2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

A Nonlinear Estimator for Dead Reckoning of Aquatic Surface Vehicles Using an IMU and a Doppler Velocity Log

Jessica Paterson, Bruno V. Adorno, Barry Lennox, Keir Groves

***Abstract*— Aquatic robots require an accurate and reliable localization system to navigate autonomously and perform practical missions. Kalman filters (KFs) and their variants are typically used in aquatic robots to combine sensor data. The two critical drawbacks of KFs are the requirement for skilled tuning of several filter parameters and the fact that changes to how the Inertial Measurement Unit (IMU) is oriented necessitate modifying the filter. To overcome those problems, this paper presents a novel method of fusing sensor data from a Doppler Velocity Log (DVL) and IMU using an adaptive nonlinear estimator to provide dead reckoning localization for a small autonomous surface vehicle. The proposed method has only one insensitive tuning parameter and is agnostic to the configuration of the IMU. The system was validated using a small ASV in a 2.4×3.6×2.4 m water tank, with a motion capture system as ground truth, and was evaluated against a state-of-the-art method based on KFs. Experiments showed that the average drift error of the nonlinear filter was 0.16 m (s.d. 0.06 m) compared to 0.15 m (s.d. 0.05 m) for the state of the art, meaning that the benefits in terms of tuning and flexible configuration do not come at the expense of performance.**

I. Introduction

In complex, confined aquatic environments such as nuclear wet-storage facilities, water storage tanks or flooded mines, inspection with autonomous Aquatic Surface Vehicles (ASVs) is increasingly desirable to reduce operational costs and the need for dangerous manual inspection by humans. However, without reliable and accurate localization, only limited levels of autonomy are achievable [1]. For inspection tasks, in particular, accurate and precise location data is imperative for the repeatability of measurements and pinpointing detected anomalies [2].

Localization methods commonly used on land or in open water, such as wheel encoders or Global Navigation Satellite Systems (GNSS), are generally unsuitable for use on ASVs operating in enclosed, confined environments, and different approaches are required. In general, the sensory components of a robot's localization system can be broken down into two broad categories: 1) those providing distance measurements relative to fixed objects or features in the environment, and therefore offering a position fix relative to the environment (world or map frame), such as LiDAR or sonar; 2) and those providing a velocity in the robot's body frame, such that odometry can be provided. Where possible, sensors from both categories are fused to improve accuracy and redundancy. A prior experimental study [3] addressing the first category concluded that 2D LiDAR-based positioning systems are well suited to ASVs in enclosed spaces. Therefore, this paper focuses on category two.

World frame displacements relative to an initial position can be derived from body frame velocity by integrating successive velocity measurements and transforming them into the initial position frame. This process, dead reckoning, is usually unsuitable as a stand-alone localization technique due to inherent drift in the system from accumulated error. Nonetheless, it may be used to supplement other localization systems, such as visual or LiDAR-based Simultaneous Localization and Mapping (SLAM), or as a backup in the case of outages in a primary localization system [4].

Sensors that provide body frame velocity depend highly on the robot type and the environment, and options for aquatic vehicles are limited. Visual odometry is possible but relies on feature-rich environments and good ambient lighting, which cannot be guaranteed [5].

On small ASVs, MEMs-based Inertial Measurement Units (IMUs) are a common sensor choice. They are largely unaffected by environmental conditions and are small in size and cost but can lack the precision required to provide reliable linear velocity measurements [4]. An IMU contains, at a minimum, a 3-axis gyroscope and a 3-axis accelerometer. The gyroscope directly provides angular velocity measurements. However, the gravity vector must be removed from raw accelerometer measurements before the resulting acceleration can be integrated to provide linear velocities. This process accumulates errors very quickly, mostly due to inaccuracies in the gravity vector's orientation and the resulting error in its removal from the raw reading.

It is, therefore, desirable to use a second sensor to measure linear velocity directly, and Doppler Velocity Logs (DVLs) are a common choice. As an acoustic sensor, DVL benefits from being immune to water turbidity yet can be affected by multi-path and echos. Until recently, most commercially available DVLs were too large for use on small ASVs, and, to date, most research into IMU/DVL localization has focused on large ocean-going vessels [6].

A. Related works

Rogne et al. [7] proposed a dead-reckoning system using two IMUs. They used different nonlinear observer models

This work was supported by the UK Research and Innovation under an EPSRC IAA fund, an EPSRC PhD studentship (Jessica Paterson) and EPSRC grant number EP/P01366X/1, and by the Royal Academy of Engineering under the Research Chairs and Senior Research Fellowships programme, grant numbers CiET1819/13 and RCSRF2324-17-58. For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) license to any Accepted Manuscript version arising.

All the authors are with the Department of Electrical and Electronic Engineering, The University of Manchester, Manchester, United Kingdom. Emails: jessica.paterson-3@postgrad.manchester.ac.uk, {bruno.adorno, keir.groves, barry.lennox}@manchester.ac.uk.

aided by position reference systems during a dynamic positioning operation in the North Sea, achieving 30 m of averaged position error in 5 minutes.

Krishnamurthy and Khorrami [8] presented a simulated self-aligning navigation system for an underwater vehicle, which fused data from multiple sensors, including a DVL and an IMU. The system used an Unscented Kalman Filter (UKF) to estimate the position, velocity, and orientation. Campos et al. [9] used an Extended Kalman Filter (EKF) to fuse data from an IMU-based dead reckoning system and GPS to provide odometry for an ASV as part of a mapping exercise for offshore wind farm infrastructure. Both [8] and [9] evaluate their systems in simulation without accounting for the negative effects of real-world measurement issues such as sensor misalignment, sensor bias, time synchronization, and non-uniform noise profiles.

The problem with methods based on system linearization, such as EKFs, is that the underlying space of rigid motions (e.g., $\mathrm{SE}(3)$) is a non-Euclidean manifold with an inherent nonlinear structure, where noises do not propagate through additive operations but multiplicative ones. Some works mitigate this problem by using UKF with nonlinear rotational error propagation [10], but those techniques usually lack formal proof of convergence. This means that despite working well in practice, there is no guarantee that those filters will not diverge. Therefore, some authors have used nonlinear estimation in systems involving rotations to formally ensure the stability of the observer error [11], [12].

