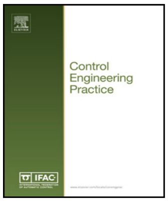
Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac



Feedback error-state Kalman filter with time-delay compensation for hydroacoustic-aided inertial navigation of underwater vehicles

Thor I. Fossen

Department of Engineering Cybernetics, Norwegian University of Science and Technology, 7491 Trondheim, Norway

ARTICLE INFO

Keywords:
Estimation and filtering
Navigation
Autonomous underwater vehicles
Acoustic-based networked control and navigation
Kalman filtering techniques in marine systems control

ABSTRACT

This article is intended as a tutorial to assist engineers who want to develop and implement low-cost underwater vehicle inertial navigation systems (INS) aided by time-delayed hydroacoustic position measurements. A discrete-time unit quaternion error-state Kalman filter (ESKF) is used for sensor fusion. The ESKF is implemented as a feedback algorithm with reset functionality. This is motivated by the need for long-endurance autonomous underwater vehicles (AUVs). Proprietary navigation systems do not allow users to add more measurement equations to the code if additional sensors are available. However, the open-source filter architecture presented in the article provides for this. The article also aims to make in-house development of strapdown INS accessible and affordable for vendors of low-cost AUV systems. Finally, a case study of an AUV with a standard sensor suite is included to demonstrate the performance of the ESKF aided by time-delayed position measurements.

1. Introduction

The growth in low-cost uncrewed systems for commercial applications has created a demand for accurate inertial navigation systems with a small computational footprint using low-cost inertial sensors. Typical applications are small uncrewed surface vehicles (USVs), autonomous underwater vehicles (AUVs), and fixed-wing uncrewed aerial vehicles (UAVs). The literature on inertial navigation systems (INS) is extensive and hard to navigate in textbooks and technical articles. Engineers who want to use an INS in their vehicle control systems often buy expensive proprietary systems with minimum interface capabilities and options for adding third-party sensor systems. A survey of underwater vehicle navigation and localization is found in Kinsey et al. (2006) and Paull et al. (2014).

The traditional Kalman filter algorithms for INS aided by global navigation satellite systems (GNSS) are described in detail by Farrell (2008) and references therein. The interested reader is also recommended to consult (Britting, 1971; Fossen, 2021; Grewal et al., 2001; Titterton & Weston, 1997). For underwater vehicles, a hydroacoustic position reference (HPR) system and a pressure meter can be used to aid (Stovner, 2018). In some cases, a Doppler velocity log (DVL) is included to improve the accuracy of the navigation solution (Jalving et al., 2007; Morgado et al., 2011; Zhao et al., 2012). The attitude of the AUV can be represented using Euler angles or unit quaternions (Fossen, 2021). The Euler angle error-state Kalman filter (ESKF) for underwater vehicle navigation is described by Miller et al. (2010). If the attitude is represented using a unit quaternion, a multiplicative extended Kalman filter (MEKF) can estimate the orientation as described by Markley and Crassidis (2014).

When designing an INS, there are two primary design philosophies (Fossen, 2021):

- Gimbal-mounted systems where the IMU platform is isolated from the vehicle's rotations through a set of gimbals, thus maintaining a fixed orientation in space. Four gimbals are used to avoid gimbal-lock, which happens when the axis of two gimbals is driven in the same direction, turning off their isolation capabilities. Mechanized systems use angular rate sensors, feedback control, and an actuated platform to keep the IMU in a fixed orientation in space.
- Strapdown systems use an IMU strapped to the vehicle. Consequently, the IMU will pick up the motions of the vehicle. This implies that the *strapdown INS equations* must be integrated online in a state estimator to accurately describe the IMUs and the vehicle's motions to separate these.

Gimbal-mounted systems are expensive compared to strapdown INS, while strapdown systems require more computational power than gimbal-mounted systems. Developing solid-state sensors based on microelectromechanical systems (MEMS) technology makes strapdown INS very affordable. However, strapdown navigation systems can have significant errors, especially in low-cost systems. This is usually handled using feedback from accurate positioning systems to remove drift.

E-mail address: thor.fossen@ntnu.no.
URL: https://www.fossen.biz.

https://doi.org/10.1016/j.conengprac.2023.105603
Received 20 March 2023; Received in revised form 15 June 2023; Accepted 15 June 2023
Available online xxxx

This article focuses on designing a low-cost strapdown INS for AUVs where drift is accurately compensated by the ESKF using hydroacoustic position measurements and a depth sensor. Attitude is parametrized using unit quaternions. The advantage of the unit quaternion to the Euler angle representation is that the ±90 degree singularity in pitch is avoided (Miller et al., 2010). In addition, unit quaternions have better numerical properties and accuracy than the Euler angle representation when used in an ESKF.

1.1. Strapdown INS aided by hydroacoustic position measurements

The position of an AUV can be determined by using hydroacoustic networks providing range measurements from known locations. A long baseline (LBL) network consists of several transducers mounted on the sea bed, and the AUV carries one transducer. An alternative is a short baseline (SBL) network where the array of transducers is mounted under a surface vessel and an AUV carrying one transducer. Finally, ultrashort baseline (USBL) systems where the transducers are fitted inside a small apparatus mounted under the surface vessel can be used. The hydroacoustic networks provide the user with range measurements from known locations, which can be used to compute the AUV's North-East-Down (NED) positions. The geometry and number of transducers highly impact the position accuracy (Vickery, 1998).

1.2. Loosely-coupled integration algorithms for hydroacoustic-aided inertial navigation

The most straightforward HPR/INS integration scheme is loosely coupled where the NED positions from a hydroacoustic reference system are used as aiding through an ESKF (Fossen, 2021). This solution is shown in Fig. 1 where the hydroacoustic reference system is referred to as the low-rate aiding sensor, typically operating at 1 Hz. The accelerometers and angular rate sensors (ARS) of the inertial measurement unit (IMU) work at a high rate (typically 100 to 1000 Hz). The acceleration measurements are integrated twice, and the ARS measurements are integrated once to obtain positions and orientation, respectively. However, the estimates will drift due to sensor biases, misalignments, and temperature variations. The drift is compensated by the ESKF, which is designed to estimate the accelerometer and ARS biases.

This article describes how a loosely-coupled strapdown INS aided by hydroacoustic position measurements can be implemented in an embedded computer system onboard an AUV (see Fig. 4). Unfortunately, the hydroacoustic position measurements will be delayed when transmitted in water. Sound travels about 1400 to 1550 meters per second in seawater. Consequently, for an AUV operating at a 1500 m range, the position measurements will be delayed by approximately one second. Hence, the ESKF must be designed to compensate for time delays. An alternative that adds complexity is to apply a tightly-coupled navigation filter where range measurements are used to aid the INS. In this approach, the unknown water speed of each range measurement must be included as additional states in the Kalman filter (Batista, 2015; Batista et al., 2010; Hegrenæs et al., 2009; Morgado et al., 2011; Stovner, 2018; Stovner et al., 2016).

A comparative study of loosely- and tightly-coupled INS architectures are presented by Falco et al. (2017) where it is concluded that the tightly-coupled algorithms provide better estimates of position and orientation than a loosely integrated system. However, this article's scope is design simplicity and low cost, which can be achieved using a loosely-coupled ESKF with time-delay compensation. Focus is placed on accelerometer and gyro bias estimation, essential for long-endurance AUV applications.

1.3. Scope and contributions of the article

The article is written as a tutorial to assist engineers who want to implement underwater hydroacoustic navigation algorithms in the navigation computer onboard the vehicle. The goal is to make the ESKF understandable to beginners and experts by concentrating on the practical aspects of strapdown INS. Important contributions are:

- Development of a loosely-coupled ESKF for underwater vehicle inertial navigation aided by slow hydroacoustic position measurements. The fast measurements are used to propagate the INS. A significant contribution is the accurate estimation of the specific force and angular rate biases and the ESKF feedback modification (reset) such that the INS can be used in long-endurance missions.
- The MEKF for attitude estimation (Markley & Crassidis, 2014) is extended to a complete INS solution aided by time-delayed positioning measurements. The unit quaternion implementation removes the Euler angle singularity and is more numerically efficient than the Euler angle representation.
- Matlab scripts for implementing the INS are made available from the GitHub repository of the Marine Systems Simulator (MSS) (Fossen & Perez, 2004). The m-files are also compatible with the scientific programming language (GNU Octave, 2023), which is free software under the terms of the GNU General Public License.
- A case study shows the strapdown INS's performance with and without time-delayed position measurements when applied to a small AUV.

1.4. Organization of the article

The rest of the article is organized as follows. Section 2 discusses the inertial measurement equations and lever-arm compensation. In Section 3, the strapdown INS equations are presented, while Section 4 derives the unit quaternion ESKF with reset functionality (feedback filter). Section 5 discusses methods for time-delay estimation and a modification of the ESKF for time-delayed hydroacoustic positioning measurements. Finally, an AUV case study is included in Section 6, while the concluding remarks are drawn in Section 7.

2. Inertial measurements and lever-arm compensation

For local navigation, two reference frames are introduced:

- $\{b\}$ Body-fixed frame with coordinate origin CO.
- $\{n\}$ North-East-Down (NED) tangent plane of the geoid acting as an approximative inertial reference frame.

Applications using the reference frames $\{b\}$ and $\{n\}$ are also referred to as "flat-Earth navigation". The NED position vector will be accurate to a smaller geographical area, typically $10\ \text{km} \times 10\ \text{km}$. Let,

$\boldsymbol{p}^n_{nb}$	Position of the CO w.r.t. $\{n\}$ expressed in $\{n\}$
$\boldsymbol{v}^n_{nb}$	Linear velocity of the CO w.r.t. $\{n\}$ expressed in $\{n\}$
$\boldsymbol{\omega}^b_{nb}$	Angular velocity of $\{b\}$ w.r.t. $\{n\}$ expressed in $\{b\}$
$\boldsymbol{\Theta}_{nb}$	Euler angles from $\{b\}$ to $\{n\}$
$\boldsymbol{q}^n_b$	Unit quaternion from $\{b\}$ to $\{n\}$

An INS uses a computer, a three-axis accelerometer, and a three-axis ARS to continuously calculate by dead reckoning the position vector $\boldsymbol{p}^n_{nb} = [x^n, y^n, z^n]^\top$, linear velocity vector $\boldsymbol{v}^n_{nb} = [\dot{x}^n, \dot{y}^n, \dot{z}^n]^\top$ and attitude (unit quaternion $\boldsymbol{q}^n_b$ and/or the roll, pitch, and yaw angles $\boldsymbol{\Theta}_{nb} = [\phi, \theta, \psi]^\top$) of a moving vehicle without the need for external reference signals. The key sensory component is the IMU, which is composed of the following sensors:

- *Three-axis accelerometer*
- *Three-axis ARS*
- *Three-axis magnetometer*

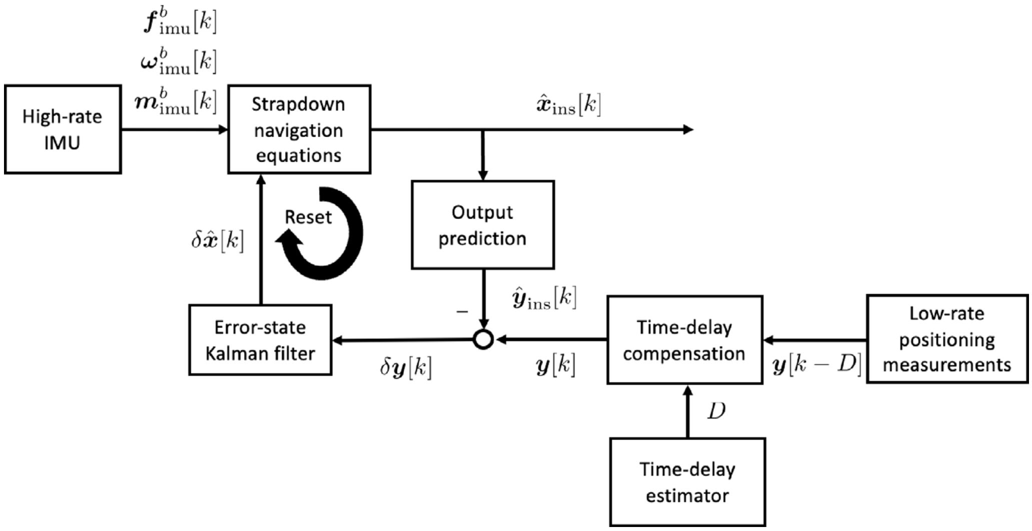

$\boldsymbol{f}^b_{\mathrm{imu}}[k]$
$\boldsymbol{\omega}^b_{\mathrm{imu}}[k]$
$\boldsymbol{m}^b_{\mathrm{imu}}[k]$
High-rate IMU
Strapdown navigation equations
$\hat{\boldsymbol{x}}_{\mathrm{ins}}[k]$
Output prediction
Reset
$\delta\hat{\boldsymbol{x}}[k]$
$\hat{\boldsymbol{y}}_{\mathrm{ins}}[k]$
−
Error-state Kalman filter
$\delta\boldsymbol{y}[k]$
$\boldsymbol{y}[k]$
Time-delay compensation
$\boldsymbol{y}[k-D]$
Low-rate positioning measurements
D
Time-delay estimator

Fig. 1. Loosely-coupled error-state feedback Kalman filter for INS. The strapdown navigation equations are propagated to obtain the estimate $\hat{\boldsymbol{x}}_{\mathrm{ins}}[k]$. The Kalman filter error-state estimate $\delta\hat{\boldsymbol{x}}[k]$ is fed back (reset mechanism) to ensure that $\hat{\boldsymbol{x}}_{\mathrm{ins}}[k] \to \boldsymbol{x}[k]$.

