Sensor Fusion using Error-State Kalman Filter to Improve Localization of Autonomous Underwater Vehicle under DVL Signal Loss

Akira Techapattaraporn[1], Vasutorn Siriyakorn[2], Peerayot Sanposh[3†],
Yodyium Tipsuwan[4], Teerasit Kasetkasem[5], and Weerawut Charubhun[6]

Abstract— **This paper concerns the development and improvement of the underwater navigation system for Autonomous Underwater Vehicles (AUVs) when faced with velocity-aiding sensor failures. The study addresses this challenge of sensor fusion by applying the Error-State Kalman Filter (ESKF), a form of indirect state filtering. Specifically, the ESKF targets the limitations of velocity measurements encountered during near-bottom operations. The proposed method was applied to the Xplorer-Mini AUV and evaluated using the Gazebo physics engine simulator in ROS 2. To assess its performance in handling the loss of velocity signals, a series of simulation-based experiments were conducted and compared against the traditional Inertial Navigation System (INS) and Extended Kalman Filter (EKF) algorithms. The results of the experiments demonstrate that the ESKF outperforms traditional INS and EKF algorithms, offering valuable insights into state estimation techniques for developing autonomous underwater vehicles in challenging environments.**

I. Introduction

Advancements in sensor technologies for autonomous underwater vehicles (AUVs) have propelled the field forward, yet accurate underwater navigation remains a significant obstacle to AUV autonomy. Near-bottom navigation is especially challenging, impeding further progress [1]. Limited sensor data from velocity sensors during near-sea floor operations may require advanced navigation techniques to mitigate data loss.

According to [2], underwater navigation techniques based on the Kalman filter can be categorized into direct state filtering and indirect state filtering. Direct filtering involves obtaining the state vector directly, while indirect filtering involves acquiring the states indirectly. The choice between these methods depends on factors such as computational complexity, accuracy requirements, and underwater environment characteristics.

Indirect state inertial navigation systems have been proposed to solve underwater navigation in situations with limited velocity measurements, as highlighted by Hegrenaes et al. [3]. By compensating for the constraints of velocity sensors, these systems enhance navigation accuracy during near-bottom operations. The improved near-bottom navigation capability enables more precise data collection, thereby facilitating further advancements in autonomous AUVs.

This paper contributes by developing and validating the Error State Kalman Filter (ESKF), which is an indirect state filtering approach for underwater navigation. The ESKF algorithm is specifically designed to better handle lossy velocity signals encountered during near-bottom operations. Through comprehensive testing, the study demonstrates the effectiveness of this approach in compensating for imperfect velocity measurements and improving navigation accuracy. This contribution expands the understanding of state estimation techniques and provides valuable insights for developing autonomous underwater vehicles in challenging environments.

The paper is organized into the following sections. Section II provides a thorough overview of underwater vehicle modeling and underwater navigation. The experimental setup and simulation details are presented in Section III. Section IV presents the results and includes a comprehensive discussion. Finally, Section V offers the conclusion and summarizes the study's key findings.

II. Backgrounds

A. Underwater Vehicle Kinematics

The focus of this research centers on the Xplorer-Mini AUV model, which incorporates crucial sensors, namely the Inertial Measurement Unit (IMU), Attitude and Heading Reference Systems (AHRS), and Doppler Velocity Log (DVL). Refer to Fig. 1 for visual representation of our AUV in ROS.

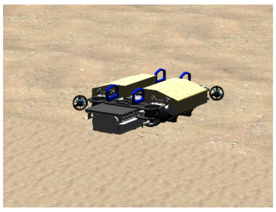

Fig. 1. Xplorer-Mini AUV in Gazebo simulator

1) Frames and Coordinates: When analyzing the motion of an AUV, it is essential to define reference frames and coordinates to facilitate analysis. These reference frames provide a spatial context for understanding the AUV's movement and its relation to the surrounding environment. Typically, three frames are commonly used: the body frame, the Earth-fixed frame, and the navigation frame. The body frame $\{b\}$

*This work was supported by Rovula (Thailand) Co., Ltd.

[1] Akira Techapattaraporn with the Department of Electrical Engineering, Faculty of Engineering, Kasetsart University, Bangkok 10903, Thailand akira.techap@gmail.com

[2] Vasutorn Siriyakorn with the Department of Electrical Engineering, Faculty of Engineering, Kasetsart University, Bangkok 10903, Thailand vasutornj@gmail.com

[3†] Peerayot Sanposh with the Department of Electrical Engineering, Faculty of Engineering, Kasetsart University, Bangkok 10903, Thailand peerayot.s@ku.ac.th

[4] Yodyium Tipsuwan with the Department of Computer Engineering, Faculty of Engineering, Kasetsart University, Bangkok 10903, Thailand yyt@ku.ac.th

[5] Teerasit Kasetkasem with the Department of Electrical Engineering, Faculty of Engineering, Kasetsart University, Bangkok 10903, Thailand fengtsk@ku.ac.th

[6] Weerawut Charubhun with Rovula (Thailand) Co., Ltd., Bangkok 10120, Thailand weerawutc@rovula.com

979-8-3503-0219-6/23/$31.00 ©2023 IEEE

is attached to the AUV itself, with its origin at the vehicle's center of mass. The Earth-fixed frame $\{e\}$ is fixed relative to the Earth's surface, allowing for absolute positioning. Lastly, the navigation frame $\{n\}$ is aligned with the Earth-fixed frame but can be translated and rotated to track the AUV's position and orientation. In this paper, the frame used for the development is the navigation frame.

