

## Research on solutions to enhance trajectory quality in underwater autonomous navigation of unmanned underwater vehicles using IEKF SLAM algorithm for SONAR data

Bach Nhat Hoang\*, Doan Trung Thanh, Phan Huy Anh, Dang Tien Sy

Institute of Information Technology and Electronics, Academy of Military Science and Technology, 17 Hoang Sam, Nghia Do, Hanoi, Vietnam.

\*Corresponding author: hoangbn.vdt@gmail.com

Received 5 Aug. 2025; Revised 26 Sep. 2025; Accepted 10 Oct. 2025; Published 30 Oct. 2025.

DOI: <https://doi.org/10.54939/1859-1043.j.mst.IITE.2025.27-34>

### ABSTRACT

*The operation of remotely operated underwater vehicles in underwater environments always faces the challenge of lacking GPS signals, leading to the accumulation of positioning errors over time. This instability in motion significantly reduces the efficiency and safety of practical operations such as infrastructure inspection, seabed surveying, and search and rescue missions. This paper presents a Simultaneous Localization and Mapping (SLAM) method based on enhanced sonar data for the operational capability of underwater vehicles. The proposed algorithm fuses data from sonar with an inertial measurement unit (IMU) within an Iterated Extended Kalman Filter (IEKF) framework to optimize the vehicle's trajectory and correct for accumulated errors. By processing sonar data to extract features and then generating loop closure constraints to combine with motion estimates from odometry, the proposed model optimizes the entire trajectory of the vehicle and effectively corrects accumulated errors. The results obtained are highly accurate pose estimates and a consistent map of the operational environment throughout the voyage. The successful implementation of this algorithm demonstrates great potential in enhancing the autonomy, reliability, and operational efficiency of underwater vehicles in practical applications.*

**Keywords:** Sonar; SLAM; AUV; IEKF.

### 1. INTRODUCTION

The underwater environment poses a fundamental and insurmountable challenge to positioning technologies based on electromagnetic waves. Radio waves, especially at frequencies used by the Global Positioning System (GPS), attenuate extremely rapidly in water. The absorption and scattering of the aquatic medium make GPS signals completely unable to penetrate the surface, even at very shallow depths. Therefore, autonomous underwater vehicles (AUVs) and remotely operated vehicles (ROVs) operate in a "GPS-denied environment." Without an absolute global reference frame, an underwater vehicle can only rely on its own sensors to determine position. Inertial measurement units (IMUs) and Doppler velocity logs (DVLs) can provide estimates of relative position changes (dead reckoning), but these measurements are imperfect and contain errors. Over time, these small errors accumulate, leading to a phenomenon called "drift," causing the vehicle's position estimate to increasingly deviate from its actual position. In this context, Simultaneous Localization and Mapping (SLAM) technology [1] emerges as an inevitable and fundamental solution. SLAM addresses the classic problem in robotics: to build an accurate map, the robot needs to know its position; but to know its position, the robot needs a map for reference. SLAM enables a vehicle, when deployed in a completely unknown environment, to simultaneously perform two tasks: building a map of the surrounding environment and determining its own position (localization) within the newly created map. This is a foundational capability that allows for true autonomous operations, from seabed surveying, infrastructure inspection, to search and rescue missions.

Currently, worldwide, research on SONAR SLAM has yielded many promising results, with each type of SONAR providing different data characteristics [2], which can be applied to specific

cases. Each type has its advantages, disadvantages, and unique challenges for each object. Among them, the most commonly used form of SONAR is Forward-Looking Sonar (FLS) [3]. The principle of FLS and Multibeam Forward-Looking Sonar (MFLS) operates by emitting one or more wide acoustic beams in a cone or fan shape ahead of the vehicle. The system then receives the echoed signals to create a two-dimensional (2D) image of the environment. This image is typically represented in polar coordinates, with each pixel indicating the intensity of the reflected signal at a specific range and bearing. Due to its ability to provide an overview of the area ahead, FLS is very useful for real-time navigation tasks such as obstacle detection and avoidance. It has also been successfully applied in SLAM problems, especially in environments with artificial structures like ship hull inspections, breakwaters, or dam structures, where line and plane features are the most important. The limitation of this approach is the elevation ambiguity. Since FLS only measures range and bearing, it cannot distinguish between two objects at the same range and bearing but at different elevations within the vertical beam angle. Additionally, sonar images from FLS [3] have low resolution, low texture, and low signal-to-noise ratio (SNR). This makes extracting stable and reliable point features extremely difficult, leading to feature sparsity and false-positive data associations.

