Lost in the Deep? Performance Evaluation of Dead Reckoning Techniques in Underwater Environments

MARKO RADETA*, Wave Labs, Faculty of Exact Sciences and Engineering, University of Madeira, Portugal, MARE – Marine and Environmental Sciences Centre / ARNET - Aquatic Research Network, Agência Regional para o Desenvolvimento da Investigação Tecnologia e Inovação (ARDITI), Portugal, Portugal, and Department of Astronomy, Faculty of Mathematics, University of Belgrade, Serbia, Portugal
CLAUDIO RODRIGUES, Wave Labs, MARE/ARNET/ARDITI, University of Madeira, Portugal
FRANCISCO SILVA, Wave Labs, MARE/ARNET/ARDITI, University of Madeira, Portugal
PEDRO ABREU, Wave Labs, MARE/ARNET/ARDITI, University of Madeira, Portugal
JOÃO PESTANA, Wave Labs, MARE/ARNET/ARDITI, University of Madeira, Portugal
NGOC THI NGUYEN, Department of Computer Science, University of Helsinki, Finland
AGUSTIN ZUNIGA, Department of Computer Science, University of Helsinki, Finland
HUBER FLORES, Institute of Computer Science, University of Tartu, Estonia
PETTERI NURMI, Department of Computer Science, University of Helsinki, Finland

Computing research is increasingly addressing underwater environments and examining how computing can support diving and other activities. Unlike on land, where well-established positioning methods are widely available, underwater environments lack a common positioning mechanism, which is a prerequisite for many applications. Dead reckoning, the use of angle and distance estimates to track position changes from a known point of origin, is a promising candidate for underwater positioning as it does not rely on wireless signals (which decay rapidly in underwater environments) and as there is a wide range of literature and algorithms freely available. Yet, currently it is unclear whether the existing techniques can be adopted in underwater environments or whether the differences in medium and environment affect the performance of the dead reckoning techniques. We contribute by evaluating and systematically analyzing the performance and trade-offs associated with dead reckoning techniques in underwater environments. We present AEOLUS, a prototype unit comprising of a low-cost microcontroller and inertial measurement unit, to perform experiments on the ground and in underwater environments to

*Corresponding author

Authors' addresses: Marko Radeta, marko@wave-labs.org, Wave Labs, Faculty of Exact Sciences and Engineering, University of Madeira, Funchal, Portugal and MARE – Marine and Environmental Sciences Centre / ARNET - Aquatic Research Network, Agência Regional para o Desenvolvimento da Investigação Tecnologia e Inovação (ARDITI), Portugal, Funchal, Portugal and Department of Astronomy, Faculty of Mathematics, University of Belgrade, Serbia, Funchal, Portugal; Claudio Rodrigues, claudio@wave-labs.org, Wave Labs, MARE/ARNET/ARDITI, University of Madeira, Funchal, Portugal; Francisco Silva, francisco@wave-labs.org, Wave Labs, MARE/ARNET/ARDITI, University of Madeira, Funchal, Portugal; Pedro Abreu, pedro@wave-labs.org, Wave Labs, MARE/ARNET/ARDITI, University of Madeira, Funchal, Portugal; João Pestana, joao@wave-labs.org, Wave Labs, MARE/ARNET/ARDITI, University of Madeira, Funchal, Portugal; Ngoc Thi Nguyen, ngoc.t.nguyen@helsinki.fi, Department of Computer Science, University of Helsinki, Helsinki, Finland; Agustin Zuniga, Department of Computer Science, University of Helsinki, Helsinki, Finland, agustin.zuniga@helsinki.fi; Huber Flores, Institute of Computer Science, University of Tartu, Tartu, Estonia, huber.flores@ut.ee; Petteri Nurmi, Department of Computer Science, University of Helsinki, Helsinki, Finland, petteri.nurmi@helsinki.fi.

2474-9567/2023/6-ART73 $15.00
https://doi.org/10.1145/3596245

assess how well the performance of different techniques translates from ground-based use cases to underwater environments. We benchmark 15 different algorithms and compare their performance in such environments to identify common patterns and dissimilarities, and identify root causes for these differences. The results show that displacement and turn errors can be estimated to within 5% error but that the best performing methods vary between land and underwater environments. We also show that the performance depends on the shape of the motion patterns with some algorithms performing better for hard turns whereas others perform better for gradual, more continuous turns.

CCS Concepts: • **Human-centered computing** → *Ubiquitous and mobile computing design and evaluation methods.*

Additional Key Words and Phrases: Sensor Fusion, Dead Reckoning, Water Sensing, Internet of Things, AHRS, Orientation Estimation, Position Estimation, Location Estimation, IMU, Underwater Positioning, Review.

ACM Reference Format:
Marko Radeta, Claudio Rodrigues, Francisco Silva, Pedro Abreu, João Pestana, Ngoc Thi Nguyen, Agustin Zuniga, Huber Flores, and Petteri Nurmi. 2023. Lost in the Deep? Performance Evaluation of Dead Reckoning Techniques in Underwater Environments. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 2, Article 73 (June 2023), 27 pages. https://doi.org/10.1145/3596245

1 INTRODUCTION

Computing research is increasingly targeting underwater environments [71]. For example, underwater sensor networks are used to monitor seabed events [91], remotely operated drones are used for monitoring pipelines and other underwater infrastructure [3, 54], and scuba divers can use underwater vehicles or smaller computing units to support underwater investigations of biodiversity, pollutants, and other factors [58]. Despite applications addressing underwater environments becoming increasingly common, building them remains highly challenging due to the fact that underwater environments severely restrict the propagation of wireless signals, making common positioning and communications solutions inaccessible [28, 73]. While the latter task, the development of underwater communications systems, has been an active research area [6, 44], the development of underwater positioning solutions has received much less attention.

Dead reckoning, also known as inertial navigation or incremental positioning, is a common technique for localization that operates by continually tracking the movements of an object from a known starting point using distance and angle estimates. Dead reckoning usually relies on inertial measurement units (IMUs) that integrate accelerometers, gyroscopes, magnetometers, and potentially also barometers, to estimate changes in locations. IMUs are highly attractive as positioning solutions as they are affordable, lightweight, and energy-efficient [16]. Dead reckoning is already widely adopted in many domains, including aerial drones [32, 47, 76], surface vessels [69], and a wide range of land-based applications [49, 86]. While dead reckoning has been able to operate reliably in these applications, it is far from trivial to generalize these results to underwater environments. First, dead reckoning solutions are prone to drift as errors in individual angle and distance estimates accumulate over time [37]. Second, currents and turbulence are unpredictable and inconsistent in underwater environments which makes it difficult to remove their effects over time, potentially affecting the accumulation of errors. Third, underwater environments lack clear reference points for re-calibrating the dead reckoning algorithms, meaning that controlling for the errors can become highly difficult over time. Regardless of these challenges, dead reckoning remains an attractive proposition for underwater environments as the main alternatives, such as sonars, lidars, and vision-based approaches, are costly and highly resource intense [93].

The present paper contributes by conducting an extensive and systematic performance evaluation of dead reckoning techniques for underwater environments. We develop a simple prototype that integrates IMUs with a low-cost microcontroller and collected measurements separately from the ground and underwater environments, using four movement patterns that are common in spatial sampling designs used to cover designated spatial areas and thus are essential for navigation (line, triangle, square, and a warp pattern) [29]. Examples of applications that benefit from these patterns include practically all monitoring applications, such as pipeline, plastic debris,

fish farm, or reef monitoring [35, 71]. We benchmark the performance of 15 different algorithms and compare their performance between the ground and underwater environments. Our results show that dead reckoning techniques can provide sufficient accuracy in underwater environments (with 5% error for displacement and turn respectively) but there are significant differences across different algorithms. The performance of different techniques also depends on the type of motion pattern, with some algorithms struggling at continually changing angles. Finally, we observe that the relative differences in the distance and angle estimates are consistent regardless of the length of the motion pattern. Based on our results, we also discuss best practices and identify gaps and open research challenges for the integration of dead reckoning techniques into underwater environments.

2 BACKGROUND AND RELATED WORK

2.1 Position Estimation with Inertial Sensor Systems

Inertial sensor systems have been thoroughly researched with the purpose of delivering position estimation of a moving body. Inertial systems are typically autonomous, independent and do not rely on external information, such as radio signals or electromagnetic waves [18]. Their navigation data can have short-term high accuracy, and elevated data update rate [1]. Some of the known techniques are listed below.

2.1.1 Pedestrian Dead Reckoning (PDR). PDR is among the most explored, which combines step detection, step length estimation, and orientation approximations to calculate the absolute position and heading of a person walking. PDR can operate with a single accelerometer, although superior precision and robustness are obtained with more sensors [26]. An IMU containing several accelerometers, gyroscopes, magnetometers, and even pressure sensors are commonly employed to recognize steps and orientation. Such sensors are typically body- 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 [33]. Ladetto et al. [43] 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. Their 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. [80] illustrated an experiment exploiting a shoe-mounted sensor prototype that calculates stride length with accelerometers and magnetometers. Their system measured angular acceleration by manipulating pairs of accelerometers as an alternative to gyroscopes. Several other studies investigated the prospect of using inertial sensor systems to estimate the absolute position and heading of a walking user for multiple purposes [12, 40, 79, 85]. With a focus on low-cost inertial motion sensors, Coyte et al. [13] applied PDR to sporting training and rehabilitation. They proposed solutions to acceleration noise accumulation and gyroscope angle error problems, improving the accuracy of displacement estimation with a low-grade IMU and a Zero-Velocity Update (ZVU) algorithm. However, the main drawback of PDR is its dependence on step prediction algorithms that must distinguish step direction and step lengths as the user changes the pace. Such algorithms remain challenging when applied to environments with aerial or underwater movements, such as in the case of identifying SCUBA diver's kicks [55].

2.1.2 Strapdown Inertial Integration (SNIS). SNIS is another prevalent position estimation method where sensors are usually tightly strapped or attached to the axes of the moving body's structure. Such a 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 $2000Hz$, as 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. [36] introduced SNIS to nautical environments, proposing a compensation method against disturbing forces affecting vessel motion caused by rough sea conditions. They aimed to resolve attitude determination offset through self-alignment of SNIS by establishing vector observations. The such 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 also described by Bird et al. [7]. Similarly to PDR, SNIS also has applications in pedestrian navigation systems, although operating in an utterly distinct fashion. Unlike PDR, the SNIS 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 [75]. This is possible because of a 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.

2.2 Sensor Fusion in Position and Orientation Estimation

Sensor fusion is defined as the blending of sensory information from two or more sources in a way that generates a more consistent and dependable understanding of the system [27]. Fusing multiple inertial systems has raised significant interest in the improvement of location and attitude estimation performance. Numerous methods arose that merge information from various systems such as inertial sensors, GNSS [84], radar, radio telescopes, signal of opportunity systems like Angle of Arrival (AOA), Time of Arrival (TOA), Received Signal Strength Indicator (RSSI) [70], and Signal to Noise Ratio (SNR). Indeed, the combination of multiple sources can help reduce noise with two different sensor types. These separate systems are integrated by sensor fusion (essentially filter) algorithms which process each input and generate a more precise and reliable output [19]. A substantial sum of distinct solutions designed to assess the orientation of a rigid body in reference three-dimensional frames exists, while two main approaches aimed at sensor fusion are Kalman and complementary related filters.

2.2.1 Kalman Filters. Several research has been conducted on Inertial Navigation System (INS) and Global Navigation Satellite System (GNSS) integration through data fusion, particularly using the Kalman Filter. To overcome the shortcomings linked to the detached functioning of GNSS and INS, Wong et al. [88], Qi et al. [68], and Nassar et al. [59] 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, the INS had a more reliable and comprehensive short-term signal. Updating INS position and velocity with GNSS data corrected error expansion and at the same time 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. 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 [53] performed an analysis on INS/GNSS alternative integration through an adaptive Kalman filtering technique. Findings revealed that their adaptive Kalman filter outperformed by almost 50% the conventional filters.