B. Statement of contributions

We propose a nonlinear estimator for dead reckoning composed of three components: the estimation of the misalignment between the IMU and body frame, the data fusion from the IMU and DVL, and a nonlinear update of the ASV pose. The key contributions of this paper can be summarized in three points:

1) A simple formulation for a highly accurate IMU/DVL dead reckoning system that makes no assumptions about the initial orientation of the IMU or the steady-state roll and pitch of the surface vehicle.
2) A formal analysis of the self-calibrating algorithm using Lyapunov theory, showing that the estimated gravity vector converges asymptotically to the actual gravity vector.
3) Experimental validation of the dead reckoning system in a confined aquatic environment using a sub-millimeter accurate ground truth.

II. MallARD's hardware architecture

MallARD is a holonomic ASV with a four-thruster, two-pontoon configuration [3]. In the present work, MallARD was fitted with a Waterlinked A50 DVL [13] and a tactical grade LORD L3DMGQ7-GNSS/INS [14]. Regarding the IMU, only the raw 6-axis, angular rate and linear acceleration measurements were used.

As shown in Fig. 1, the DVL was mounted underneath the rear pontoon attached via an adapter that holds the DVL below the thrusters to minimize the risk of acoustic interference. The IMU, fitted to an internal electronics tray in the rear pontoon, is not guaranteed to be in the same orientation if, for example, the electronics tube is opened to allow access to the hardware. Hence, the IMU is unlikely to be in the same orientation each time the robot is deployed. For simplicity, the robot's body frame $\mathcal{F}_B$ was chosen to align with the VICON object frame, although this is not a system requirement.

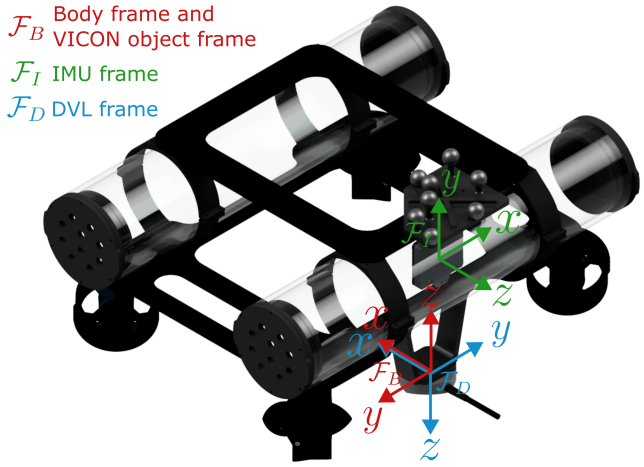

Fig. 1. Rendered image of the MallARD ASV with DVL, IMU, and VICON object fitted

III. Mathematical Background

We use dual quaternion algebra, which provides a computationally efficient and compact way of representing poses and twists with a strong geometrical meaning [15]. Quaternions are a straightforward extension of complex numbers [16], where the imaginary units obey $\hat{\imath}^2 = \hat{\jmath}^2 = \hat{k}^2 = \hat{\imath}\hat{\jmath}\hat{k} = -1$. The quaternion set is defined as $\mathbb{H} \triangleq \{a + b\hat{\imath} + c\hat{\jmath} + d\hat{k} : a, b, c, d \in \mathbb{R}\}$. The multiplication and addition operations are the same as in complex numbers, and one must only respect the properties of imaginary units. Given $\boldsymbol{h} \in \mathbb{H}$, such that $\boldsymbol{h} = a + b\hat{\imath} + c\hat{\jmath} + d\hat{k}$, the real part is given by $\mathrm{Re}(\boldsymbol{h}) \triangleq a$ and the imaginary part by $\mathrm{Im}(\boldsymbol{h}) \triangleq b\hat{\imath} + c\hat{\jmath} + d\hat{k}$. The quaternion norm is defined as $\|\boldsymbol{h}\| \triangleq \sqrt{\boldsymbol{h}\boldsymbol{h}^*}$, where $\boldsymbol{h}^* \triangleq \mathrm{Re}(\boldsymbol{h}) - \mathrm{Im}(\boldsymbol{h})$ is the quaternion conjugate.

The subset $\mathbb{H}_p \triangleq \{\boldsymbol{h} \in \mathbb{H} : \mathrm{Re}(\boldsymbol{h}) = 0\}$ contains elements that represent points in the tridimensional Euclidean space, in the sense that addition of elements of $\mathbb{H}_p$ (e.g., $x\hat{\imath} + y\hat{\jmath} + z\hat{k}$) is analogous to the addition of their counterparts in $\mathbb{R}^3$ (e.g., (x, y, z)). The subset $\mathbb{S}^3 \triangleq \{\boldsymbol{h} \in \mathbb{H} : \|\boldsymbol{h}\| = 1\}$ has elements that represent tridimensional rotations. For instance, a unit-norm quaternion $\boldsymbol{r} \in \mathbb{S}^3$ can always be written as $\boldsymbol{r} = \cos(\phi/2) + \boldsymbol{n}\sin(\phi/2)$, which represents a rotation of an angle $\phi \in \mathbb{R}$ around the rotation axis $\boldsymbol{n} \in \mathbb{H}_p \cap \mathbb{S}^3$.

Given the rotation $\boldsymbol{r}_B^A \in \mathbb{S}^3$ from frame $\mathcal{F}_A$ to frame $\mathcal{F}_B$, and a point $\boldsymbol{p}^B \in \mathbb{H}_p$ expressed in $\mathcal{F}_B$, we can express it in $\mathcal{F}_A$ using the adjoint operation:

$$\boldsymbol{p}^A = \mathrm{Ad}\left(\boldsymbol{r}_B^A\right)\boldsymbol{p}^B \triangleq \boldsymbol{r}_B^A \boldsymbol{p}^B \left(\boldsymbol{r}_B^A\right)^*. \tag{1}$$

The derivative of $\boldsymbol{r}_B^A$ is given by

$$\dot{\boldsymbol{r}}_B^A = \frac{1}{2}\boldsymbol{\omega}_{A,B}^A \boldsymbol{r}_B^A, \tag{2}$$

where $\boldsymbol{\omega}_{A,B}^A \in \mathbb{H}_p$ is the angular velocity of $\mathcal{F}_B$ with respect to $\mathcal{F}_A$, expressed in $\mathcal{F}_A$.