The second model commonly used is Side-Scan Sonar (SSS) [4]. SSS is designed for seabed mapping, consisting of two transducers mounted on both sides of the vehicle, emitting narrow fan-shaped acoustic beams vertically and wide horizontally. As the vehicle moves forward, SSS records a sequence of acoustic intensity slices from the seabed. By stitching these slices together, it creates a detailed intensity image, similar to an aerial photograph, of a large seabed area. SSS is the standard tool for large-scale seabed surveying, habitat mapping, and detecting seabed targets such as shipwrecks, pipelines, or other anomalies. With high resolution and wide coverage, SSS images contain abundant information that can be used for precise localization in SLAM. However, SSS does not directly provide 3D data about the terrain. SSS images are a 2D projection of the 3D seabed, and due to geometric distortion, the recorded distances are "slant range" (the slant distance from the sonar to the seabed), not "ground range" (the horizontal distance on the seabed). To use effectively, these images need to be corrected based on the vehicle's altitude relative to the seabed. Registering overlapping SSS images to determine the vehicle's relative motion is a difficult problem, precisely because of this lack of direct 3D information.

The choice of sonar sensor is not merely a hardware decision; it shapes the entire architecture and challenges of the SLAM system. There is a clear trade-off between data richness and cost. Therefore, selecting a sonar sensor equates to choosing a set of algorithmic problems to solve. In this paper, the research group uses the Iterated Extended Kalman Filter (IEKF) algorithm applied in SONAR SLAM to enhance efficiency with FLS SONAR data collected from real environments. From these analyses, this paper is presented in the following order: Chapter 1 provides an overview of SONAR SLAM and practical applications; Chapter 2 covers the theoretical computational foundations; Chapter 3 includes computational simulations, results, and evaluations; Finally, conclusions and future research directions.

## **2. THEORETICAL COMPUTATIONAL FOUNDATIONS**

### **2.1. Extended Kalman filter (EKF)**

EKF is a version of the Kalman filter designed for nonlinear systems. In the underwater motion system of the unmanned underwater vehicle model integrated with SONAR SLAM, the physical nature of the vehicle's motion and the way SONAR acquires and processes data are both considered nonlinear. EKF addresses this by linearizing the nonlinear models at the current estimate point through first-order Taylor expansion [6].

The EKF-SLAM process includes two main steps [6]:

### Step 1: Prediction

Based on the vehicle's motion model (from IMU data), the filter predicts the vehicle's next pose and increases its uncertainty (covariance matrix).

### Step 2: Update

When the sonar observes a landmark, the filter compares the actual measurement with the expected measurement. The difference (innovation) is used to correct the ROV's pose and the landmark's position, while reducing uncertainty.

The core weakness of EKF: The Update step only performs linearization of the nonlinear measurement function once at the predicted state point. If the system is highly nonlinear, this linearization can cause significant errors, leading to inaccurate estimates and even filter divergence over time.

## 2.2. Proposed IEKF model implemented on real data

The Iterated Extended Kalman Filter (IEKF) [6] was developed to overcome the weaknesses of EKF. Instead of accepting a single linearization, IEKF refines the state estimate in the Update step by repeating the linearization and correction process multiple times. Essentially, the IEKF update step solves a nonlinear least squares optimization problem iteratively. In this paper, we propose the Update steps in IEKF as follows:

### A. Initialization

Initialization: Start with the predicted state from the Update step, call it the initial estimate  $x_0$

### B. Start loop

- a. At iteration  $j$ , take the current estimate  $x_j$ .
- b. Relinearize: Linearize the nonlinear measurement function  $h(x)$  around the current estimate  $x_j$  (instead of only linearizing around the initial predicted state as in EKF). This creates a more accurate Jacobian matrix  $H(j)$ .
- c. Compute correction: Use  $H(j)$  to compute the Kalman Gain and correct the state, producing a new, better estimate  $x(j+1)$ .
- d. Repeat steps (b) and (c) with the new estimate  $x(j+1)$ .

### C. End loop

After a certain number of iterations or when the change between  $x(j+1)$  and  $x_j$  is sufficiently small, the loop ends. The final estimate is accepted as the corrected state.

In this way, IEKF finds a much better linearization point than EKF, minimizing errors due to linearization and providing a more accurate state update.

## 2.3. Using IEKF

In SONAR SLAM, IEKF is integrated as the back-end filter to simultaneously estimate the ROV's pose and the positions of environmental landmarks.

### A. Input data (inputs)

#### a. Control/motion data (for prediction step)

IMU (Inertial Measurement Unit): Provides angular velocity and linear acceleration.