Among tests on Unmanned Aerial Vehicles (UAVs), the use of sensor fusion has grown 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. Hetényi et al. [32] applied a Kalman Filter to fuse the sonar and accelerometer signals, obtaining a considerably improved altitude estimation with minimal error. Similarly, Luo et al. [47] combined the UAV sensor system and RSSI 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 radio frequency signals, further reducing the noise component. With their application on robots, Sharma et al. [76] presented an experiment of Kalman filter-based sensor fusion for extrapolation of 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.

2.2.2 Complementary Filter (CF). Madgwick et al. outlined the formulation of an Extended Complementary Filter (ECF) algorithm and exhibited its applicability as a human motion monitoring wearable [49]. 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. Authors claimed an improved computational efficiency of over 30% when compared against standard alternative algorithms. A complementary filter designed for sensor fusion in quadrotor UAV employing a low-cost inertial measurement system was also proposed [60]. The CF filtered high-frequency signals associated with the gyroscope and low-frequency signals linked to the accelerometer. Their findings demonstrated that the complementary filter technique overcame the over-drift conundrum related to gyroscopes, being capable of efficiently computing attitude angles. Similarly, Euston et al. conducted an analogous study with a non-linear CF for attitude estimation in a UAV utilizing a low-cost IMU [21]. 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 (EKF) that fused GPS/INS at a far less computational cost.

2.2.3 Other Position Estimators. Among the plethora of existing estimation algorithms, most of them are typically based on quaternions, and focus on integrating the differential equations to obtain a local rotation rate [78]. For instance, the Algebraic Quaternion Algorithm (AQUA) estimates also a quaternion using magnetic observations, by separately calculating the "tilt" and "heading", avoiding the impact of magnetic disturbances on the pitch and roll components of the orientation [83]. Davenport also uses quaternions for rotation between two coordinate systems [15]. Fast Accelerometer-Magnetometer Combination (FAMC) integrates the local gravity and the Earth's magnetic field [89], providing simplification of Davenport's approach, resulting in the reduction in time consumption. Fast Linear Attitude Estimator (FLAE) uses the attitude quaternion with an eigenvalue-based solution, obtaining polynomial derived for a higher computation speed [89]. Fourati uses the values from an accelerometer, magnetometer and gyroscope combined and based on the time integral of the angular velocity using Earth's magnetic field and gravity vector to compensate for what was predicted by the gyroscope [22]. Madgwick also stands on quaternions, providing orientations in three dimensions, avoiding gimbal locks traditionally made by the Euler angle representation [48]. Other estimators such as Mahony require accelerometer and gyroscope for providing good estimates as well as online gyro bias computation [51]. Optimal Linear Estimator of Quaternion (OLEQ) further enhances quaternions, applying least-squares to the set of pre-computed single rotated quaternions [95]. QUEST is an algorithm that determines the attitude from the best-weighted overlap of reference and observation vectors [77]. Additionally, the Recursive Optimal Linear Estimator of Quaternion (ROLEQ) is a modified version of OLEQ, with the attitude measuring the angular velocity [95], simplifying further the rotational operations. Super-fast Attitude from Accelerometer and Magnetometer (SAAM) is streamlined Davenport's solution, being reduced to floating point operations to spare computational time [89]. TILT provides further attitude estimation using gravity acceleration measurements [82].

Overall, some studies have conducted comparisons between the sensor fusion algorithms in distinct settings to assess their performance in unique conditions. For instance, Ludwig et al. compared Madgwick and Mahony in a foot-mounted experiment [45], finding that Madgwick achieved better heading orientation than Mahony. Nonetheless, the overall performance of Mahony was still superior to Madgwick [45]. The same authors tested on quadcopters the EKF, Madgwick, and Mahony filters [46]. Results showed that Mahony delivered a more precise orientation estimation and faster execution time than Madgwick and EKF. Also, Diaz et al. presented a comparison among Madgwick and Mahony, a basic Attitude Heading Reference System (AHRS) estimation algorithm [17]. The study centered around comparing the performance of Madgwick, Mahony, EKF, and their own sensor fusion algorithm, emphasizing the behavior under magnetic perturbations. Various examples of the

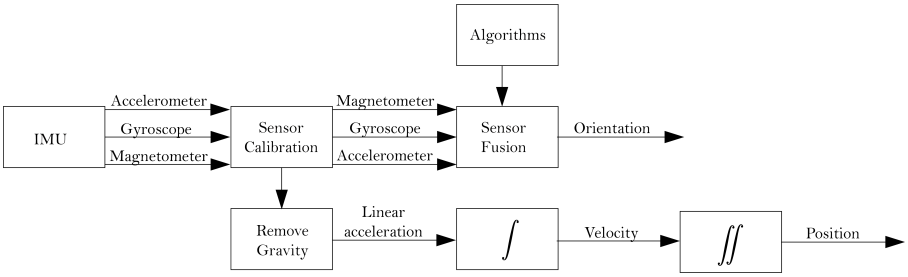


Fig. 1. AEOLUS Position estimation pipeline.

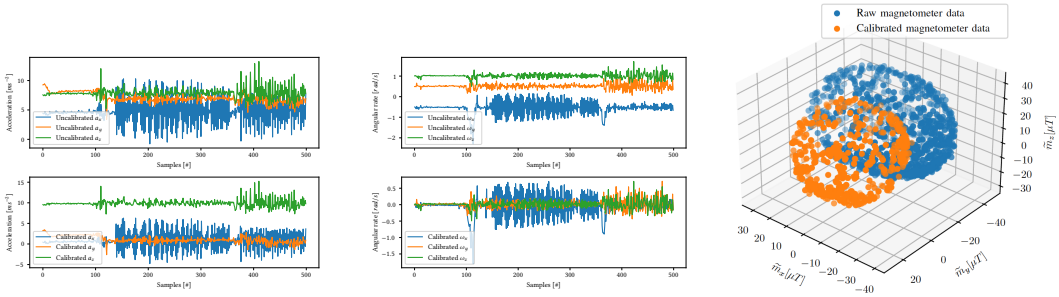


Fig. 2. Calibration procedure during sampling (from left to right): (a) Accelerometer: raw (top), calibrated (bottom); (b) Gyroscope: raw (top), calibrated (bottom); and (c) Magnetometer: raw (blue), calibrated (orange).

movement were analyzed, from carrying the sensor at separate places such as pockets, shoes, and hands. Their algorithm was slightly less influenced by magnetic perturbations than the others, but overall, the algorithms performed similarly. Most of these state-of-the-art algorithms will be further benchmarked in this study, and contested both in underwater and on-the-ground environments, simulating the frictionless environment. It remains unknown whether the existing techniques can be adopted in underwater environments or whether the differences in medium and environment affect the performance of the dead reckoning techniques.

3 AEOLUS POSITION ESTIMATION PIPELINE

We next describe the procedure to assess the system's performance describing procedures for IMU sensor calibration. Moreover, steps for orientation, displacement, and position estimation are shown. The selection of sampling frequency is described. Orientation is obtained using sensor fusion while the displacement uses double integration of accelerometer readings, removing the gravity influence (Figure 1). The position method combined orientation and displacement using known dead reckoning techniques, mentioned in the related work.

3.1 Sensor Calibration

Calibration of inertial sensors is essential for the compensation of their systematic errors, biases, and scale factors [90]. Prior to any data collection, the inertial sensor is calibrated while the system is stationary and stabilized to compensate for static errors that might corrupt the measurements. The used IMU throughout this study is MPU9250 and will be detailed in the next section.

3.1.1 Accelerometer. 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 [87]. 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 obtain a linear fit between the three values and the output values accelerometer axis. The result of the accelerometer's calibration offset correction (Figure 2a), where the calibrated output shows an offset correction with the a_x and a_y readings centered at 0 m/s^2 while the a_z oscillating around 9.8 m/s^2.

3.1.2 Gyroscope. The simplest IMU calibration consists in calculating the offset for each gyroscope axis. The gyroscope is the easiest among other sensors for calibration due to the expected readings in steady conditions. Each of the three axes of the gyro should read 0 radians-per-second (rad/s) when the IMU is not moving. The offsets can be measured by first taking some readings while the IMU is not moving, then using those values as 'offsets' when reading the gyro values in the future [61]. The result of the gyroscope's calibration offset correction indicates shows adequate offsets with the ω_x, ω_y, and ω_z readings centered at 0 rad/s (Figure 2b).

3.1.3 Magnetometer. To calibrate the magnetometer, a series of measurements of the nearby magnetic field are taken while holding the sensor in an "eight/infinity" figure pattern, passing through every orientation possible [74]. Ideally, the measurements should portray a perfect sphere centered at the coordinate origin and where the radius of the sphere is the magnetic field's strength [41]. Plotting the default magnetometer's raw measurements (as seen in Figure 2c), it is visible that these do not form a perfect sphere, nor that they are centered at the origin. These are referred to as "hard iron" and "soft iron" errors or biases, respectively [90]. If no magnetic anomaly algorithm is applied to such distortions in the magnetic field, it will cause changes in orientation.

Hard iron biases are commonly the largest and the simplest errors to rectify [23]. 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, resulting in the re-centering of the field at the origin of coordinate system. Expression 1 illustrates how a hard iron correction can be implemented:

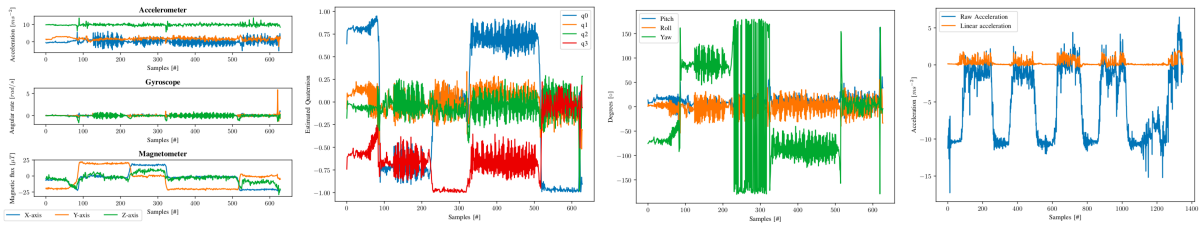
$$offset_x = (max_x + min_x)/2 \tag{1}$$

It is possible now to filter any soft iron biases by taking the minimum and maximum of the measured field and manipulating it to rescale the magnetometer data to normalize the output across $\widetilde{m}_x$, $\widetilde{m}_y$ and $\widetilde{m}_z$. Such is done 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 also known as orthogonal rescaling, identical to a diagonalized 3×3 calibration matrix while enabling further scale bias correction as in expression 2. Rescaling is depicted in Figure 2c.

$$\begin{gathered}\overline{\Delta_x} = \frac{(max_x - min_x)}{2}, \overline{\Delta_y} = \frac{(max_y - min_y)}{2}, \overline{\Delta_z} = \frac{(max_z - min_z)}{2}, \overline{\Delta} = (\overline{\Delta_x} + \overline{\Delta_y} + \overline{\Delta_z})/3 \\ scale_x = \overline{\Delta}/\overline{\Delta_x}, scale_y = \overline{\Delta}/\overline{\Delta_y}, scale_z = \overline{\Delta}/\overline{\Delta_z}\end{gathered} \tag{2}$$

3.2 Orientation Estimation

After performing calibration, the next step is to obtain the correct orientation using a fusion of the three sensor raw outputs (Figure 3a). The accelerometer provides the device proper acceleration; the gyroscope supplies the device's angular rate, and the magnetometer presents the detected magnetic flux. To successfully estimate the orientation from the raw sensor data, two approaches can be performed: (i) *real-time* – applying a sensor fusion algorithm directly in the inertial system's microcontroller, which estimates orientation in real-time while the measurements were taken. The Madgwick algorithm [49] with gradient descent correction was utilized for its



(a) (b) (c) (d)

Fig. 3. Sensor Fusion procedure (from left to right): (a) Collected raw sensor data with 9 Degrees of Freedom (DoF) IMU: accelerometer, gyroscope and magnetometer; (b) Quaternions obtained using Madgwick's algorithm; (c) Euler angles derived from quaternions; and (d) Raw (blue) and linear acceleration measurements (orange) with gravity compensation. The gravity's acceleration approximates 9.8 m/s^2 when tilting the sensor.

well-recognized proficiency to merge accuracy with computational cost and simplicity of implementation; (ii) *post-processing* – storing the raw sensor data in an SD card to be processed later with the sensor fusion algorithm. Throughout this study, the second approach was used as it facilitated a benchmark of how different sensor fusion algorithms perform under the same collected data. Plots in Figures 3 and 5 are based on data collection and microcontroller running the Madgwick real-time sensor fusion, while the same principle applies to any arbitrary sensor fusion algorithm based on quaternions. The choice of using Madgwick algorithm in the below plots was for its well-recognized proficiency to merge accuracy with computational cost and simplicity of implementation [50].