Given $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{H}_p$, the cross product and inner product have the same meaning as their counterparts in $\mathbb{R}^3$ and are defined, respectively, as

$$\boldsymbol{a} \times \boldsymbol{b} \triangleq \frac{1}{2}\left(\boldsymbol{a}\boldsymbol{b} - \boldsymbol{b}\boldsymbol{a}\right), \quad \langle \boldsymbol{a}, \boldsymbol{b} \rangle \triangleq -\frac{1}{2}\left(\boldsymbol{a}\boldsymbol{b} + \boldsymbol{b}\boldsymbol{a}\right). \tag{3}$$

Dual quaternion algebra extends the set of quaternions by introducing the dual unit ε [17]. The set of dual quaternions is defined as $\mathcal{H} \triangleq \{\boldsymbol{h} + \varepsilon \boldsymbol{h}' : \boldsymbol{h}, \boldsymbol{h}' \in \mathbb{H}, \varepsilon^2 = 0, \varepsilon \neq 0\}$. Given $\underline{\boldsymbol{h}} \in \mathcal{H}$, such that $\underline{\boldsymbol{h}} = \boldsymbol{h} + \varepsilon \boldsymbol{h}'$, the primary part is given by $\mathcal{P}\left(\underline{\boldsymbol{h}}\right) = \boldsymbol{h}$ and the dual part is given by $\mathcal{D}\left(\underline{\boldsymbol{h}}\right) = \boldsymbol{h}'$.

The set of pure dual quaternions, used to represent twists, is given by $\mathcal{H}_p \triangleq \{\underline{\boldsymbol{h}} \in \mathcal{H} : \mathrm{Re}\left(\mathcal{P}\left(\underline{\boldsymbol{h}}\right)\right) = \mathrm{Re}\left(\mathcal{D}\left(\underline{\boldsymbol{h}}\right)\right) = 0\}$. The set of unit-norm dual quaternions containing elements representing rigid motions is given by $\underline{\boldsymbol{\mathcal{S}}} \triangleq \{\underline{\boldsymbol{h}} \in \mathcal{H} : \|\underline{\boldsymbol{h}}\| = 1\}$, where $\|\underline{\boldsymbol{h}}\| \triangleq \sqrt{\underline{\boldsymbol{h}}\underline{\boldsymbol{h}}^*}$ is the dual quaternion norm and $\underline{\boldsymbol{h}}^* = \boldsymbol{h}^* + \varepsilon \boldsymbol{h}'^*$ is the dual quaternion conjugate. An element $\underline{\boldsymbol{x}} \in \underline{\boldsymbol{\mathcal{S}}}$ can always be written as $\underline{\boldsymbol{x}} = \boldsymbol{r} + (1/2)\varepsilon \boldsymbol{p}\boldsymbol{r}$, where $\boldsymbol{r} \in \mathbb{S}^3$ represents a rotation and $\boldsymbol{p} \in \mathbb{H}_p$ represents a translation.

Consider a time-varying rigid motion $\underline{\boldsymbol{x}}_B^A \in \underline{\boldsymbol{\mathcal{S}}}$ from $\mathcal{F}_A$ to $\mathcal{F}_B$, such that $\underline{\boldsymbol{x}}_B^A = \boldsymbol{r}_B^A + (1/2)\varepsilon \boldsymbol{p}_{A,B}^A \boldsymbol{r}_B^A$, in which $\boldsymbol{p}_{A,B}^A \in \mathbb{H}_p$ is the translation from $\mathcal{F}_A$ to $\mathcal{F}_B$, expressed in $\mathcal{F}_A$, and $\boldsymbol{r}_B^A \in \mathbb{S}^3$ is the rotation from $\mathcal{F}_A$ to $\mathcal{F}_B$. The time derivative of $\underline{\boldsymbol{x}}_B^A$ is given by

$$\dot{\underline{\boldsymbol{x}}}_B^A = \frac{1}{2}\underline{\boldsymbol{\xi}}_{A,B}^A \underline{\boldsymbol{x}}_B^A = \frac{1}{2}\underline{\boldsymbol{x}}_B^A \underline{\boldsymbol{\xi}}_{A,B}^B,$$

where $\underline{\boldsymbol{\xi}}_{AB}^A = \boldsymbol{\omega}_{A,B}^A + \varepsilon\left(\dot{\boldsymbol{p}}_{A,B}^A + \boldsymbol{p}_{A,B}^A \times \boldsymbol{\omega}_{A,B}^A\right)$ is the twist in frame $\mathcal{F}_A$, in which $\boldsymbol{\omega}_{A,B}^A \in \mathbb{H}_p$ is the angular velocity that satisfies (2), and $\underline{\boldsymbol{\xi}}_{AB}^B = \boldsymbol{\omega}_{A,B}^B + \varepsilon \dot{\boldsymbol{p}}_{A,B}^B$ is the twist in $\mathcal{F}_B$.

IV. Nonlinear estimator for dead reckoning

The nonlinear estimator for dead reckoning is composed of three components: the estimation of the misalignment between the IMU and body frame, the data fusion from the IMU and DVL, and a nonlinear update of the vehicle pose. For the estimator design, we have the following assumptions: **A1)** The transformation between the DVL and body frames is known; **A2)** The vehicle's acceleration does not affect, on average, the accelerometer measurements; **A3)** There is neither roll nor pitch, and the gravity vector is parallel to the yaw axis; i.e., $\boldsymbol{g}^B = \boldsymbol{g}^W \in \mathbb{H}_p$.

The body, VICON, and DVL frames are assumed to be at the same position on the robot, with the DVL frame rotated 180 degrees about the x-axis compared to the other two frames. For the derivation of the estimator of the misalignment between the IMU and body frame, we also assume that we have perfect knowledge of the static gravity vector in the IMU frame. Nonetheless, this assumption is relaxed, and we design a recursive expression for the estimation of the zero-frequency component of the gravity vector in the IMU frame, which converges asymptotically to the actual gravity vector.

A. Estimation of the misalignment between the IMU and body frame

Consider a perfect[1] (i.e., without noise) static gravity vector $\boldsymbol{g}^I \in \mathbb{H}_p$ in the IMU frame, $\mathcal{F}_I$. Assume that the body frame and the world frame have a parallel common axis aligned with the gravity vector, such that $\boldsymbol{g}^B = \boldsymbol{g}^W \in \mathbb{H}_p$. For example, if the gravity vector is aligned with the z axis, then $\boldsymbol{g}^B = \boldsymbol{g}^W = 9.81\hat{k}\,\mathrm{m/s^2}$; if it is aligned with the x axis, then $\boldsymbol{g}^B = \boldsymbol{g}^W = 9.81\hat{\imath}\,\mathrm{m/s^2}$. Given an unknown constant rotation $\boldsymbol{r}_I^B \in \mathbb{S}^3$ between the body frame $\mathcal{F}_B$ and $\mathcal{F}_I$, the goal is to find an estimation $\hat{\boldsymbol{r}}_I^B(t)$ such that the estimated gravity vector in the body frame, $\hat{\boldsymbol{g}}^B(t) = \mathrm{Ad}\left(\hat{\boldsymbol{r}}_I^B(t)\right)\boldsymbol{g}^I$, converges asymptotically to the actual constant gravity vector $\boldsymbol{g}^B$; that is,

$$\lim_{t\to\infty} \hat{\boldsymbol{g}}^B(t) = \lim_{t\to\infty} \mathrm{Ad}\left(\hat{\boldsymbol{r}}_I^B(t)\right)\boldsymbol{g}^I = \boldsymbol{g}^B. \tag{4}$$