2) Vehicle States: The AUV state vector consists of its position, linear velocity, quaternion, accelerometer bias, gyroscope bias, and gravity vector; therefore, it can be written as

$$\boldsymbol{x} = \left[\boldsymbol{p}^\top \; \boldsymbol{v}^\top \; \boldsymbol{q}^\top \; \boldsymbol{a}_b^\top \; \boldsymbol{\omega}_b^\top \; \boldsymbol{g}^\top\right]^\top \tag{1}$$

where $\boldsymbol{p}$ denotes the position, $\boldsymbol{v}$ represents the linear velocity, $\boldsymbol{q}$ is the quaternion, $\boldsymbol{a}_b$ stands for the accelerometer bias, $\boldsymbol{\omega}_b$ denotes the gyroscope bias, and $\boldsymbol{g}$ represents the gravity vector. All state variables are expressed in the navigation frame.

3) Nominal state kinematics: The nominal state kinematics represents the vehicle model without any noise or perturbation terms [4]. The true state vector, denoted as $\boldsymbol{x}_t$, can be written as the composition of the nominal state vector, represented as $\boldsymbol{x}$, and the error-state vector, denoted as $\delta\boldsymbol{x}$, i.e.,

$$\boldsymbol{x}_t = \boldsymbol{x} \oplus \delta\boldsymbol{x} \tag{2}$$

where $\oplus$ denotes the composition operator. Furthermore, the state variable with the subscript t denotes its true values, the state variable without the subscript t denotes its nominal values, and the state variable with δ in the front denotes its error.

The vehicle has the nominal-kinematic equations in discrete time as follows:

$$\begin{aligned} \boldsymbol{p}_k &= \boldsymbol{p}_{k-1} + \boldsymbol{v}_{k-1}\Delta t_{k-1} \\ &\quad + \frac{1}{2}(\boldsymbol{R}_{k-1}(\boldsymbol{a}_{m,k-1} - \boldsymbol{a}_{b,k-1}) + \boldsymbol{g}_{k-1})\Delta t_{k-1}^2 \end{aligned} \tag{3}$$

$$\begin{aligned} \boldsymbol{v}_k &= \boldsymbol{v}_{k-1} \\ &\quad + (\boldsymbol{R}_{k-1}(\boldsymbol{a}_{m,k-1} - \boldsymbol{a}_{b,k-1}) + \boldsymbol{g}_{k-1})\Delta t_{k-1} \end{aligned} \tag{4}$$

$$\boldsymbol{q}_k = \boldsymbol{q}_{k-1} \otimes \boldsymbol{q}\big\{(\boldsymbol{\omega}_{m,k-1} - \boldsymbol{\omega}_{b,k-1})\Delta t_{k-1}\big\} \tag{5}$$

$$\boldsymbol{a}_{b,k} = \boldsymbol{a}_{b,k-1} \tag{6}$$

$$\boldsymbol{\omega}_{b,k} = \boldsymbol{\omega}_{b,k-1} \tag{7}$$

$$\boldsymbol{g}_k = \boldsymbol{g}_{k-1} \tag{8}$$

The subscripts k and $k-1$ indicate the indices of the current and previous samples, respectively. Moreover, Δt_{k-1} is the time difference between two sample times, i.e.,

$$\Delta t_{k-1} = t_k - t_{k-1}. \tag{9}$$

The matrix $\boldsymbol{R}$ is the rotation matrix transforming a vector from the body frame to the navigation frame. The symbol $\otimes$ denotes the quaternion product, and $\boldsymbol{q}\{\boldsymbol{\phi}\}$ denotes the quaternion of the rotation vector $\boldsymbol{\phi}$.

The measured linear acceleration from the accelerometer is represented by $\boldsymbol{a}_m$, and the measured angular velocity from the gyroscope is denoted by $\boldsymbol{\omega}_m$. Any variable with the subscript m denotes its measured quantity.

4) Error-state kinematics: The error-state kinematics [4] is determined by

$$\delta\boldsymbol{p}_k = \delta\boldsymbol{p}_{k-1} + \delta\boldsymbol{v}_{k-1}\Delta t_{k-1} \tag{10}$$

$$\begin{aligned} \delta\boldsymbol{v}_k &= \delta\boldsymbol{v}_{k-1} + (-\boldsymbol{R}_{k-1}\left[\boldsymbol{a}_{m,k-1} - \boldsymbol{a}_{b,k-1}\right]_\times \delta\boldsymbol{\theta}_{k-1} \\ &\quad - \boldsymbol{R}_{k-1}\delta\boldsymbol{a}_{b,k-1} + \delta\boldsymbol{g}_{k-1})\Delta t_{k-1} + \boldsymbol{v}_{i,k-1} \end{aligned} \tag{11}$$

$$\begin{aligned} \delta\boldsymbol{\theta}_k &= \boldsymbol{R}^\top\big\{(\boldsymbol{\omega}_{m,k-1} - \boldsymbol{\omega}_{b,k-1})\Delta t_{k-1}\big\}\,\delta\boldsymbol{\theta}_{k-1} \\ &\quad - \delta\boldsymbol{\omega}_{b,k-1}\Delta t_{k-1} + \boldsymbol{\theta}_{i,k-1} \end{aligned} \tag{12}$$

$$\delta\boldsymbol{a}_{b,k} = \delta\boldsymbol{a}_{b,k-1} + \boldsymbol{a}_{i,k-1} \tag{13}$$

$$\delta\boldsymbol{\omega}_{b,k} = \delta\boldsymbol{\omega}_{b,k-1} + \boldsymbol{\omega}_{i,k-1} \tag{14}$$

$$\delta\boldsymbol{g}_k = \delta\boldsymbol{g}_{k-1} \tag{15}$$

where $[\;\cdot\;]_\times$ is the skew operator producing the cross-product skew-symmetric matrix. The function $\boldsymbol{R}\{\cdot\}$ maps the rotation vector $\boldsymbol{\phi}$ to the rotation matrix $\boldsymbol{R}\{\boldsymbol{\phi}\}$.