Sensor fusion algorithm takes as an input the raw sensor measurements from the IMU (Figure 3a) and outputs the estimated orientation as a set of four quaternions (Figure 3b), in an effort to obtain three Euler angles, commonly known as pitch, roll and yaw (Figure 3c). A good theoretical background of understanding quaternions may be found in literature [42]. For brevity, we showcase the derivation of typical quaternion:

$$q = q_0 + q_1 i + q_2 j + q_3 k, \tag{3}$$

where q_0 is the real and q_1, q_2, and q_3 are the imaginary components. Rotation from coordinate frame to any arbitrary frame (e.g. from frame A to B) is given by the conjugation operation, so the expression $qB = qR \times qA \times qR^*$ as a matrix is:

$$M = 2 \begin{bmatrix} q_0^2 + q_0^2 - \frac{1}{2} & q_1 q_2 - q_0 q_3 & q_0 q_2 + q_1 q_3 \\ q_0 q_3 + q_1 q_2 & q_0^2 + q_2^2 - \frac{1}{2} & q_2 q_3 - q_0 q_1 \\ q_1 q_3 - q_0 q_2 & q_0 q_1 + q_2 q_3 & q_0^2 + q_3^2 - \frac{1}{2} \end{bmatrix}. \tag{4}$$

From the obtained matrix, the trace is calculated as:

$$Trace(M) = M_{11} + M_{22} + M_{33} = 4q_0^2 - 1. \tag{5}$$

Using trace, all individual quaternions are found as:

$$\begin{aligned} |q_0| &= \sqrt{\frac{Trace(M)+1}{4}}, \\ |q_1| &= \sqrt{\frac{M_{11}}{2} + \frac{1-Trace(M)}{4}}, \\ |q_2| &= \sqrt{\frac{M_{22}}{2} + \frac{1-Trace(M)}{4}}, \\ |q_3| &= \sqrt{\frac{M_{33}}{2} + \frac{1-Trace(M)}{4}} \end{aligned} \tag{6}$$

From obtained quaternions, we finally obtain the Euler angles and device heading by next expression (Figure 3c):

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \arctan \frac{2(q_0 q_1 + q_2 q_3)}{1-2(q_1^2+q_2^2)} \\ \arcsin(2(q_0 q_2 - q_3 q_1)) \\ \arctan \frac{2(q_0 q_1 + q_1 q_2)}{1-2(q_2^2+q_3^2)} \end{bmatrix} \tag{7}$$

3.3 Obtaining Linear Acceleration

As input accelerometer measurements are affected by the Earth's gravitational field (as seen on X-axis in Figure 3a) such considerably amplify numerical errors. If the gravity component is not subtracted from the physical acceleration of the device, it 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. We rotate the accelerometer reading by the obtained quaternion into the Earth frame of reference (Earth's spherical coordinate system), then subtract the gravity from it. The residual acceleration is the acceleration of the sensor in the Earth frame of reference often called the linear acceleration. Through orientation estimation determined previously, it is possible to find the orientation of the Earth frame with respect to the sensor frame. Therefore, we compute the expected direction of gravity and then subtract that from the accelerometer readings (Figure 3d).

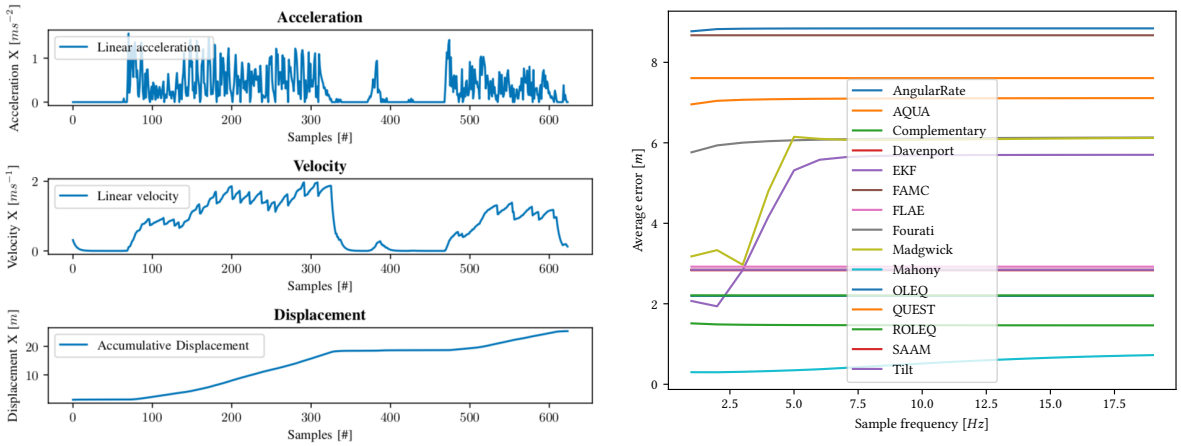
3.4 Sampling Rate Selection

Prior empirical testing was conducted to select the sampling rate by performing the estimation of line geometrical shape trajectory of 4 m, reaching a maximum 20Hz due to hardware limitations of the used inertial system. Selected sampling frequency was chosen (6 Hz), resulting in a frequency where all of the state-of-the-art estimation algorithms reached a plateau (Figure 4b). Hereinafter, we assume that such sampling frequency holds true when applied to all tested environments. This frequency also made sense considering that the magnetometer samples were obtained at a rate of 8 Hz which is a technical constraint that we selected for this study. Used hardware and AEOLUS inertial system will be described in the next section.

3.5 Displacement Estimation

Once the heading is obtained, we next estimate the displacement. As accelerometer readings measure proper acceleration forces, a common practice is to determine device velocity and position relative to a starting point [39]. From obtained linear acceleration, numerical integration yields velocity, and double integration yield the position vector (Figure 4a). Inertial frame velocity v_i and position x_i we calculate as:

$$v_i = v_0 + \int^t a_i \, \Delta t \quad \text{and} \quad x_i = x_0 + \int^t v_i \, \Delta t. \tag{8}$$



(a) (b)

Fig. 4. (a) Example of displacement estimation obtained from acceleration measurements in X-axis; (b) Selection of sampling frequency for each algorithm based on their average error.

where i indicates any of the vector three axis in Euclidean space x, y, z. From prior empirical tests, using the existing setup showed that the system would come to a stationary state. However, the software would keep adding and summing the position vectors. This is a known fact that perceived velocity will be increased (and hence accumulated displacement) since double integrations of acceleration (even if the sensed acceleration in the known axis is 0m/s^2) can cause drifts [92]. Thus, in our drift reduction Algorithm 1, we added additional factors to zero out the accumulated perceived velocity when accelerometer was indicating no acceleration using $Vx* = 0.55$, $Vy* = 0.55$ and $Vz* = 0.80$, which were calibrated from the range of 0 to 1 for yielding minimal distances between the estimated corner positions and those of the geometry baseline. The selection of these constants was based on prior empirical tests by manually matching the obtained displacements with baseline shape positions.

Algorithm 1 AEOLUS Drift Reduction Algorithm

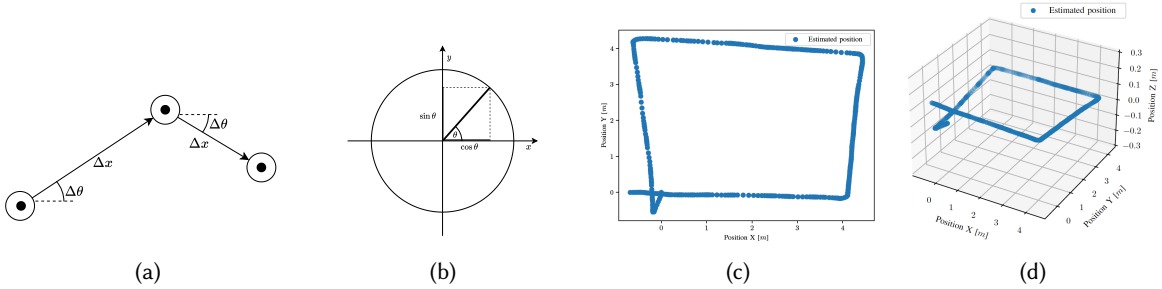
if $acc_X == 0.00$ **then**
  $vel_X \leftarrow vel_X * 0.55$
else
  $vel_X \leftarrow acc_X * \Delta_t$
end if

if $acc_Y == 0.00$ **then**
  $vel_Y \leftarrow vel_Y * 0.55$
else
  $vel_Y \leftarrow acc_Y * \Delta_t$
end if

if $acc_Z == 0.0$ **then**
  $vel_Z \leftarrow vel_Z * 0.80$
else
  $vel_Z \leftarrow acc_Z * \Delta_t$
end if

3.6 Position Estimation

To obtain the exact position of the object, the dead reckoning technique (expression 9, Figure 5a) is applied so the new position x_i, is given by applying a direction vector to the previously determined position x_{i-1}. The norm of the direction vector is given by the displacement in that axis over the period from the last sample (Δx) and its angle from the decomposition of attitude estimation (yaw θ) into two components, $\cos\theta$ for the x axis, and $\sin\theta$ for the y axis (as shown in Figure 5b). An example of integrations using two axes is depicted in Figure 5c.



(a) (b) (c) (d)

Fig. 5. Dead reckoning approach, from left to right: (a) Current object's position is determined by the last known position (heading angle $\Delta\theta$ and traveled distance Δx); (b) Attitude angle θ seen in polar coordinates; (c) Estimated position using Madgwick algorithm of an arbitrary trajectory in Cartesian coordinate system seen in 2D; and (d) Same trajectory seen in 3D.

Expanding the same method for three-dimensional estimation is seen in Figure 5d.

$$x_i = \Delta x \cos(\theta) + x_{i-1} \quad \text{and} \quad y_i = \Delta y \sin(\theta) + y_{i-1} \tag{9}$$

4 APPARATUS: AEOLUS INERTIAL SYSTEM

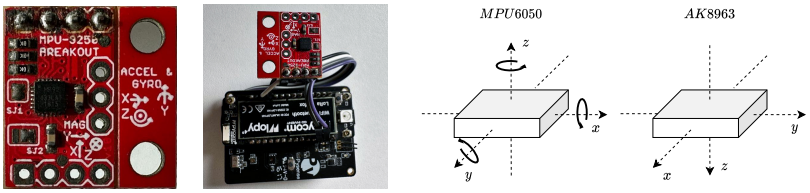
We next describe the used hardware and software to leverage the AEOLUS position estimation pipeline.