We define the gravity estimation error

$$\tilde{\boldsymbol{g}}^B(t) \triangleq \hat{\boldsymbol{g}}^B(t) - \boldsymbol{g}^B, \tag{5}$$

such that (4) is equivalent to

$$\lim_{t\to\infty} \tilde{\boldsymbol{g}}^B = 0. \tag{6}$$

Therefore, $\tilde{\boldsymbol{g}}^B = \mathrm{Ad}\left(\hat{\boldsymbol{r}}_I^B\right)\boldsymbol{g}^I - \boldsymbol{g}^B$. Because we assume that there is neither roll nor pitch, the gravity vector is parallel to the yaw axis, thus $\dot{\boldsymbol{g}}^B = 0$. Using (1), (2), and (3) we obtain

$$\dot{\tilde{\boldsymbol{g}}}^B = \dot{\hat{\boldsymbol{r}}}_I^B \boldsymbol{g}^I \left(\hat{\boldsymbol{r}}_I^B\right)^* + \hat{\boldsymbol{r}}_I^B \boldsymbol{g}^I \left(\dot{\hat{\boldsymbol{r}}}_I^B\right)^* = \hat{\boldsymbol{\omega}}_{B,I}^B \times \hat{\boldsymbol{g}}^B. \tag{7}$$

Given the estimation error dynamics (7), we want to find a suitable adaptation signal $\hat{\boldsymbol{\omega}}_{B,I}^B(t)$ such that the estimated rotation $\hat{\boldsymbol{r}}_I^B(t)$ ensures that (6) is fulfilled, that is, the estimated gravity vector in the body frame converges to its actual vector.

Theorem 1: Given the adaptation signal

$$\hat{\boldsymbol{\omega}}_{B,I}^B(t) = \lambda\left(\hat{\boldsymbol{g}}^B(t) \times \boldsymbol{g}^B\right), \tag{8}$$

where $\lambda \in (0, \infty)$, the gravity estimation error converges to zero whenever $\hat{\boldsymbol{g}}^B(0) \neq -\boldsymbol{g}^B$.

Proof: Consider the Lyapunov candidate $V\left(\tilde{\boldsymbol{g}}^B\right) = \frac{1}{2}\left\|\tilde{\boldsymbol{g}}^B\right\|^2 = \frac{1}{2}\langle \tilde{\boldsymbol{g}}^B, \tilde{\boldsymbol{g}}^B \rangle$. Its time derivative is given by

$$\dot{V}\left(\tilde{\boldsymbol{g}}^B\right) = \langle \tilde{\boldsymbol{g}}^B, \dot{\tilde{\boldsymbol{g}}}^B \rangle = \langle \tilde{\boldsymbol{g}}^B, \hat{\boldsymbol{\omega}}_{B,I}^B(t) \times \hat{\boldsymbol{g}}^B \rangle, \tag{9}$$

where the last equality holds due to (7). We replace (8) into (9), use (5) and the fact that $\lambda\left(\hat{\boldsymbol{g}}^B \times \boldsymbol{g}^B\right) \times \hat{\boldsymbol{g}}^B = \lambda\left(\boldsymbol{g}^B \langle \hat{\boldsymbol{g}}^B, \hat{\boldsymbol{g}}^B \rangle - \hat{\boldsymbol{g}}^B \langle \boldsymbol{g}^B, \hat{\boldsymbol{g}}^B \rangle\right)$ to obtain

$$\dot{V}\left(\tilde{\boldsymbol{g}}^B\right) = \lambda\left(\left\|\boldsymbol{g}^B\right\| \left\|\hat{\boldsymbol{g}}^B\right\|\right)^2 \left(\cos^2 \phi_{g,\hat{g}} - 1\right) \leq 0, \tag{10}$$

[1]The IMU provides noisy measurements, and the surface robot oscillates due to water motion. Those main assumptions are relaxed in Section IV-C.

where $\phi_{g,\hat{g}}$ is the angle between $\boldsymbol{g}^B$ and $\hat{\boldsymbol{g}}^B$. Therefore, $\dot{V}\left(\tilde{\boldsymbol{g}}^B\right) < 0$ for all $\hat{\boldsymbol{g}}^B \neq \pm\boldsymbol{g}^B$ and $\dot{V}\left(\tilde{\boldsymbol{g}}^B\right) = 0$ if and only if $\hat{\boldsymbol{g}}^B = \pm\boldsymbol{g}^B$.[2] Thus, whenever $\hat{\boldsymbol{g}}^B(0) \neq -\boldsymbol{g}^B$, the adaptation signal (8) ensures that $\lim_{t\to\infty} V\left(\tilde{\boldsymbol{g}}^B\right) = 0$, which implies $\lim_{t\to\infty} \tilde{\boldsymbol{g}}^B = 0$. ■

The closed-loop system composed of the error dynamics (7) under the adaptation signal (8) has two equilibrium points, namely $\hat{\boldsymbol{g}}^B = \pm\boldsymbol{g}^B$. Because the gravity estimation error norm always decreases under the adaptation signal (8), except when the initial estimation is $\hat{\boldsymbol{g}}^B(0) = \pm\boldsymbol{g}^B$, the estimated value $\hat{\boldsymbol{g}}^B$ will always converge to the actual value $\boldsymbol{g}^B$ when $\hat{\boldsymbol{g}}^B(0) \neq -\boldsymbol{g}^B$.

