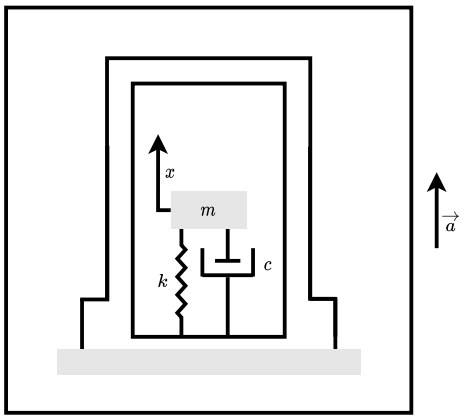


Fig. 7: Schematic demonstrating the working principle of an accelerometer.

Often, the structure incorporates a dashpot (mechanical device which resists motion via viscous friction). The dashpot has a resistive coefficient (c) and is tied to the mass parallelly 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. A conventional gyroscope accommodates a spinning wheel mounted on two gimbals letting it spin in all three axes, as shown in figure 8. 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.

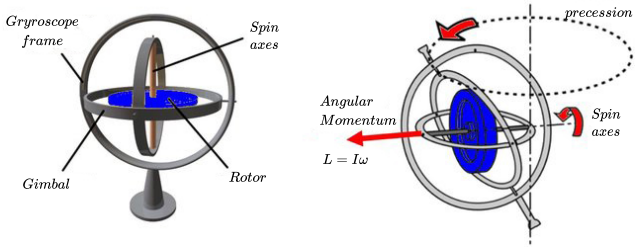


Fig. 8: Representation of a conventional gyroscope accommodating a spinning wheel mounted on two gimbals.

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. 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. 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 9):

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

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.

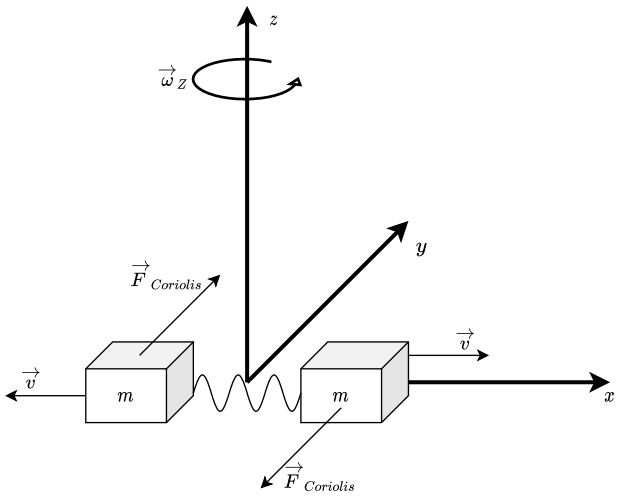


Fig. 9: System where two masses are oscillating in opposite directions. When a rotation is applied, the masses are affected by the Coriolis force and the displacement is measured by a change in capacitance.

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

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

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:

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

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):

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

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.

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

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. 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. There are two categories of magnetic distortions: Hard iron and soft iron. 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. 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:

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

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. 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 take into account, the major and the minor axes.

The major axis is the axis that runs along the x axes. 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.

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

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

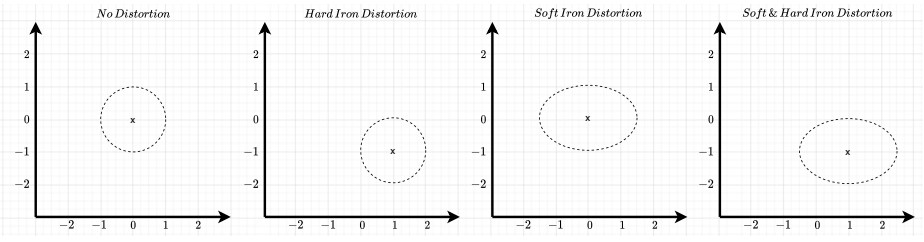


Fig. 10: Representation of a magnetometer's different types of disturbances.

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. 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 lower accuracy and precision. This originates from a drift build up on the gyro's readings, where error accumulates exponentially on the integrated output. 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. 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. 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 [42]. 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 11) 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 12) 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 [29].

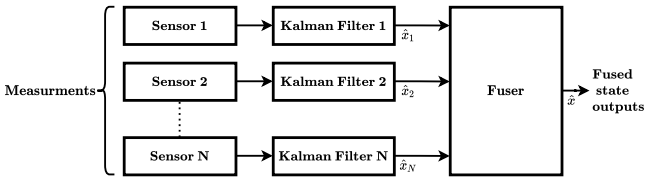


Fig. 11: Kalman-filter-based multi-sensor state-vector fusion. [29]

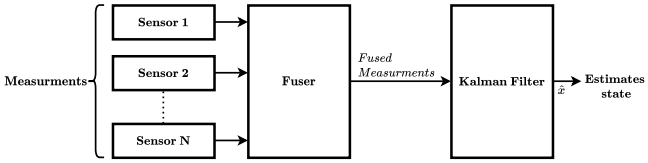


Fig. 12: Kalman-filter-based multi-sensor easurement fusion. [29]

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 [43].

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 [14]. 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 13.

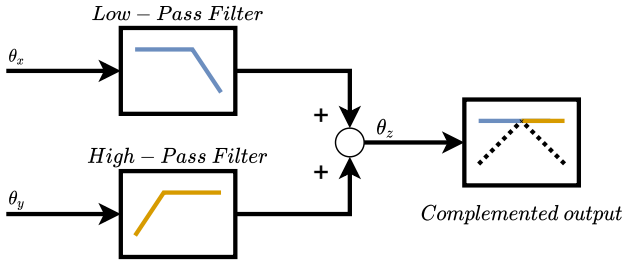


Fig. 13: Basic complementary filter [14] - Two different measurement sources for estimating one variable. The noise properties of the two measurements are such that one source gives good information only in low frequency region while the other is good only in high frequency region.

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 complete the signal is subject to a low pass filter (equation 29) or a high pass filter (equation 30), where θ_t represents current orientation, α is a filter coefficient, ω the gyro's angular velocity and a the acceleration obtained from the accelerometer:

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

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

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 [43]. 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 Filters

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 [27]. 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 [23]. Additionally, vector rotations

are easily executed by quaternion multiplications. Madgwick et al. [26] 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.

2.4.3.1 Other Filters

3 Related Work

The literature review is divided into 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 Inertial Measurement Unit (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. Pedestrian dead reckoning is commonly reviewed in the literature, being subject to studies in various settings.

Ladetto et al. [21] 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. [39] 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 [38] [41] [19] [5]. 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, ROVs, projectiles) and are a topic of study among scholars.

Jameian et al. [16] introduced strapdown inertial navigation system 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. [3]. Similar to pedestrian dead reckoning, 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 algorithm (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. [6] 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 zero-velocity update 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 [11]. 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 [9]. 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 Systems and Global Navigation Satellite System 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. [44] Qi et al. [36], and Nassar et al. [30] 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 [28] 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 Unmanned Aerial Vehicles (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. [13] applied a Kalman Filter to fuse the sonar and accelerometer signals, obtaining a considerably improved altitude estimate with minimal error. Similarly, Luo et al. [25] 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. [37] 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. [27] outlined the formulation of an extended complementary filter 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. [32]. 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. [10] 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 extended Kalman filter that fused GPS/INS at a far less computational cost.

3.2.3 Optimization filters

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 [27]. 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 [23]. Additionally, vector rotations are easily executed by quaternion multiplications. Madgwick et al. [26] 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.

3.2.4 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. [23] 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 [24] quadcopters the Extended Kalman Filter, 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. [8] 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, Extended Kalman Filter, 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.

3.3 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 Extended Kalman Filter, Madgwick, and Mahony algorithms. Furthermore, a quaternion-based gravity compensation filter will be presented, diminishing the influence of the gravity component on acceleration readings.