4.1 Hardware

LoPy4 microcontroller (based on ESP32) was selected as the navigational computing device of the inertial system since it provides integrated WiFi which will be leveraged for data collection [62]. The LoPy expansion board was used to interface with the external physical inertial sensor through connection pins (Figure 6b). The used IMU was MPU-9250 (Figure 6a), being one of the most widely available low-cost commercial sensors. This chip is extensively employed in wearable sensors for health [2, 52], fitness [4], and sports [20, 56, 72], motion-based game controllers [30], and portable gaming [8, 57]. This IMU is operated to estimate motion by identifying the presence of acceleration vectors, rotational rates, and local magnetic field direction. It encompasses an MPU-6050 (3-axis accelerometer and 3-axis gyroscope combination) and an AK8963 3-axis magnetometer [65]. The MPU-6050 coordinate system is based on a traditional Cartesian coordinate system with a counter-clockwise rotation as the positive rotation direction [64]. The AK8963 shifts the x and y-axis directions while reversing the direction of the z-axis [63], as depicted in Figure 6c.

4.2 Underwater Enhancement

For the mechanical housing of the sensors tailored for water environments, we used existing apparatus made of a transparent acrylic sphere capable of withstanding pressure at a depth of 10m [14, 71]. The IMU, MCU, and the power bank were attached to the inner base of the sphere using J-Hook adapters for sports cameras. For easy handling of the sphere underwater, we further added a 50 mm rigid Chlorinated Hydrocarbon Polymer (PVC) tube extension being 3 mm thick. The plate had a diameter of 30 cm, on which the holes were been manually drilled at one end to facilitate attachment to the plastic base of the sphere using zip ties. At the opposite end of the tube, lead diving weights were attached to the line, creating a downward force from the sphere, resulting in 5 kg weight for neutral buoyancy at 5 m depth. Image of the housing and sensor is seen in Figure 8c.



(a) (b) (c)

Fig. 6. Microcontroller and IMU used in the experiments: (a) IMU (MPU-9250); (b) Pin connection between microcontroller and IMU; and (c) Coordinate system of the 3-axis accelerometer, 3-axis gyroscope (MPU-6050) and 3-axis magnetometer (AK8963).

4.3 Software

The microcontroller ran MicroPython, already tested in constrained environments [81]. The inertial module's raw measurements were interpreted by the microcontroller through the i2c MicroPython driver serial, allowing it to read the peripheral's memory addresses synchronously. The i2c communication makes use of the SDA and SCL pins of the MPU9250. The SCL offered the serial clock, and the SDA provided the raw serial data. The memory address of the MPU9250 was at address 0x68 when the pin was connected to the ground and was set by the pin AD0. The AK8963's address is by default at $0x0C$. Prior to the communication begins the SCL and SDA are equally high, triggered by the pull-up resistors. These pull-up resistors are arranged to make specific pins turn out to be high when not being defined by the LoPy, this avoids communication difficulties when the LoPy is waiting for inbound serial data. When the transmission between the LoPy (master) and the MPU9250 (slave) initializes the start order is produced. This is accomplished by lowering SDA. Later the SCL commences to produce clock pulses and the initial byte is sent out by the master. This first byte holds the 7-bit address of the slave and a read/write bite. If the read/write bit is 0 the master writes to the slave. If the read/write bit is 1 the master reads from the slave. The bits can only be sent when the serial clock is high. Each byte sent should be followed by an acknowledge bit. When the master is done writing/reading then the stop condition is created. The readings of each sensor are later averaged and linearized to better detect and reduce the presence of outlier readings [38]. Code interfacing with the IMU including the AEOLUS estimation pipeline is available online repository[1].

4.4 Additional Underwater Considerations

We next take into consideration the possible influence of the underwater environment (e.g. high currents, greater depths, magnetic sources) on the position estimation performance. Prior works already showcase the feasibility of AEOLUS hardware system during recreational SCUBA diving [71], being representative of the needs of real-world underwater data science applications. More recently, the same apparatus was further expanded with multiple units and demonstrated in underwater edge computing [14]. These studies addressed the usage of accelerometer, CPU, amd memory and found no influences on sensor readings at greater depths, given that the encasing pressure was the same as of those at the sea surface. In fact, a temperature drop of 4 degrees Celsius has been observed during the 20-minute SCUBA diving duration and depth at 8 m, showing an ideal passive cooling. Same studies also found a negative correlation between high water motion using water pumps and communication, whereas, in a high turbulent environment it is challenging to use the accelerometer readings and hence, perform the integrations. Still, we foresee the usage of AEOLUS at greater depths during SCUBA or Autonomous Unmanned Vehicles (AUV) surveys, as swells decay exponentially with greater depth [11]. Moreover, ship wreckage sites

[1] https://bitbucket.org/wave_labs/aeolus/src/master/

may also influence the readings of magnetometer [5], and in such conditions, the pipeline should stop integrating position.

5 EXPERIMENTAL PROCEDURE

We next conduct rigorous experiments to assess the position estimation performance of AEOLUS inertial system under different contexts (ground, underwater). The system was moved following predetermined geometric shapes (line, square, triangle, warp). Such geometric shapes were used to assess the position estimation performance of the system against a ground truth. All conducted experiments stored the raw data onto the AEOLUS SD card, where the position estimation algorithms were conducted in post-processing. We investigate the effects of distance on position estimation at six different distance magnitudes, having 4, 5, 16, 20, 24 and 28 meters for the side lengths of the next geometric shapes: line (with 4, 16 and 28 m total distance), triangle (with 12, 48 and 84 m total distance), square (with distances of 16, 64 and 112 m respectively (Figure 7a), and warps (with total distances of 136, 150 and 448 m respectively, seen in Figure 7b. The rationale for using such distances was made of practical reasons, to fit into constrained public space on the ground (Figure 7a) or to fit into the wooden frame structure in an underwater setting (Figure 8a).

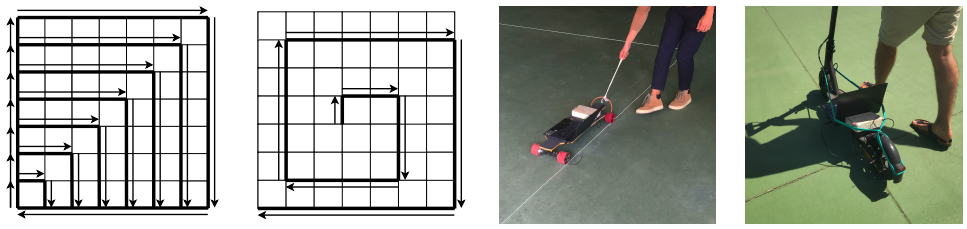
5.1 On the Ground Setup

Geared with simulating the friction-less environment, we initially challenged our estimation pipeline in a controlled environment, using carpet floor inside the building space and by pulling the motion device on predeterment shape trajectory (Figure 7c). We used 4 wheel motion device by placing on top of the AEOLUS hardware being kept inside of a cardboard container. The IMU sensor was placed outside of a cardboard container, facing upwards. The motion device was pulled with the rope, following the trajectory until the rear wheels reached the corner of the ground geometrical figure. The motion device was then rotated by hand and aligned with the next side of the ground geometrical figure. We noticed that the 4 wheels and carpet further contributed to the braking of the system, and we thus moved to a public outdoor space (Figure 7d). The motion device was then based on two wheels and was pushed using the steering wheel, facilitating the rotation at the corner. The AEOLUS hardware was placed onto the top side of the motion device, being further secured with the rubber band. The outdoor setting was a public space with a floor made of 49 concrete square tiles of 4×4 m each, providing the open space of 28×28 m. The choice of this space was as it was the only possible open space providing us with greater dimensions.

Such shapes were formed using a guiding rope laid on the ground, acting as the baselines. We placed the inertial system on a movable platform (an electrical scooter or a skateboard) and pushed/pulled the platform at a constant walking speed along the planned path marked by the rope, pausing at the corners when the direction changed; see Figures 7c and 7d. The system periodically estimates the attitude and position and stores data on an SD card for post-processing.

5.2 Underwater Setup

Underwater experiments were carried out at a local dive spot, where AEOLUS was placed in an underwater spherical housing built for 360 cameras rated until 10 m with, including a portable 1350 mAh power-bank (Figure 8c). Since the hardware was inaccessible inside the housing, we configured the software to support the remote triggering of the system via a wireless hotspot using a mobile phone. The entrance to the water marked the start time of integration. The underwater experiments were conducted to evaluate the estimation quality of the proposed solution underwater by a SCUBA diver (Figure 8d). A 5-meter square net from wood was being held by two persons above water, serving as a guideline to the scuba diver to follow the trajectory during diving (Figure 8a). The total diving trajectory was estimated to be 150 m, comprising the 6 times that the diver completes



(a) (b) (c) (d)

Fig. 7. On the Ground experimental setup: (a) Squares of increasing side length; (b) Warp-based shape with increasing length sides; (c) IMU being placed on a skateboard while pulling; and (d) IMU on a scooter while pushing.

(a) (b) (c) (d)

Fig. 8. Underwater experimental setup. (a) Baseline wooden structure of 5x5 meters at sea surface; (b) Diver performing trajectory beneath the surface baseline structure; (c) Acrylic sphere with IoT components on underwater environment; and (d) IoT components with housing carried by a scuba diver at a depth of 4.1 m.

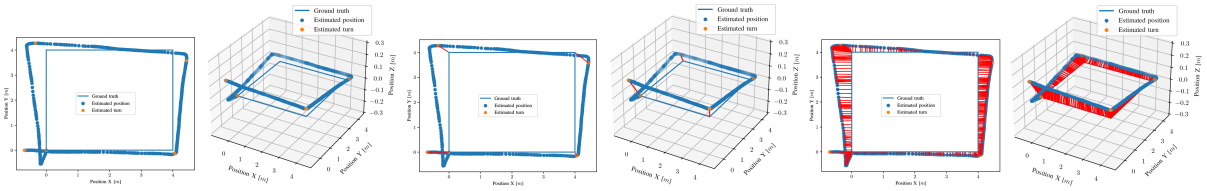
the same square shape beneath the baseline structure. The diver held the spherical housing while swimming along the guideline maintaining a constant speed while slowly increasing the depth (Figure 8d). The total dive time took 20 minutes and was conducted at a depth between 4 to 5 m.

5.3 Ground Truth

To quantify how well a test performed in estimating position, we compare the computed position against geometric markers in physical space. On the ground, guided geometry trajectories (line, triangle, square, warp) were formed using existing physical block marks on the ground with side lengths of 4 *m*. Such was combined with thin flat ropes laid on the ground following the markings of geometrical trajectories (Figure 7d). In an underwater setting, a floating square made of a wooden frame with side lengths of 5 *m* deployed at the sea surface acted as a guided trajectory for the scuba diver, as seen in Figures 8a and 8b. Such structure was with positive buoyancy, and to avoid the surface drifting, it was further attached to the other surface buoys and support lines which are typically used by SCUBA divers for descending and starting the dive, and was additionally supported by swimmers during the data collection as seen in Figure 8a.