The IMU is mounted onboard the vehicle in a body-fixed measurement frame $\{m_{\mathrm{I}}\}$ with coordinate origin CM_{I} located at

$$\boldsymbol{r}^b_{bm_{\mathrm{I}}} = [x_{m_{\mathrm{I}}},\, y_{m_{\mathrm{I}}},\, z_{m_{\mathrm{I}}}]^{\top} \tag{1}$$

with respect to the $\{b\}$ frame coordinate origin CO. It is assumed that the axes of $\{m_{\mathrm{I}}\}$ and $\{b\}$ point in the same directions. Since the IMU is rigidly attached to the body and a lightweight digital computer propagates the strapdown INS equations, this is called a *strapdown system*. Thus the need for a mechanical-gimbal system is eliminated.

2.1. Attitude rate sensors

The classic ARS is a gyro, a spinning wheel that utilizes conservation of momentum to detect rotation. ARS based on MEMS dominate for low- and medium-cost applications (Barbour & Schmidt, 1998). For strapdown applications, optical gyros such as ring-laser gyros (RLG) and fiber-optic gyros (FOG) have been used for some time. They are also expected to be the standard for high-accuracy strapdown INS for the foreseeable future.

The IMU measurement equation for a three-axis ARS is

$$\boldsymbol{\omega}^b_{\mathrm{imu}} = \boldsymbol{\omega}^b_{nb} + \boldsymbol{b}^b_{\mathrm{ars}} + \boldsymbol{w}^b_{\mathrm{ars}} \tag{2}$$

$$\dot{\boldsymbol{b}}^b_{\mathrm{ars}} = \boldsymbol{w}^b_{b,\,\mathrm{ars}} \tag{3}$$

where the ARS bias vector is denoted as $\boldsymbol{b}^b_{\mathrm{ars}}$. Additive zero-mean Gaussian white noise terms $\boldsymbol{w}^b_{\mathrm{ars}}$ and $\boldsymbol{w}^b_{b,\,\mathrm{ars}}$ are used to model the measurement and bias noise, respectively. It is necessary to estimate $\boldsymbol{b}^b_{\mathrm{ars}}$ in the Kalman filter since the ARS bias will grow over time. The ARS measurement (2) is only valid for low-speed applications such as an AUV since it assumes that $\{n\}$ is nonrotating, that is $\boldsymbol{\omega}^b_{ib} \approx \boldsymbol{\omega}^b_{nb}$ where the subscript $\{i\}$ denotes the inertial frame. For terrestrial navigation, the Earth's rotation will affect the results, and it is necessary to use an inertial frame instead of the approximate inertial frame $\{n\}$.

2.2. Accelerometers

The IMU accelerometer is a device that measure three-axis specific force, i.e., non-gravitational force per unit mass

$$\boldsymbol{f}^b_{\mathrm{imu}} = \boldsymbol{R}^{\top}_q(\boldsymbol{q}^n_b)(\boldsymbol{a}^n_{nm_{\mathrm{I}}} - \boldsymbol{g}^n) + \boldsymbol{b}^b_{\mathrm{acc}} + \boldsymbol{w}^b_{\mathrm{acc}} \tag{4}$$

$$\dot{\boldsymbol{b}}^b_{\mathrm{acc}} = \boldsymbol{w}^b_{b,\,\mathrm{acc}} \tag{5}$$

where $\boldsymbol{g}^n = [0,\, 0,\, g]^{\top}$ is the gravity vector, g is the acceleration of gravity, and $\boldsymbol{R}(\boldsymbol{q}^n_b)$ is the unit quaternion rotation matrix from $\{b\}$ to $\{n\}$ given by Fossen (2021)

$$\boldsymbol{R}_q(\boldsymbol{q}^n_b) = \begin{bmatrix} 1-2(\varepsilon_2^2+\varepsilon_3^2) & 2(\varepsilon_1\varepsilon_2-\varepsilon_3\eta) & 2(\varepsilon_1\varepsilon_3+\varepsilon_2\eta) \\ 2(\varepsilon_1\varepsilon_2+\varepsilon_3\eta) & 1-2(\varepsilon_1^2+\varepsilon_3^2) & 2(\varepsilon_2\varepsilon_3-\varepsilon_1\eta) \\ 2(\varepsilon_1\varepsilon_3-\varepsilon_2\eta) & 2(\varepsilon_2\varepsilon_3+\varepsilon_1\eta) & 1-2(\varepsilon_1^2+\varepsilon_2^2) \end{bmatrix} \tag{6}$$

The vector $\boldsymbol{q}^n_b = [\eta, \varepsilon_1, \varepsilon_2, \varepsilon_3]^{\top}$ defines the unit quaternion. The accelerometer bias is denoted as $\boldsymbol{b}^b_{\mathrm{acc}}$, while $\boldsymbol{w}^b_{\mathrm{acc}}$ and $\boldsymbol{w}^b_{b,\,\mathrm{acc}}$ are additive Gaussian white measurement and bias noise, respectively.

2.3. Magnetometer

The magnetic field strength can be measured by a three-axis magnetometer, usually included in the sensor suite of commercial available IMUs. Mathematically this can be expressed as

$$\boldsymbol{m}^b_{\mathrm{imu}} = \boldsymbol{R}^{\top}_q(\boldsymbol{q}^n_b)\boldsymbol{m}^n + \boldsymbol{w}^b_{\mathrm{mag}} \tag{7}$$

where $\boldsymbol{m}^b_{\mathrm{imu}}$ is the IMU measurement, $\boldsymbol{m}^n$ is the strength and direction of Earth's magnetic field expressed in $\{n\}$ and $\boldsymbol{w}^b_{\mathrm{mag}}$ is additive zero-mean white noise expressed in $\{b\}$.

2.4. IMU lever-arm compensation

Instead of transforming the IMU measurements $\boldsymbol{f}^b_{\mathrm{imu}}$, $\boldsymbol{\omega}^b_{\mathrm{imu}}$, and $\boldsymbol{m}^b_{\mathrm{imu}}$ to the coordinate origin CO, the state estimator is formulated in the origin CM_{I}. Hence, the state estimates $\hat{\boldsymbol{v}}^b_{nm_{\mathrm{I}}}$ and $\hat{\boldsymbol{\omega}}^b_{nm_{\mathrm{I}}}$ (output of the Kalman filter) must be transformed to the CO using the lever-arm vector $\boldsymbol{r}^b_{bm_I}$ from the CO to the CM_{I} according to

$$\hat{\boldsymbol{v}}^b_{nb} = \hat{\boldsymbol{v}}^b_{nm_{\mathrm{I}}} - \hat{\boldsymbol{\omega}}^b_{nm_{\mathrm{I}}} \times \boldsymbol{r}^b_{bm_I} \tag{8}$$

$$\hat{\boldsymbol{\omega}}^b_{nb} = \hat{\boldsymbol{\omega}}^b_{nm_{\mathrm{I}}} \tag{9}$$

The motivation for this is that it is not straightforward to transform the IMU-specific force measurement $\boldsymbol{f}^b_{\mathrm{imu}}$ to the CO and express the state estimator in the CO. This is seen from

$$\boldsymbol{f}^b_{nb} = \boldsymbol{f}^b_{nm_{\mathrm{I}}} + \dot{\boldsymbol{\omega}}^b_{nm_{\mathrm{I}}} \times \boldsymbol{r}^b_{m_{\mathrm{I}}b} + \boldsymbol{\omega}^b_{nm_{\mathrm{I}}} \times (\boldsymbol{\omega}^b_{nm_{\mathrm{I}}} \times \boldsymbol{r}^b_{m_{\mathrm{I}}b}) \tag{10}$$

where $\boldsymbol{r}^b_{m_{\mathrm{I}}b} = -\boldsymbol{r}^b_{bm_{\mathrm{I}}}$. The angular acceleration vector $\dot{\boldsymbol{\omega}}^b_{nm_{\mathrm{I}}}$ in (10) requires that the angular rate $\boldsymbol{\omega}^b_{nm_{\mathrm{I}}} = \boldsymbol{\omega}^b_{\mathrm{imu}} - \boldsymbol{b}^b_{\mathrm{ars}}$ is numerical differentiated. Since this signal is not available as a direct measurement, it is recommended to use $\{m_{\mathrm{I}}\}$ as the reference frame in combination with the transformations (8)–(9) when implementing the ESKF.

3. Strapdown INS equations

Indirect filtering implies that the Kalman filter is formulated as an ESKF. In this context, the INS state $\boldsymbol{x}_{\text{ins}}$ and error state $\delta\boldsymbol{x}$ are related to the true state $\boldsymbol{x}$ by

$$\boldsymbol{x} = \boldsymbol{x}_{\text{ins}} + \delta\boldsymbol{x} \tag{11}$$

The estimate of the state vector is denoted by $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}_{\text{ins}} + \delta\hat{\boldsymbol{x}}$. The ESKF in Section 4 computes the estimate $\delta\hat{\boldsymbol{x}}$, while $\hat{\boldsymbol{x}}_{\text{ins}}$ is obtained by propagating a 16-state system known as the *strapdown INS equations*

$$\boxed{\begin{aligned}
\dot{\hat{\boldsymbol{p}}}^n_{\text{ins}} &= \hat{\boldsymbol{v}}^n_{\text{ins}} &&\qquad (12)\\
\dot{\hat{\boldsymbol{v}}}^n_{\text{ins}} &= \boldsymbol{R}_q(\hat{\boldsymbol{q}}_{\text{ins}})\boldsymbol{f}^b_{\text{ins}} + \boldsymbol{g}^n &&\qquad (13)\\
\dot{\hat{\boldsymbol{b}}}^b_{\text{ins, acc}} &= \boldsymbol{0} &&\qquad (14)\\
\dot{\hat{\boldsymbol{q}}}_{\text{ins}} &= \boldsymbol{T}_q(\hat{\boldsymbol{q}}_{\text{ins}})\boldsymbol{\omega}^b_{\text{ins}} &&\qquad (15)\\
\dot{\hat{\boldsymbol{b}}}^b_{\text{ins, ars}} &= \boldsymbol{0} &&\qquad (16)
\end{aligned}}$$

where $\hat{\boldsymbol{q}}_{\text{ins}} \equiv \hat{\boldsymbol{q}}^n_b = [\hat{\eta}, \hat{\varepsilon}_1, \hat{\varepsilon}_2, \hat{\varepsilon}_3]^\top$ denote the unit quaternion. The INS position $\hat{\boldsymbol{p}}^n_{\text{ins}}$ and linear velocity $\hat{\boldsymbol{v}}^n_{\text{ins}}$ of the origin CM_I with respect to $\{n\}$ are expressed in $\{n\}$. The inertial sensor bias states are denoted by $\hat{\boldsymbol{b}}^b_{\text{ins, acc}}$ and $\hat{\boldsymbol{b}}^b_{\text{ins, ars}}$. Let $\boldsymbol{S}(\boldsymbol{\lambda})$ denote the skew-symmetric matrix operator

$$\boldsymbol{S}(\boldsymbol{\lambda}) = -\boldsymbol{S}^\top(\boldsymbol{\lambda}) = \begin{bmatrix} 0 & -\lambda_3 & \lambda_2 \\ \lambda_3 & 0 & -\lambda_1 \\ -\lambda_2 & \lambda_1 & 0 \end{bmatrix} \tag{17}$$

where $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \lambda_3]^\top$. Hence, the cross product of two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ satisfy $\boldsymbol{a} \times \boldsymbol{b} = \boldsymbol{S}(\boldsymbol{a})\boldsymbol{b}$, and Fossen (2021)

$$\boldsymbol{T}_q(\hat{\boldsymbol{q}}_{\text{ins}}) = \frac{1}{2}\begin{bmatrix} -\hat{\boldsymbol{\varepsilon}}^\top \\ \hat{\eta}\boldsymbol{I}_3 + \boldsymbol{S}(\hat{\boldsymbol{\varepsilon}}) \end{bmatrix} \tag{18}$$

where $\hat{\boldsymbol{\varepsilon}} = [\hat{\varepsilon}_1, \hat{\varepsilon}_2, \hat{\varepsilon}_3]^\top$. The inputs, $\boldsymbol{f}^b_{\text{ins}}$ and $\boldsymbol{\omega}^b_{\text{ins}}$ to the strapdown INS Eqs. (13) and (15), respectively, are bias-compensated IMU measurements

$$\boxed{\begin{aligned}
\boldsymbol{f}^b_{\text{ins}} &:= \boldsymbol{f}^b_{\text{imu}} - \hat{\boldsymbol{b}}^b_{\text{acc ,ins}} &&\qquad (19)\\
\boldsymbol{\omega}^b_{\text{ins}} &:= \boldsymbol{\omega}^b_{\text{imu}} - \hat{\boldsymbol{b}}^b_{\text{ars, ins}} &&\qquad (20)
\end{aligned}}$$

The biases are initialized as $\hat{\boldsymbol{b}}^b_{\text{acc, ins}}[0]=\boldsymbol{0}$ and $\hat{\boldsymbol{b}}^b_{\text{ars, ins}}[0]=\boldsymbol{0}$. However, the ESKF will update the biases and remove the drift by feedback as described in Section 4.