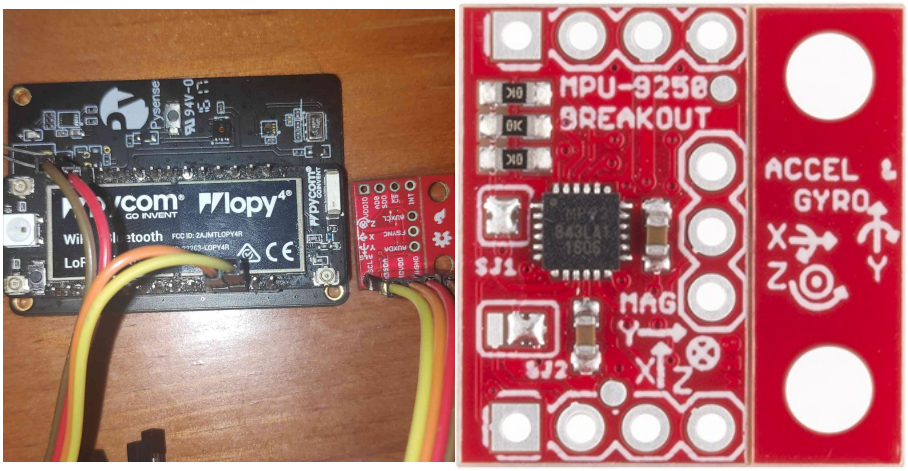
4 Experimental Setup

This section provides a hardware and software experimental setup outline, describing and specifying development and implementation of the low-cost Inertial Navigation System 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.

4.1 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. 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. The LoPy development board interfaces with the external physical inertial sensor through connection pins (figure 14a and communicates remotely by LoRa protocol. A PySense expansion board connects with the LoPy module providing a programable interface for the microcontroller. The inertial navigation system encompasses a small MPU-9250 (figure 14b).

MPU-9250 inertial measurement unit 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 as well temperature and pressure sensors. This chip is extensively employed in wearable sensors for health, fitness, and sports, motion-based game controllers, and portable gaming. 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 magnetometer.The MPU-6050 coordinate



(a) INS Hardware (b) MPU-9250 Breakout

Fig. 14: Pin connection between the microcontroller (left) and the IMU (right). Modules are linked through SCL, CLK, VDD and GND pins.

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

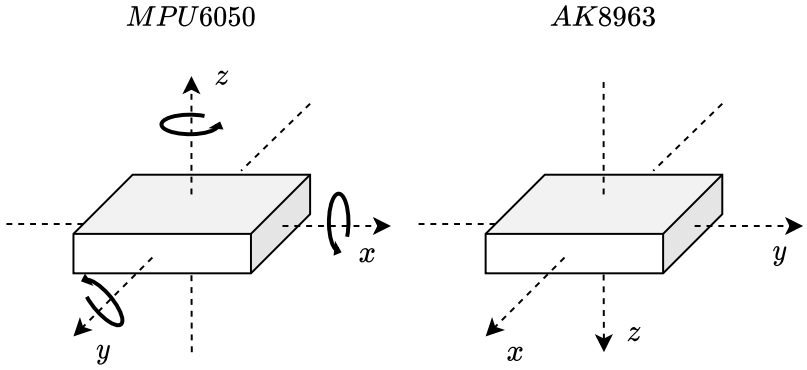


Fig. 15: Coordinate system of an MPU-6050 (3-axis accelerometer and 3-axis gyroscope combination) and AK8963 3-axis magnetometer.

4.1.1 Accelerometer

The triple-axis MEMS accelerometer in MPU-9250 includes a wide range of features:

- 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 3: Accelerometer Specifications.

4.1.2 Gyroscope

The triple-axis MEMS gyroscope in the MPU-9250 includes a wide range of features:

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

4.1.3 Magnetometer

The triple-axis MEMS accelerometer in MPU-9250 includes a wide range of features:

- 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 bit (15$\mu T/LSB$)
- Full scale measurement range is ±4800μT
- Magnetometer normal operating current: 280μA 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 5: Magnetometer specification.

4.2 Software

The microcontroller operates MicroPython, a barebones and efficient implementation of Python 3, which incorporates a small subset of the Python standard library. It is optimized to run on microcontrollers and in constrained environments. The inertial module's raw measurements are interpreted by the microcontroller through Inter-Integrated Circuit (I2C) MicroPython driver serial allowing to read the peripherals memory addresses synchronously. The readings of each sensor are later averaged and linearized to better detect and reduce the presence of outlier readings. 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. Merging the AHRS with an accumulative position allows tracking a moving body in three dimensions over time.

The microcontroller operates MicroPython, a barebones and efficient implementation of Python 3, which incorporates a small subset of the Python standard library. It is optimized to run on microcontrollers and in constrained environments. The inertial module's raw measurements are interpreted by the microcontroller through Inter-Integrated Circuit (I2C) MicroPython driver serial allowing to read the peripherals memory addresses synchronously. The

Algorithm	Accelerometer	Gyroscope	Magnetometer
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
FQA	YES	NO	Optional
Integration	YES	YES	NO
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
TRIAD	YES	NO	YES

Table 6: Available sensor fusion algorithms in AHRS library.

readings of each sensor are later averaged and linearized to better detect and reduce the presence of outlier readings. 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. Merging the AHRS with an accumulative position allows tracking a moving body in three dimensions over time.

4.3 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.

4.3.1 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. Additionally, we can also position the IMU perpendicular to gravity in order to acquire a third calibration point. This results in three unique values 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
```

4.3.2 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 degrees-per-second (dps, ř/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. There are a range of higher-order gyroscope calibration routines, which can be found at the following links (with application descriptions):

```
    count=256
```

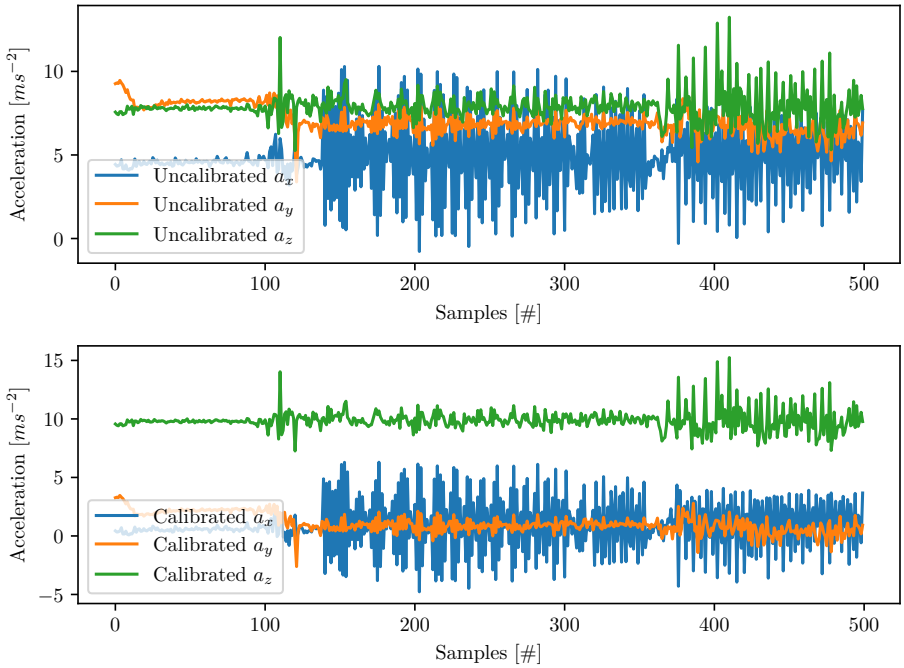


Fig. 16: Accelerometer calibration offset correction.

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

while count:
    # taking samples for a period of time
    utime.sleep_ms(delay)
    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 / n, goy / n, goz / n)
```

```
return self._gyro_offset, self._accelerometer_offset
```