The above equations describe the evolution of the error-state components, namely position error $\delta\boldsymbol{p}$, linear velocity error $\delta\boldsymbol{v}$, orientation error $\delta\boldsymbol{\theta}$, accelerometer bias error $\delta\boldsymbol{a}_b$, gyroscope bias error $\delta\boldsymbol{\omega}_b$, and gravity vector error $\delta\boldsymbol{g}$.

In addition to these equations, random impulses $\boldsymbol{v}_i$, $\boldsymbol{\theta}_i$, $\boldsymbol{a}_i$, and $\boldsymbol{\omega}_i$ are introduced, which represent random effects applied to velocity, orientation, and bias estimates. These impulses are based on the integration of zero-mean white Gaussian processes.

To write the error-state kinematic equations in compact form, the error state vector $\delta\boldsymbol{x}$, input vector $\boldsymbol{u}_m$, and impulse vector $\boldsymbol{i}$ are defined as

$$\delta\boldsymbol{x} = \left[\delta\boldsymbol{p}^\top \; \delta\boldsymbol{v}^\top \; \delta\boldsymbol{\theta}^\top \; \delta\boldsymbol{a}_b^\top \; \delta\boldsymbol{\omega}_b^\top \; \delta\boldsymbol{g}^\top\right]^\top \tag{16}$$

$$\boldsymbol{u}_m = \left[\boldsymbol{a}_m^\top \; \boldsymbol{\omega}_m^\top\right]^\top \tag{17}$$

$$\boldsymbol{i} = \left[\boldsymbol{v}_{\boldsymbol{i}}^\top \; \boldsymbol{\theta}_{\boldsymbol{i}}^\top \; \boldsymbol{a}_{\boldsymbol{i}}^\top \; \boldsymbol{\omega}_{\boldsymbol{i}}^\top\right]^\top . \tag{18}$$

The error-sate kinematic equation is then compactly written as

$$\delta\boldsymbol{x}_k = \boldsymbol{F}_{\boldsymbol{x},k-1}\,\delta\boldsymbol{x}_{k-1} + \boldsymbol{F}_{\boldsymbol{i},k-1}\,\boldsymbol{i}_{k-1} \tag{19}$$

where $\boldsymbol{F}_{\boldsymbol{x},k} = \boldsymbol{F}_{\boldsymbol{x},k}\left(\boldsymbol{x}_k, \boldsymbol{u}_{m,k}\right)$ is given by

$$\begin{aligned} \boldsymbol{F}_{\boldsymbol{x},k} &= \begin{bmatrix} \boldsymbol{F}_{11,k} & \boldsymbol{F}_{12,k} \\ \boldsymbol{F}_{21,k} & \boldsymbol{F}_{22,k} \end{bmatrix} \\ \boldsymbol{F}_{11,k} &= \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{I}_3\Delta t_k & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_3 & -\boldsymbol{R}_k\left[\boldsymbol{a}_{m,k} - \boldsymbol{a}_{b,k}\right]_\times \Delta t_k \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{R}_k^\top\left\{\left(\boldsymbol{\omega}_{m,k} - \boldsymbol{\omega}_{b,k}\right)\Delta t_k\right\} \end{bmatrix} \\ \boldsymbol{F}_{12,k} &= \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ -\boldsymbol{R}_k\Delta t_k & \boldsymbol{0} & \boldsymbol{I}_3\Delta t_k \\ \boldsymbol{0} & -\boldsymbol{I}_3\Delta t_k & \boldsymbol{0} \end{bmatrix} \\ \boldsymbol{F}_{21,k} &= \boldsymbol{0}_{9\times 9} \\ \boldsymbol{F}_{22,k} &= \boldsymbol{I}_{\boldsymbol{9}} \end{aligned} \tag{20}$$

and $\boldsymbol{F}_{\boldsymbol{i},k}$ is the Jacobian matrix of the state function with respect to the error state and perturbation vectors given by

$$\boldsymbol{F}_{\boldsymbol{i},k} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{I}_3 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_3 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_3 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_3 \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} . \tag{21}$$

B. Sensors

1) IMU: An IMU, which consists of accelerometers, gyroscopes, and sometimes magnetometers [5], is a sensor for measuring motion-related parameters. The output of an IMU is typically the raw data which are linear acceleration $\boldsymbol{a}_m$, angular velocity $\boldsymbol{\omega}_m$, and magnetic field $\boldsymbol{H}$.

2) AHRS: An AHRS is a sensor-based system used to precisely determine an object's orientation (given by the measured quaternion $\boldsymbol{q}_m$) and heading. These systems integrate data from multiple sensors, including accelerometers, gyroscopes, and magnetometers, to calculate and provide attitude parameters. Measurements can be obtained through the navigation frame.

3) DVL: DVL is an acoustic sensor that measures relative velocity to the sea floor. It emits acoustic pings, analyzes returned signals to determine echo intensity, and detects the sea floor. It provides velocity measurements in different modes: Bottom Track Mode for seabed speed, Water Track Mode for water current speed, and Water Profile Mode for water column velocity. By tracking the bottom, it calculates its own velocity relative to the seabed and estimates absolute water velocity, aided by a heading sensor [6].

C. Indirect State Filtering

Indirect state filtering, used in state estimation filters like the Kalman filter, estimates the error state directly instead of the full system state. In this approach, the Kalman filter is formulated as an error state filter, estimating deviations from the true system state. These errors are commonly referred to as the error state or error vector. The estimated error state is then used to correct and update the predicted state. This approach offers benefits such as numerical stability, reduced computational complexity, and the ability to handle nonlinear system dynamics. To obtain the estimated full state of the system, the estimated error state is typically combined with the predicted state.