The discrete-time Eqs. (12)–(16) can be implemented using exact discretization. Let

$$\boldsymbol{a}^n_{\text{ins}}[k] = \boldsymbol{R}_q(\hat{\boldsymbol{q}}_{\text{ins}}[k])\boldsymbol{f}^b_{\text{ins}}[k] + \boldsymbol{g}^n \tag{21}$$

denote the linear acceleration expressed in $\{n\}$. Then (12), (13), and (15) can be propagated using

$$\hat{\boldsymbol{p}}^n_{\text{ins}}[k+1] = \hat{\boldsymbol{p}}^n_{\text{ins}}[k] + h\hat{\boldsymbol{v}}^n_{\text{ins}}[k] + \frac{1}{2}h^2\boldsymbol{a}^n_{\text{ins}}[k] \tag{22}$$

$$\hat{\boldsymbol{v}}^n_{\text{ins}}[k+1] = \hat{\boldsymbol{v}}^n_{\text{ins}}[k] + h\boldsymbol{a}^n_{\text{ins}}[k] \tag{23}$$

$$\hat{\boldsymbol{q}}_{\text{ins}}[k+1] = \mathrm{e}^{\boldsymbol{T}_\omega(\boldsymbol{\omega}^b_{\text{ins}}[k])h}\,\hat{\boldsymbol{q}}_{\text{ins}}[k] \tag{24}$$

where h is the sampling time. The matrix exponential in (24) is computed by Sola (2016)

$$\boldsymbol{T}_w(\boldsymbol{\omega}^b_{\text{ins}})[k] = \frac{1}{2}\begin{bmatrix} 0 & -\boldsymbol{\omega}^b_{\text{ins}}[k]^\top \\ \boldsymbol{\omega}^b_{\text{ins}}[k] & -\boldsymbol{S}(\boldsymbol{\omega}^b_{\text{ins}}[k]) \end{bmatrix} \tag{25}$$

Finally, the unit quaternion estimate must be normalized each time the state is propagated. This is mathematically equivalent to

$$\hat{\boldsymbol{q}}_{\text{ins}}[k+1] \leftarrow \frac{\hat{\boldsymbol{q}}_{\text{ins}}[k+1]}{\hat{\boldsymbol{q}}_{\text{ins}}[k+1]^\top\hat{\boldsymbol{q}}_{\text{ins}}[k+1]} \tag{26}$$

4. Error-state Kalman filter

This section describes the ESKF for estimating the error state $\delta\boldsymbol{x}$ in (11). It is well known that estimating the biases of inertial sensors is non-trivial, particularly if the vehicle is at rest or moving with little excitation in the roll, pitch, and yaw modes. In particular, the accelerometer biases are hard to estimate for certain maneuvers. However, by monitoring the excitation level of the vehicle, a strategy could be to estimate the biases during agile maneuvers to ensure the convergence of biases. For this purpose, an online metric for observability with a threshold value can be used to trigger bias estimation.

4.1. Three-parameter attitude dynamics

The MEKF is an ESKF where orientation is parametrized by a four-dimensional *unit quaternion*. However, the *attitude error* is uniquely defined by three parameters, which is the minimal representation for the 3-DOF rotational motion of a rigid body (Crassidis et al., 2007; Markley & Crassidis, 2014). It is well known that it is impossible to represent attitude by three parameters globally. Three-parameter representations, such as the Euler angles, all have singular points. It is tempting to use the four-parameter unit quaternion in the ESKF to avoid singular points. However, the unit constraint will make the covariance matrix rank deficient. Another problem is the ESKF injection term. The standard ESKF with the additive-error injection cannot create an unbiased estimator with the unit quaternion in the estimated state, as the additive-error injection would also violate the quaternion unit constraint. To overcome these difficulties, the MEKF formalism is used. The main idea of the MEKF is that the unit quaternion error

$$\delta\boldsymbol{q}^n_b = \begin{bmatrix} \delta\eta \\ \delta\boldsymbol{\varepsilon} \end{bmatrix} \tag{27}$$

is parametrized using a three-parameter attitude representation where $\delta\eta$ and $\delta\boldsymbol{\varepsilon}$ are the real and imaginary components of the unit quaternion $\delta\boldsymbol{q}^n_b$. There are many candidates for the representation of the unit quaternion error. A frequently used representation is the *Rodrigues parameters* (Gibbs vector)

$$\delta\boldsymbol{a}_g = \frac{\delta\boldsymbol{\varepsilon}}{\delta\eta}, \qquad \delta\boldsymbol{q}^n_b = \frac{1}{\sqrt{1 + \delta\boldsymbol{a}_g^\top\delta\boldsymbol{a}_g}}\begin{bmatrix} 1 \\ \delta\boldsymbol{a}_g \end{bmatrix} \tag{28}$$

It is convenient to scale Gibbs vector by a factor 2 such that the ESKF covariance estimates will be given in radians squared, which is equivalent to angle errors using a first-order approximation. This is mathematically equivalent to

$$\delta\boldsymbol{a} := 2\delta\boldsymbol{a}_g, \qquad \delta\boldsymbol{q}^n_b = \frac{1}{\sqrt{4 + \delta\boldsymbol{a}^\top\delta\boldsymbol{a}}}\begin{bmatrix} 2 \\ \delta\boldsymbol{a} \end{bmatrix} \tag{29}$$

The differential equation for $\delta\boldsymbol{a}$ is derived in Appendix A, where it is shown that

$$\delta\dot{\boldsymbol{a}} = -\frac{1}{2}\boldsymbol{S}(\boldsymbol{\omega}^b_{nb} + \boldsymbol{\omega}^b_{\text{ins}})\delta\boldsymbol{a} + \boldsymbol{\omega}^b_{nb} - \boldsymbol{\omega}^b_{\text{ins}} \tag{30}$$

where $\boldsymbol{\omega}^b_{\text{ins}} := \boldsymbol{\omega}^b_{\text{imu}} - \hat{\boldsymbol{b}}^b_{\text{ars}}$ is the bias-compensated IMU measurement. Since $\delta\boldsymbol{\omega}^b_{nb} = \boldsymbol{\omega}^b_{nb} - \boldsymbol{\omega}^b_{\text{ins}}$, application of the IMU measurement Eq. (2) gives

$$\begin{aligned}
\boldsymbol{\omega}^b_{\text{ins}} &:= \boldsymbol{\omega}^b_{\text{imu}} - \hat{\boldsymbol{b}}^b_{\text{ars}}\\
&= (\boldsymbol{\omega}^b_{nb} + \boldsymbol{b}^b_{\text{ars}} + \boldsymbol{w}^b_{\text{ars}}) - \hat{\boldsymbol{b}}^b_{\text{ars}}\\
&= \boldsymbol{\omega}^b_{nb} + \delta\boldsymbol{b}^b_{\text{ars}} + \boldsymbol{w}^b_{\text{ars}}
\end{aligned} \tag{31}$$

From this it follows that

$$\delta\dot{\boldsymbol{a}} = -\frac{1}{2}\boldsymbol{S}(2\boldsymbol{\omega}^b_{\text{ins}} + \delta\boldsymbol{\omega}^b_{nb})\delta\boldsymbol{a} - \delta\boldsymbol{b}^b_{\text{ars}} - \boldsymbol{w}^b_{\text{ars}} \tag{32}$$

Finally, neglecting the second-order term $\boldsymbol{S}(\delta\boldsymbol{\omega}^b_{nb})\delta\boldsymbol{a}$ in (32) gives

$$\boxed{\begin{aligned}
\delta\dot{\boldsymbol{a}} &\approx -\boldsymbol{S}(\boldsymbol{\omega}^b_{\text{ins}})\delta\boldsymbol{a} - \delta\boldsymbol{b}^b_{\text{ars}} - \boldsymbol{w}^b_{\text{ars}} &&\qquad (33)\\
\delta\dot{\boldsymbol{b}}^b_{\text{ars}} &= -\frac{1}{T_{\text{ars}}}\delta\boldsymbol{b}^b_{\text{ars}} + \boldsymbol{w}^b_{b,\text{ ars}} &&\qquad (34)
\end{aligned}}$$

Note that the ARS bias is modeled as a first-order system with user-specified time constant $T_{\rm ars}$, while $\boldsymbol{w}^b_{b,\,\rm ars}$ is Gaussian white noise. The exponential convergence of $\delta\boldsymbol{b}^b_{\rm ars}$ to zero is essential during sensor failure (dead reckoning).

4.2. Translational motion error dynamics

Consider the position and velocity error states $\delta\boldsymbol{p}^n_{nm_{\rm I}} = \boldsymbol{p}^n_{nm_{\rm I}} - \hat{\boldsymbol{p}}^n_{\rm ins}$ and $\delta\boldsymbol{v}^n_{nm_{\rm I}} = \boldsymbol{v}^n_{nm_{\rm I}} - \hat{\boldsymbol{v}}^n_{\rm ins}$. The vehicle's linear acceleration is obtained from (4), which yields

$$\boldsymbol{a}^n_{nm_{\rm I}} = \boldsymbol{R}_q(\boldsymbol{q}^n_b)(\boldsymbol{f}^b_{\rm imu} - \hat{\boldsymbol{b}}^b_{\rm acc} - \delta\boldsymbol{b}^b_{\rm acc} - \boldsymbol{w}^b_{\rm acc}) + \boldsymbol{g}^n \tag{35}$$