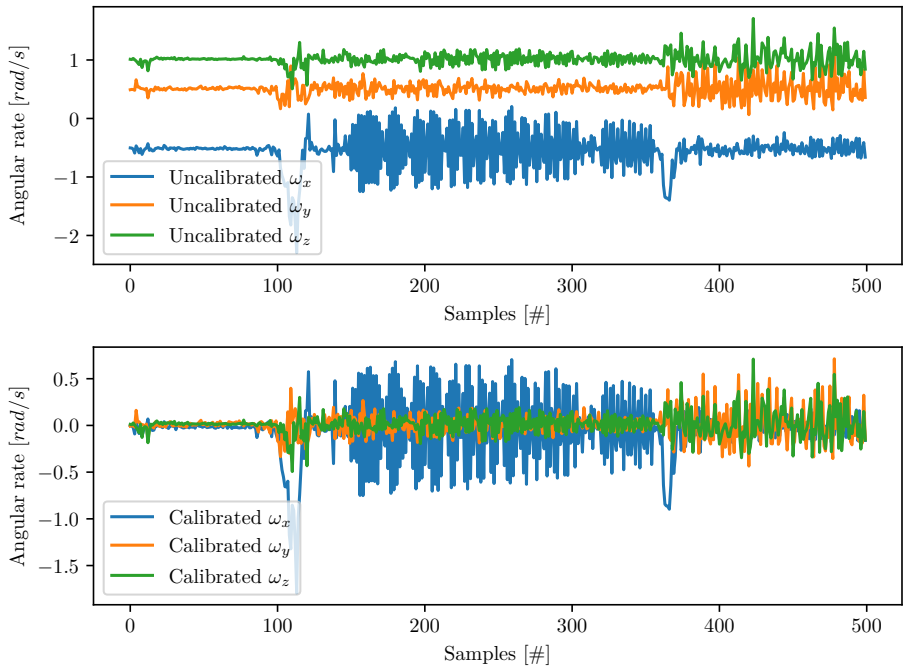
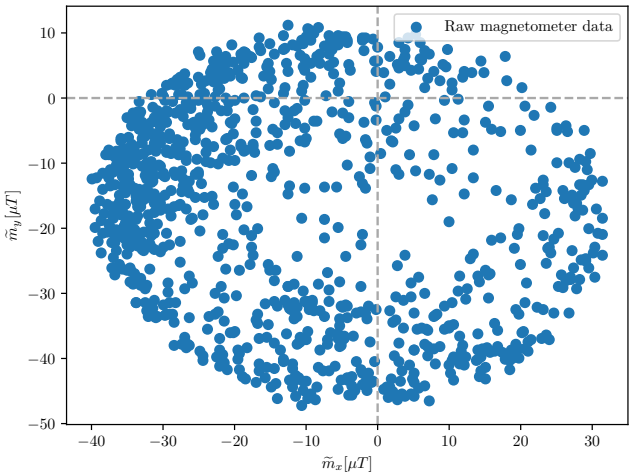


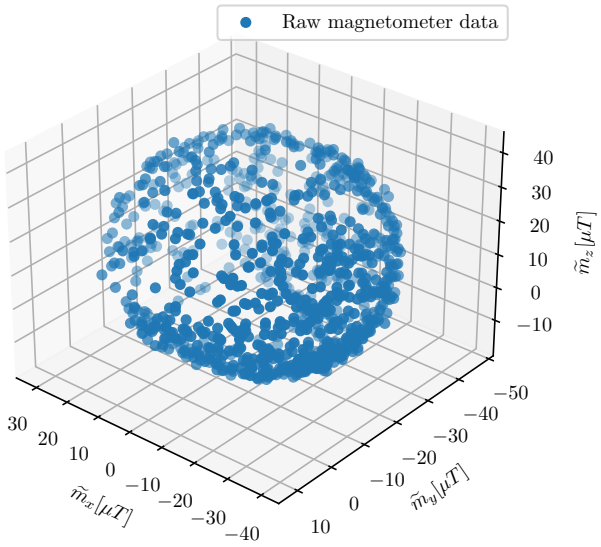
Fig. 17: Gyroscope calibration offset correction.

4.3.3 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. Plotting the magnetometer's raw measurements (as seen in figure 18a and figure 18b), 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.



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



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

Fig. 18: Raw magnetometer measurements.

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 mx, my and mz. 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 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.

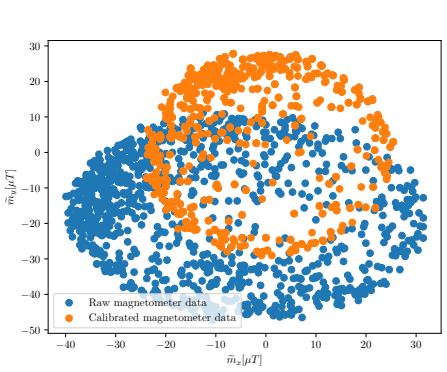
```
# Soft iron correction
# get average x axis max chord length in counts
avg_delta_x = (maxx - minx) / 2
# get average y axis max chord length in counts
avg_delta_y = (maxy - miny) / 2
# get average z axis max chord length in counts
avg_delta_z = (maxz - minz) / 2

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

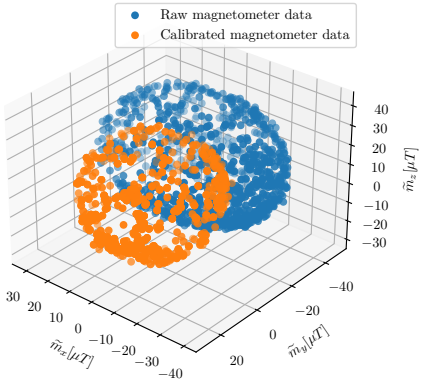
```
    scale_x = avg_delta / avg_delta_x
    scale_y = avg_delta / avg_delta_y
    scale_z = avg_delta / avg_delta_z

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

The result of rescaling the MPU9250 magnetometer data is shown at figure 19:



(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. 19: Raw and calibrated magnetometer measurements.

4.4 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.

4.5 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 and circle. To better understand how distance affects estimation of position, for every geometric figure, tests were made under two different distance magnitudes: 4 and 8 meters.

Fig. 20: 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. 21: Skateboard being dragged along the guideline.

4.6 Outdoor Experiments

Outdoor experiments were conducted at praça do povo's square (figure 22), 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. 22: Platform where outdoors experiments were conducted.

Here, the sports platform consisted of one 28mx28m square, composed of 4m squares, forming a 7x7 grid (figure 23). This was helpful since this knowledge could also be used to benchmark the position estimation of the system and easily calculate error margins.

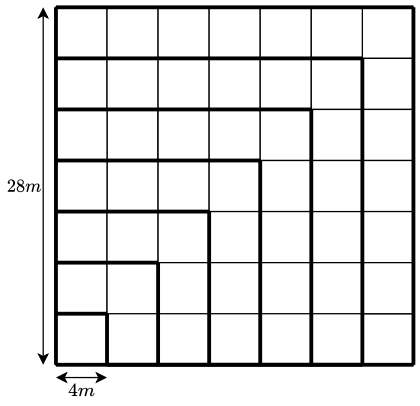
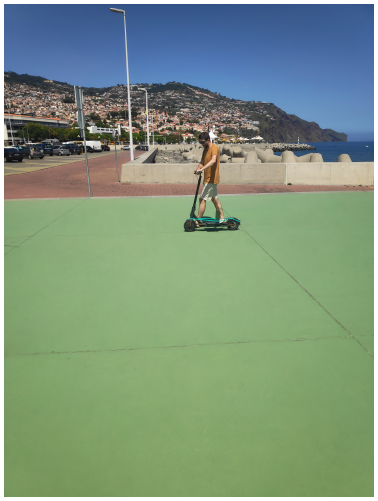


Fig. 23: Bird's-eye view representation of the 7x7 grid platform where outdoor experiments were conducted.

The inertial system was placed on an electrical scooter (figure 24a) in this case, being powered by a laptop computer (figure 24b). The scooter was dragged at walking pace (figure 24c) along the planned path marked by the rope.

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

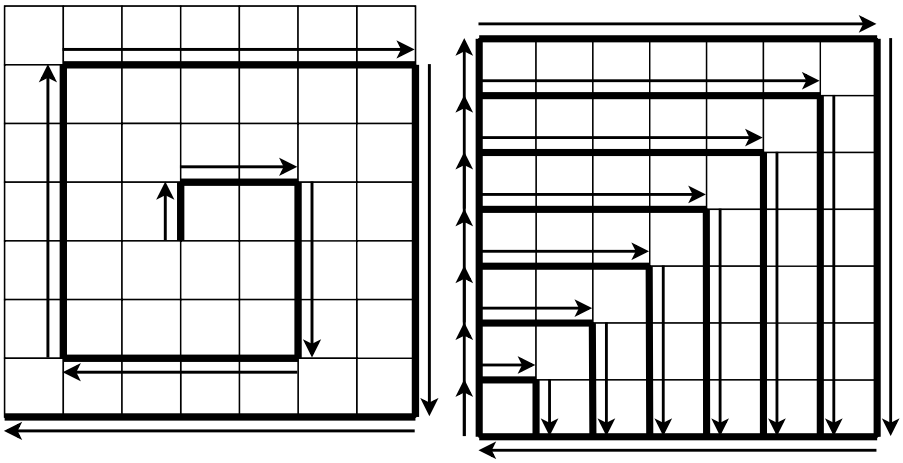
(b) Close-up view of the inertial system (white box) attach with an elastic band to the scooter's body frame.