The indirect state filtering technique can be classified into two distinct types: feedforward and feedback systems [2]. In the feedforward approach, error corrections are integrated into the state of the Inertial Navigation System (INS). Conversely, the feedback approach utilizes error estimates to directly update the INS, thereby mitigating the propagation and accumulation of INS errors. This feedback-based indirect state filtering approach is commonly referred to as the ESKF.

The ESKF algorithm is a variation of the Extended Kalman Filter (EKF) that estimates the state indirectly [7]. It involves three important aspects: the true state, the nominal state, and the error state values [4]. The true state is a combination of the nominal state, which represents a large signal and is integrable in a nonlinear manner, and the error state, which is small and linearly integrable, as shown in (2).

Using strap-down navigation [8], high-frequency IMU data can be constructed to INS. The ESKF incorporates INS data into the nominal state, ignoring noise and imperfections in the model that lead to error accumulation. These errors are captured in the error state and estimated using the ESKF, accounting for noise and disturbances. It represents small-signal magnitudes and follows a time-variant linear dynamic system. The dynamic, control, and measurement matrices for this system are computed based on the values of the nominal state. The ESKF predicts the error state using a Gaussian estimate while integrating the nominal state but lacks additional measurements for correction. Correction occurs when supplementary information, such as Global Navigation Satellite System (GNSS), velocity input, or heading input becomes available, allowing for error observation. This correction, occurring at a lower rate than the integration phase, yields a posterior Gaussian estimate of the error state. After correction, the error state's mean is injected into the nominal state and reset to zero, while the covariance matrix is updated accordingly. This iterative process continues indefinitely in this manner.

The ESKF consists of 4 steps: prediction, update, injection, and reset. The prediction and update steps in the ESKF closely resemble those of a conventional Kalman filter. However, the ESKF differs from the standard Kalman filter approach in the injection and reset steps.

First, let the initial error-state estimate be zero, i.e., $\widehat{\delta \boldsymbol{x}}_0 = \mathbf{0}$. Then, the ESKF is described as follows: for $k = 1, 2, 3, \ldots$

1) Prediction: The predicted equations of ESKF can be represented by

$$\begin{aligned} \widehat{\delta \boldsymbol{x}}_k^- &= \boldsymbol{F}_{\boldsymbol{x},k-1}\, \widehat{\delta \boldsymbol{x}}_{k-1} \\ \boldsymbol{P}_k^- &= \boldsymbol{F}_{\boldsymbol{x},k-1} \boldsymbol{P}_{k-1} \boldsymbol{F}_{\boldsymbol{x},k-1}^\top + \boldsymbol{F}_{\boldsymbol{i},k-1} \boldsymbol{Q}_{\boldsymbol{i},k-1} \boldsymbol{F}_{\boldsymbol{i},k-1}^\top \end{aligned} \tag{22}$$

where $\widehat{\delta \boldsymbol{x}}_k^-$ and $\boldsymbol{P}_k^-$ represent the priori error-state estimate and state covariance matrix, respectively. Similarly, $\widehat{\delta \boldsymbol{x}}_{k-1}$ and $\boldsymbol{P}_{k-1}$ represent the posteriori error-state estimate and covariance matrix in the previous step, respectively. Since the posteriori error-state $\widehat{\delta \boldsymbol{x}}_{k-1}$ in the previous step is zero, the priori error-state estimate $\widehat{\delta \boldsymbol{x}}_k^-$ is always zero.

Finally, $\boldsymbol{Q}_{\boldsymbol{i},k}$ is the covariance of the perturbation matrix represented by

$$\boldsymbol{Q}_{\boldsymbol{i},k} = \mathrm{diag}\{\boldsymbol{V}_{\boldsymbol{i},k}, \boldsymbol{\Theta}_{\boldsymbol{i},k}, \boldsymbol{A}_{\boldsymbol{i},k}, \boldsymbol{\Omega}_{\mathrm{i},k}\} \tag{23}$$

where $\boldsymbol{V}_{\boldsymbol{i},k}$, $\boldsymbol{\Theta}_{\boldsymbol{i},k}$, $\boldsymbol{A}_{\boldsymbol{i},k}$, and $\boldsymbol{\Omega}_{\boldsymbol{i},k}$ are the integration of the variances of components of $\boldsymbol{i}$, which are obtained from sensor datasheets.

2) Update: Note that the measured output $\boldsymbol{y} = \boldsymbol{y}_m$ from one of the sensors can be written as

$$\boldsymbol{y}_k = \boldsymbol{y}_{m,k} = \boldsymbol{h}(\boldsymbol{x}_{t,k}) + \boldsymbol{n}_k \tag{24}$$

where $\boldsymbol{h}$ is an appropriate measurement function of the true state, and $\boldsymbol{n}_k$ is a zero-mean white Gaussian noise with covariance $\boldsymbol{R}_k$.

Consequently, the update equations are given as follows:

$$\boldsymbol{K}_k = \boldsymbol{P}_k^- \boldsymbol{H}_k^\top \left(\boldsymbol{H}_k \boldsymbol{P}_k^- \boldsymbol{H}_k^\top + \boldsymbol{R}_k\right)^{-1} \tag{25}$$

$$\widehat{\boldsymbol{x}_{t}}_{,k}^- = \boldsymbol{x}_k \oplus \widehat{\delta \boldsymbol{x}}_k^- \tag{26}$$

$$\widehat{\delta \boldsymbol{x}}_k = \boldsymbol{K}_k \left(\boldsymbol{y}_k - h(\widehat{\boldsymbol{x}_{t}}_{,k}^-)\right) \tag{27}$$

$$\widehat{\boldsymbol{x}_{t}}_{,k} = \boldsymbol{x}_k \oplus \widehat{\delta \boldsymbol{x}}_k \tag{28}$$

$$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) \boldsymbol{P}_k^- \tag{29}$$

where $\boldsymbol{K}_k$ is the Kalman gain. Moreover, $\widehat{\delta \boldsymbol{x}}_k$ and $\boldsymbol{P}_k$ represent the posteriori estimated error state and state covariance matrix at the current step, respectively.