5.4 Error Assessments

An error was quantified in two ways. *Turn error* was calculated by assessing how far were the estimated shape corners from the ground truth's vertices (Figure 9a). A turning detection algorithm Ramer-Douglas-Peucke (RDP) [94]) was used to estimate when a vertice was present in the estimated path (Figure 9c). By measuring how far the estimated turn point is from the closest real shape vertice point (using the Euclidean distance between two points in three-dimensional space using the expression below), it was possible to gain insights into how well



(a) (b) (c) (d) (e) (f)

Fig. 9. Error estimation in Cartesian coordinate system in 2D and 3D: (a,b) estimated trajectory over the ground truth; (c,d) turn error detection using distance to corners; and (e,f) displacement error using the distance between sampled points and perpendicular closest points on the ground truth.

a specific shape was estimated by the inertial solution.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \tag{10}$$

Displacement error used distance from every estimated point to the closest neighbor in the ground truth's path (Figure 9e). Each reading point from the estimated trajectory is connected perpendicularly to the point on the ground truth trajectory. To assess the position estimation performance, we sum and average each pair of ground truth/estimation points, computing the traditional Mean Absolute Error (MAE) as:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |x_i - x| \,. \tag{11}$$

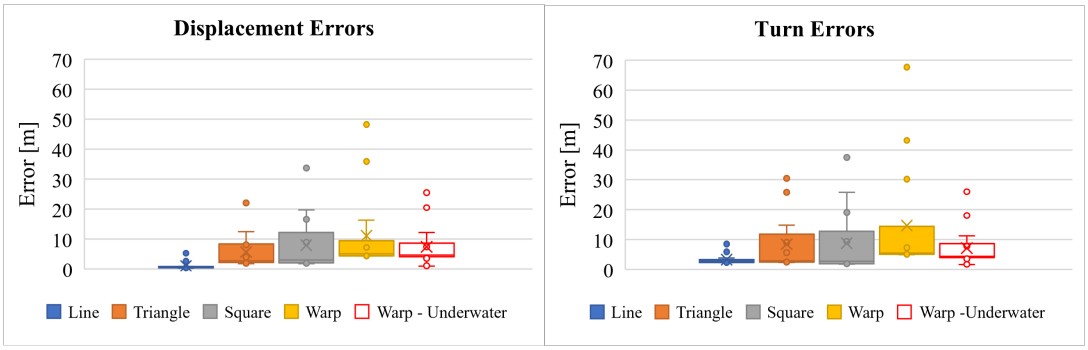
Figure 9 depicts an example output when applying AEOLUS in an experiment following the trajectory of the $4m \times 4m$ square geometrical shape. The figure shows a succession of position points, each calculated by the last known position, the heading angle, and the traveled distance. The position given is relative to the sensor's own body frame, and not the real world's frame.

6 RESULTS

We next present the effects of trajectory shapes, Degree-of-Freedom (DoF) of IMUs, and position estimation algorithms on the errors of displacement and turn. We also include the qualitative analysis of the best-performing algorithms, with displacement and turn error, seen for each geometric shape (lines: Figure 14, triangles: Figure 15, squares: Figure 16 and warps: Figures 17 and 18).

6.1 Effects of Trajectory Shapes on the Errors of Displacement and Turn

By evaluating the magnitude of the error in displacement and turn for different shapes, it is possible to better understand the effect of trajectory on the drift. Figure 10 shows the variance of error of different trajectory shapes on the ground. The errors of displacement (line: 1.02 ± 1.27 m, triangle: 6.26 ± 5.55 m, square: 8.86 ± 9.16 m, warp: 12.85 ± 12.62 m) and turn (line: 3.3 ± 1.67 m, triangle: 9.65 ± 8.29 m, square: 10.02 ± 1.48 m, warp: 17.47 ± 18.42 m) for the line trajectory are the lowest with the least variance compared to other geometric trajectories. When comparing the displacement errors between the experiments performed on the ground and underwater for the warp trajectory, we observe that the average errors of displacement underwater (7.44 ± 6.30 m) are similar to that of trajectory paths with lower turning on the ground (i.e., triangle, square), and are much lower compared to the similar trajectory on the ground (warp). Analogous to prior studies in PDR with walking, this can be further explained by the difference in medium. Specifically, in the underwater environment, the dominant motion results from "kicks" by the scuba diver [25] and from underwater currents which typically have a regular and periodic



(a) (b)

Fig. 10. Effects of trajectory shapes on errors of displacement and turn (m).

structure, which makes it easier to isolate them. In contrast, on the ground, the motion intensity is higher and more frequent, and also the surface of the ground induces variations in the signals [33], unlike in the underwater environment. On the ground, there are also more variations along the vertical axis (due to constant motion and uneven surface). The magnitude of errors depends on the intensity of motion and thus the accumulated error is higher during periods of intense motion, further explaining the higher errors in the ground setting. A good indicator of these errors is the integration drifts seen along the Z-axis of all complex shapes in the ground setting (triangle in Figure 15, square in Figure 16 and warp in Figures 17).

ANOVA test considering different shapes as experimental conditions show statistically significant differences in displacement errors ($F(3,137)=13.301$, $p<0.001$, $\eta^2=0.119$) and turn errors ($F(3,137)=8.358$, $p<0.001$, $\eta^2=0.079$). Posthoc tests (using the Holm correction) indicate the complexity of trajectory shapes produces higher significant errors for all pairs except the triangle-square pair. This can be explained due to the geometric similarity between triangle and square in terms of the perimeter and the number of turns (triangle: 3 turns, square: 4 turns).

6.2 Effects of Degree-of-Freedom (DoF) on the Errors of Displacement and Turn

Figure 11 shows the variances in displacement errors and turn errors when the samplings were performed using different sensor combinations, and thus also different numbers of degrees-of-freedom: 6-DoF level (acc+mag) and 9-DoF level (acc+gyr+mag). When the experiments were performed on the ground, we observe slightly higher magnitudes of displacement and turn errors in the 9-DoF level compared to the 6-DoF level. In contrast, the average errors for the warp trajectory in the underwater environment at the 9-DoF level (displacement error: 6.28 ± 3.19 m, turn error: 6.01 ± 2.91) is lower in both magnitude and variance compared to those at the 6-DoF level (displacement error: 8.31 ± 9.28 m, turn error: 8.01 ± 8.98). The results from the ANOVA test show these differences are not statistically significant, suggesting that either the accelerometer or gyroscope can be removed in the interest of saving power consumption for the sensing units as using both induces higher electricity consumption compared to using the magnetometer and only one of the other sensors. This finding is in line with [24]. The results also indirectly indicate that the benefits of a magnetometer depend on the setting. On the ground, the turns are faster and there are more sources of magnetic disturbances whereas underwater settings typically have fewer sources of magnetic noise – or at least the sources are generally more constant in the underwater setting. For example, while wrecks, anchors, moorings, drone engines, or moving ships can cause magnetic fields, these are typically fairly consistent and highly localized, enabling common algorithms to compensate against them.

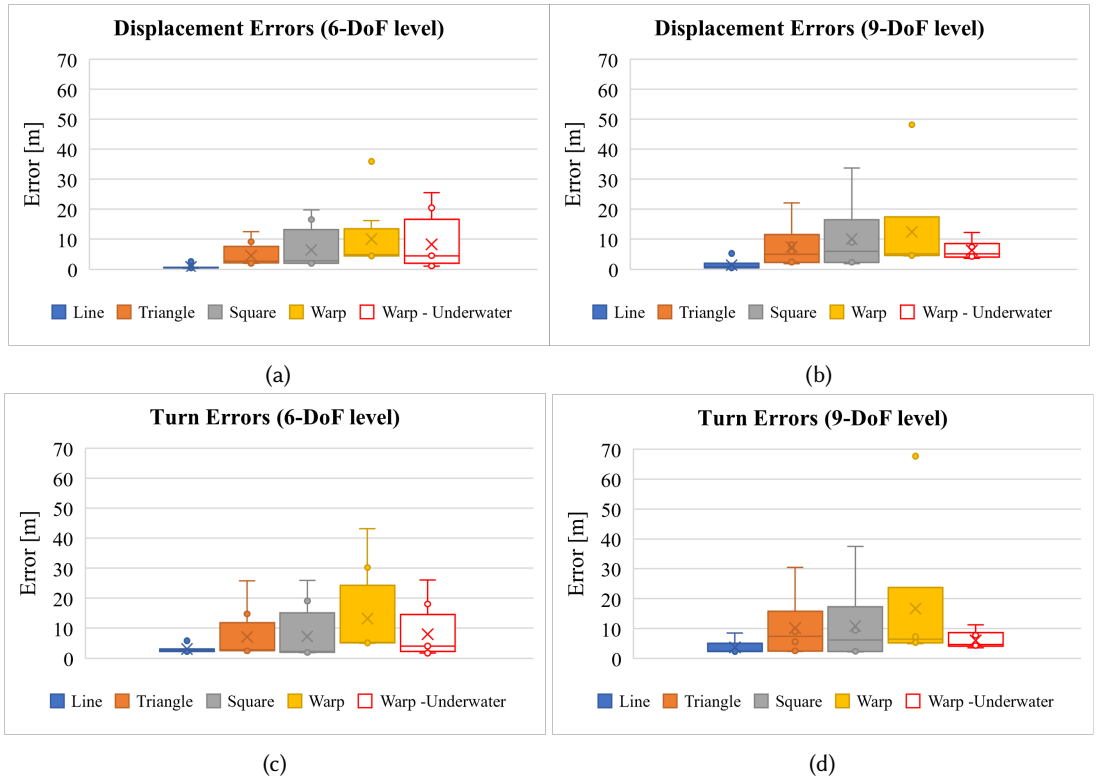
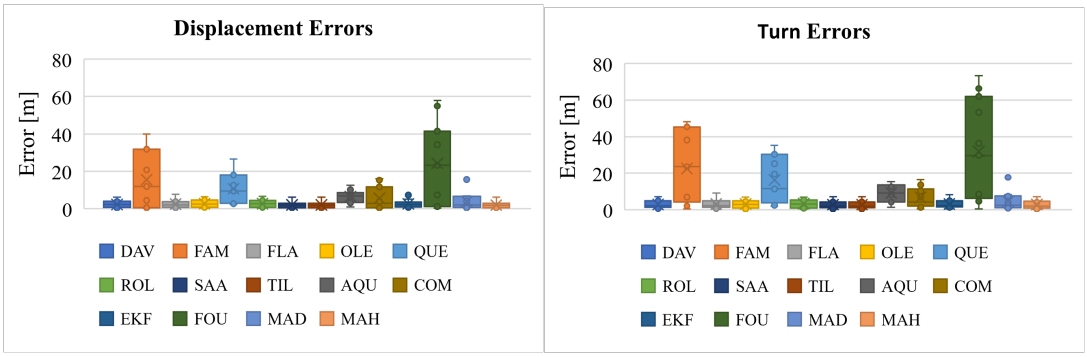


Fig. 11. Effects of DoF levels on the errors of displacement and turn (m).

This contrasts with ground scenarios where magnetic disturbances typically have abnormal and dynamically changing patterns [10].

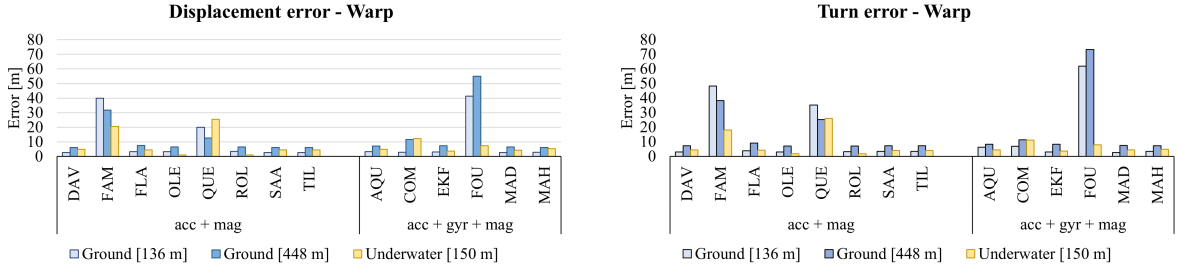
6.3 Effects of Position Estimation Algorithms on the Errors of Displacement and Turn

Figure 12 shows the error variance of different position algorithms. High magnitudes of errors in displacement and turn are observed with FAMC, QUEST, and Fourati algorithms compared to other position estimation algorithms. These significant differences in displacement and turning are confirmed from ANOVA tests, with displacement errors ($F(13,137)=9.697$, $p<0.001$, $\eta^2=0.416$) and turn errors ($F(3,137)=10.259$, $p<0.001$, $\eta^2=0.451$), and posthoc tests using the Holm correction. While FAMC and QUEST use an accelerometer and magnetometer (6-DoF) for location estimation, Fourati algorithm requires an accelerometer, gyroscope, and magnetometer (9-DoF), suggesting that the high estimation error would be associated with the input requirements of the algorithms, e.g., QUEST algorithm has the capability of handling very simple dynamic models but it requires the attitude rate time history as an input [67]. Compared with other algorithms, QUEST and FAMC may be over-simplistic and fast for the computation of rotation matrices while Fouriati needs a high-bandwidth rate for gyro when there is constant propulsion. Indeed, the errors of displacement estimation performed by QUEST increase multiple folds with trajectory complexity, i.e., the average displacement error of the triangle is approximately 3 times higher compared to the line (5.86 m).