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

Fig. 24: Photos illustrating hoe 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 25b and figure 25a.



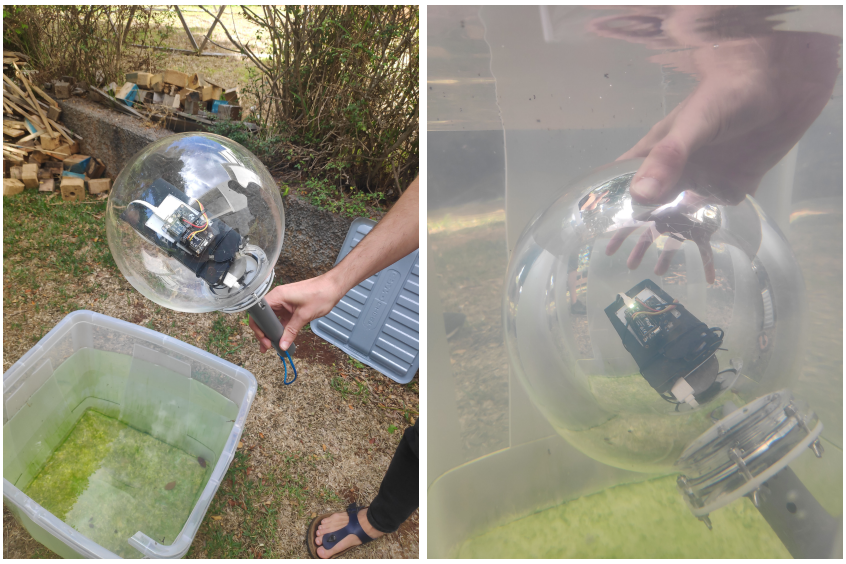
(a) Spiral-based shape with increasing length sides.

(b) Squares of increasing side length.

Fig. 25: Bird's-eye view representation of the more complex path shapes taken.

4.7 Underwater Experiments

Underwater experiments were carried out at Madeira Carlton hotel divepoint, where the INS was placed in an underwater spherical housing built for 360ž cameras with a portable battery pack (figure 26a). 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 26b).



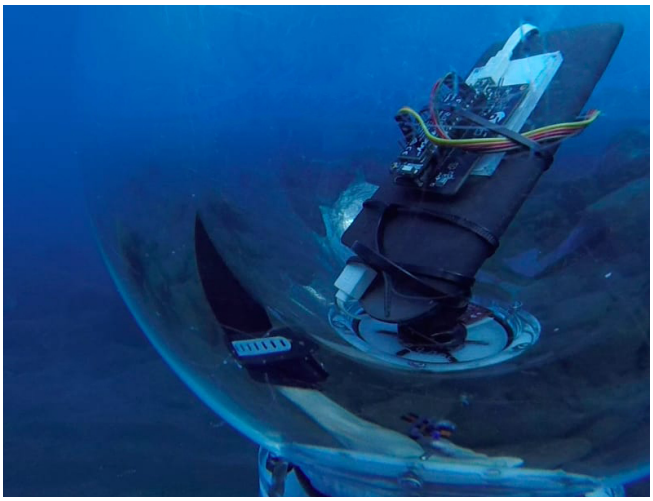
(a) Spherical housing encompassing the inertial system and portable battery pack.

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

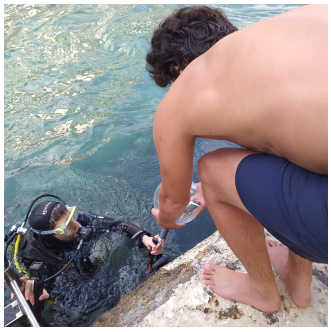
Fig. 26: View of the tests made to assure the underwater housing was water proof.

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 27a). A 5-meter square net being held by two people above water served as guideline for the diver performing the experiment underwater. The diver held the spherical housing while swimming along the

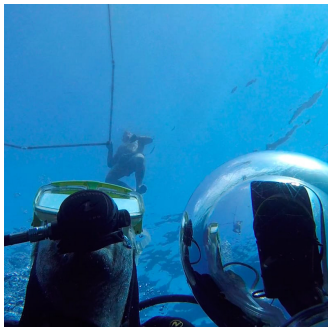
guideline maintaining a constant speed while slowly increasing depth (figure 27c).



(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. 27: Photos of the underwater tests conducted at Madeira Carlton hotel divepoint.

5 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 to remover 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 to measure how well did the estimation perform by describing how to measuring an experiment's estimation error.

5.0.1 Estimating Orientation

The first step in methodology is to obtain orientation (AHRS) from the fusion of the three sensor raw outputs (figure 28). The accelerometer provided the system's proper acceleration; the gyroscope supplied the body's angular rate, and the magnetometer presented the detected magnetic flux.

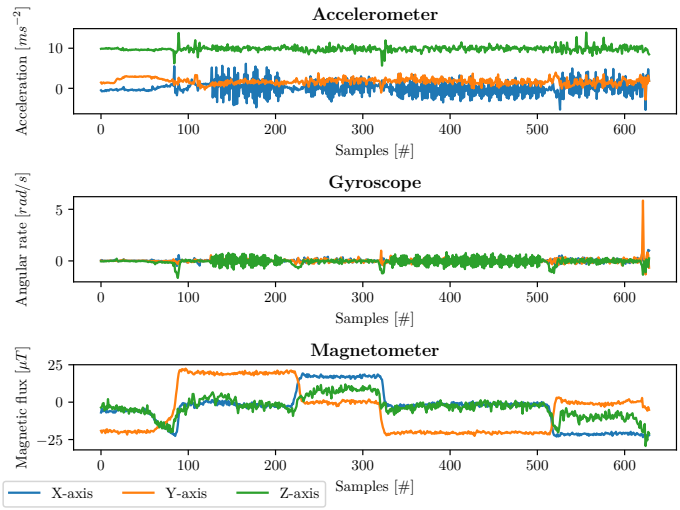


Fig. 28: Raw sensor data on each axis obtained by the accelerometer, gyroscope, and magnetometer.

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 the same principle applies exactly 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 29).

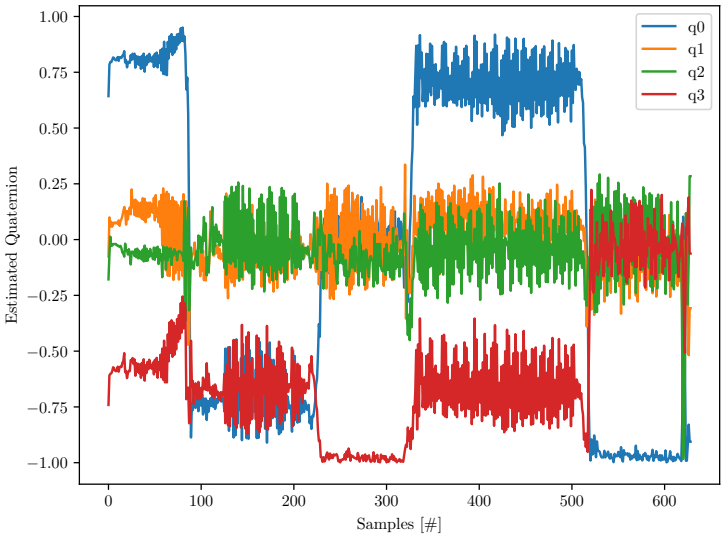


Fig. 29: Madgwick's algorithm sensor fusion output of the raw measurement data of figure 28.