The Jacobian matrix $\boldsymbol{H}_k$ represents the sensitivity of the measurement function $\boldsymbol{h}$ to the error state $\delta\boldsymbol{x}$ and should be evaluated at the posteriori true-state estimate $\widehat{\boldsymbol{x}_{t,k}} = \boldsymbol{x}_k \oplus \widehat{\delta\boldsymbol{x}_k}$. However, $\boldsymbol{H}_k$ must be calculated first before finding $\boldsymbol{K}_k$ and then $\widehat{\delta\boldsymbol{x}_k}$. Since the error-state mean is zero, the nominal state can be used as the evaluating point, $\widehat{\boldsymbol{x}_{t,k}} \approx \boldsymbol{x}_k$ for $\boldsymbol{H}_k$. This simplification leads to the expression:

$$\begin{aligned} \boldsymbol{H}_k = \boldsymbol{H}_{\delta\boldsymbol{x},k} &= \left.\frac{\partial \boldsymbol{h}}{\partial \delta\boldsymbol{x}}\right|_{\boldsymbol{x}_t=\boldsymbol{x}_k} \\ &= \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_t}\right|_{\boldsymbol{x}_t=\boldsymbol{x}_k} \left.\frac{\partial \boldsymbol{x}_t}{\partial \delta\boldsymbol{x}}\right|_{\boldsymbol{x}_t=\boldsymbol{x}_k} = \boldsymbol{H}_{\boldsymbol{x}_t,k}\boldsymbol{X}_{\delta\boldsymbol{x},k} \end{aligned} \tag{30}$$

where $\boldsymbol{H}_{\boldsymbol{x}_t,k}$ is the Jacobian matrix of the measurement function with respect to the true state evaluated at $\boldsymbol{x}_k$, and $\boldsymbol{X}_{\delta\boldsymbol{x},k}$ is the Jacobian matrix of the true state with respect to the error state evaluated at $\boldsymbol{x}_k$ and is calculated by

$$\boldsymbol{X}_{\delta\boldsymbol{x},k} = \text{diag}\{\boldsymbol{I}_6, \boldsymbol{Q}_{\delta\boldsymbol{\theta},k}, \boldsymbol{I}_9\} \tag{31}$$

$$\boldsymbol{Q}_{\delta\boldsymbol{\theta},k} = \frac{1}{2}\begin{bmatrix} -q_{x,k} & -q_{y,k} & -q_{z,k} \\ q_{w,k} & -q_{z,k} & q_{y,k} \\ q_{z,k} & q_{w,k} & -q_{x,k} \\ -q_{y,k} & q_{x,k} & q_{w,k} \end{bmatrix}. \tag{32}$$

3) Injection: After the update, the nominal state is injected using

$$\boldsymbol{x}_k \leftarrow \boldsymbol{x}_k \oplus \widehat{\delta\boldsymbol{x}_k}. \tag{33}$$

4) Reset: The error reset function $\boldsymbol{g}$ is given by

$$\boldsymbol{g}(\delta\boldsymbol{x}_k) = \delta\boldsymbol{x}_k \ominus \widehat{\delta\boldsymbol{x}_k} \tag{34}$$

where $\ominus$ is the composition inverse operator of $\oplus$. The error state is reset by the error reset function as

$$\delta\boldsymbol{x}_k \leftarrow \boldsymbol{g}(\delta\boldsymbol{x}_k) = \delta\boldsymbol{x}_k \ominus \widehat{\delta\boldsymbol{x}_k} \tag{35}$$

Therefore, the error state estimate and the covariance matrix can be reset by the following expressions:

$$\widehat{\delta\boldsymbol{x}_k} \leftarrow \boldsymbol{0}_{18\times 1} \tag{36}$$

$$\boldsymbol{P}_k \leftarrow \boldsymbol{G}_k\boldsymbol{P}_k\boldsymbol{G}_k^{\top} \tag{37}$$

where $\boldsymbol{G}_k$ is the Jacobian matrix of the error reset function $\boldsymbol{g}$ with respect to the error state $\delta\boldsymbol{x}$ evaluated at the posteriori error-state estimate $\widehat{\delta\boldsymbol{x}_k}$ and is calculated by

$$\begin{aligned} \boldsymbol{G}_k = \boldsymbol{G}_{\delta\boldsymbol{x},k} &= \left.\frac{\partial \boldsymbol{g}}{\partial \delta\boldsymbol{x}}\right|_{\delta\boldsymbol{x}=\widehat{\delta\boldsymbol{x}_k}} \\ &= \text{diag}\{\boldsymbol{I}_6, \boldsymbol{I}_3 - \left[\frac{1}{2}\widehat{\delta\boldsymbol{\theta}_k}\right]_{\times}, \boldsymbol{I}_9\} \end{aligned} \tag{38}$$

Note that since the term $\widehat{\delta\boldsymbol{\theta}_k}$ is very small and can be neglected, this leads to the simplification of $\boldsymbol{G}_k = \boldsymbol{I}_{18}$.

III. Implementation and Experiment

A. Underwater Navigation

The system diagram of the ESKF implementation is illustrated in Fig. 2. The design of this implementation follows the indirect state filtering approach discussed previously, utilizing both AHRS and DVL as aiding sensors. The AHRS contributes orientation and angular velocity data, while the DVL measures linear velocity.