Then the translational motion errors dynamics becomes

$$\delta\dot{\boldsymbol{p}}^n_{nm_{\rm I}} = \delta\boldsymbol{v}^n_{nm_{\rm I}} \tag{36}$$

$$\begin{aligned}
\delta\dot{\boldsymbol{v}}^n_{nm_{\rm I}} &= \boldsymbol{R}_q(\boldsymbol{q}^n_b)(\boldsymbol{f}^b_{\rm imu} - \hat{\boldsymbol{b}}^b_{\rm acc} - \delta\boldsymbol{b}^b_{\rm acc} - \boldsymbol{w}^b_{\rm acc}) + \boldsymbol{g}^n \\
&\quad - \left(\boldsymbol{R}_q(\hat{\boldsymbol{q}}^n_b)(\boldsymbol{f}^b_{\rm imu} - \hat{\boldsymbol{b}}^b_{\rm acc}) + \boldsymbol{g}^n\right) \\
&= \left(\boldsymbol{R}_q(\boldsymbol{q}^n_b) - \boldsymbol{R}_q(\hat{\boldsymbol{q}}^n_b)\right)(\boldsymbol{f}^b_{\rm imu} - \hat{\boldsymbol{b}}^b_{\rm acc}) \\
&\quad - \boldsymbol{R}_q(\boldsymbol{q}^n_b)(\delta\boldsymbol{b}^b_{\rm acc} + \boldsymbol{w}^b_{\rm acc}) \\
&\approx \boldsymbol{R}_q(\hat{\boldsymbol{q}}_{\rm ins})\boldsymbol{S}(\delta\boldsymbol{a})(\boldsymbol{f}^b_{\rm imu} - \hat{\boldsymbol{b}}^b_{\rm acc}) \\
&\quad - \boldsymbol{R}_q(\hat{\boldsymbol{q}}_{\rm ins})\left(\boldsymbol{I}_3 + \boldsymbol{S}(\delta\boldsymbol{a})\right)(\delta\boldsymbol{b}^b_{\rm acc} + \boldsymbol{w}^b_{\rm acc})
\end{aligned} \tag{37}$$

where $\hat{\boldsymbol{q}}_{\rm ins} \equiv \hat{\boldsymbol{q}}^n_b$ and

$$\begin{aligned}
\boldsymbol{R}_q(\boldsymbol{q}^n_b) &= \boldsymbol{R}_q(\hat{\boldsymbol{q}}_{\rm ins} \otimes \delta\boldsymbol{q}^n_b) \\
&= \boldsymbol{R}_q(\hat{\boldsymbol{q}}_{\rm ins})\boldsymbol{R}_q(\delta\boldsymbol{q}^n_b) \\
&\approx \boldsymbol{R}_q(\hat{\boldsymbol{q}}_{\rm ins})\left(\boldsymbol{I}_3 + \boldsymbol{S}(\delta\boldsymbol{a})\right)
\end{aligned} \tag{38}$$

have been applied (see Appendix A). The second-order cross-product term $\boldsymbol{S}(\delta\boldsymbol{a})\delta\boldsymbol{b}^b_{\rm acc}$ in (37) is neglected such that

$$\boxed{\begin{aligned}
\delta\dot{\boldsymbol{p}}^n_{nm_{\rm I}} &= \delta\boldsymbol{v}^n_{nm_{\rm I}} && (39)\\
\delta\dot{\boldsymbol{v}}^n_{nm_{\rm I}} &\approx -\boldsymbol{R}_q(\hat{\boldsymbol{q}}_{\rm ins})\boldsymbol{S}(\boldsymbol{f}^b_{\rm ins})\delta\boldsymbol{a} \\
&\quad - \boldsymbol{R}_q(\hat{\boldsymbol{q}}_{\rm ins})(\delta\boldsymbol{b}^b_{\rm acc} + \boldsymbol{w}^b_{\rm acc}) && (40)\\
\delta\dot{\boldsymbol{b}}^b_{\rm acc} &= -\frac{1}{T_{\rm acc}}\delta\boldsymbol{b}^b_{\rm acc} + \boldsymbol{w}^b_{b,\,\rm acc} && (41)
\end{aligned}}$$

where $\boldsymbol{f}^b_{\rm ins} := \boldsymbol{f}^b_{\rm imu} - \hat{\boldsymbol{b}}^b_{\rm acc}$ is the bias-compensated IMU measurement. The bias is modeled as a first-order system (41) with time constant $T_{\rm acc}$ and Gaussian white noise $\boldsymbol{w}^b_{b,\,\rm acc}$ as driving input. The exponential convergence of the bias $\delta\boldsymbol{b}^b_{\rm acc}$ to zero is important during sensor failure (dead reckoning).

4.3. Kalman filter state-space model

The goal is to estimate the discrete-time state vector $\boldsymbol{x}[k]$ given by (11). The state estimate is denoted by

$$\hat{\boldsymbol{x}}[k] = \hat{\boldsymbol{x}}_{\rm ins}[k] + \delta\hat{\boldsymbol{x}}[k] \tag{42}$$

where the strapdown INS equations in Section 3 are used to compute $\hat{\boldsymbol{x}}_{\rm ins}[k]$. The ESKF estimates the errors state vector $\delta\boldsymbol{x}$ using the discrete-time state-space model

$$\delta\boldsymbol{x}[k+1] = \boldsymbol{A}_d[k]\delta\boldsymbol{x}[k] + \boldsymbol{E}_d[k]\boldsymbol{w}[k] \tag{43}$$

$$\delta\boldsymbol{y}[k] = \boldsymbol{C}_d[k]\delta\boldsymbol{x}[k] + \boldsymbol{\varepsilon}[k] \tag{44}$$

where $\boldsymbol{A}_d[k]$, $\boldsymbol{C}_d[k]$ and $\boldsymbol{E}_d[k]$ are the discrete-time system matrices, and $\boldsymbol{w}[k]$ and $\boldsymbol{\varepsilon}[k]$ are Gaussian white noise vectors. The ESKF equations are given by Table 1. This is a special version of the ESKF known as a *feedback filter* due to the reset function. The error-state model derived in Sections 4.1 and 4.2 is a 15-states nonlinear system

$$\delta\dot{\boldsymbol{x}} = \boldsymbol{f}(\delta\boldsymbol{x}, \boldsymbol{w}) \tag{45}$$

$$\delta\boldsymbol{y} = \boldsymbol{h}(\delta\boldsymbol{x}) + \boldsymbol{\varepsilon} \tag{46}$$

with state and process noise vectors

$$\delta\boldsymbol{x} = [(\delta\boldsymbol{p}^n_{nm_{\rm I}})^\top, (\delta\boldsymbol{v}^n_{nm_{\rm I}})^\top, (\delta\boldsymbol{b}^b_{\rm acc})^\top, \delta\boldsymbol{a}^\top, (\delta\boldsymbol{b}^b_{\rm ars})^\top]^\top \tag{47}$$

$$\boldsymbol{w} = [(\boldsymbol{w}^b_{\rm acc})^\top, (\boldsymbol{w}^b_{b,\,\rm acc})^\top, (\boldsymbol{w}^b_{\rm ars})^\top, (\boldsymbol{w}^b_{b,\,\rm ars})^\top]^\top \tag{48}$$

Note that $\delta\boldsymbol{a}$ replaces the unit quaternion as an error state in (47). Consequently, linearization of (33)–(34) and (39)–(41) about $\delta\boldsymbol{x}[k] = \boldsymbol{0}$ and $\boldsymbol{w}[k] = \boldsymbol{0}$, and application of Euler's integration method gives

$$\boldsymbol{A}_d[k] \approx \boldsymbol{I}_{15} + h\left.\frac{\partial\boldsymbol{f}(\delta\boldsymbol{x}[k], \boldsymbol{w}[k])}{\partial\delta\boldsymbol{x}[k]}\right|_{\delta\boldsymbol{x}[k]=\boldsymbol{0},\,\boldsymbol{w}[k]=\boldsymbol{0}} \tag{49}$$

$$\boldsymbol{E}_d[k] \approx h\left.\frac{\partial\boldsymbol{f}(\delta\boldsymbol{x}[k], \boldsymbol{w}[k])}{\partial\boldsymbol{w}[k]}\right|_{\delta\boldsymbol{x}[k]=\boldsymbol{0},\,\boldsymbol{w}[k]=\boldsymbol{0}} \tag{50}$$

where h is the sampling time and

$$\begin{aligned}
&\left.\frac{\partial\boldsymbol{f}(\delta\boldsymbol{x}[k], \boldsymbol{w}[k])}{\partial\delta\boldsymbol{x}[k]}\right|_{\delta\boldsymbol{x}[k]=\boldsymbol{0},\,\boldsymbol{w}[k]=\boldsymbol{0}} = \\
&\begin{bmatrix}
\boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\
\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & -\boldsymbol{R}_q(\hat{\boldsymbol{q}}_{\rm ins}[k]) & -\boldsymbol{R}_q(\hat{\boldsymbol{q}}_{\rm ins}[k])\boldsymbol{S}(\boldsymbol{f}^b_{\rm ins}[k]) & \boldsymbol{0}_{3\times3} \\
\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & -\frac{1}{T_{\rm acc}}\boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\
\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & -\boldsymbol{S}(\boldsymbol{\omega}^b_{\rm ins}[k]) & -\boldsymbol{I}_3 \\
\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & -\frac{1}{T_{\rm ars}}\boldsymbol{I}_3
\end{bmatrix}
\end{aligned} \tag{51}$$

and

$$\begin{aligned}
&\left.\frac{\partial\boldsymbol{f}(\delta\boldsymbol{x}[k], \boldsymbol{w}[k])}{\partial\boldsymbol{w}[k]}\right|_{\delta\boldsymbol{x}[k]=\boldsymbol{0},\,\boldsymbol{w}[k]=\boldsymbol{0}} = \\
&\begin{bmatrix}
\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\
-\boldsymbol{R}_q(\hat{\boldsymbol{q}}_{\rm ins}[k]) & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\
\boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\
\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & -\boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} \\
\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3
\end{bmatrix}
\end{aligned} \tag{52}$$

Unfortunately, the error state estimates $\delta\hat{\boldsymbol{x}}[k]$ will grow too large for long-time applications. Hence, a reset technique (feedback filter) is used to regulate the state $\delta\hat{\boldsymbol{x}}[k]$ to zero such that $\hat{\boldsymbol{x}}_{\rm ins}[k] \to \hat{\boldsymbol{x}}[k]$.

Algorithm 1 (*Feedback filter*)**.** *The reset for the ESKF is implemented in two steps:*

Step 1. *After every slow position measurement, the INS states are corrected by setting the error-state vector $\delta\hat{\boldsymbol{x}}[k]$ to zero. This is mathematically equivalent to*

$$\hat{\boldsymbol{x}}_{ins}[k] \leftarrow \hat{\boldsymbol{x}}_{ins}[k] + \delta\hat{\boldsymbol{x}}[k] \tag{53}$$

Step 2. *ensures that the error-state vector will be zero before applying fast IMU measurements. Therefore, the state predictor of the filter becomes redundant since*

$$\delta\hat{\boldsymbol{x}}^-[k+1] = \boldsymbol{A}_d[k]\boldsymbol{0} \equiv \boldsymbol{0} \tag{54}$$

Consequently,

$$\begin{aligned}
\delta\hat{\boldsymbol{x}}[k] &= \delta\hat{\boldsymbol{x}}^-[k] + \boldsymbol{K}[k]\big(\delta\boldsymbol{y}[k] - \boldsymbol{C}_d[k]\delta\hat{\boldsymbol{x}}^-[k]\big) \\
&\overset{reset}{=} \boldsymbol{K}[k]\big(\boldsymbol{y}[k] - \boldsymbol{C}_d[k]\hat{\boldsymbol{x}}_{ins}[k]\big)
\end{aligned} \tag{55}$$

Note that $\boldsymbol{A}_d[k]$ is still necessary to compute since it is used in the ESKF covariance prediction step; see Table 1.

4.4. Measurement equations

The $\boldsymbol{C}_d[k]$ matrix in the measurement model (44) depends on the sensor suit. For underwater vehicles, the primary sensors used for aided INS are

$\delta\boldsymbol{y}_h[k]$	- Hydroacoustic position vector
$\delta\boldsymbol{y}_p[k]$	- Pressure meter
$\delta\boldsymbol{y}_v[k]$	- DVL (optionally)
$\delta\boldsymbol{y}_g[k]$	- Gravity reference vector
$\delta\boldsymbol{y}_m[k]$	- Magnetometer reference vector

(56)