(a) (b)

Fig. 12. Effects of attitude estimation algorithms on errors of displacement and turn (m). DAV: Davenport, FAM: Famc, FLA: Flae, OLE: Oleq, QUE: Quest, ROL: Roleq, SAA: Saam, TIL: Tilt, AQU: Aqua, COM: Complementary, EKF: Ekf, FOU: Fourati, MAD: Madgwick, MAH: Mahony.



(a) (b)

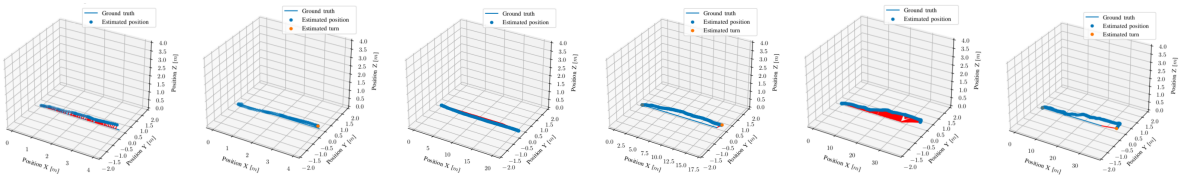
Fig. 13. Comparison of average errors of trajectory displacement and turning (m) on the ground and underwater environments. DAV: Davenport, FAM: Famc, FLA: Flae, OLE: Oleq, QUE: Quest, ROL: Roleq, SAA: Saam, TIL: Tilt, AQU: Aqua, COM: Complementary, EKF: Ekf, FOU: Fouriati, MAD: Madgwick, MAH: Mahony. acc: accelerometer, gyr: gyroscope, mag: magnetometer.

Figure 13 shows the comparison of average displacement and turn errors estimated by different algorithms for the warp trajectory being the most complex shapes in both settings. In underwater environments, attitude estimation is best computed using OLEQ (displacement error=1.04m, turn error=1.71m) or ROLEQ whereas FAMC, QUEST and Fourati algorithms produce high errors. Interestingly, attitude estimation computed using some algorithms (e.g., FAMC, Fourati) result in high errors on the ground but lower errors when the experiments were performed in the underwater environment. Such may further enhance the time the dead-reckoning can be used in one run without resetting the values to a known baseline.

6.4 Qualitative Assessments

6.4.1 Estimating Line (on the ground). The line shape consisted of moving the inertial system in a straight line for a determined distance. Three line distances were tested: 4, 16, and 28 meters (Figure 14). For the 4-meter line experiment, the OLEQ algorithm which had the lowest displacement error with an average of 0.13 meters

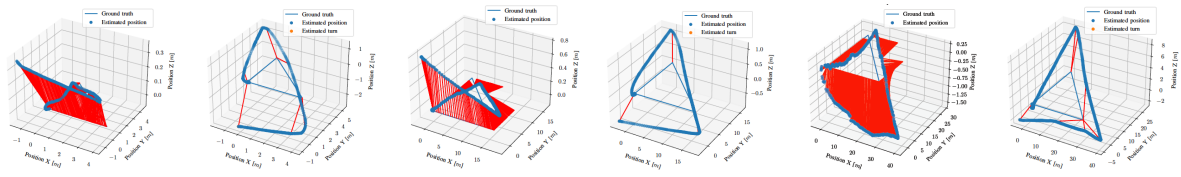
(3.24% of error margin), and ROLEQ with an average of 0.24 meters of turn error (6.06% of error margin). For the 16-meter line experiment, the FAMC algorithm had the lowest displacement error with an average of 0.43 meters (2.69% of error margin), and Mahony with an average of 2.11 meters of turn error (13.21% of error margin). For the 28-meter line experiment, the Complementary algorithm had the lowest displacement error with an average of 0.52 meters (1.85% of error margin), and SAAM with an average of 4.23 meters of turn error (15.09% of error margin).



(a) OLEQ (b) ROLEQ (c) FAMC (d) Mahony (e) Complementary (f) SAAM

Fig. 14. Line Estimation (on the ground). Position estimation for different line sizes using the algorithms with the lowest displacement error (a) 4-meter, (c) 16-meter, (e) 28-meter; and with the lowest turn error (b) 4-meter, (d) 16-meter, (f) 28-meter.

6.4.2 Estimating Triangle (on the ground). The triangle shape consisted of moving the inertial system in a triangle pattern for a determined distance. We tested three (3) distances: 4, 16, and 28 meters (Figure 15). For the 4-meter triangle experiment, the Mahony algorithm had the lowest displacement error with an average of 0.90 meters (7.50% of error margin), and EKF with an average of 0.49 meters of turn error (4.09% of error margin). For the 16-meter triangle experiment, the Mahony algorithm had the lowest displacement error with an average of 1.79 meters (3.74% of error margin), and EKF with an average of 1.59 meters of turn error (3.32% of error margin). For the 28-meter triangle experiment, the Mahony algorithm had the lowest displacement error with an average of 2.93 meters (3.49% of error margin), and EKF with an average of 5.04 meters of turn error (5.99% of error margin).

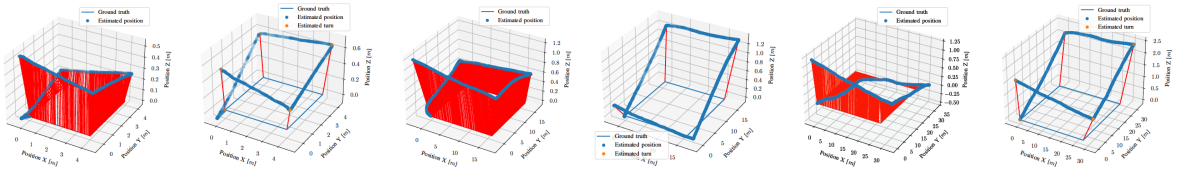


(a) Mahony (b) EKF (c) Mahony (d) EKF (e) Mahony (f) EKF

Fig. 15. Triangle Estimation (on the ground). Position estimation for different triangle sides using the algorithms with the lowest displacement error (a) 4-meter, (c) 16-meter, (e) 28-meter; and with the lowest turn error (b) 4-meter, (d) 16-meter, (f) 28-meter.

6.4.3 Estimating Square (on the ground). The square shape consisted of moving the inertial system in a square pattern for a determined distance. We tested three (3) line distances: 4, 16, and 28 meters (Figure 16). For the 4-meter square experiment, the Mahony algorithm had the lowest displacement error with an average of 0.53 meters (3.34% of error margin), and FLAE with an average of 0.38 meters of turn error (2.37% of error margin). For the 16-meter square experiment, the Madgwick algorithm had the lowest displacement error with an average of 1.93 meters (3.02% of error margin), and FLAE with an average of 1.82 meters of turn error (2.84% of error

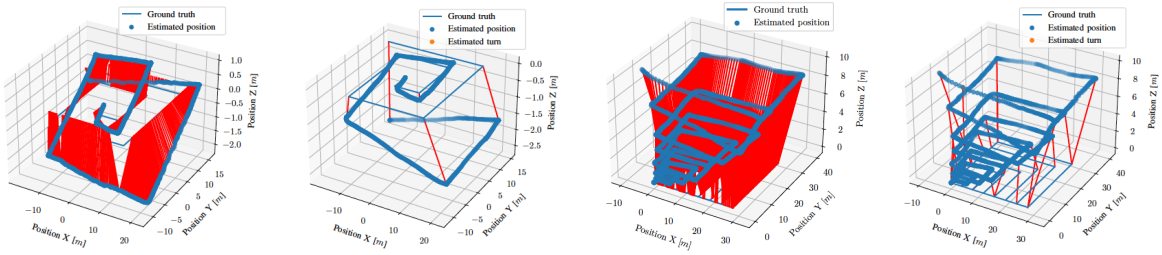
margin). For the 28-meter square experiment, the Mahony algorithm had the lowest displacement error with an average of 2.97 meters (2.65% of error margin), and FLAE with an average of 3.23 meters of turn error (2.89% of error margin).



(a) Mahony (b) FLAE (c) Madgwick (d) FLAE (e) Mahony (f) FLAE

Fig. 16. Square Estimation (on the ground). Position estimation for different square sides using the algorithms with the lowest displacement error (a) 4-meter, (c) 16-meter, (e) 28-meter; and with the lowest turn error (b) 4-meter, (d) 16-meter, (f) 28-meter.

6.4.4 Estimating Warps (on the ground). Warp shapes included a spiral and growing squares (Figure 17). For the spiral experiment, Davenport's algorithm had the lowest displacement error with an average of 2.64 meters (1.78% of error margin), and Madgwick with an average of 2.46 meters of turn error (1.66% of error margin). For the maximum 28-meter line experiment of growing squares, the Tilt algorithm which had the lowest displacement error with an average of 6.11 meters (4.88% of error margin), and SAAM with an average of 7.23 meters of turn error (4.77% of error margin).

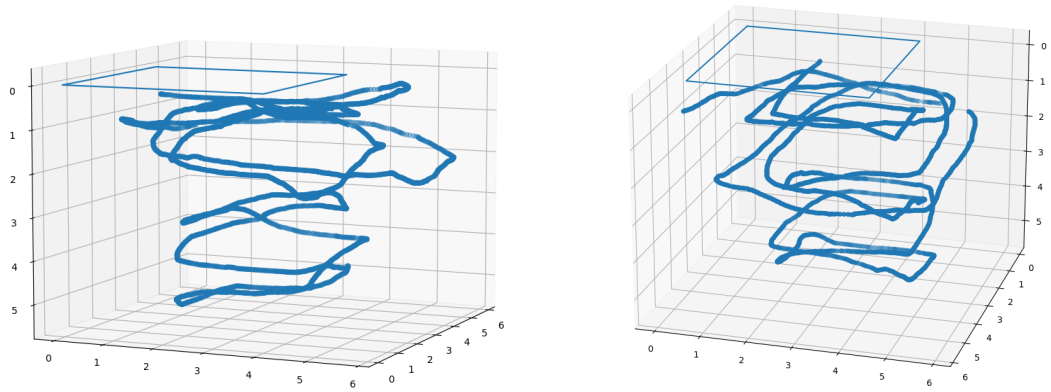


(a) Davenport (b) Madgwick (c) TILT (d) SAAM

Fig. 17. Warp Estimation (on the ground). Position estimation using the algorithms with (a) the lowest displacement error and (b) the lowest turn error for warp tests; (c) the lowest displacement error; and (d) the lowest turn error for squares tests.

6.4.5 Estimating Warps (underwater). Obtained warps[2] showcase the scuba diving performing the trajectory beneath the wooden base structure (Figure 18). Trajectory indicates the clear 6 passages of the square patterns with the depth ranging up to 5 m. OLEQ and ROLEQ had similar results with both of them achieving a displacement error of close to $1m$ or 0.001% and a turn error of nearly $1.7m$ or 0.01% with the use of only two sensors which indicates the robust performance of the estimator, while the worst performing was QUEST with $4.51m$ and $4.07m$ displacement and turn error respectively which translates to a 0.03% error for both of them.

[2]The shape trajectory by the diver is square, while the estimated trajectory resembles the warp shape.



(a) OLEQ (b) ROLEQ

Fig. 18. Warp Estimation (underwater). Position estimation using the algorithms: (a) the lowest displacement error and (b) the lowest turn error. The thin rectangle indicates the base wooden surface structure, while the thick line indicates the trajectory taken by the scuba diver. The vertical axis indicates the estimated depth.