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

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

```
# 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)

if (abs(sinp) >= 1):
    # use 90 degrees if out of range
    angles.pitch = math.copysign(M_PI / 2, sinp)
else
    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
```

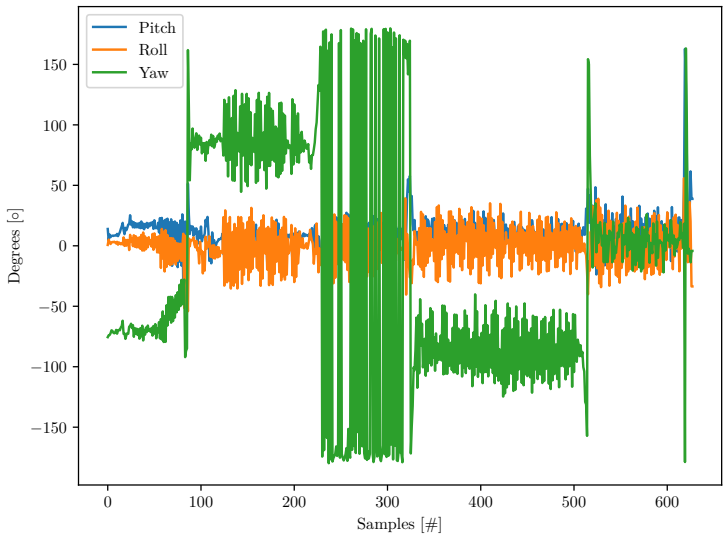


Fig. 30: Madgwick's algorithm sensor fusion output of the raw measurement data of 28 converted into euler angles.

5.0.2 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 28 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

user_acceleration = a_rotated - gravity
```

You need to rotate the accelerometer reading by the quaternion into the Earth frame of reference (into the coordinate system of the room if you like), then subtract gravity. The remaining acceleration is the acceleration of the sensor in the Earth frame of reference often referred to as linear acceleration or user 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 31).

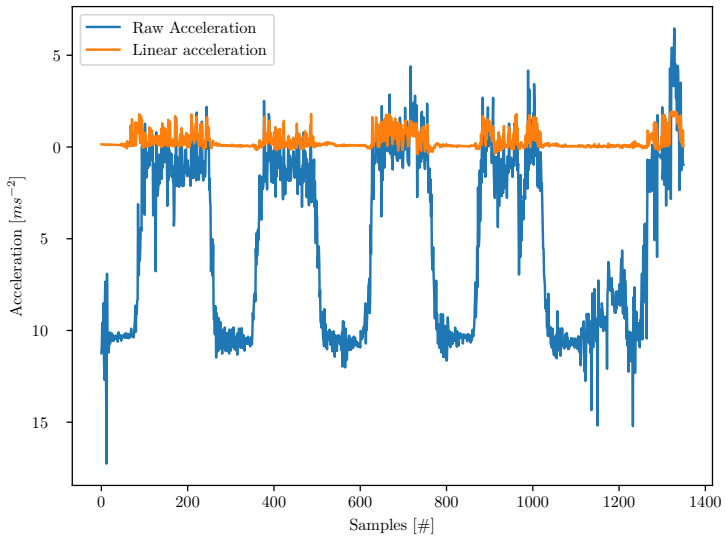


Fig. 31: Comparison between raw acceleration (red) and measurements with gravity compensation (blue). When tilting the sensor, the accelerometer will measure gravitys acceleration (around 9.8 ms^{-2}). The gravity compensation filter can counteract virtually all the gravitational influence on the sensor readings.

5.0.3 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 32).

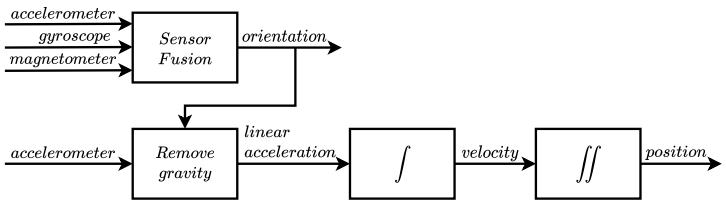


Fig. 32: Overview of the position estimation methodology.

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.

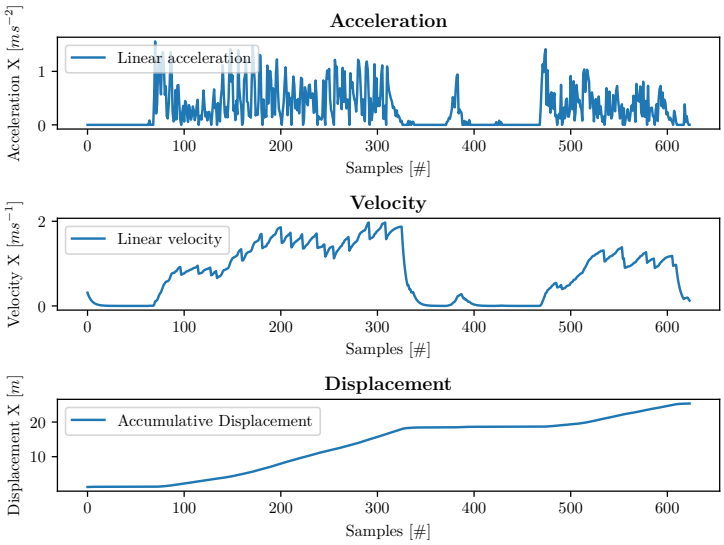


Fig. 33: Plot showing how acceleration is integrated into velocity and displacement.

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 33, where the final observed accumulative position was 25 meters.

Combining the displacement over a period of time 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 of time 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 34):

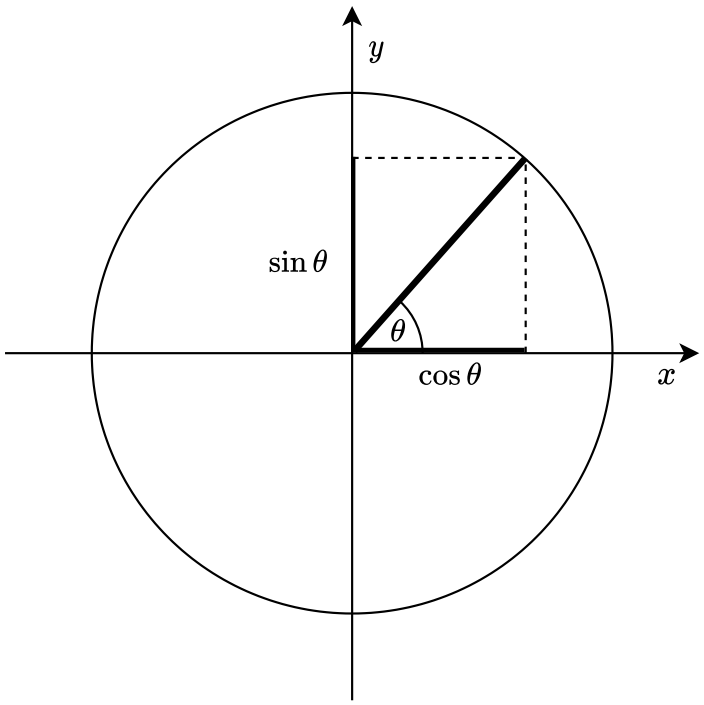


Fig. 34: Illustration of a unit circle where θ.

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

```
# 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 = displacement * sin(radians(pitch)) + lastZ

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

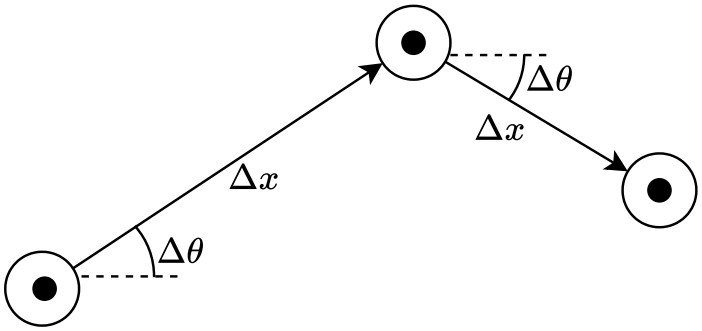


Fig. 35: Illustration of the dead reckoning approach. The object's position at a given moment is given by the last known position, the heading angle $\Delta\theta$, and the travelled distance Δx that occur during the last sample.

5.0.4 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 36 shows the output of applying the position methodology to an experiment of a $4m \times 4m$ square.