B. Combining data from the IMU and DVL

The DVL provides the velocities $\boldsymbol{\nu}^D$ along the x and y axes of its own frame, $\mathcal{F}_D$, whereas the IMU gives the angular velocities $\boldsymbol{\omega}^I$ expressed in its own frame, $\mathcal{F}_I$. Assuming a known constant rotation $\boldsymbol{r}_D^B$ between the body frame $\mathcal{F}_B$ and the DVL frame $\mathcal{F}_D$, our goal is to combine data from the DVL and the IMU in a single estimated twist

$$\hat{\underline{\boldsymbol{\xi}}}_{W,B}^W = \hat{\boldsymbol{\omega}}_{W,B}^W + \varepsilon\left(\dot{\hat{\boldsymbol{p}}}_{W,B}^W + \hat{\boldsymbol{p}}_{W,B}^W \times \hat{\boldsymbol{\omega}}_{W,B}^W\right) \tag{11}$$

of the body frame with respect to the world frame $\mathcal{F}_W$, where $\hat{\boldsymbol{p}}_{W,B}^W \in \mathbb{H}_p$ is the translation associated with the estimated pose

$$\dot{\hat{\underline{\boldsymbol{x}}}}_B^W = \hat{\boldsymbol{r}}_B^W + \varepsilon\frac{1}{2}\hat{\boldsymbol{p}}_{W,B}^W\hat{\boldsymbol{r}}_B^W. \tag{12}$$

Since $\hat{\boldsymbol{r}}_I^B$ and $\hat{\boldsymbol{\omega}}_{B,I}^B$ satisfy the relationship in (2), then

$$\dot{\hat{\boldsymbol{r}}}_I^B = \frac{1}{2}\hat{\boldsymbol{\omega}}_{B,I}^B\hat{\boldsymbol{r}}_I^B. \tag{13}$$

Thus, after we calculate $\hat{\boldsymbol{\omega}}_{B,I}^B$ using (8), we integrate (13) to obtain an updated estimate of $\hat{\boldsymbol{r}}_I^B$, and use it to obtain the estimated angular velocity of the body frame with respect to the world frame, $\hat{\boldsymbol{\omega}}_{W,B}^W = \mathrm{Ad}\left(\hat{\boldsymbol{r}}_B^W\hat{\boldsymbol{r}}_I^B\right)\boldsymbol{\omega}^I$. Analogously, $\dot{\hat{\boldsymbol{p}}}_{W,B}^W = \mathrm{Ad}\left(\hat{\boldsymbol{r}}_B^W\boldsymbol{r}_D^B\right)\boldsymbol{\nu}^D$. The time-varying estimated body pose (12) and the estimated body twist (11) satisfy

$$\dot{\hat{\underline{\boldsymbol{x}}}}_B^W = \frac{1}{2}\hat{\underline{\boldsymbol{\xi}}}_{W,B}^W\hat{\underline{\boldsymbol{x}}}_B^W, \tag{14}$$

which is integrated to update the body pose estimate with respect to $\mathcal{F}_W$.

C. Practical implementation

In Section IV-A, we assume that the gravity vector in the body frame is aligned with the gravity vector in the world frame (i.e., $\boldsymbol{g}^B = \boldsymbol{g}^W$), meaning that there is neither pitch nor roll. We also did not consider the IMU noise. Both assumptions are, in practice, false. The surface robot oscillates due to the water motion, and the IMU is noisy; hence, the gravity vector $\boldsymbol{g}^I$ measured in $\mathcal{F}_I$ is not constant. Nonetheless, because the IMU noise is Gaussian and the robot tends to oscillate symmetrically with respect to the vertical axis, we can use the zero-frequency component of the gravity vector in the IMU frame, denoted by $\bar{\boldsymbol{g}}^I \in \mathbb{H}_p$.

The zero-frequency gravity vector is given by the average value across all IMU measurements in the discrete-time domain, which can be efficiently calculated using a recursive formulation, namely

$$\bar{\boldsymbol{g}}^I[n] = \frac{1}{n}\boldsymbol{g}^I[n] + \frac{n-1}{n}\bar{\boldsymbol{g}}^I[n-1], \tag{15}$$

where $\bar{\boldsymbol{g}}^I[0] \triangleq 0$ and $n \in \{1, 2, 3, \ldots\}$. This way, when calculating the estimated gravity vector $\hat{\boldsymbol{g}}^B$ in the adaptation signal (8), we use the most up-to-date value of $\bar{\boldsymbol{g}}^I$, namely

$$\hat{\boldsymbol{g}}^B[n] = \mathrm{Ad}\left(\hat{\boldsymbol{r}}_I^B[n]\right)\bar{\boldsymbol{g}}^I[n]. \tag{16}$$

Fig. 2 presents the three axes of the measured gravity vector in the IMU frame $\mathcal{F}_I$, and the corresponding zero-frequency components, showing that the filtered measurements converge to a constant value corresponding to the actual gravity vector.

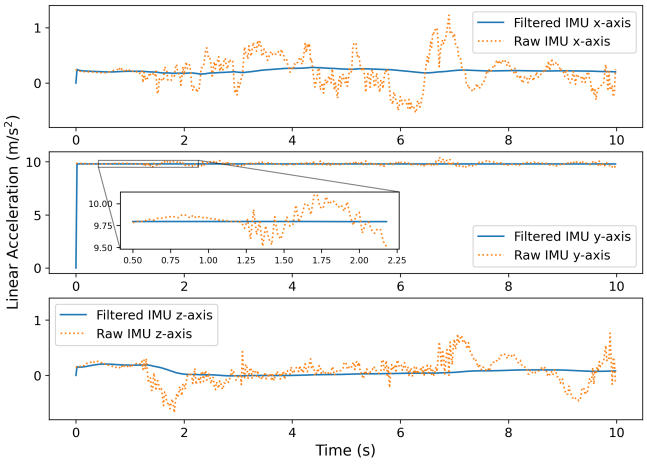

Fig. 2. Example of the raw measurement of the gravity vector using an IMU (*dotted lines*) and the corresponding zero-frequency components calculated using the recursive average (15) (*solid lines*).

Solving the differential equations (13) and (14) in continuous time to update the estimate (12) is notoriously hard [18], as the underlying space of rigid motions is a non-Euclidean manifold [19], [20]. Fortunately, those equations can be easily integrated numerically in the unit dual quaternion manifold [15] to obtain the estimates $\hat{\boldsymbol{r}}_I^B$ and $\hat{\underline{\boldsymbol{x}}}_B^W$ at time n according to

$$\hat{\boldsymbol{r}}_I^B[n] = \exp\left(\frac{T}{2}\hat{\boldsymbol{\omega}}_{B,I}^B[n]\right)\hat{\boldsymbol{r}}_I^B[n-1] \tag{17}$$

and

$$\hat{\underline{\boldsymbol{x}}}_B^W[n] = \hat{\underline{\boldsymbol{x}}}_B^W[n-1]\exp\left(\frac{T}{2}\hat{\underline{\boldsymbol{\xi}}}_{W,B}^B[n]\right), \tag{18}$$

respectively, where T is the sampling time and $\hat{\boldsymbol{r}}_I^B$ is given by an initial guess (it might be the identity 1) and $\hat{\underline{\boldsymbol{x}}}_B^W[0] = 1$.