When comparing all of the obtained results in an underwater setting for a 150 m baseline distance, a 5% error for displacement and a 5% error for the turn were obtained. On the ground with a 136 m baseline distance, an error of 8% for displacement and an error of 12% for the turn was obtained. Similarly, on the ground with 448 m, the error was smaller (having 3% of an error for displacement estimation and an error of 4% for the turn). On the ground, from every test made, Mahony ranks best at displacement with an average of 1.94 meters of absolute displacement error and 3.72% relative error. Closely followed by SAAM and TILT with 2 meters of displacement error and 4.12% of relative displacement error. SAAM and TILT ranked best at turn error both with an average of 2.79 meters of absolute turn error and 6.93% of relative turn error. Closely followed by Mahony with 6.97% meters of relative turn error. On the other hand, Fourati had the worst general performance among all algorithms, with a 24.19 (36.14% relative displacement error) meter error average displacement and 31.87 (48.16% relative turn error) meter turn error average. FAMC, QUEST, and AQUA also had poor overall results generally performing under average to the other sensor fusion algorithms.

7 DISCUSSION

7.1 Implications

The results demonstrated that IMU-based dead reckoning can achieve sufficient performance in most underwater applications but that the choice of sensors, the choice of algorithms, and target patterns all affect which methods should be chosen. Most techniques have been designed for ground scenarios and while the overall error tends to be higher on the ground – due to more frequent and intense motion than in underwater settings – the variation in performance across techniques, sensors, or shapes is smaller than in the underwater setting. Another key difference comes from the type of error as on the ground the displacement error tends to dominate the overall error, whereas in the underwater case displacement error tends to be smaller than the turn error. The general trend of errors is the same for both ground and underwater settings with the absolute error increasing with the length of the pattern whereas the relative error decreases. In terms of algorithms, those that rely on faster turn detection tended to result in worse performance in the underwater setting, highlighting the effect of turn estimation errors. Specifically, techniques that are widely used in pedestrian settings, such as complementary

filtering, EKF, and Madwick, had worse performance than OLEQ, ROLEQ, SAAM, TILT, FLAE, and Davenport techniques.

Besides this, Figure 11, Figure 12 and Figure 13, it is possible to draw a correlation between what sensors are used by each algorithm and how they performed at estimating position. Results depicted there show that the worst performing algorithms on the surface are generally the ones that do not use every sensor available, when the algorithm allowed for the optional use of a sensor it was used, for instance, FAMC, QUEST and AQUA do not fuse the gyro's readings to estimate orientation. Fourati's poor performance might be explained by the fact that it depicts a rigid body's attitude in space with respect to the navigation frame (X_N,Y_N,Z_N), where the navigation frame follows the convention North East Down, opposite of the convention used by the other algorithms, East North Up. In the underwater experience, the use of a third sensor did not seem to improve the results since the values from the algorithms that only use two sensors when compared with the ones that use all three sensors are similar with the best-performing ones being the ones limited to two sensors.

7.2 Comparison with Literature

Prior studies have mostly relied on acoustic sensors [66] and computer vision [34] while in our article we contribute with an overview of the principles and the review of existing dead reckoning algorithms, analyzing their performance with data obtained from inertial sensors, how complex and different number of degrees of freedom can influence such results, and determining guidelines for what should be used in different situations. Acoustic sensors are sensitive to underwater noise, which is becoming a significant concern [9], whereas computer vision requires support for computations that are difficult to provide [14]. Ladetto et al. [43] 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, and determining a user's step signature. They verified an average position error of 5% of distance traveled which is significantly smaller than our results of 12% average traveled error. Nonetheless, Ladetto et al. [43] had GPS receivers and other sensors to help improve absolute position, which is not our case. When compared to other scholars, Stirling et al. [80] illustrates an experiment exploiting a shoe-mounted sensor prototype that calculates stride length with accelerometers and magnetometers. Their system measures angular acceleration by manipulating pairs of accelerometers as an alternative to gyroscopes. The foot pod stride length measurement was generally less than 10% different from the distances given by their control point surveyed path. With a total distance estimated by the foot pod of 978 meters, compared to 862 meters for the survey which goes into conformity with our own results of 12% displacement error. Other experiments conducted by Ludwig et al. [45] using quadcopter data and results show that Mahony provides better orientation estimation than both Madgwick and EKF when using optimum parameters. The same authors tested on quadcopters the EKF, Madgwick, and Mahony filters [46]. Results showed that Mahony delivered a more precise orientation estimation and faster execution time than Madgwick and EKF which also goes in compliance with our results.

7.3 Generalization

The patterns that we considered were chosen as common patterns that are considered in underwater monitoring scenarios and practically any application that relies on spatial sampling designs benefits from our approach (see, e.g., [35] for thorough coverage of possible scenarios). One limitation of our study, however, was the reliance on scuba divers as the source of motion, and in practice also underwater vehicles should be supported. In these cases, the general effects of water as the medium are the same but the propulsion patterns can have an effect on the motion characteristics, and thus on the performance of the algorithms. Underwater vehicles also incorporate ferrous components which generate magnetic fields and thus require integrating magnetometer

compensation techniques before the inertial navigation techniques can be used. In practice, these propulsion patterns of underwater vehicles apply a consistent force on the vehicle and thus the variation in the motion patterns is even smaller than in the diving scenario – and clearly smaller than in ground-based navigation. Thus, we would expect the performance to be largely similar or even better than in the scuba diving case. While further research is needed to validate this more accurately, our work serves as an important starting point in bringing pervasive sensing and ubicomp solutions into underwater domains.

7.4 Challenges

Naturally we also encountered some challenges while carrying out the measurement campaigns and we next briefly highlight some of them with the aim of pointing out further research directions that would need addressing. First, low-cost sensors for micro-controllers have varying reliability and consistency and thus it is essential to verify the sensor performance separately prior to carrying out measurement campaigns. The casing for underwater settings is also challenging as there are no plug-and-play solutions but custom designs that can potentially affect the performance of the sensors are needed. The sensors also need calibration before data collection. On the ground, the calibration can be done opportunistically by identifying periods where the sensors are sufficiently stable [31] but generalizing these techniques to underwater settings is not straightforward as currents and other factors can result in difficulties in identifying suitable rest periods for calibration. Finally, the choice of sensors also poses limitations on the algorithms in terms of sampling rate and data availability. For example, pressure sensors could further improve accuracy by providing more accurate depth estimates, whereas higher sampling rates could potentially reduce sensor drift. Note that even low sampling rates are sufficient for scuba diving scenarios as the frequency of motion patterns is typically low (below one hertz) to preserve oxygen. Thus faster sampling rates are more important for other scenarios, such as navigating underwater vehicles.

ACKNOWLEDGMENTS

The research was supported by FCT grant INTERWHALE (PTDC/CCI-COM/0450/2020), FCT strategic project UIDB/04292/2020 awarded to MARE, and through project LA/P/0069/2020 granted to the Associate Laboratory ARNET. It has been also financed by the EU Horizon Europe project CLIMAREST: Coastal Climate Resilience and Marine Restoration Tools for the Arctic Atlantic basin (grant agreement: 101093865), the Academy of Finland (grant number: 339614), the European Social Fund via “ICT programme” measure, Estonian Center of Excellence in ICT Research (TK148 EXCITE), and the Nokia Foundation (grant number: 20220138). The authors thank the participants of our studies and the anonymous reviewers for their insightful comments.

REFERENCES

[1] Norhafizan Ahmad, Raja Ariffin Raja Ghazilla, Nazirah M Khairi, and Vijayabaskar Kasi. 2013. Reviews on various inertial measurement unit (IMU) sensor applications. *International Journal of Signal Processing Systems* 1, 2 (2013), 256–262.

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

[3] John Allen and Brian Walsh. 2008. Enhanced oil spill surveillance, detection and monitoring through the applied technology of unmanned air systems. In *International oil spill conference*, Vol. 2008. American Petroleum Institute, 113–120.

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

[5] J Barto Arnold. 1996. Magnetometer survey of La Salle’s ship the Belle. *International Journal of Nautical Archaeology* 25, 3-4 (1996), 243–249.

[6] Khalid Mahmood Awan, Peer Azmat Shah, Khalid Iqbal, Saira Gillani, Waqas Ahmad, and Yunyoung Nam. 2019. Underwater wireless sensor networks: A review of recent issues and challenges. *Wireless Communications and Mobile Computing* 2019 (2019).

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

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

[9] Abir Chahouri, Nadia Elouahmani, and Hanan Ouchene. 2022. Recent progress in marine noise pollution: A thorough review. *Chemosphere* 291 (2022), 132983.

[10] Jaewoo Chung, Matt Donahoe, Chris Schmandt, Ig-Jae Kim, Pedram Razavai, and Micaela Wiseman. 2011. Indoor location sensing using geo-magnetism. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*. 141–154.

[11] Allan J Clarke and Stephen Van Gorder. 2018. The relationship of near-surface flow, stokes drift and the wind stress. *Journal of Geophysical Research: Oceans* 123, 7 (2018), 4680–4692.

[12] Jussi Collin, G Lachapelle, and Jani Käppi. 2002. MEMS-IMU for personal positioning in a vehicle–A gyro-free approach. In *Proceedings of ION GPS*. 24–27.

[13] James L Coyte, David Stirling, Montserrat Ros, Haiping Du, and Andrew Gray. 2013. 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*. IEEE, 1290–1295.

[14] Farooq Dar, Mohan Liyanage, Marko Radeta, Zhigang Yin, Agustin Zuniga, Sokol Kosta, Sasu Tarkoma, Petteri Nurmi, and Huber Flores. 2022. Upscaling Fog Computing in Oceans for Underwater Pervasive Data Science using Low-Cost Micro-Clouds. *ACM Transactions on Internet of Things* (2022).

[15] Paul B Davenport. 1968. *A vector approach to the algebra of rotations with applications*. Vol. 4696. National Aeronautics and Space Administration.

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

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

[18] Naser El-Sheimy and Ahmed Youssef. 2020. Inertial sensors technologies for navigation applications: State of the art and future trends. *Satellite Navigation* 1, 1 (2020), 1–21.

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

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

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

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

[23] Eric Foxlin. 2005. Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Computer graphics and applications* 25, 6 (2005), 38–46.

[24] Pierre Gogendeau, Sylvain Bonhommeau, Hassen Fourati, Denis De Oliveira, Virgil Taillandier, Andrea Goharzadeh, and Serge Bernard. 2022. Dead-reckoning configurations analysis for marine turtle context in a controlled environment. *IEEE Sensors Journal* 22, 12 (2022), 12298–12306.

[25] Benjamin H Groh, Tobias Cibis, Ralph O Schill, and Bjoern M Eskofier. 2015. IMU-based pose determination of scuba divers' bodies and shanks. In *2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*. IEEE, 1–6.

[26] Kenneth Gustavsson. 2015. UAV pose estimation using sensor fusion of inertial, sonar and satellite signals.

[27] David L Hall and James Llinas. 1997. An introduction to multisensor data fusion. *Proc. IEEE* 85, 1 (1997), 6–23.

[28] John Heidemann, Wei Ye, Jack Wills, Affan Syed, and Yuan Li. 2006. Research challenges and applications for underwater sensor networking. In *IEEE Wireless Communications and Networking Conference, 2006. WCNC 2006.*, Vol. 1. IEEE, 228–235.

[29] John N Heine. 2000. Scientific diving techniques. *Marine Technology Society Journal* 34, 4 (2000), 23–37.

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

[31] Samuli Hemminki, Petteri Nurmi, and Sasu Tarkoma. 2014. Gravity and linear acceleration estimation on mobile devices. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. 50–59.

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

[33] Xinyu Hou and Jeroen Bergmann. 2020. Pedestrian dead reckoning with wearable sensors: A systematic review. *IEEE Sensors Journal* 21, 1 (2020), 143–152.