The hydroacoustic position measurement relates to the error states as follows

$$\boxed{\begin{aligned}\delta\boldsymbol{y}_h[k] &= (\boldsymbol{p}^n_{nm_{\mathrm{I}}}[k] + \boldsymbol{\varepsilon}_h[k]) - \hat{\boldsymbol{p}}^n_{\mathrm{ins}}[k]\\ &= \delta\boldsymbol{p}^n_{nm_{\mathrm{I}}}[k] + \boldsymbol{\varepsilon}_h[k]\end{aligned}} \tag{57}$$

where $\boldsymbol{\varepsilon}_h[k]$ is Gaussian white measurement noise. For AUVs, a pressure meter can be used to improve the accuracy of the vertical position. This is mathematically equivalent to

$$\boxed{\delta y_p[k] = [0, 0, 1]\,\delta\boldsymbol{p}^n_{nm_{\mathrm{I}}}[k] + \varepsilon_p[k]} \tag{58}$$

where $\varepsilon_p[k]$ is Gaussian white measurement noise. The DVL measurement equation is

$$\boxed{\delta\boldsymbol{y}_v[k] = \delta\boldsymbol{v}^n_{nm_{\mathrm{I}}}[k] + \boldsymbol{\varepsilon}_v[k]} \tag{59}$$

where $\boldsymbol{\varepsilon}_v[k]$ is Gaussian white measurement noise. For underwater vehicles, the bias-compensated specific force vector $\boldsymbol{f}^b_{\mathrm{ins}} := \boldsymbol{f}^b_{\mathrm{imu}} - \hat{\boldsymbol{b}}^b_{\mathrm{acc}}$ can also be used to aid the INS. Let the reference vector $\boldsymbol{v}^n_{01} = [0, 0, 1]^\top$ expressed in $\{n\}$ denote the normalized gravity vector, pointing downwards. Then the normalized specific force and its estimate become

$$\boldsymbol{v}^b_1 = -\frac{1}{g}\boldsymbol{f}^b_{\mathrm{ins}}, \qquad \hat{\boldsymbol{v}}^b_1 = \boldsymbol{R}^\top_q(\hat{\boldsymbol{q}}^n_b)\boldsymbol{v}^n_{01} \tag{60}$$

where g is the acceleration of gravity. Since $\hat{\boldsymbol{q}}^n_b \equiv \hat{\boldsymbol{q}}_{\mathrm{ins}}$, the measurement equation for the gravity reference vector $\boldsymbol{v}^n_{01}$ takes the following form (see Appendix B)

$$\boxed{\delta\boldsymbol{y}_g[k] = \boldsymbol{S}(\boldsymbol{R}^\top_q(\hat{\boldsymbol{q}}_{\mathrm{ins}}[k])\boldsymbol{v}^n_{01})\delta\boldsymbol{a}[k] + \boldsymbol{\varepsilon}_g[k]} \tag{61}$$

where $\boldsymbol{\varepsilon}_g[k]$ is Gaussian white measurement noise. The IMU magnetometer can be treated as a second reference vector $\boldsymbol{v}^n_{02} = \boldsymbol{m}^n/\|\boldsymbol{m}^n\|$, where $\boldsymbol{m}^n$ is the strength and direction of Earth's magnetic field expressed in $\{n\}$. Hence,

$$\boldsymbol{v}^b_2 = \frac{\boldsymbol{m}^b_{\mathrm{imu}}}{\|\boldsymbol{m}^n\|}, \qquad \hat{\boldsymbol{v}}^b_2 = \boldsymbol{R}^\top_q(\hat{\boldsymbol{q}}^n_b)\boldsymbol{v}^n_{02} \tag{62}$$

Consequently, from Appendix B it follows that

$$\boxed{\delta\boldsymbol{y}_m[k] = \boldsymbol{S}(\boldsymbol{R}^\top_q(\hat{\boldsymbol{q}}_{\mathrm{ins}}[k])\boldsymbol{v}^n_{02})\delta\boldsymbol{a}[k] + \boldsymbol{\varepsilon}_m[k]} \tag{63}$$

where $\boldsymbol{\varepsilon}_m[k]$ is Gaussian white measurement noise. The resulting measurement matrix (44) corresponding to the measurements (56) becomes

$$\boldsymbol{C}_d[k] \approx \left.\frac{\partial\boldsymbol{h}(\delta\boldsymbol{x}[k])}{\partial\delta\boldsymbol{x}[k]}\right|_{\delta\boldsymbol{x}[k]=\boldsymbol{0}}$$

$$\approx \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3}\\ [0,0,1] & \boldsymbol{0}_{1\times3} & \boldsymbol{0}_{1\times3} & \boldsymbol{0}_{1\times3} & \boldsymbol{0}_{1\times3}\\ \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3}\\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{S}(\boldsymbol{R}^\top_q(\hat{\boldsymbol{q}}_{\mathrm{ins}}[k])\boldsymbol{v}^n_{01}) & \boldsymbol{0}_{3\times3}\\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{S}(\boldsymbol{R}^\top_q(\hat{\boldsymbol{q}}_{\mathrm{ins}}[k])\boldsymbol{v}^n_{02}) & \boldsymbol{0}_{3\times3} \end{bmatrix} \tag{64}$$

and the Gaussian white noise measurement vector is

$$\boldsymbol{\varepsilon}[k] = [\boldsymbol{\varepsilon}_h[k]^\top, \varepsilon_p[k], \boldsymbol{\varepsilon}_v[k]^\top, \boldsymbol{\varepsilon}_g[k]^\top, \boldsymbol{\varepsilon}_m[k]^\top]^\top \tag{65}$$

4.5. Programming code for computer implementations

The discrete-time ESKF is implemented as a while-loop where the hydroacoustic position measurements are much slower than the IMU measurements; see Fig. 2. The Matlab function *ins.m* (see Appendix C) is called with all the 11 arguments to update the ESKF estimates when a slow position measurement arrives. This activates the reset function of the ESKF, and the INS states are updated accordingly. When there are no position measurements, the function *ins.m* is called with 7 arguments to propagate the INS states using only the fast IMU measurements.

Table 1
Discrete-time error-state Kalman filter (ESKF) with reset (feedback) modification.

Initial values	$\delta\hat{\boldsymbol{x}}^-[0] = \delta\boldsymbol{x}_0$ $\hat{\boldsymbol{P}}^-[0] = \mathrm{E}[(\delta\boldsymbol{x}[0] - \delta\hat{\boldsymbol{x}}^-[0])(\delta\boldsymbol{x}[0] - \delta\hat{\boldsymbol{x}}^-[0])^\top] = \boldsymbol{P}_0$
ESKF gain	$\boldsymbol{K}[k] = \hat{\boldsymbol{P}}^-[k]\boldsymbol{C}^\top_d[k](\boldsymbol{C}_d[k]\hat{\boldsymbol{P}}^-[k]\boldsymbol{C}^\top_d[k] + \boldsymbol{R}_d[k])^{-1}$
State corrector	$\delta\hat{\boldsymbol{x}}[k] = \delta\hat{\boldsymbol{x}}^-[k] + \boldsymbol{K}[k]\big(\delta\boldsymbol{y}[k] - \boldsymbol{C}_d[k]\delta\hat{\boldsymbol{x}}^-[k]\big) \overset{\text{reset}}{=} \boldsymbol{K}[k]\left(\boldsymbol{y}[k] - \boldsymbol{C}_d[k]\hat{\boldsymbol{x}}_{\mathrm{ins}}[k]\right)$
Covariance corrector	$\hat{\boldsymbol{P}}[k] = \left(\boldsymbol{I}_n - \boldsymbol{K}[k]\boldsymbol{C}_d[k]\right)\hat{\boldsymbol{P}}^-[k]\left(\boldsymbol{I}_n - \boldsymbol{K}[k]\boldsymbol{C}_d[k]\right)^\top + \boldsymbol{K}[k]\boldsymbol{R}_d[k]\boldsymbol{K}^\top[k]$
State predictor	$\delta\hat{\boldsymbol{x}}^-[k+1] = \boldsymbol{A}_d[k]\,\delta\hat{\boldsymbol{x}}[k] \overset{\text{reset}}{=} \boldsymbol{0}$
Covariance predictor	$\delta\hat{\boldsymbol{P}}^-[k+1] = \boldsymbol{A}_d[k]\hat{\boldsymbol{P}}[k]\boldsymbol{A}^\top_d[k] + \boldsymbol{E}_d[k]\boldsymbol{Q}_d[k]\boldsymbol{E}^\top_d[k]$

where
$\boldsymbol{Q}_d[k]$, $\boldsymbol{R}_d[k]$ Covariance matrices for the process and measurement noises.
$\delta\hat{\boldsymbol{x}}^-[k]$, $\hat{\boldsymbol{P}}^-[k]$ *A priori* error state and covariance matrix estimates (before update).
$\delta\hat{\boldsymbol{x}}[k]$, $\hat{\boldsymbol{P}}[k]$ *A posteriori* error state and covariance matrix estimates (after update).

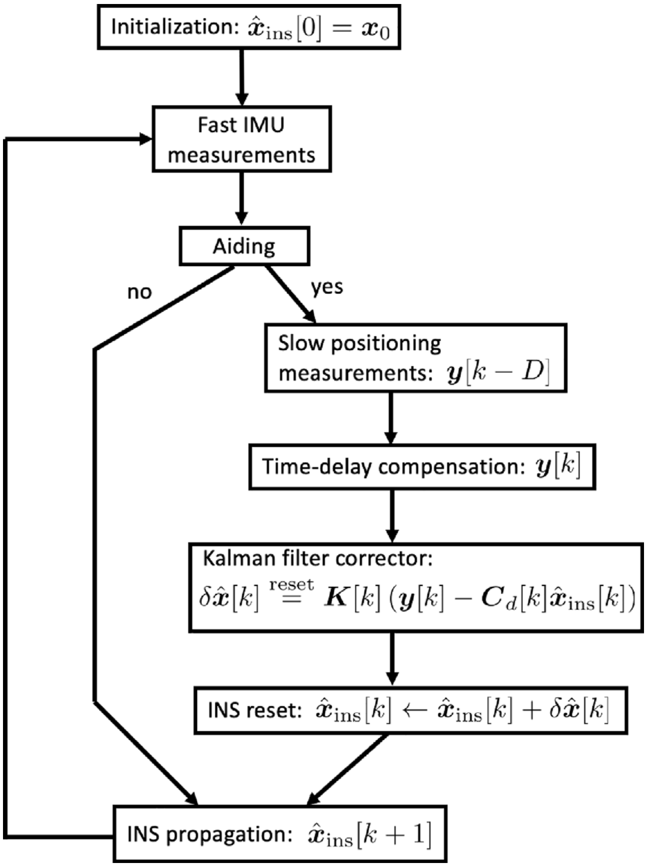

Initialization: $\hat{\boldsymbol{x}}_{\mathrm{ins}}[0] = \boldsymbol{x}_0$
Fast IMU measurements
Aiding
no
yes
Slow positioning measurements: $\boldsymbol{y}[k-D]$
Time-delay compensation: $\boldsymbol{y}[k]$
Kalman filter corrector:
$\delta\hat{\boldsymbol{x}}[k] \overset{\text{reset}}{=} \boldsymbol{K}[k]\,(\boldsymbol{y}[k] - \boldsymbol{C}_d[k]\hat{\boldsymbol{x}}_{\mathrm{ins}}[k])$
INS reset: $\hat{\boldsymbol{x}}_{\mathrm{ins}}[k] \leftarrow \hat{\boldsymbol{x}}_{\mathrm{ins}}[k] + \delta\hat{\boldsymbol{x}}[k]$
INS propagation: $\hat{\boldsymbol{x}}_{\mathrm{ins}}[k+1]$

Fig. 2. Flow chart showing the ESKF with reset mechanism. The fast IMU measurements propagate the INS strapdown navigation equations, while the slow time-delayed position measurements are used to correct the estimates.

5. Error-state Kalman filter with time-delay compensation

The feedback ESKF is shown in Fig. 1. The fast IMU measurements propagate the strapdown INS equations, while the hydroacoustic positioning measurements represent the slow corrections used to remove drift. The section describes the methods for time-delay estimation and correction.

5.1. Time-delay estimation

The hydroacoustic position measurements can be significantly delayed when transmitted in water. Sound travels about 1400 to 1550 meters per second in seawater; see Fig. 3. Hence, the time delay can be seconds for underwater vehicles operating at great depths. When

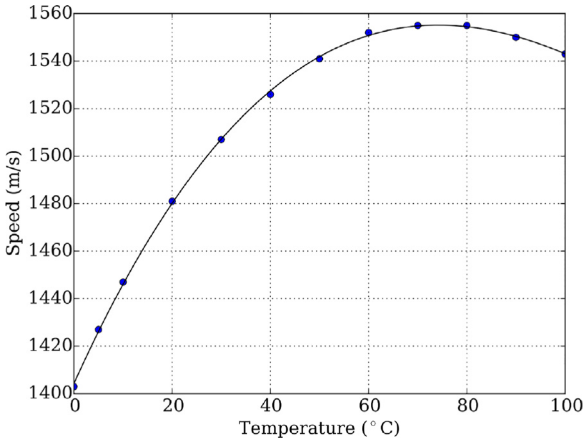


Fig. 3. Speed of sound in water as a function of temperature. Courtesy to K. Krallis, SV1XV, CC BY 1.0, https://creativecommons.org/licenses/by/1.0, Wikimedia Commons.

designing a loosely-coupled position-aided ESKF, it is necessary to have an accurate estimate of the time delay. This can be obtained by combing the non-delayed pressure measurement with the time-delayed hydroacoustic position measurement. The classical method computes the cross-correlation peak between the output and the input where the output is the delayed version of the input (Hero et al., 1998). A survey of methods for time-delay estimation, including time-delay approximation model methods, explicit time-delay parameter methods, area and moment methods, and higher-order statistics (HOS) methods, are found in Björklund and Ljung (2003).

5.2. Time-delay compensation

The ESKF must be redesigned to compensate for time-delayed hydroacoustic positioning measurements. The alternative to a loosely-coupled ESKF aided by position measurements is to apply a tightly-coupled navigation filter where range measurements are used to aid the INS (Batista, 2015; Batista et al., 2010; Hegrenæs et al., 2009; Morgado et al., 2011; Stovner, 2018; Stovner et al., 2016). However, the scope of this article is the loosely-coupled ESKF due to design simplicity.