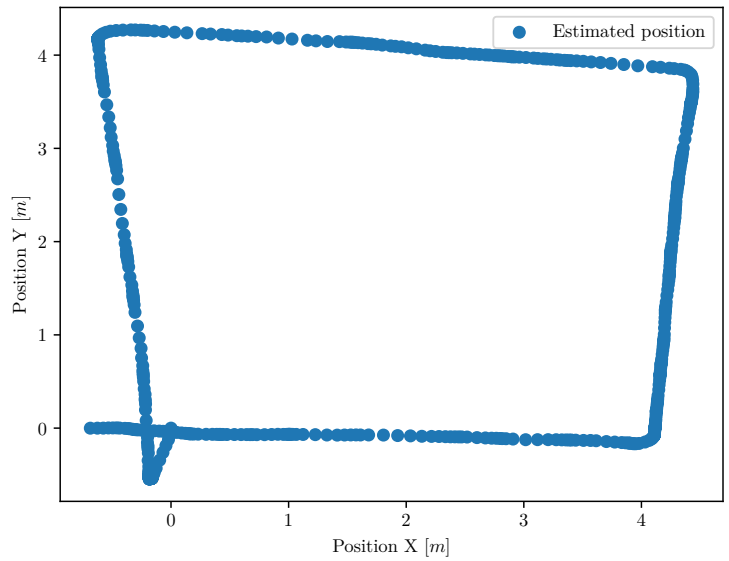


Fig. 36: Result after applying the position estimation methodology to the raw sensor data.

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 the world's frame.

5.0.4.1 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-Peucker Algorithm, was used to estimate when a vertice is present in the estimated path.

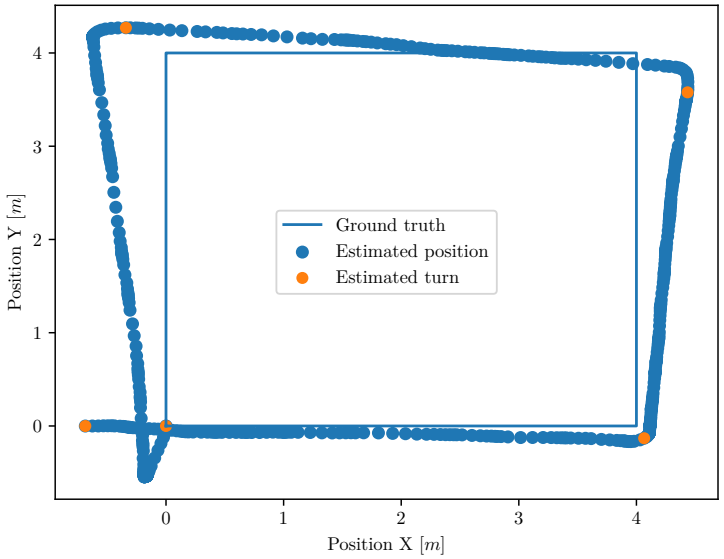
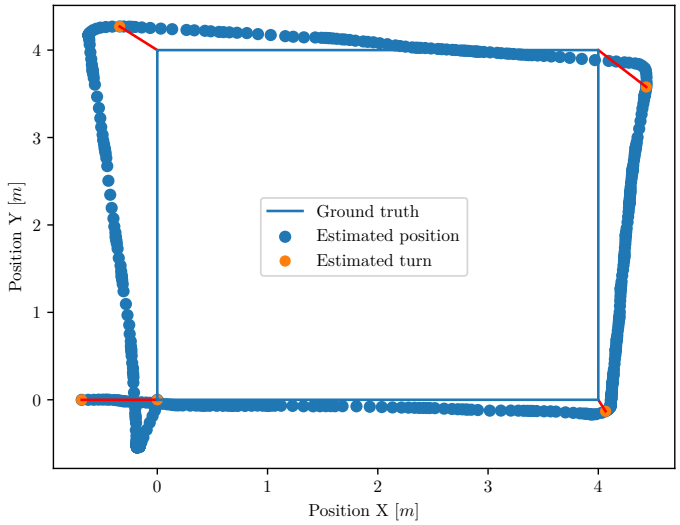
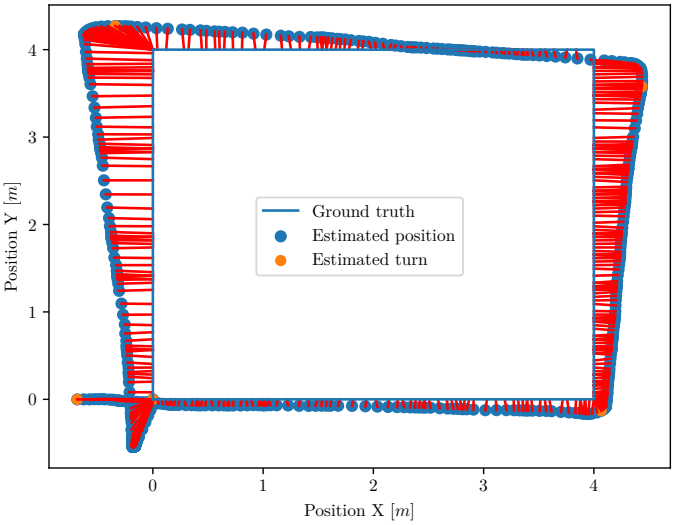


Fig. 37: Illustration of the real world test ground truth (represented by the 4mx4m blue line square) and how the estimated path compares. Orange dots represent the estimated corners of the Ramer-Douglas-Peucker Algorithm.

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. By averaging every distance, it is possible to have an overall understanding of how the estimation performed.



(a) Estimated shape corners are connected to their closest ground truth vertice, averaging all this 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.

5.0.5 Applying methodology in 3D

Finally, by applying the aforementioned 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 39.

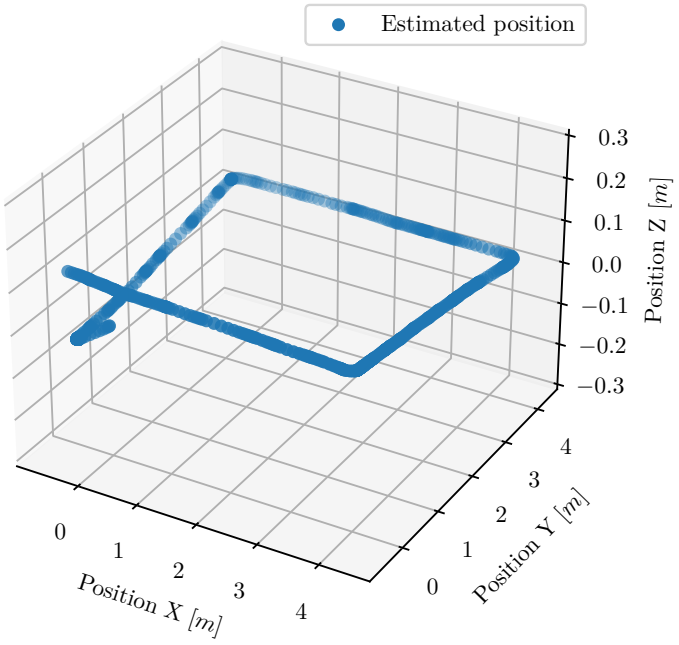


Fig. 39: Three dimensional view of the experiment.

The same methodology mentioned in the previous chapter can be applied in three dimensions. The Ramer-Douglas-Peucker 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.

5.0.5.1 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-Peucker Algorithm, was used to estimate when a vertice is present in the estimated path.