[34] Zengshi Huang, Naijie Gu, Jianlin Hao, and Jie Shen. 2018. 3DLoc: 3D Features for Accurate Indoor Positioning. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4, Article 141 (jan 2018), 26 pages. https://doi.org/10.1145/3161409

[35] Mohammad Jahanbakht, Wei Xiang, Lajos Hanzo, and Mostafa Rahimi Azghadi. 2021. Internet of underwater things and big marine data analytics—a comprehensive survey. *IEEE Communications Surveys & Tutorials* 23, 2 (2021), 904–956.
[36] Hassan Jameian, Behrouz Safarinejadian, and Mokhtar Shasadeghi. 2019. A robust and fast self-alignment method for strapdown inertial navigation system in rough sea conditions. *Ocean Engineering* 187 (2019), 106196.
[37] Wei-Wen Kao. 1991. Integration of GPS and dead-reckoning navigation systems. In *Vehicle Navigation and Information Systems Conference, 1991*, Vol. 2. IEEE, 635–643.
[38] Matenat Khamphroo, Natavut Kwankeo, Kamol Kaemarungsi, and Kazuhiko Fukawa. 2017. MicroPython-based educational mobile robot for computer coding learning. In *2017 8th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)*. IEEE, 1–6.
[39] Junhee Kim, Kiyoung Kim, and Hoon Sohn. 2014. Autonomous dynamic displacement estimation from data fusion of acceleration and intermittent displacement measurements. *Mechanical Systems and Signal Processing* 42, 1 (2014), 194–205. https://doi.org/10.1016/j.ymssp.2013.09.014
[40] Jeong Won Kim, Han Jin Jang, Dong-Hwan Hwang, and Chansik Park. 2004. A step, stride and heading determination for the pedestrian navigation system. *Journal of Global Positioning Systems* 3, 1-2 (2004), 273–279.
[41] Manon Kok and Thomas B Schön. 2016. Magnetometer calibration using inertial sensors. *IEEE Sensors Journal* 16, 14 (2016), 5679–5689.
[42] Jack B Kuipers. 2000. Quaternions and rotation sequences. *Geometry, Integrability and Quantization* 1 (2000), 127–143.
[43] Quentin Ladetto and Bertrand Merminod. 2002. In step with INS navigation for the blind, tracking emergency crews. *Gps World* 13, ARTICLE (2002), 30–38.
[44] Yuzhou Li, Shengnan Wang, Cheng Jin, Yu Zhang, and Tao Jiang. 2019. A survey of underwater magnetic induction communications: Fundamental issues, recent advances, and challenges. *IEEE Communications Surveys & Tutorials* 21, 3 (2019), 2466–2487.
[45] Simone A Ludwig and Kaleb D Burnham. 2018. Comparison of Euler Estimate using Extended Kalman Filter, Madgwick and Mahony on Quadcopter Flight Data. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 1236–1241.
[46] Simone A Ludwig, Kaleb D Burnham, Antonio R Jiménez, and Pierre A Touma. 2018. 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*, Vol. 10598. International Society for Optics and Photonics, 105982L.
[47] Chunbo Luo, Sally I McClean, Gerard Parr, Luke Teacy, and Renzo De Nardi. 2013. UAV position estimation and collision avoidance using the extended Kalman filter. *IEEE Transactions on Vehicular Technology* 62, 6 (2013), 2749–2762.
[48] Sebastian Madgwick. 2010. An efficient orientation filter for inertial and inertial/magnetic sensor arrays. *Report x-io and University of Bristol (UK)* 25 (2010), 113–118.
[49] Sebastian OH Madgwick, Samuel Wilson, Ruth Turk, Jane Burridge, Christos Kapatos, and Ravi Vaidyanathan. 2020. An extended complementary filter for full-body MARG orientation estimation. *IEEE/ASME Transactions on Mechatronics* 25, 4 (2020), 2054–2064.
[50] Sebastian O. H. Madgwick, Andrew J. L. Harrison, and Ravi Vaidyanathan. 2011. Estimation of IMU and MARG orientation using a gradient descent algorithm. In *2011 IEEE International Conference on Rehabilitation Robotics*. 1–7. https://doi.org/10.1109/ICORR.2011.5975346
[51] Robert Mahony, Tarek Hamel, and Jean-Michel Pflimlin. 2008. Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on automatic control* 53, 5 (2008), 1203–1218.
[52] Sumit Majumder, Tapas Mondal, and M Jamal Deen. 2017. Wearable sensors for remote health monitoring. *Sensors* 17, 1 (2017), 130.
[53] AH Mohamed and KP Schwarz. 1999. Adaptive Kalman filtering for INS/GPS. *Journal of geodesy* 73, 4 (1999), 193–203.
[54] Nader Mohamed, Jameela Al-Jaroodi, Imad Jawhar, Hassan Noura, and Sara Mahmoud. 2017. UAVFog: A UAV-based fog computing for Internet of Things. In *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. IEEE, 1–8.
[55] Robert Mooney, Gavin Corley, Alan Godfrey, Leo R Quinlan, and Gearóid ÓLaighin. 2015. Inertial sensor technology for elite swimming performance analysis: A systematic review. *Sensors* 16, 1 (2015), 18.
[56] Deirdre Morris, Benjamin Schazmann, Yangzhe Wu, Shirley Coyle, Sarah Brady, Jer Hayes, Conor Slater, Cormac Fay, King Tong Lau, Gordon Wallace, and Dermot Diamond. 2008. Wearable sensors for monitoring sports performance and training. In *2008 5th International Summer School and Symposium on Medical Devices and Biosensors*. IEEE, 121–124.
[57] Bobak Mortazavi, Suneil Nyamathi, Sunghoon Ivan Lee, Thomas Wilkerson, Hassan Ghasemzadeh, and Majid Sarrafzadeh. 2013. Near-realistic mobile exergames with wireless wearable sensors. *IEEE Journal of Biomedical and Health Informatics* 18, 2 (2013), 449–456.
[58] Hannah M Murphy and Gregory P Jenkins. 2010. Observational methods used in marine spatial monitoring of fishes and associated habitats: a review. *Marine and Freshwater Research* 61, 2 (2010), 236–252.
[59] S Nassar, A Noureldin, and N El-Sheimy. 2004. Improving positioning accuracy during kinematic DGPS outage periods using SINS/DGPS integration and SINS data de-noising. *Survey Review* 37, 292 (2004), 426–438.
[60] A Noordin, MAM Basri, and Z Mohamed. 2018. Sensor fusion algorithm by complementary filter for attitude estimation of quadrotor with low-cost IMU. *Telkomnika* 16, 2 (2018), 868–875.

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

[62] Online. [n. d.]. AHRS: Attitude and Heading Reference Systems. https://docs.pycom.io/datasheets/development/lopy/. Accessed: 2021-04-29.

[63] Online. [n. d.]. AK8963 3-axis Electronic Compass. https://download.mikroe.com/documents/datasheets/ak8963c-datasheet.pdf. Accessed: 2021-04-29.

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

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

[66] Swadhin Pradhan, Ghufran Baig, Wenguang Mao, Lili Qiu, Guohai Chen, and Bo Yang. 2018. Smartphone-Based Acoustic Indoor Space Mapping. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 2, Article 75 (jul 2018), 26 pages. https://doi.org/10.1145/3214278

[67] M Psiaki. 1999. Extended quest attitude determination filtering. In *NASA CONFERENCE PUBLICATION*. NASA, 1–16.

[68] Honghui Qi and John B Moore. 2002. Direct Kalman filtering approach for GPS/INS integration. *IEEE Trans. Aerospace Electron. Systems* 38, 2 (2002), 687–693.

[69] Marko Radeta, Miguel Ribeiro, Dinarte Vasconcelos, Jorge Lopes, Michael Sousa, João Monteiro, and Nuno Jardim Nunes. 2019. Seamote-interactive remotely operated apparatus for aquatic expeditions. In *IFIP Conference on Human-Computer Interaction*. Springer, 237–248.

[70] Marko Radeta, Miguel Ribeiro, Dinarte Vasconcelos, Hildegardo Noronha, and Nuno Jardim Nunes. 2020. LoRaquatica: Studying range and location estimation using LoRa and IoT in aquatic sensing. In *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, 1–6.

[71] Marko Radeta, Agustin Zuniga, Naser Hossein Motlagh, Mohan Liyanage, Ruben Freitas, Moustafa Youssef, Sasu Tarkoma, Huber Flores, and Petteri Nurmi. 2022. Deep learning and the oceans. *Computer* 55, 5 (2022), 39–50.

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

[73] Dinesh Kumar Sah, Tu N Nguyen, Manjusha Kandulna, Korhan Cengiz, and Tarachand Amgoth. 2022. 3D Localization and Error Minimization in Underwater Sensor Networks. *ACM Transactions on Sensor Networks (TOSN)* (2022).

[74] Sarvenaz Salehi, Navid Mostofi, and Gabriele Bleser. 2012. A practical in-field magnetometer calibration method for IMUs. In *Proceedings of the IROS Workshop on Cognitive Assistive Systems: Closing the Action-Perception Loop*. 39–44.

[75] Paul G. Savage. 1998. Strapdown Inertial Navigation Integration Algorithm Design Part 2: Velocity and Position Algorithms. *Journal of Guidance, Control, and Dynamics* 21, 2 (1998), 208–221. https://doi.org/10.2514/2.4242 arXiv:https://doi.org/10.2514/2.4242

[76] K Rahul Sharma, Daniel Honc, and František Dušek. 2014. 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*. IEEE, 263–266.

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

[78] Joan Sola. 2017. Quaternion kinematics for the error-state Kalman filter. *arXiv preprint arXiv:1711.02508* (2017).

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

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

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

[82] Sebastian Trimpe and Raffaello D'Andrea. 2012. The Balancing Cube: A dynamic sculpture as test bed for distributed estimation and control. *IEEE Control Systems Magazine* 32, 6 (2012), 48–75.

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

[84] Bjørnar Vik and Thor I Fossen. 2001. A nonlinear observer for GPS and INS integration. In *Proceedings of the 40th IEEE conference on decision and control (cat. no. 01CH37228)*, Vol. 3. IEEE, 2956–2961.

[85] Harvey Weinberg. 2002. Using the ADXL202 in pedometer and personal navigation applications. *Analog Devices AN-602 application note* 2, 2 (2002), 1–6.

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

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

[88] RVC Wong, KP Schwarz, and ME Cannon. 1988. High-Accuracy Kinematic Positioning by GPS-INS. *Navigation* 35, 2 (1988), 275–287.
[89] Jin Wu, Zebo Zhou, Hassen Fourati, and Yuhua Cheng. 2018. A super fast attitude determination algorithm for consumer-level accelerometer and magnetometer. *IEEE Transactions on Consumer Electronics* 64, 3 (2018), 375–381.
[90] Yuanxin Wu and Wei Shi. 2015. On calibration of three-axis magnetometer. *IEEE Sensors Journal* 15, 11 (2015), 6424–6431.
[91] Guobao Xu, Weiming Shen, and Xianbin Wang. 2014. Applications of wireless sensor networks in marine environment monitoring: A survey. *Sensors* 14, 9 (2014), 16932–16954.
[92] Yanli Yang, Yanfei Zhao, and Dali Kang. 2016. Integration on acceleration signals by adjusting with envelopes. *Journal of Measurements in Engineering* 4, 2 (2016), 117–121.
[93] Song Zhang, Shili Zhao, Dong An, Jincun Liu, He Wang, Yu Feng, Daoliang Li, and Ran Zhao. 2022. Visual SLAM for underwater vehicles: A survey. *Computer Science Review* 46 (2022), 100510.
[94] Xuehua Zhang, Hong zhe Liu, Jiazheng Yuan, Xiankai Huang, and Hanyu Xuan. 2017. A Non-parametric RDP Algorithm Based on Leap Motion. *DEStech Transactions on Computer Science and Engineering* (2017).
[95] Zebo Zhou, Jin Wu, Jinling Wang, and Hassen Fourati. 2018. Optimal, recursive and sub-optimal linear solutions to attitude determination from vector observations for GNSS/accelerometer/magnetometer orientation measurement. *Remote sensing* 10, 3 (2018), 377.