Assume that the time-delay D has been estimated using one of the methods described in Section 5.1. Then there exist several methods for filter time-delay compensation (Comellini et al., 2020):

Filter Recalculation (FR): The FR method goes back to the time step when the delayed measurement is taken, incorporating the measurement and recomputing the entire trajectory of the state until the current step. Hence, the whole time history will be optimal. The estimation requires two filters: a principal one, which operates at a constant rate by processing fast measurements, and a second one, which is activated any time a delayed, slow measurement arrives.

Extrapolation methods: The Alexander (1991) and Larsen et al. (1998) extrapolation methods compute a correction term, which is added to the filter estimate when the delayed measurement becomes available.

State augmentation: The classical approach (Gelb, 1974) is to augment the state vector with the delayed state to incorporate the delayed measurement. This is only convenient for constant and small time delays as the dimension of the state vector is significantly increased.

Fig. 4. The Remus 100 at the Applied Underwater Robotics Laboratory (AUR-Lab) at NTNU.

The approach in this article is to extrapolate the delayed hydroacoustic position measurements $\boldsymbol{y}_h[k-D]$ to $\boldsymbol{y}_h[k]$ where D is the known time delay. The propagated states will be based on the strapdown INS Eqs. (12)–(16). The INS states and specific force measurements are stored in a table from time t_{k-D} to t_k. Let h denote the sampling time, and $N = \text{round}(D/h)$ is the total number of samples rounded downwards. Hence, the bias-compensated specific force and angular velocity vectors (fast measurements) can be computed for future samples $i = 1, 2, \dots, N$ using

$$\boldsymbol{f}^b_{\text{ins}}[i] = \boldsymbol{f}^b_{\text{imu}}[i] - \hat{\boldsymbol{b}}^b_{\text{ins, acc}}[i] \tag{66}$$

The linear acceleration vector expressed in $\{n\}$ is

$$\boldsymbol{a}^n_{\text{ins}}[i] = \boldsymbol{R}_q(\hat{\boldsymbol{q}}_{\text{ins}}[i])\boldsymbol{f}^b_{\text{ins}}[i] + \boldsymbol{g}^n \tag{67}$$

For each slow hydroacoustic measurement, initialize the state $\hat{\boldsymbol{p}}^n_h[0] = \boldsymbol{y}_h[k-D]$. Then $\boldsymbol{y}_h[k-D]$ can be extrapolated to $\boldsymbol{y}_h[k]$ using the fast acceleration measurements

$$\hat{\boldsymbol{p}}^n_h[i+1] = \hat{\boldsymbol{p}}^n_h[i] + h\hat{\boldsymbol{v}}^n_{\text{ins}}[i] + \frac{1}{2}h^2\boldsymbol{a}^n_{\text{ins}}[i] \tag{68}$$

until $\hat{\boldsymbol{y}}_h[k] = \hat{\boldsymbol{p}}^n_h[N+1]$ is reached. Consequently, the ESKF in Section 4 can be run without time-delayed measurements by replacing the unknown measurement $\boldsymbol{y}^n_h[k]$ with the estimate $\hat{\boldsymbol{y}}^n_h[k]$. The price being paid is that the INS states $(\hat{\boldsymbol{q}}_{\text{ins}}, \hat{\boldsymbol{v}}^n_{\text{ins}}, \hat{\boldsymbol{b}}^b_{\text{ins, acc}})$ and specific force measurement $\boldsymbol{f}^b_{\text{ins}}$ must be stored in a lookup table to execute (68) when a slow measurement arrives. If the time delay is 1 s and the IMU runs at 100 Hz, the table will contain 13 × 100 entries. Even at 1000 Hz, this will not be a problem for a commercial low-cost embedded computer.

Note that white noise measurement noise will also be propagated when applying (68) for extrapolation. The position and velocity terms are white, while the acceleration term contains a squared white noise signal resulting in a bias term. However, the bias term will be small when propagating the state vector only 1 to 2 s into the future. If these results are unsatisfactory, the FR or the state augmentation methods can be implemented at the price of increased computational loads.

6. AUV case study

The case study aims to demonstrate the performance of the feedback ESKF with and without time-delay compensation. The IMU is assumed to be onboard a Remus 100 AUV, as shown in Fig. 4. The specific force vector measurement is simulated using

$$\boldsymbol{f}^b_{\text{imu}} = \begin{bmatrix} a_h \cos(\psi) \\ a_h \sin(\psi) \\ a_v \end{bmatrix} - \boldsymbol{R}_q(\boldsymbol{q}^n_b)^{\top}\boldsymbol{g}^n + \boldsymbol{b}^b_{\text{acc}} + \boldsymbol{w}^b_{\text{acc}} \tag{69}$$

where $\boldsymbol{b}^b_{\text{acc}} = [0.05, 0.1, -0.1]^{\top}$ is the unknown accelerometer bias vector and $\boldsymbol{w}^b_{\text{acc}}$ is a vector of Gaussian white measurement noise. The acceleration of gravity is chosen as $\boldsymbol{g}^n = [0, 0, 9.82]^{\top}$, while the horizontal and vertical acceleration components are chosen as $a_h = 0.1$ and $a_v = 0.01$, respectively for time $t \leq 30$ s. Both acceleration components are zero for time $t > 30$ s. The ARS measurements vector is

$$\boldsymbol{\omega}^b_{\text{imu}} = \begin{bmatrix} 0.01\cos(0.2t) \\ -0.005\sin(0.1t) \\ 0.02\sin(0.1t) \end{bmatrix} + \boldsymbol{b}^b_{\text{ars}} + \boldsymbol{w}^b_{\text{ars}} \tag{70}$$

where $\boldsymbol{b}^b_{\text{ars}} = [0.05, 0.1, -0.05]^{\top}$ rad/s is the unknown ARS bias vector and $\boldsymbol{w}^b_{\text{ars}}$ is vector of Gaussian white measurement noise. The gravity

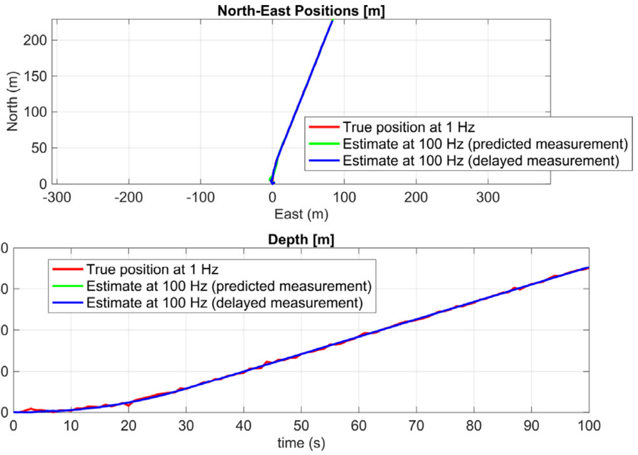


Fig. 5. Upper plot: North-East positions versus time. Lower plot: Down position (depth) versus time.

reference vector $\boldsymbol{v}^n_{01} = [0, 0, 1]^\top$ is used in (60). The second reference vector $\boldsymbol{v}^n_{02} = \boldsymbol{m}^n/\|\boldsymbol{m}^n\|$ is computed for Trondheim, Norway using the (NOAA, 2023) magnetic field calculator. This yields the magnetic field vector $\boldsymbol{m}^n = [13536.8, 1086.8, 50355.8]^\top$ nT. Hence, the time-varying IMU measurement vector becomes

$$\boldsymbol{m}^b_{\text{imu}} = \boldsymbol{R}_q(\boldsymbol{q}^n_b)^\top \boldsymbol{m}^n + \boldsymbol{w}^b_{\text{mag}} \tag{71}$$

where $\boldsymbol{w}^b_{\text{mag}}$ is vector of Gaussian white measurement noise. Finally, the AUV state vector is obtained by exact numerical integration of the differential equations

$$\dot{\boldsymbol{p}}^n = \boldsymbol{v}^n \tag{72}$$

$$\dot{\boldsymbol{v}}^n = \boldsymbol{R}_q(\boldsymbol{q}^n_b)\boldsymbol{f}^b_{\text{imu}} + \boldsymbol{g}^n \tag{73}$$

$$\dot{\boldsymbol{q}}^n_b = \boldsymbol{T}_q(\boldsymbol{q}^n_b)\boldsymbol{\omega}^b_{\text{imu}} \tag{74}$$

at 100 Hz. The fast IMU measurements are assumed to be available at 100 Hz, while the hydroacoustic position measurement frequency is chosen as 1 Hz. The time delay D is assumed to be known, and it is chosen as 1.0 s implying that 100 samples of INS data must be stored to apply the predictor (68). The following cases are shown in the plots:

True measurements: The true position measurement $\boldsymbol{y}[k]$ obtained by numerical integration of (72)–(74) driven by the IMU measurements $\boldsymbol{f}^b_{\text{imu}}[k]$ and $\boldsymbol{\omega}^b_{\text{imu}}[k]$.

KF aided by predicted measurements: Aided INS using the predicted position measurements $\hat{\boldsymbol{y}}[k]$ defined by (68).

KF aided by delayed measurement: Aided INS using the time-delayed position measurements $\boldsymbol{y}[k - D]$ instead of $\boldsymbol{y}[k]$.

Figs. 5–7 confirm that the estimated positions, linear velocities, and Euler angles are close to their actual values for the strapdown INS when aided by predicted position and time-delayed position measurements. However, the difference in performance is clear when zooming in on the plots. In Fig. 8, it is seen that position accuracy is improved by a couple of meters when using the predicted values. This also improves the accuracy of the attitude estimates, as shown in Fig. 9. Note that attitude is estimated using unit quaternions, but it is straightforward to map a unit quaternion to the roll-pitch-yaw Euler angles using (Fossen, 2021)

$$\phi = \text{atan2}\left(2(\varepsilon_2\varepsilon_3 + \varepsilon_1\eta),\ 1 - 2(\varepsilon_1^2 + \varepsilon_2^2)\right) \tag{75}$$

$$\theta = -\text{asin}\left(2(\varepsilon_1\varepsilon_3 - \varepsilon_2\eta)\right) \tag{76}$$

$$\psi = \text{atan2}\left(2(\varepsilon_1\varepsilon_2 + \varepsilon_3\eta),\ 1 - 2(\varepsilon_2^2 + \varepsilon_3^2)\right) \tag{77}$$

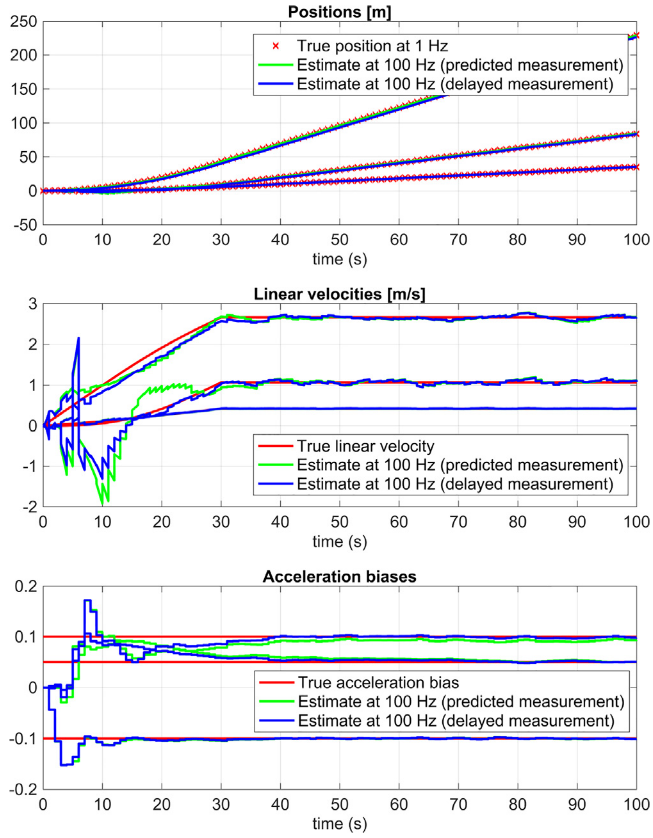


Fig. 6. North-East-Down positions, linear velocities, and acceleration biases versus time.