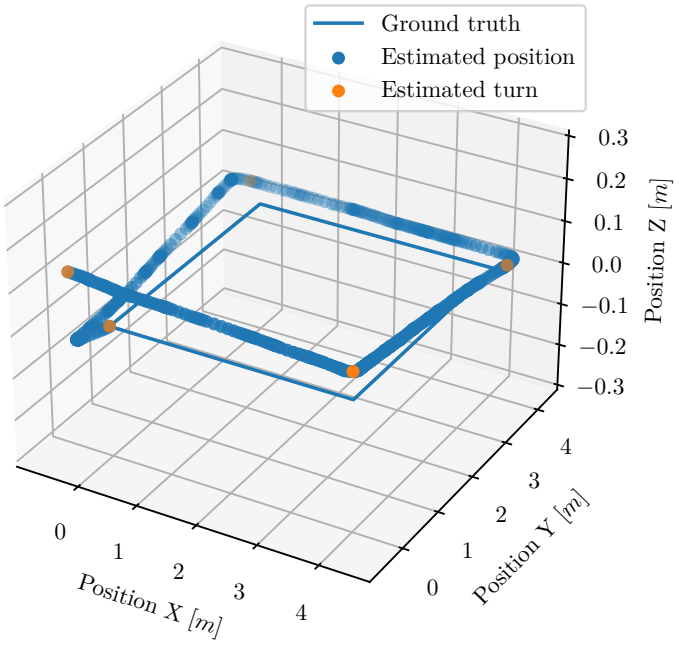
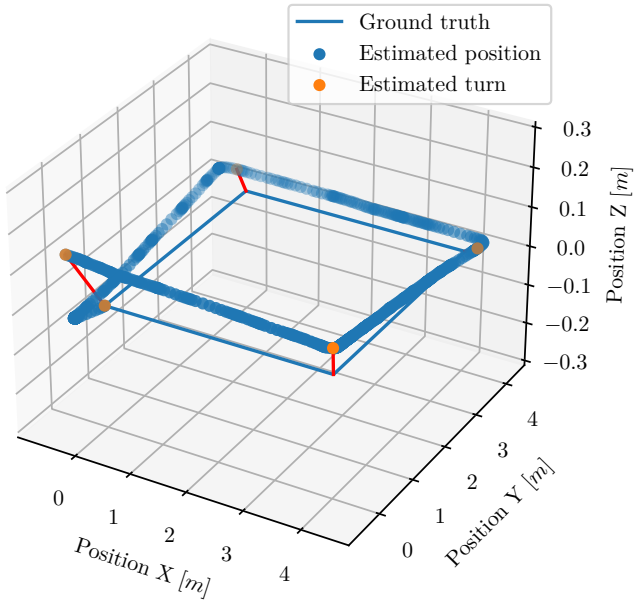
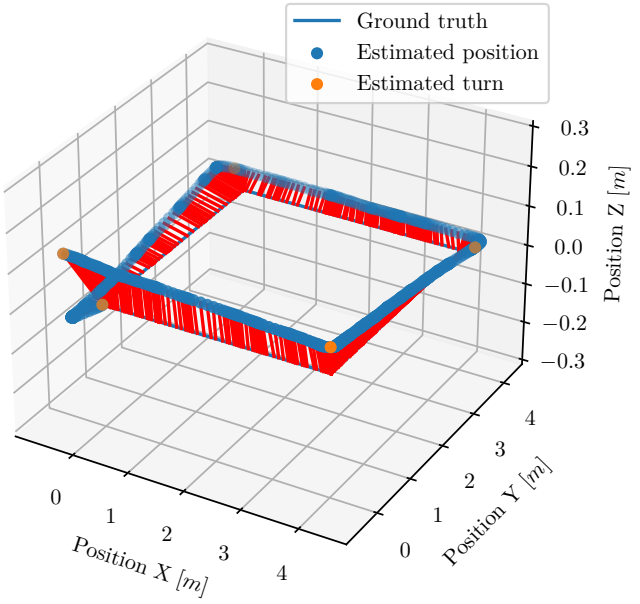


Fig. 40: The same methodology applies to three dimensional estimation. The real world test ground truth (represented by the 4mx4m blue line square) and how the estimated path compares. Orange dots represent the estimated corners of the Ramer-Douglas-Peucker Algorithm.

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, it is possible to have an overall understanding of how the estimation performed.

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


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


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

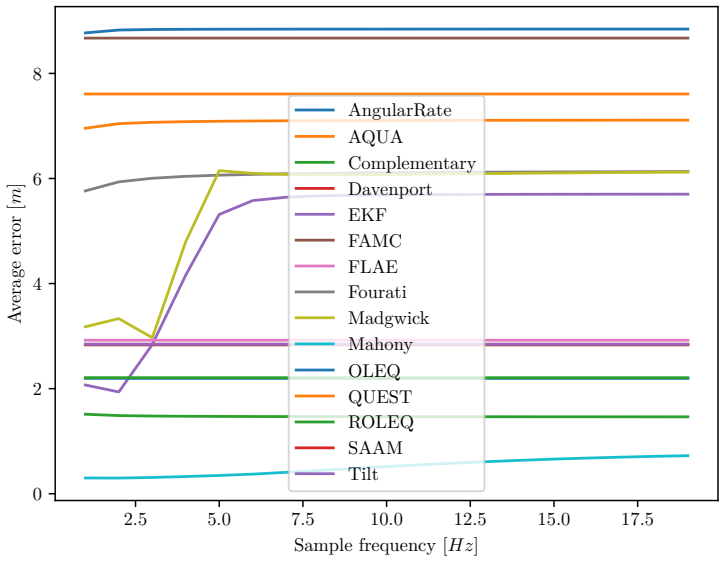


Fig. 42: The second error measurement technique where every point in the estimated position path is connected their closest neighbor in the ground truth's line. Averaging every distance here can give a better understanding of how precise the estimation was.

6 Results

This section presents results from the different set of experiments mentioned in chapter 4.4 Experiments. For every experiment, a table comparing every algorithm's displacement and turn error is present. And a error percentage for that particular shape. And lastly, a 2D and 3D visualization of the best performing algorithm for that particular experiment is present for every test.

6.1 Line

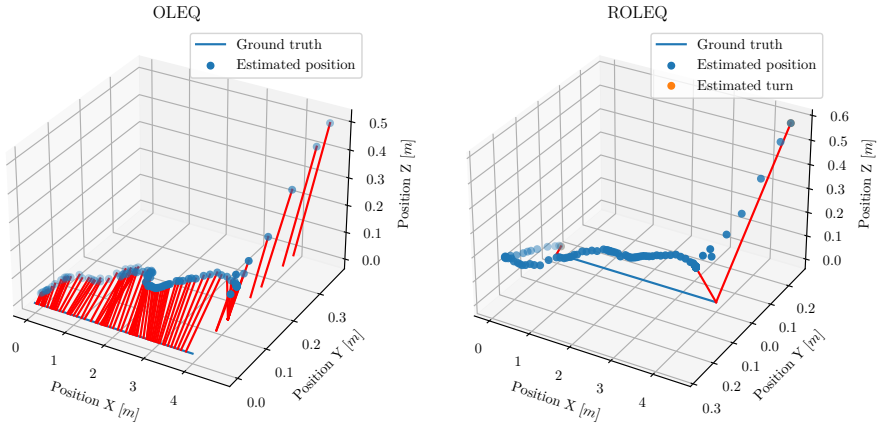
The line shape consisted of moving the inertial system in a straight line for a determined distance. 3 line distances were tested: 4, 16, and 28 meter. The results are shown bellow:

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 7: Accelerometer Specifications.



(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. 43: 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 Mahony algorithm which had the lowest displacement error with an average of 0.48 meters (16% of error margin), and ROLEQ with an average of 0.24 meters of turn error (7% of error margin).

6.1.3 28 meter

For the 28 meter line experiment, the OLEQ algorithm which had the lowest displacement error with an average of 0.16 meters (16% of error margin), and ROLEQ with an average of 0.24 meters of turn error (7% of error margin).

6.2 Triangle

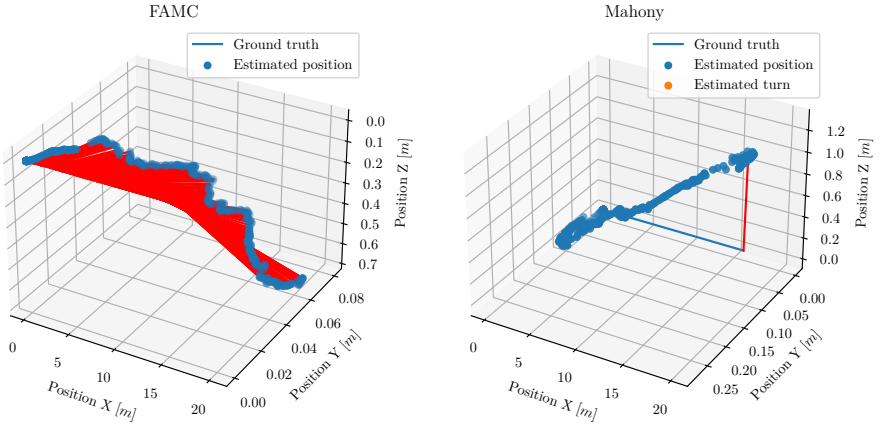
The line shape consisted of moving the inertial system in a straight line for a determined distance. 3 line distances were tested: 4, 16, and 28 meter. The results are shown bellow:

6.2.1 4 meter

For the 16 meter line experiment, the Mahony algorithm which had the lowest displacement error with an average of 0.48 meters (16% of error margin), and ROLEQ with an average of 0.24 meters of turn error (7% 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 8: Accelerometer Specifications.