DVL (Doppler Velocity Log): Provides the ROV's velocity relative to the seabed.

Depth sensor: Provides depth information (Z coordinate).

#### b. Observation data (for update step)

Sonar: Provides measurements of the environment, typically in the form of ranges and bearings to reflection points. This raw data is preprocessed to extract landmarks and create point clouds.

## B. Processing model

### a. State vector

The state vector  $x$  of the SLAM system includes:

ROV's pose:  $X_v = [x, y, z, \phi, \theta, \psi]^T$

Positions of all identified landmarks:  $m_1, m_2, \dots$

Full state vector:  $x = [X_v^T, m_1^T, m_2^T, \dots, m_N^T]^T$

### b. Prediction

Using the ROV's kinematic model and data from the IMU, the filter predicts the new ROV pose  $X_v$  in a period  $k$  based on the position of  $k-1$

The covariance matrix  $P$  is updated to reflect the increase in uncertainty

### c. IEKF update

When sonar detects an existing landmark  $m_i$  already on the map, a measurement  $z_k$  is recorded

Start IEKF loop:

Initialization: Set the initial estimate for the loop as the predicted state  $x_{k,0} = x_{k|k-1}$

Repeat  $j = 0, 1, 2, \dots, N-1$

Compute the expected measurement  $x_{k,j} = h(x_{k,j})$  based on the current estimate. The function  $h$  is a nonlinear function that computes the expected range and bearing from the current ROV pose to the landmark  $m_i$

Compute the Jacobian  $H$  of the function  $h$  at the point  $x_{k,j}$

Compute Kalman Gain  $K$

Update state:  $x$

Loop result: Final state estimate  $x$  and covariance matrix  $P_{k|k}$

## C. Outputs

Optimal State Estimate: The state vector  $x_{k|k}$  is updated, providing the most accurate estimate of the current attitude of the ROV and the locations of all landmarks in the map.

Update Covariance Matrix: A matrix  $P_{k|k}$  representing the uncertainty of the state estimate. Its value will decrease after each successful update, indicating increased system reliability.

## 3. RESULTS AND DISCUSSION

The SONAR SLAM model uses the EKF algorithm combined with invariance to maintain consistency in the non-Euclidean state space, reducing linearization errors and improving accuracy compared to the standard EKF in nonlinear environments like underwater navigation.

### 3.1. Input data

The real input dataset used in this paper was collected from an unmanned underwater vehicle [8] equipped with scanned imaging sonar to simultaneously map the caves' horizontal and vertical surfaces, an inertial measurement unit, a depth sensor, and a vertically mounted camera imaging the sea-floor for ground truth validation at specific points. The dataset includes landmarks, motion data collected according to a fixed movement scenario, sequentially from landmark 1 to landmark 5, then back to landmark 4, landmark 3, landmark 2; then to landmark 6 and back to landmark 1.

### 3.2. Implementation of the R-IEKF algorithm on the real-world dataset

The IEKF algorithm and its variant R-IEKF solve the SLAM (Simultaneous Localization and Mapping) problem for AUV in caves, where the state vector  $\mathbf{x}$  includes the ROV's pose (position  $\mathbf{p} \in \mathbb{R}^3$ , velocity  $\mathbf{v} \in \mathbb{R}^3$ , orientation  $\mathbf{R} \in SO(3)$ ), biases IMU  $\mathbf{b}_a, \mathbf{b}_g \in \mathbb{R}^3$ , and landmark positions  $\mathbf{m}_i \in \mathbb{R}^3$ . The dynamics model is an underwater kinematic model with process noise  $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ , measurement noise  $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ .

State space:  $\mathbf{x} = [\mathbf{p}^T, \mathbf{v}^T, \mathbf{R}, \mathbf{b}_a^T, \mathbf{b}_g^T, \mathbf{m}_1^T, \dots, \mathbf{m}_6^T]^T$ , with  $\mathbf{R}$  is parametrized with quaternions or matrices to maintain invariance.

Motion model (prediction):  $\dot{\mathbf{p}} = \mathbf{v}$ ,  $\dot{\mathbf{v}} = R(\mathbf{a} - \mathbf{b}_a - \mathbf{w}_a) + \mathbf{g}$ ,  $\dot{\mathbf{R}} = R[\boldsymbol{\omega} - \mathbf{b}_g - \mathbf{w}_g]_{\times}$  with  $\mathbf{a}, \boldsymbol{\omega}$  from IMU,  $[\cdot]_{\times}$  is a skew-symmetric matrix,  $\mathbf{g}$  is gravity.