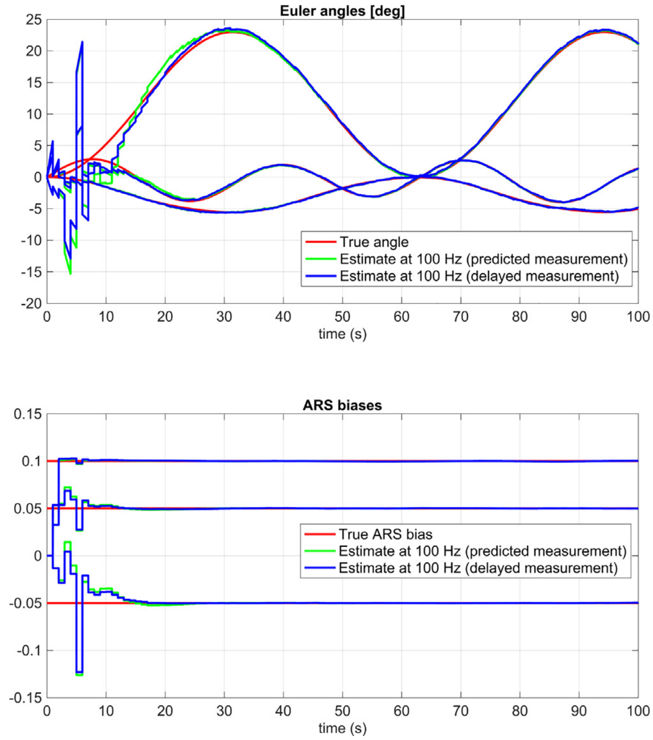


Fig. 7. Euler angles (roll-pitch-yaw) and ARS biases versus time.

Finally, the case study also confirmed that the ESKF manages to estimate the acceleration biases (lower plot in Fig. 6) and ARS bi-

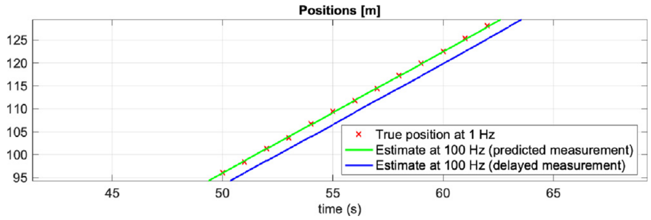


Fig. 8. Position zoom showing the performance improvement of the INS aided by the predicted positions to the 1 Hz delayed position measurements. The predicted values (green) are close to the measurements (red cross). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

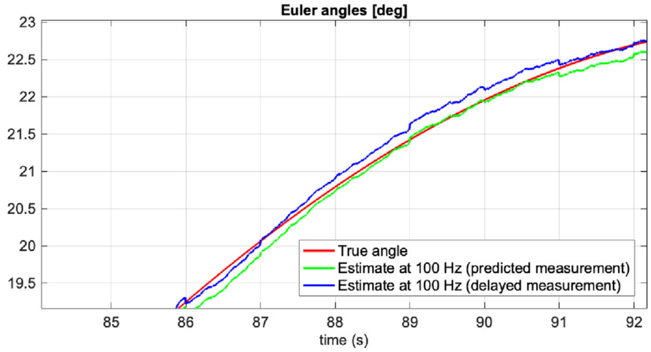


Fig. 9. Euler angle zoom showing the performance improvement of the INS aided by the predicted positions to the 1 Hz delayed position measurements. The predicted values (green) are close to the true values (red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

ases (lower plot in Fig. 7) with great accuracy. This is important for long-endurance underwater vehicle applications.

7. Conclusions

A tutorial article for the design of inertial navigation systems (INS) for underwater vehicles aided by time-delayed hydroacoustic positioning measurements has been presented. The state estimator was implemented using an error-state Kalman filter (ESKF) and a loosely-coupled integration scheme where the North-East-Down (NED) positions from a hydroacoustic reference system were used to aid the INS. The discrete-time unit quaternion ESKF, known as the multiplicative extended Kalman filter (MEKF), was implemented as a feedback algorithm with reset functionality. This is necessary for long-endurance autonomous underwater vehicle (AUV) operations. Proprietary navigation systems do not allow the user to add more measurement equations to the code if additional sensors are available for aided INS. However, the open-source filter architecture presented in the article provides for this. Hence, the presented ESKF is useful for vendors of low-cost AUV systems who aim to make in-house development of strapdown INS accessible and affordable. Finally, a case study of an AUV with a standard sensor suite confirmed that ESKF could accurately estimate the NED positions, linear velocities, and attitude when correcting the time-delayed position measurements. The ESKF also estimated the biases of the inertial measurements, which is essential for performance improvement and dead-reckoning operations.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Attitude dynamics

The unit quaternion error $\delta\boldsymbol{q}_b^n$ satisfies (Sola, 2016)

$$\boldsymbol{q}_b^n = \hat{\boldsymbol{q}}_{\text{ins}} \otimes \delta\boldsymbol{q}_b^n \tag{78}$$

where $\hat{\boldsymbol{q}}_{\text{ins}} \equiv \hat{\boldsymbol{q}}_b^n$. The unit quaternion differential equations are

$$\dot{\boldsymbol{q}}_b^n = \frac{1}{2}\boldsymbol{q}_b^n \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}_{nb}^b \end{bmatrix} = \frac{1}{2}\hat{\boldsymbol{q}}_{\text{ins}} \otimes \delta\boldsymbol{q}_b^n \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}_{nb}^b \end{bmatrix} \tag{79}$$

Differentiating (78) gives

$$\begin{aligned} \dot{\boldsymbol{q}}_b^n &= \dot{\hat{\boldsymbol{q}}}_{\text{ins}} \otimes \delta\boldsymbol{q}_b^n + \hat{\boldsymbol{q}}_{\text{ins}} \otimes \delta\dot{\boldsymbol{q}}_b^n \\ &= \frac{1}{2}\hat{\boldsymbol{q}}_{\text{ins}} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}_{\text{ins}}^b \end{bmatrix} \otimes \delta\boldsymbol{q}_b^n + \hat{\boldsymbol{q}}_{\text{ins}} \otimes \delta\dot{\boldsymbol{q}}_b^n \end{aligned} \tag{80}$$

Combining (79) and (80), and at the same time isolating the term $\hat{\boldsymbol{q}}_{\text{ins}}$ gives

$$\frac{1}{2}\delta\boldsymbol{q}_b^n \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}_{nb}^b \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0 \\ \boldsymbol{\omega}_{\text{ins}}^b \end{bmatrix} \otimes \delta\boldsymbol{q}_b^n + \delta\dot{\boldsymbol{q}}_b^n \tag{81}$$

Neglecting second-order terms in (29) implies that

$$\delta\boldsymbol{q}_b^n = \begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{a} \end{bmatrix} \Longrightarrow \delta\dot{\boldsymbol{q}}_b^n = \begin{bmatrix} 0 \\ \frac{1}{2}\delta\dot{\boldsymbol{a}} \end{bmatrix} \tag{82}$$

Hence, (82) inserted into (81) gives

$$\begin{bmatrix} 0 \\ \delta\dot{\boldsymbol{a}} \end{bmatrix} = \delta\boldsymbol{q}_b^n \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}_{nb}^b \end{bmatrix} - \begin{bmatrix} 0 \\ \boldsymbol{\omega}_{\text{ins}}^b \end{bmatrix} \otimes \delta\boldsymbol{q}_b^n \tag{83}$$

The product of two unit quaternions (*Hamiltonian product*) is defined as

$$\boldsymbol{q}_1 \otimes \boldsymbol{q}_2 = \begin{bmatrix} \eta_{q_1}\eta_{q_2} - \boldsymbol{\varepsilon}_{q_1}^{\top}\boldsymbol{\varepsilon}_{q_2} \\ \eta_{q_1}\boldsymbol{\varepsilon}_{q_2} + \eta_{q_2}\boldsymbol{\varepsilon}_{q_1} + \boldsymbol{S}(\boldsymbol{\varepsilon}_{q_1})\boldsymbol{\varepsilon}_{q_2} \end{bmatrix} \tag{84}$$

Expanding the second line in (83) gives

$$\begin{aligned} \delta\dot{\boldsymbol{a}} &= \boldsymbol{\omega}_{nb}^b + \frac{1}{2}\boldsymbol{S}(\delta\boldsymbol{a})\boldsymbol{\omega}_{nb}^b - \boldsymbol{\omega}_{\text{ins}}^b - \boldsymbol{S}(\boldsymbol{\omega}_{\text{ins}}^b)\delta\boldsymbol{q}_b^n \\ &= \boldsymbol{\omega}_{nb}^b - \frac{1}{2}\boldsymbol{S}(\boldsymbol{\omega}_{nb}^b)\delta\boldsymbol{a} - \boldsymbol{\omega}_{\text{ins}}^b - \frac{1}{2}\boldsymbol{S}(\boldsymbol{\omega}_{\text{ins}}^b)\delta\boldsymbol{a} \\ &= -\frac{1}{2}\boldsymbol{S}(\boldsymbol{\omega}_{nb}^b + \boldsymbol{\omega}_{\text{ins}}^b)\delta\boldsymbol{a} + \boldsymbol{\omega}_{nb}^b - \boldsymbol{\omega}_{\text{ins}}^b \end{aligned} \tag{85}$$

where $\boldsymbol{S}(\delta\boldsymbol{a})\boldsymbol{\omega}_{nb}^b = -\boldsymbol{S}(\boldsymbol{\omega}_{nb}^b)\delta\boldsymbol{a}$ and $\delta\boldsymbol{q}_b^n = (1/2)\delta\boldsymbol{a}$ have been exploited.

Appendix B. Reference vector measurement equation

Assume that $\boldsymbol{v}_i^b$ is a measured vector expressed in $\{b\}$. Consequently, the estimated vector is $\hat{\boldsymbol{v}}_i^b = \boldsymbol{R}_q^{\top}(\hat{\boldsymbol{q}}_b^n)\boldsymbol{v}_{0i}^n$ where $\boldsymbol{v}_{0i}^n$ is the *reference vector*. From this, it follows that

$$\begin{aligned} \delta\boldsymbol{y}_i &= (\boldsymbol{v}_i^b + \boldsymbol{\varepsilon}_i) - \boldsymbol{R}_q^{\top}(\hat{\boldsymbol{q}}_b^n)\boldsymbol{v}_{0i}^n \\ &= \boldsymbol{R}_q^{\top}(\boldsymbol{q}_b^n)\boldsymbol{v}_{0i}^n + \boldsymbol{\varepsilon}_i - \boldsymbol{R}_q^{\top}(\hat{\boldsymbol{q}}_b^n)\boldsymbol{v}_{0i}^n \\ &\approx \big(\boldsymbol{I}_3 - \boldsymbol{S}(\delta\boldsymbol{a})\big)\,\boldsymbol{R}_q^{\top}(\hat{\boldsymbol{q}}_{\text{ins}})\boldsymbol{v}_{0i}^n - \boldsymbol{R}_q^{\top}(\hat{\boldsymbol{q}}_{\text{ins}})\boldsymbol{v}_{0i}^n + \boldsymbol{\varepsilon}_i \\ &= -\boldsymbol{S}(\delta\boldsymbol{a})\boldsymbol{R}_q^{\top}(\hat{\boldsymbol{q}}_{\text{ins}})\boldsymbol{v}_{0i}^n + \boldsymbol{\varepsilon}_i \\ &= \boldsymbol{S}(\boldsymbol{R}_q^{\top}(\hat{\boldsymbol{q}}_{\text{ins}})\boldsymbol{v}_{0i}^n)\delta\boldsymbol{a} + \boldsymbol{\varepsilon}_i \end{aligned} \tag{86}$$

where (38) and $\hat{\boldsymbol{q}}_b^n = \hat{\boldsymbol{q}}_{\text{ins}}$ have been used on line three.