Lastly, because the norm of the gravity vector is constant, we need only to estimate the gravity vector direction.

[2]Notice that $\hat{\boldsymbol{g}}^B = \pm\boldsymbol{g}^B$ implies $\phi_{g,\hat{g}} = \pm 1$. Furthermore, $\|\hat{\boldsymbol{g}}^B\| = \|\boldsymbol{g}^B\|$ because we assume a noiseless gravity vector. This means that it is unnecessary to consider the general case $\hat{\boldsymbol{g}}^B = \pm\alpha\boldsymbol{g}^B$ for $\alpha \in (0, \infty)$ which would also result in $\dot{V} = 0$.

Therefore, in practice, it is convenient to use the *normalized* gravity vector; that is $\boldsymbol{g}^B_{\text{norm}} \triangleq \boldsymbol{g}^B / \left\|\boldsymbol{g}^B\right\|$. This helps to eliminate measurement noises that affect the gravity vector norm.

The nonlinear estimation process is summarized in Algorithm 1 and was implemented using the DQ Robotics [21] module for Python. The computational cost is very low and suitable for real-time, with a total of floating-point operations summarized as follows: 173 additions, 279 multiplications, four trigonometric operations (sine and cosine), four divisions, and two square roots. Those operations can be found by analyzing the cost of addition, multiplication, adjoint operation, and exponential mapping of quaternions and dual quaternions [15].

Algorithm 1 Nonlinear estimator.

1: $T \leftarrow$ sampling time; $n \leftarrow 1$
2: $\boldsymbol{g}^I[0] \leftarrow$ initial guess; $\hat{\underline{\boldsymbol{x}}}^W_B[0] \leftarrow 1$; $\hat{\boldsymbol{r}}^B_I[0] \leftarrow$ initial guess
3: **while** deadReckoning() **do**
4: ▷ *Get measurements from sensors*
5: $\boldsymbol{g}^I[n], \boldsymbol{\omega}^I[n], \boldsymbol{\nu}^D[n] \leftarrow$ getDataFromIMUandDVL()
6: ▷ *Filter* $\boldsymbol{g}^I[n]$ *and update the estimate* $\hat{\boldsymbol{r}}^B_I$
7: $\bar{\boldsymbol{g}}^I[n] \leftarrow \frac{1}{n}\boldsymbol{g}^I[n] + \frac{n-1}{n}\bar{\boldsymbol{g}}^I[n-1]$
8: $\hat{\boldsymbol{g}}^B[n] \leftarrow \text{Ad}\left(\hat{\boldsymbol{r}}^B_I[n-1]\right)\bar{\boldsymbol{g}}^I[n]$
9: $\hat{\boldsymbol{\omega}}^B_{B,I}[n] \leftarrow \lambda\left(\hat{\boldsymbol{g}}^B[n] \times \boldsymbol{g}^B\right)$
10: $\hat{\boldsymbol{r}}^B_I[n] \leftarrow \exp\left(\frac{T}{2}\hat{\boldsymbol{\omega}}^B_{B,I}[n]\right)\hat{\boldsymbol{r}}^B_I[n-1]$
11: ▷ *Project the measurements onto the body frame*
12: $\hat{\boldsymbol{\omega}}^B_{W,B}[n] \leftarrow \text{Ad}\left(\hat{\boldsymbol{r}}^B_I[n]\right)\boldsymbol{\omega}^I[n]$
13: $\hat{\dot{\boldsymbol{p}}}^B_{W,B}[n] \leftarrow \text{Ad}\left(\boldsymbol{r}^B_D\right)\boldsymbol{\nu}^D[n]$
14: ▷ *Update the body twist and current pose w.r.t world*
15: $\hat{\underline{\boldsymbol{\xi}}}^B_{W,B}[n] \leftarrow \hat{\boldsymbol{\omega}}^B_{W,B}[n] + \varepsilon\left(\hat{\dot{\boldsymbol{p}}}^B_{W,B}[n]\right)$
16: $\hat{\underline{\boldsymbol{x}}}^W_B[n] \leftarrow \hat{\underline{\boldsymbol{x}}}^W_B[n-1]\exp\left(\frac{T}{2}\hat{\underline{\boldsymbol{\xi}}}^B_{W,B}[n]\right)$
17: $n \leftarrow n+1$
18: **end while**

V. Experiments

We compared our proposed nonlinear filter with the work by Fukuda et al. [4], who used an IMU and DVL with a Kalman Filter (KF) for dead reckoning during GNSS outages onboard a 50 m long ship in Tokyo Bay. A camera-based motion capture system (VICON) was used instead of the global navigation satellite system (GNSS) to initialize the filter and as a ground truth. Values for the covariance matrices were chosen according to sensor datasheets where possible and KF's $\mathbf{Q}$ matrix values were manually tuned to give the lowest drift possible in the experiments.

A. Data Collection

A VICON motion capture system was used to record MallARD's body frame orientation and position. The DVL frame was collocated with the body frame to reduce the number of transformations but rotated 180 degrees about the x axis. The VICON system consisted of 8 cameras mounted around a $2.4\,\text{m} \times 3.6\,\text{m}$ metallic pool. The vehicle was tethered to receive joystick commands and transfer sensor data to a base station computer.

MallARD was manually driven around the pool in different motion patterns in ten experiments, including figure-of-eight, stripes, squares, and pure rotation. Each experiment was capped at two minutes as the dead reckoning system is not intended to be used stand-alone for extended periods.

B. Experimental Method and Results

For brevity, the figures in this section present data from one experiment, figure-of-eight, unless otherwise stated. The results for all experiments can be verified using the open-source code and data sets available on GitHub [22].

Before dead reckoning was started, the IMU to body-frame rotation estimation, described in lines 7–10 in Algorithm 1, was allowed a four-second period to align the gravity vectors automatically. The tuning factor was set to $\lambda = 10$ for all experiments.