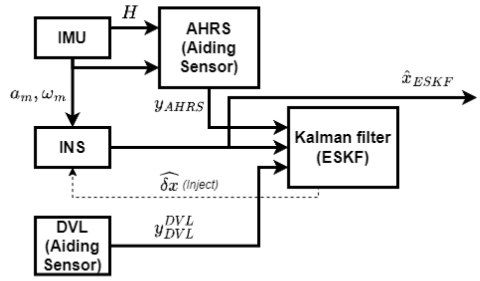

Fig. 2. INS-AHRS-DVL aided navigation system

The update step of the ESKF follows (25)-(29). Additionally, the update equations for AHRS and DVL are represented in (41) and (45), respectively. The update is performed whenever information from one of the aiding sensors is acquired.

1) AHRS: The AHRS measurement update is described by

$$\boldsymbol{y}_{\text{AHRS},k} = \boldsymbol{q}_{m,k} \tag{39}$$

$$\boldsymbol{H}_{\boldsymbol{x}_t,k} = \boldsymbol{H}_{\text{AHRS}} = \begin{bmatrix}\boldsymbol{0}_{4\times 6} & \boldsymbol{I}_4 & \boldsymbol{0}_{4\times 9}\end{bmatrix} \tag{40}$$

$$\widehat{\delta\boldsymbol{x}_k} = \boldsymbol{K}_{\text{AHRS},k}\left(\boldsymbol{y}_{\text{AHRS},k} - \boldsymbol{H}_{\text{AHRS}}\widehat{\boldsymbol{x}_{t,k}^{-}}\right). \tag{41}$$

2) DVL: The DVL measurement update is described by

$$\boldsymbol{y}_{\text{DVL},k}^{\text{DVL}} = \boldsymbol{v}_{m,k}^{\text{DVL}} \tag{42}$$

$$\boldsymbol{y}_{\text{DVL},k} = \boldsymbol{R}_b^n\boldsymbol{R}_{\text{DVL}}^b\boldsymbol{y}_{\text{DVL},k}^{\text{DVL}} \tag{43}$$

$$\boldsymbol{H}_{\boldsymbol{x}_t,k} = \boldsymbol{H}_{\text{DVL}} = \boldsymbol{R}_b^n\boldsymbol{R}_{\text{DVL}}^b\begin{bmatrix}\boldsymbol{0}_{3\times 3} & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times 13}\end{bmatrix} \tag{44}$$

$$\widehat{\delta\boldsymbol{x}_k} = \boldsymbol{K}_{\text{DVL},k}\left(\boldsymbol{y}_{\text{DVL},k} - \boldsymbol{H}_{\text{DVL}}\widehat{\boldsymbol{x}_{t,k}^{-}}\right). \tag{45}$$

Since the DVL measures the vehicle's linear velocity in the DVL frame {DVL}, it must be transformed to the linear velocity in the navigation frame $\{n\}$ by $\boldsymbol{R}_b^n\boldsymbol{R}_{\text{DVL}}^b$ as shown in (43).

B. Experimental Setup

1) Simulation Tools: The simulation tool used is Gazebo [9], which enables virtual robot simulation with physics and sensor capabilities. In Gazebo, the provided position serves as the crucial ground truth reference for evaluating the technique's performance. The underwater environment was simulated using Plankton [10], an open-source ROS package. The true state of the AUV is denoted as $\boldsymbol{x}_t$.

2) Velocity aiding sensor signal loss capturing and simulating: The experiment was conducted in the 1.2-meter depth swimming pool where the DVL is attached to an AUV. The position of the AUV was controlled through the joystick to maneuver freely in the water. An obstacle was placed on the pool floor to replicate poor bathymetry. The AUV was then guided through the designated area with the obstacle, shown in Fig. 3. Consequently, the collected DVL data exhibited signal loss due to unfavorable bathymetry.

The DVL's lossy signal was simulated using the Gilbert-Elliott (GE) channel model with a geometric distribution, as shown in Fig. 4 [11]. This widely used model represents binary channels with intermittent errors, characterized by two states: a normal state, where data is transmitted without

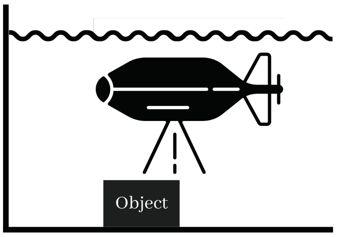

Fig. 3. Illustration of DVL signal loss capturing experiment

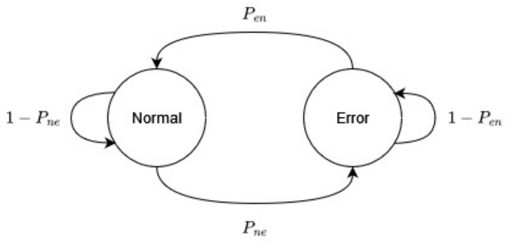

Fig. 4. Gilbert-Elliott channel model state diagram

errors, and an error state, where errors occur. By incorporating the geometric distribution, the model accurately captures the durations of each state, providing valuable insights into channel properties. This approach effectively simulated the erroneous DVL signal. The probabilities P_{ne} (transition from normal to error) and P_{en} (transition from error to normal) allows control over the duration and frequency of state changes.

3) Lawn mower path: To validate the technique, a lawn mower path was employed in a validation experiment. The ground truth trajectory of the AUV during the experiment is illustrated in Figs. 7, 8, and 9, representing the desired trajectory. The selection of this path is based on its frequent usage in AUV surveying operations. The path exhibits horizontal position variations while maintaining a fixed vertical position.