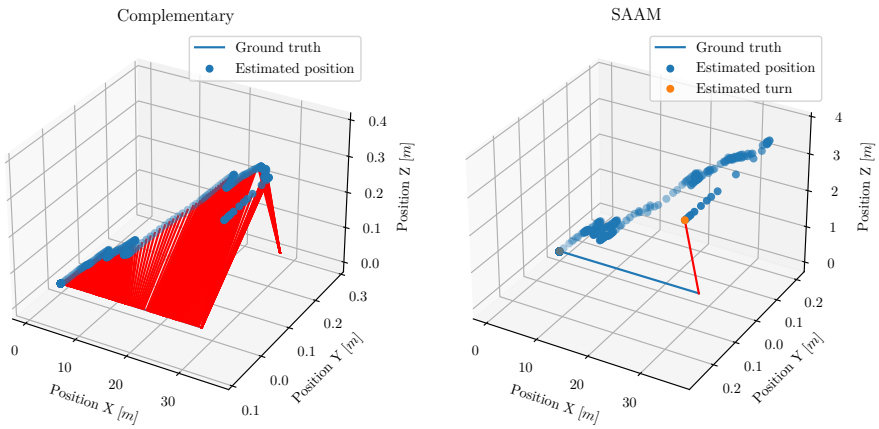
(a) ROLEQ's 3D position estimation had the lowest turn error for the 4 meter line experiment.

(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.

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 9: Accelerometer Specifications.



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

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

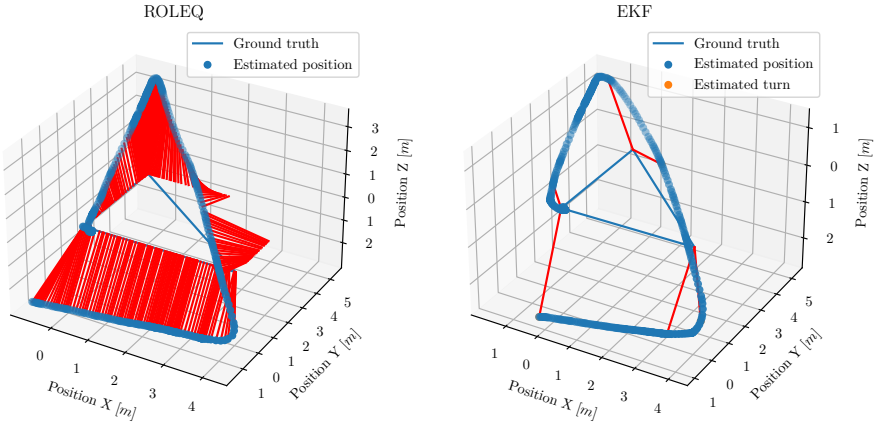
Fig. 45: Position estimation by the best performing algorithms in the 4 meter line experiment.

6.2.2 16 meter

For the 16 meter line experiment, the Mahony algorithm which had the lowest displacement error with an average of 0.48 meters (16% of error margin), and ROLEQ with an average of 0.24 meters of turn error (7% of error margin).

Algorithm	Displacement Error[m]	Displacement Error[%]	Turn Error[m]	Turn Error[%]
AngularRate	4.50	37.47	6.51	54.28
AQUA	2.91	24.24	5.66	47.18
Complementary	0.85	7.08	1.46	12.14
Davenport	0.93	7.75	1.36	11.35
EKF	1.02	8.50	0.92	7.66
FAMC	4.23	35.23	6.19	51.56
FLAE	0.91	7.54	1.30	10.82
Fourati	7.04	58.67	8.32	69.35
Madgwick	1.68	14.03	2.00	16.63
Mahony	0.79	6.60	0.97	8.11
OLEQ	0.79	6.57	1.02	8.53
QUEST	2.46	20.54	3.30	27.49
ROLEQ	0.78	6.54	0.84	6.98
SAAM	0.87	7.24	1.10	9.16
Tilt	0.87	7.24	1.10	9.16
Average	2.04	17.02	2.80	23.36

Table 10: Accelerometer Specifications.



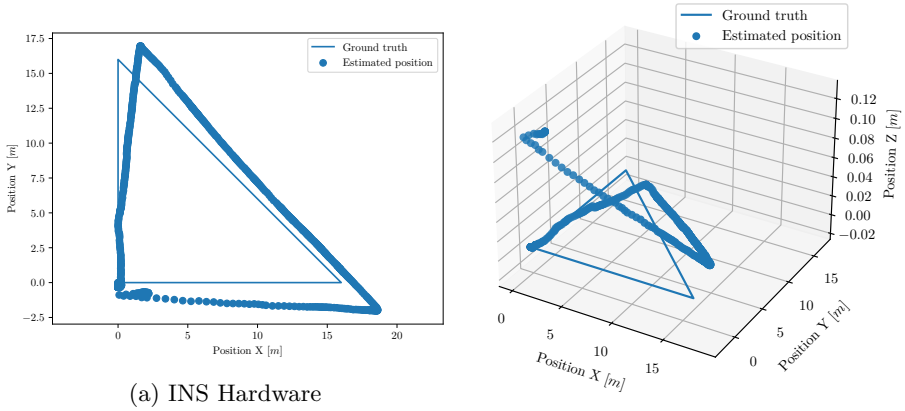
(a) INS Hardware

(b) MPU-9250 Breakout

Fig. 46: Pin connection between the microcontroller (left) and the IMU (right). Modules are linked through SCL, CLK, VDD and GND pins.

Algorithm	Displacement Error[m]	Displacement Error[%]	Turn Error[m]	Turn Error[%]
AngularRate	12.53	26.10	18.91	39.40
AQUA	7.48	15.58	12.02	25.05
Complementary	2.35	4.90	3.72	7.75
Davenport	1.85	3.86	2.57	5.35
EKF	2.03	4.22	2.45	5.10
FAMC	11.55	24.05	19.62	40.87
FLAE	1.85	3.86	2.49	5.18
Fourati	21.75	45.32	29.43	61.32
Madgwick	6.25	13.02	7.85	16.36
Mahony	1.64	3.43	2.20	4.58
OLEQ	1.66	3.46	3.76	7.84
QUEST	8.39	17.48	17.65	36.77
ROLEQ	1.82	3.80	4.36	9.07
SAAM	1.67	3.48	2.13	4.43
Tilt	1.67	3.48	2.13	4.43
Average	5.63	11.74	8.75	18.23

Table 11: Accelerometer Specifications.



(a) INS Hardware

(b) MPU-9250 Breakout

Fig. 47: Pin connection between the microcontroller (left) and the IMU (right). Modules are linked through SCL, CLK, VDD and GND pins.

7 Discussion

To develop and determine an accurate positioning system is a challenge on many levels as it requires computationally advanced electronic components

that can perform highly accurate calculations. The difficulties faced when solving this problem arise when the determining the orientation of an object as it produces many sources of error. The downside of having a well oriented system is at the expense of detection of low velocities and vise versa. The positioning system performs at high accuracy (cm precision) during non-rotational movements but has trouble with drift when the object is rotated. This is due to the low convergence rate of the orientation filter which is clearly a big system limitation. A suggestion on solving this problem could be making the tuning parameters of the filter more dynamic which would make the system more adaptable to extreme rotations and low velocities.

Common problems faced when trying to implement an accurate position lies in the gyroscope because it tends to drift significantly over time. The observations made during the studying of other positioning systems have been that they all suffer in accuracy due the gyroscope drift. These findings lead to the consideration of reducing the gyroscope drift significantly. Methods and algorithms have been presented in this thesis which evidently reduced the drift components and these methods have helped increase the accuracy of the positioning system. Similarly to other systems, this system suffers from inaccuracies. Due to the gyroscope drift reduction algorithms developed, this system stands out from other previously developed systems in that sense. To develop a very accurate positioning system using low cost MEMS sensors is a challenging subject that comes with many complications. The system requirements are very strict due to the environmental, economical and accuracy specifications. Therefore implementing a system of very high accuracy, low cost and high mobility is very difficult. In order to achieve a higher accuracy, the electronic components need to have a higher performance, higher resolution and a higher sampling frequency. The micro controller unit which is the heart of the whole system needs to have a faster processor but in the end, it is the sensors that limit the system's performance.

8 Conclusion

Conceptualizing, building, and developing an Inertial Navigation System for position estimation was a challenging problem. This work permitted me to gain a deeper comprehension, understanding and appreciation of how 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. 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

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

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

[12] 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.

[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

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

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

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

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

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

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

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

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

[43] 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.

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

[45] 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.

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