To assess the efficacy of the algorithm, it was first run with the initial estimated orientation of the IMU in the body frame defined to be close to its nominal orientation (i.e., $\hat{\boldsymbol{r}}^B_I[0] = \cos(\pi/3) + \boldsymbol{n}\sin(\pi/3)$, with $\boldsymbol{n} = (\sqrt{3}/3)(\hat{\imath} - \hat{\jmath} - \hat{k})$). Then, the same data was used with the initial condition $\hat{\boldsymbol{r}}^B_I[0] = 1$. Fig. 3 shows that, in both cases, the normalized error between the estimated gravity vector and the actual gravity vector in the body frame quickly converges to zero. Therefore, in practice, we do not need to know the nominal IMU to body-frame rotation, thanks to the quick convergence of the estimation error.

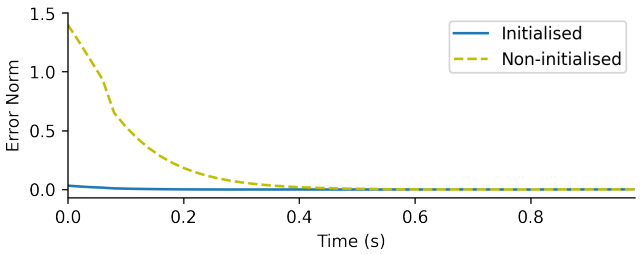

Fig. 3. The normalized error between the estimated and actual gravity vectors in the body frame during a four-second calibration. The initialised estimated rotation is $\hat{\boldsymbol{r}}^B_I[0] = \cos(\pi/3) + \boldsymbol{n}\sin(\pi/3)$ and the non-initialized is $\hat{\boldsymbol{r}}^B_I[0] = 1$.

Fig. 4 shows the yaw component (around the body-frame z-axis) of the aligned gyroscope data (i.e., already considering the IMU to body-frame rotation estimation) against the yaw rate extracted from the VICON's orientation quaternion. The alignment between the rotated gyroscope data and robot's body frame is held over the full length of the experiment, highlighting the quality of the IMU to body-frame rotation estimation, which is given by the adaptation law (8) and the rotation propagation (17).

To assess the dead reckoning, the full estimated trajectory is plotted in Fig. 5 along with the results from the state of the art and VICON ground truth. Over the course of the two-minute experiment, the dead reckoning estimation from the nonlinear filter gives a final position drift of 0.04 m. The state of the art is similarly positioned with a final drift of 0.06 m. Here, 'drift' is defined as the Euclidean distance

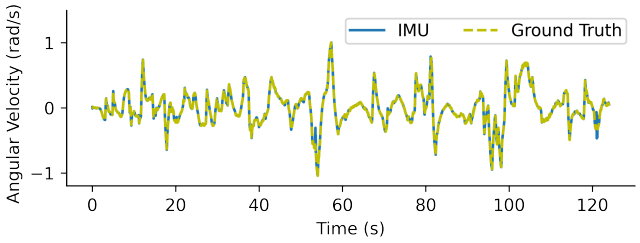

Fig. 4. The angular velocity from the aligned gyroscope data and the VICON ground truth around the body frame z-axis.

between the final position as determined by the ground truth and the final position as estimated using dead reckoning.

The relatively accurate final position demonstrates that the system would operate well as a short-term backup (i.e., less than two minutes) in case of outages or as an odometry source for a SLAM package. Averaged over all ten experiments, summarized in Fig. 6, the average drift for the nonlinear filter was 0.16 m with a standard deviation (s.d.) of 0.06 m vs. an average drift of 0.15 m (s.d. 0.05 m) for the state of the art.

The experiments were repeated with the IMU data rotated by fixed amounts in degrees to further evaluate the effects of IMU misalignment in the electronics tube. The KF-based method is not re-tuned for each rotation and shows a noticeable increase in final horizontal drift, even for small ($\leq$ 10 °) rotations of the IMU, whereas the nonlinear filter is practically unaffected. Fig. 7 shows this rotation has an effect on both methods for the figure-of-eight experiment. A similar relationship between angle and drift was observed for each method across all experiments.

VI. Conclusions and Future Work

This paper has proposed a dual-quaternion-based nonlinear estimator for dead reckoning and a comprehensive experimental evaluation using real-world experiments. The estimator's performance is comparable to the state of the art if the state-of-the-art system has accurate sensor placement information. However, the nonlinear estimator has several advantages. First, knowledge of the rotation between the IMU and the robot's body frame is unnecessary because it is calculated online by the estimator, and the estimation converges within a few seconds of startup. This allows flexibility in robot configuration that is useful in space-constrained robot platforms where components are regularly swapped and repositioned. Second, unlike state-of-the-art approaches, often based on KFs and challenging to tune, the proposed method has only one tuning parameter determining the estimation convergence rate. Third, a formal analysis has proven closed-loop stability of the estimation error for the nonlinear filter; formal convergence guarantees are not available with KF-based approaches because the system is inherently nonlinear. Lastly, the nonlinear estimator is computationally inexpensive, requiring 173 additions, 279 multiplications, four trigonometric operations (sine and cosine), four divisions, and two square roots, which makes it suitable for real-time applications and implementation in onboard processing units.

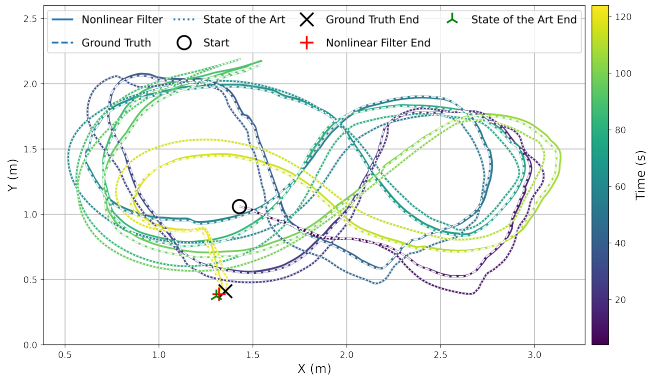

Fig. 5. Trajectory in x and y in the world frame for both methods and ground truth.

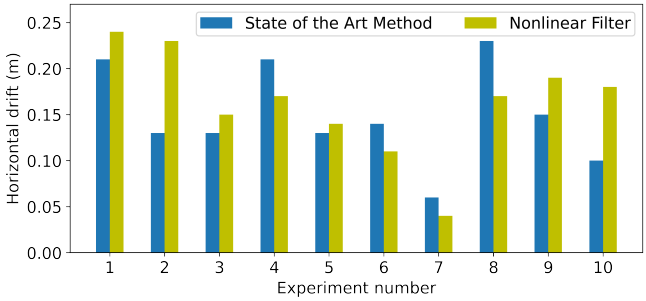

Fig. 6. Final horizontal drift for each experiment.