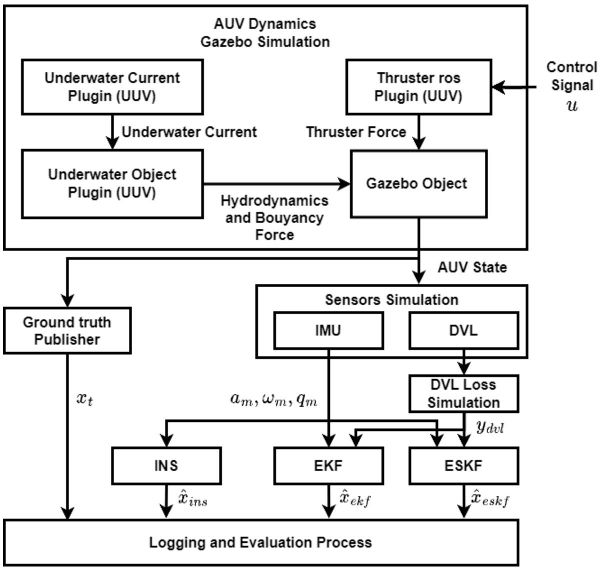

Fig. 5. Overview of AUV system using Gazebo simulation

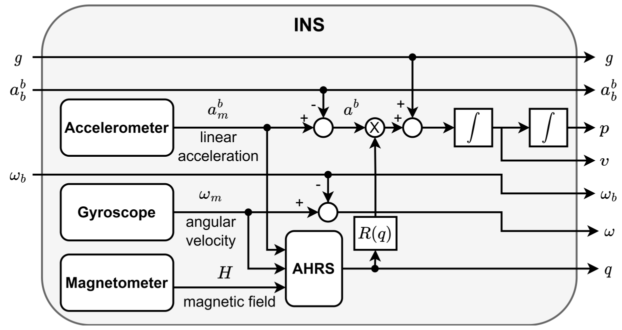

Fig. 6. INS system diagram

C. Implementation

The implementation of the underwater system, as depicted in Fig. 5, involves the crucial role of the Gazebo physics engine in generating the underwater environment for the AUV. Sensors like the INS and lossy DVL were simulated to evaluate the performance of the developed ESKF against other navigation techniques. The logged data provides valuable insights into the system's accuracy and reliability under varying conditions.

1) INS Method: The INS module is responsible for integrating the IMU measurements, which are provided by the Gazebo simulation, and estimating the vehicle's position and orientation, was implemented separately using a custom software solution. The development of INS followed the strapdown navigation approach. As illustrated in Fig. 6, the INS module gives the vehicle's estimated state $\widehat{\boldsymbol{x}}_{\text{INS}}$ including the estimated linear velocity, position, angular velocity, and orientation.

2) EKF Method: The EKF was developed to obtain $\widehat{\boldsymbol{x}}_{\text{EKF}}$ using the robot_localization package [12]. The package offers a comprehensive set of tools and algorithms for sensor fusion and state estimation in robotic systems, making it an excellent choice for implementing the EKF.

3) ESKF Method: First, the ESKF, as explained in Section II, was implemented to obtain $\widehat{\delta\boldsymbol{x}}_{\text{ESKF}}$ using the nominal state $\widehat{\boldsymbol{x}}_{\text{INS}}$ from the INS module. Then, in this ESKF method, the estimated state $\widehat{\boldsymbol{x}}_{\text{ESKF}}$ is found by adding the correction term $\widehat{\delta\boldsymbol{x}}_{\text{ESKF}}$ to the nominal state $\widehat{\boldsymbol{x}}_{\text{INS}}$, as shown below: $\widehat{\boldsymbol{x}}_{\text{ESKF}} = \widehat{\boldsymbol{x}}_{\text{INS}} + \widehat{\delta\boldsymbol{x}}_{\text{ESKF}}$.

IV. Result and Discussion

The experiment evaluated the impact of varying the error-to-normal state transition probability on DVL signal loss, comparing the INS, EKF, and ESKF navigation techniques. The tested probabilities were 0.01, 0.033, and 0.1, with a fixed error-to-normal transition probability of 0.1. Table I summarizes the horizontal position errors. The INS method yielded errors of 8.09741, 8.06606, and 7.73550; the EKF method reported errors of 0.08371, 0.13573, and 0.67363; and the ESKF method achieved the lowest errors of 0.04077, 0.04667, and 0.37127. Figs. 7, 8, and 9 visually depict the horizontal position plots for the different navigation methods. The green lines represent the output from the INS method, the orange lines indicate the EKF method, the blue lines indicate the ESKF method, and the red lines represent the ground truth signal.

The experiment results highlight the ESKF method's exceptional performance compared to the INS and EKF methods. The ESKF method achieved the lowest horizontal position errors, showing its robustness in estimating the vehicle's position even with the DVL signal loss. The ESKF method possesses several advantageous characteristics, such as its indirect state estimation approach, improved numerical stability, reduced computational complexity due to the operating point being close to zero, and its ability to effectively handle system nonlinearities. These key factors significantly contribute to its superior performance.