Appendix C. Strapdown INS Matlab function

The flow chart for implementing the strapdown INS in a computer language is shown in Fig. 2. The Matlab function *ins.m* calls the functions *Smtrx.m*, *Rquat.m*, and *Tquat.m*, which can be downloaded from the GitHub repository of the Marine Systems Simulator (MSS) (Fossen & Perez, 2004). The Matlab scripts are also compatible with the scientific programming language (GNU Octave, 2023), which is free software under the terms of the GNU General Public License.

```
function [x_ins, P_prd] = ins(x_ins, P_prd,...
   h, Qd, Rd,f_imu, w_imu, m_imu, m_ref, y_pos, y_pre)
% [x_ins, P_prd] = ins() updates the INS states x_ins
%  and covariance matrix P_prd at each sample.
%
% INPUTS:
%  x_ins[k], [p_ins, v_ins, b_acc_ins, q_ins, b_ars_ins]
%  P_prd[k],  15 x 15 covariance matrix
%  h sampling time [s]
%  Qd, Rd, ESKF process/measurement covariance matrices
%  f_imu[k], w_imu[k], m_imu[k] fast IMU measurements
%  m_ref, magnetometer NED reference vector, compute
%     m_ref = R' * m_imu when theta = phi = psi = 0
%  y_pos[k], slow position measurement
%  y_pre[k], slow pressure measurement
%
% OUTPUTS:
%  x_ins[k+1] INS state vector (16 states)
%  P_prd[k+1] ESKF covariance matrix (15 x 15)
% User-specified constants
T_acc = 1000;   % acceleration bias time constant
T_ars = 500;    % ARS bias time constant
rho = 1026;     % density of water
g = 9.82;       % acceleration of gravity
g_n = [0 0 g]'; % NED gravity vector

% INS states
p_ins = x_ins(1:3);
v_ins = x_ins(4:6);
b_acc_ins = x_ins(7:9);
q_ins = x_ins(10:13);
b_ars_ins = x_ins(14:16);

% Unit quaternion rotation matrix
R = Rquat(q_ins);

% Bias-compensated IMU measurements
f_ins = f_imu - b_acc_ins;
w_ins = w_imu - b_ars_ins;

% Normalized gravity vector
v10 = [0 0 1]';                              % NED
v1 = -f_ins/g;                               % BODY
v1 = v1 / sqrt( v1' * v1 );

% Normalized magnetic field vector
v20 = m_ref / sqrt( m_ref' * m_ref );        % NED
v2  = m_imu / sqrt( m_ref' * m_ref );        % BODY

% Discrete-time Kalman filter matrices
O3 = zeros(3,3);   O31 = zeros(3,1);
O13 = zeros(1,3);  I3 = eye(3);

A = [ O3 O3  O3              O3               O3
      O3 O3 -R              -R*Smtrx(f_ins)   O3
      O3 O3  -(1/T_acc)*I3   O3               O3
      O3 O3  O3             -Smtrx(w_ins)    -I3
      O3 O3  O3              O3               -(1/T_ars)*I3 ];

Ad = eye(15) + h * A + 0.5 * (h * A)^2;

Ed = h * [  O3 O3     O3 O3
           -R  O3     O3 O3
            O3 I3     O3 O3
            O3 O3    -I3 O3
            O3 O3     O3 I3  ];

Cd = [ I3  O3 O3   O3 O3                    % NED positions
       O13 [0  0   1] O13 O13 O13           % pressure
       O3  O3 O3   Smtrx(R'*v10) O3         % gravity
       O3  O3 O3   Smtrx(R'*v20) O3  ];     % magnetic field

if (nargin == 7),              % no aiding

    P_hat = P_prd;

else                           % aiding (slow measurement)

   K = P_prd * Cd' * inv(Cd * P_prd * Cd' + Rd);
   IKC = eye(15) - K * Cd;

   % Estimation error: eps[k]
   eps_pos = y_pos - p_ins;
   eps_g   = v1 - R' * v10;
   eps_mag = v2 - R' * v20;
   eps_pre = y_pre / (rho * g) - p_ins(3);

   eps = [eps_pos; eps_pre; eps_g; eps_mag];

   % Kalman filter corrector: delta_x_hat[k] and P_hat[k]
   delta_x_hat = K * eps;
   P_hat = IKC * P_prd * IKC' + K * Rd * K';

   % Error quaternion (2 x Gibbs vector): delta_q_hat[k]
   delta_a = delta_x_hat(10:12);
   delta_q_hat = 1 / sqrt(4 + delta_a' * delta_a) * ...
      [2 delta_a']';

   % INS reset: x_ins[k]
   p_ins = p_ins + delta_x_hat(1:3);               % position
   v_ins = v_ins + delta_x_hat(4:6);               % velocity
   b_acc_ins = b_acc_ins + delta_x_hat(7:9);       % acc bias
   b_ars_ins = b_ars_ins + delta_x_hat(13:15);     % ars bias
   q_ins = quatprod(q_ins, delta_q_hat);   % Schur product
      q_ins = q_ins / sqrt(q_ins' * q_ins);  % normalization

 end

 % Kalman filter predictor: P_prd[k+1]
 P_prd = Ad * P_hat * Ad' + Ed * Qd * Ed';

 % INS propagation: x_ins[k+1]
 a_ins = R * f_ins + g_n;
 p_ins = p_ins + h * v_ins + h^2/2 * a_ins;
 v_ins = v_ins + h * a_ins;
 q_ins = expm( Tquat(w_ins) * h ) * q_ins;
 q_ins = q_ins / sqrt(q_ins' * q_ins);

 x_ins = [p_ins; v_ins; b_acc_ins; q_ins; b_ars_ins];

 end
```

References

Alexander, H. L. (1991). State estimation for distributed systems with sensing delay. In V. Libby (Ed.), *Data structures and target classification, vol. 1470* (pp. 103–111). International Society for Optics and Photonics, SPIE, http://dx.doi.org/10.1117/12.44843.

Barbour, N., & Schmidt, G. (1998). Inertial sensor technology trends. In *Proceedings of the workshop on autonomous underwater vehicles* (pp. 52–55).

Batista, P. (2015). GES long baseline navigation with unknown sound velocity and discrete-time range measurements. *IEEE Transactions on Control Systems Technology*, *23*(1), 219–230. http://dx.doi.org/10.1109/TCST.2014.2321973.

Batista, P., Silvestre, C., & Oliveira, P. (2010). A sensor-based long baseline position and velocity navigation filter for underwater vehicles. *IFAC Proceedings Volumes*, *43*(14), 302–307. http://dx.doi.org/10.3182/20100901-3-IT-2016.00173, 8th IFAC Symposium on Nonlinear Control Systems.

Björklund, S., & Ljung, L. (2003). A review of time-delay estimation techniques. In *42nd IEEE international conference on decision and control (IEEE cat. no.03CH37475), vol. 3* (pp. 2502–2507 Vol.3). http://dx.doi.org/10.1109/CDC.2003.1272997.

Britting, K. R. (1971). *Inertial navigation systems analysis*. New York, NY: Wiley-Interscience.

Comellini, A., Casu, D., Zenou, E., Dubanchet, V., & Espinosa, C. (2020). Incorporating delayed and multi-rate measurements in navigation filter for autonomous space rendezvous. *Journal of Guidance Control and Dynamics*, *43*(6), 1164–1188. http://dx.doi.org/10.2514/1.G005034.

Crassidis, J. L., Markley, F. L., & Cheng, Y. (2007). Survey of nonlinear attitude estimation methods. *Journal of Guidance, Control, and Dynamics*, *30*(11), 12–28. http://dx.doi.org/10.2514/1.22452.

Falco, G., Pini, M., & Marucco, G. (2017). Loose and tight GNSS/INS integrations: Comparison of performance assessed in real urban scenarios. *Sensors 2017(17):27*, *17*(255), 1–25. http://dx.doi.org/10.3390/s17020255.

Farrell, J. A. (2008). *Aided navigation: GPS with high rate sensors*. New York, NY: McGraw-Hill.

Fossen, T. I. (2021). *Handbook of marine craft hydrodynamics and motion control* (2nd ed.). Chichester, UK: John Wiley & Sons, Ltd..

Fossen, T. I., & Perez, T. (2004). Marine Systems Simulator (MSS). https://github.com/cybergalactic/MSS.

Gelb, A. (1974). *Applied optimal estimation*. Cambridge, MA: MIT Press.

GNU Octave (2023). Scientific programming language. https://octave.org/. (Accessed 01 March 2023).

Grewal, M. S., Weill, L. R., & Andrews, A. P. (2001). *Global positioning systems, inertial navigation and integration*. New York, NY: John Wiley & Sons, Ltd.

Hegrenæs, Ø., Gade, K., Hagen, O. K., & Hagen, P. E. (2009). Underwater transponder positioning and navigation of autonomous underwater vehicles. In *OCEANS 2009* (pp. 1–7). http://dx.doi.org/10.23919/OCEANS.2009.5422358.

Hero, A., Messer, H., Goldberg, J., Thomson, D., Amin, M., Giannakis, G., Swami, A., Tugnait, J., Nehorai, A., Swindlehurst, A., Cardoso, J. F., Tong, L., & Krolik, J. (1998). Highlights of statistical signal and array processing. *IEEE Signal Processing Magazine*, *15*(5), 21–64. http://dx.doi.org/10.1109/79.708539.

Jalving, B., Gade, K., Svartveit, K., Willumsen, A., & Srhagen, R. (2007). DVL velocity aiding in the HUGIN 1000 integrated inertial navigation system. *Modeling, Identification and Control*, *25*(4), 223–236. http://dx.doi.org/10.4173/mic.2004.4.2.

Kinsey, J. C., Eustice, R. M., & Whitcomb, L. L. (2006). A survey of underwater vehicle navigation: Recent advances and new challenges. In *Proceedings of the 7th conference on maneuvering and control of marine craft*.

Larsen, T., Andersen, N., Ravn, O., & Poulsen, N. (1998). Incorporation of time delayed measurements in a discrete-time Kalman filter. In *Proceedings of the 37th IEEE conference on decision and control (Cat. no.98CH36171), vol. 4* (pp. 3972–3977 vol.4). http://dx.doi.org/10.1109/CDC.1998.761918.

Markley, F. L., & Crassidis, J. L. (2014). *Space Technology Library*: *vol.33*, *Fundamentals of spacecraft attitude determination and control* (1st ed.). New York, NY: Springer-Verlag.

Miller, P. A., Farrell, J. A., Zhao, Y., & Djapic, V. (2010). Autonomous underwater vehicle navigation. *IEEE Journal of Oceanic Engineering*, *35*(3), 663–678. http://dx.doi.org/10.1109/JOE.2010.2052691.

Morgado, M., Batista, P., Oliveira, P., & Silvestre, C. (2011). Position and velocity USBL/IMU sensor-based navigation filter. *IFAC Proceedings Volumes*, *44*(1), 13642–13647. http://dx.doi.org/10.3182/20110828-6-IT-1002.01127, 18th IFAC world congress.

NOAA (2023). Magnetic field calculator, the national centers for environmental information. https://www.ngdc.noaa.gov/geomag/calculators/magcalc.shtml. (Accessed 01 March 2023).

Paull, L., Saeedi, S., Seto, M., & Li, H. (2014). AUV navigation and localization: A review. *IEEE Journal of Oceanic Engineering*, *39*(1), 131–149. http://dx.doi.org/10.1109/JOE.2013.2278891.

Sola, J. (2016). *Quaternion kinematics for the error-state kalman filter*: *Report, IRI-TR-16-02*, Institut de Robotica i Informatica Industrial, CSIC-UPC.

Stovner, B. N. (2018). *Aided inertial navigation of underwater vehicles* (Ph.D. thesis), Trondheim, Norway: Department of Marine Technology, Norwegian University of Science and Technology.

Stovner, B. N., Johansen, T. A., Fossen, T. I., & Schjølberg, I. (2016). Three-stage filter for position and velocity estimation from long baseline measurements with unknown wave speed. In *2016 American control conference* (pp. 4532–4538). http://dx.doi.org/10.1109/ACC.2016.7526066.

Titterton, D. H., & Weston, J. L. (1997). *Strapdown inertial navigation technology*. London, UK: IEE.

Vickery, K. (1998). Acoustic positioning systems. A practical overview of current systems. In *Proceedings of the 1998 workshop on autonomous underwater vehicles (Cat. no.98CH36290)* (pp. 5–17). http://dx.doi.org/10.1109/AUV.1998.744434.

Zhao, B., Blanke, M., & Skjetne, R. (2012). Particle filter ROV navigation using hydroacoustic position and speed log measurements. In *2012 American control conference* (pp. 6209–6215). http://dx.doi.org/10.1109/ACC.2012.6315511.

Thor I. Fossen is a naval architect and a cyberneticist. He received an M.Sc. in Marine Technology in 1987 and a Ph.D. in Engineering Cybernetics in 1991 from the Norwegian University of Science and Technology (NTNU), Trondheim. He is currently a guidance, navigation, and control professor at the Department of Engineering Cybernetics, NTNU. Fossen's academic background, besides cybernetics, is computer science, cybersecurity, aerospace engineering, marine technology, and inertial navigation systems. He has authored three textbooks and three editorials. Fossen is one of the co-founders and former Vice President of R&D of the company Marine Cybernetics, which DNV acquired in 2012. He is also a co-founder of ScoutDI. The Institute of Electrical and Electronics Engineers elevated him to IEEE Fellow in 2016. He received the Automatica Prize Paper Award in 2002 and the Arch T. Colwell Merit Award in 2008 at the SAE World Congress. He was elected to the Norwegian Academy of Technological Sciences (1998) and the Norwegian Academy of Science and Letters (2022).