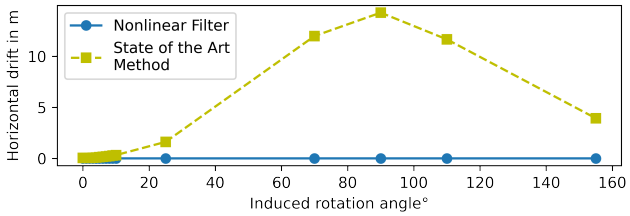

Fig. 7. IMU data rotated by fixed amounts to simulate a misalignment of the IMU in the electronics tube. The nonlinear estimator estimates this misalignment online, thus being practically unaffected by it.

In ten two-minute-long experiments, the average drift was 0.16 m (s.d. of 0.06 m), suitable for use in tandem with a complementary absolute localization system, providing estimates between readings and as a backup system in case of outages.

Although this work has used a DVL for body-frame velocity measurements, the proposed nonlinear estimator is agnostic to the type of sensor providing this measurement and could be used on a wheeled vehicle incorporating velocity measurements from wheel encoders.

Future work will expand on the ideas presented here to provide high-accuracy self-tuning and self-aligning estimators, thus not requiring exact transformations between sensors.

REFERENCES

[1] S. Watson, D. A. Duecker, and K. Groves, "Localisation of Unmanned Underwater Vehicles (UUVs) in Complex and Confined Environments: A Review," *Sensors*, vol. 20, no. 21, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/21/6203

[2] J. Zhang, V. Ila, and L. Kneip, "Robust Visual Odometry in Underwater Environment," in *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO)*, 2018, pp. 1–9.

[3] K. Groves, B. Lennox, A. West, K. Gornicki, S. Watson, and J. Carrasco, "MallARD: An Autonomous Aquatic Surface Vehicle for Inspection and Monitoring of Wet Nuclear Storage Facilities," *Robotics*, vol. 8, no. 2, 2019. [Online]. Available: https://www.mdpi.com/2218-6581/8/2/47

[4] G. Fukuda, D. Hatta, X. Guo, and N. Kubo, "Performance Evaluation of IMU and DVL Integration in Marine Navigation," *Sensors*, vol. 21, no. 4, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/4/1056

[5] M. O. A. Aqel, M. H. Marhaban, M. I. Saripan, and N. B. Ismail, "Review of visual odometry: types, approaches, challenges, and applications," *SpringerPlus*, vol. 5, no. 1, p. 1897, Oct 2016. [Online]. Available: https://doi.org/10.1186/s40064-016-3573-7

[6] J. Snyder, "Doppler Velocity Log (DVL) navigation for observation-class ROVs," in *OCEANS 2010 MTS IEEE SEATTLE*, 2010, pp. 1–9.

[7] R. H. Rogne, T. H. Bryne, T. I. Fossen, and T. A. Johansen, "MEMS-based Inertial Navigation on Dynamically Positioned Ships: Dead Reckoning," *IFAC-PapersOnLine*, vol. 49, no. 23, pp. 139–146, 2016, 10th IFAC Conference on Control Applications in Marine SystemsCAMS 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896316319218

[8] P. Krishnamurthy and F. Khorrami, "A self-aligning underwater navigation system based on fusion of multiple sensors including DVL and IMU," in *2013 9th Asian Control Conference (ASCC)*, 2013, pp. 1–6.

[9] D. F. Campos, A. Matos, and A. M. Pinto, "Multi-domain Mapping for Offshore Asset Inspection using an Autonomous Surface Vehicle," in *2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2020, pp. 221–226.

[10] J. Kelly and G. S. Sukhatme, "Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-calibration," *The International Journal of Robotics Research*, vol. 30, pp. 56–79, 1 2011. [Online]. Available: http://journals.sagepub.com/doi/10.1177/0278364910382802

[11] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Nonlinear Complementary Filters on the Special Orthogonal Group," *IEEE Transactions on Automatic Control*, vol. 53, pp. 1203–1218, 6 2008. [Online]. Available: http://ieeexplore.ieee.org/document/4608934/

[12] G. G. Scandaroli, P. Morin, and G. Silveira, "A nonlinear observer approach for concurrent estimation of pose, IMU bias and camera-to-IMU rotation," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 3335–3341.

[13] Waterlinked, *The world's smallest DVL*, 2022. [Online]. Available: https://www.waterlinked.com/hubfs/Product_Assets/DVL_A50/wl-21035-3_DVL_A50.pdf

[14] MicroStrain Sensing, *MicroStrain Sensing Product Datasheet*, 2022. [Online]. Available: https://www.microstrain.com/sites/default/files/3DMGQ7 20GNSSINS 20Datasheet 20 288400-0133 29 20Rev 20A.pdf

[15] B. V. Adorno, "Robot Kinematic Modeling and Control Based on Dual Quaternion Algebra — Part I: Fundamentals," 2017.

[16] W. R. Hamilton, "II. On quaternions; or on a new system of imaginaries in algebra," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 25, pp. 10–13, 7 1844. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/14786444408644923

[17] J. M. Selig, *Geometric fundamentals of robotics*, 2nd ed., D. Gries and F. B. Schneider, Eds. Springer-Verlag New York Inc., 2005.

[18] A. Müller, "Singularity-free lie group integration and geometrically consistent evaluation of multibody system models described in terms of standard absolute coordinates," *Journal of Computational and Nonlinear Dynamics*, vol. 17, pp. 1–10, 5 2022.

[19] S. P. Bhat and D. S. Bernstein, "A topological obstruction to continuous global stabilization of rotational motion and the unwinding phenomenon," *Systems & Control Letters*, vol. 39, pp. 63–70, 1 2000. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167691199000900

[20] H. T. Kussaba, L. F. Figueredo, J. Y. Ishihara, and B. V. Adorno, "Hybrid kinematic control for rigid body pose stabilization using dual quaternions," *Journal of the Franklin Institute*, vol. 354, pp. 2769–2787, 5 2017. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0016003217300522

[21] B. V. Adorno and M. M. Marinho, "DQ Robotics: A Library for Robot Modeling and Control," *IEEE Robotics & Automation Magazine*, vol. 28, no. 3, pp. 102–116, Sept. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9136790/

[22] J. Paterson, B. V. Adorno, B. Lennox, and K. Groves, "ICRA2023 PatersonAdornoLennoxGroves," 2022. [Online]. Available: https://github.com/EEEManchester/ICRA2023-PatersonAdornoLennoxGroves