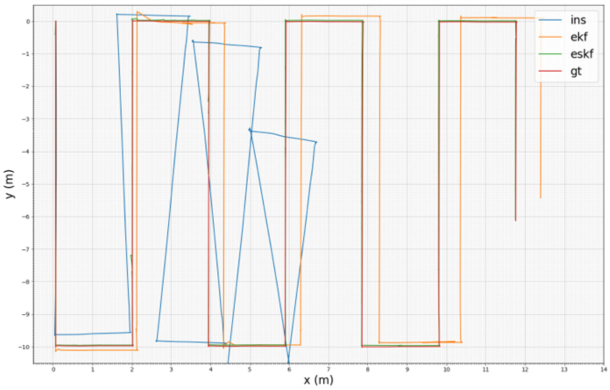

Fig. 7. XY-position of the AUV with $P_{ne} = 0.01$

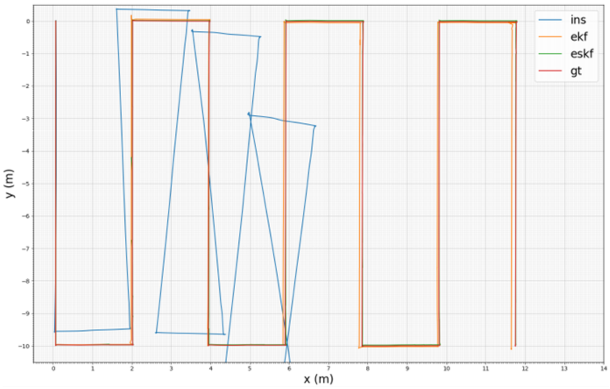

Fig. 8. XY-position of the AUV with $P_{ne} = 0.033$

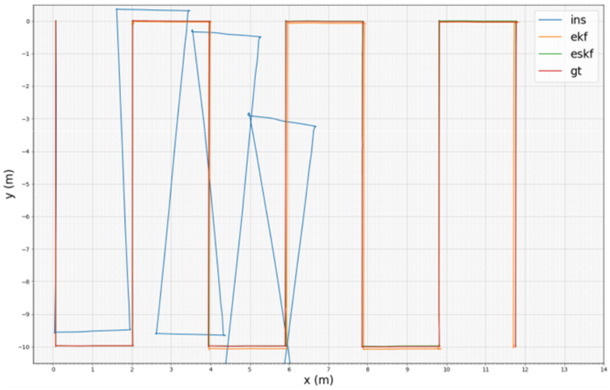

Fig. 9. XY-position of the AUV with $P_{ne} = 0.1$

TABLE I
HORIZONTAL POSITION ERRORS

Case	Method	Horizontal Position Error			
		Max	*Mean*	*SD*	*Final*
$P_{ne} = 0.01$ $P_{en} = 0.1$	**ESKF**	**0.51838**	**0.19195**	**0.13057**	**0.04077**
	EKF	0.64652	0.20032	0.13993	0.08371
	INS	8.22199	2.61992	2.34148	8.09741
$P_{ne} = 0.033$ $P_{en} = 0.1$	**ESKF**	**0.52441**	**0.19011**	**0.12986**	**0.04667**
	EKF	0.61178	0.19742	0.14906	0.13573
	INS	8.21971	2.61209	2.33278	8.06606
$P_{ne} = 0.1$ $P_{en} = 0.1$	**ESKF**	**0.50959**	**0.18163**	**0.12704**	**0.37127**
	EKF	1.70157	0.43583	0.24092	0.67363
	INS	7.73871	2.36539	2.16057	7.73550

V. CONCLUSION

Overall, the ESKF proved to be a highly effective choice for underwater state estimation, even with the variation of the DVL signal loss. The effectiveness in reducing errors validates its potential to enhance underwater navigation systems. For future work, the vehicle model or alternative sensors such as GNSS, pressure sensors, or hydroacoustic sensors can provide additional data sources to improve accuracy. Furthermore, implementing output smoothing techniques can minimize sudden jumps and discontinuities of the navigation output, resulting in more consistency of the vehicle states. Implementing these enhancements can significantly improve underwater vehicles' output, thereby advancing their autonomy and capabilities.

ACKNOWLEDGMENT

This work would not be possible without the support of Rovula (Thailand) Co., Ltd. The Faculty of Engineering and Chulabhornwalailak Swimming Pool, Sport, Art, and Culture Division, Kasetsart University were also acknowledged for their facilities.

REFERENCES

[1] J. C. Kinsey, R. Eustice, and L. L. Whitcomb, "A Survey of Underwater Vehicle Navigation: Recent Advances and New Challenges," *7th Conference on Manoeuvring and Control of Marine Craft (MCMC'2006)*, pp. 1–12, 11 2006.
[2] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*, 2nd ed. John Wiley & Sons, Ltd, 2021.
[3] Hegrenaes and O. Hallingstad, "Model-aided INS with sea current estimation for robust underwater navigation," *IEEE Journal of Oceanic Engineering*, vol. 36, no. 2, pp. 316–337, 4 2011.
[4] J. Solà, "Quaternion kinematics for the error-state Kalman filter," 11 2017. [Online]. Available: https://arxiv.org/abs/1711.02508v1
[5] L. VectorNav Technologies, *Inertial Navagation Primer*. VectorNav, 2020.
[6] Teledyne Marine, "Explorer Operation Manual," 2014.
[7] K. Lu, D. Tang, and K. Kou, "Analysis and Evaluation of MEMS-IMU Attitude Estimation Algorithm," in *Advances in Guidance, Navigation and Control*, L. Yan, H. Duan, and Y. Deng, Eds. Singapore: Springer Nature Singapore, 2023, pp. 3110–3119.
[8] P. G. Savage, *Strapdown Analytics*, 2nd ed. Strapdown Associates, Inc., 2007.
[9] "Gazebo." [Online]. Available: https://gazebosim.org/home
[10] "GitHub - Liquid-ai/Plankton: Open source simulator for maritime robotics researchers." [Online]. Available: https://github.com/Liquid-ai/Plankton
[11] A. Goldsmith, *Wireless communications*. Cambridge University Press, 1 2005, vol. 9780521837.
[12] T. Moore and D. Stouch, "A Generalized Extended Kalman Filter Implementation for the Robot Operating System," in *Intelligent Autonomous Systems 13*, E. Menegatti, N. Michael, K. Berns, and H. Yamaguchi, Eds. Cham: Springer International Publishing, 2016, pp. 335–348.