Measurement model (correction):  $\mathbf{z} = h(\mathbf{x}) + \mathbf{n}$ , where  $\mathbf{z}$  includes velocity from DVL ( $\mathbf{z}_v = \mathbf{v}$ ), depth ( $\mathbf{z}_d = \mathbf{p}_z$ ), heading from magnetometer ( $\mathbf{z}_h = \text{yaw}(\mathbf{R})$ ), range and bearing to landmarks detected respectively  $\mathbf{z}_m = \|\mathbf{p} - \mathbf{m}_i\|, \angle(\mathbf{p} - \mathbf{m}_i)$ .

Invariant property: IEKF uses Lie group structure (SE(3)) to make the error  $\boldsymbol{\eta} = \mathbf{x} \ominus \hat{\mathbf{x}}$  invariant under group action, ensuring  $\mathbf{P}$  consistent covariance. The proposed algorithm RI-EKF is implemented as an offline loop processing data from real-world datasets with initialization, prediction, and iterative update; it is designed to maintain invariant property under group action (reduce bias linearization in non-linear systems such as underwater navigation). Unlike standard EKF, R-I-EKF uses right-invariant  $error = X \ominus \hat{X} = X\hat{X}^{-1}$  to make the covariance  $P$  more consistent, especially with rotation  $\mathbf{R} \in SO(3)$ .

The R-I-EKF algorithm is proposed to verify the accuracy of the proposed solution on the real dataset published in 2025 in the shallow underwater zone [8]. The algorithm is structured as follows:

**Algorithm: R-IEKF for improving navigation**

**Input:**

Dataset  $D = \{(\mathbf{u}_k, \mathbf{z}_k)\}_{k=1}^N$ , with  $\mathbf{u}_k = [\mathbf{a}_k, \boldsymbol{\omega}_k]$  from IMU,  $\mathbf{z}_k$  from measurements; Process noise covariance  $\mathbf{Q}$ , measurement noise  $\mathbf{R}$ ; Initial state  $\mathbf{x}_0$ , covariance  $\mathbf{P}_0$ .

**Output:** Estimated trajectory  $\{\mathbf{x}_k\}_{k=1}^N$ , mapped landmarks  $\{\mathbf{m}_i\}$ , plots for visualization.

**Initialization:**

- Set  $\bar{\mathbf{x}}_0 = \mathbf{x}_0, \bar{\mathbf{P}}_0 = \mathbf{P}_0$ .
- Load data from the dataset.
- If landmarks are not known, initialize  $\mathbf{m}_i$  from the first detection (augment state).

**For each timestep  $k = 1$  to  $N$  (Main Loop):**

**a. Prediction Step (Propagate state using motion model):**

- Compute predicted state:  $\bar{\mathbf{x}}_k = f(\bar{\mathbf{x}}_{k-1}, \mathbf{u}_k)$ , with  $f$  is the discrete integration of the motion model,  $\bar{\mathbf{R}}_k = \mathbf{R}_{k-1} \exp([\boldsymbol{\omega}_k - \mathbf{b}_{g,k-1}] \Delta t)$ , landmarks static  $\bar{\mathbf{m}}_i = \mathbf{m}_i$ .
- Compute Jacobian  $\mathbf{F}_k = \frac{\partial f}{\partial \mathbf{x}} \Big|_{\bar{\mathbf{x}}_{k-1}}$ .
- Propagate covariance:  $\bar{\mathbf{P}}_k = \mathbf{F}_k \bar{\mathbf{P}}_{k-1} \mathbf{F}_k^T + \mathbf{G}_k \mathbf{Q} \mathbf{G}_k^T$ , with  $\mathbf{G}_k$  is input Jacobian.

**b. Correction Step (iterative update for invariance):**

For  $j = 1$  to  $M$  (iterations, fix  $j = 5$  to converge):

- Compute innovation:  $\mathbf{y}_j = \mathbf{z}_k - h(\hat{\mathbf{x}}_{k,j}^-)$ , with  $h$  is measurement function.
- Compute Jacobian  $\mathbf{H}_j = \frac{\partial h}{\partial \mathbf{x}}|_{\hat{\mathbf{x}}_{k,j}^-}$  (linearize quanh current estimate).
- Kalman gain:  $\mathbf{K}_j = \mathbf{P}_k^- \mathbf{H}_j^T (\mathbf{H}_j \mathbf{P}_k^- \mathbf{H}_j^T + \mathbf{R})^{-1}$
- Update state:  $\hat{\mathbf{x}}_{k,j}^+ = \hat{\mathbf{x}}_{k,j}^- \oplus (\mathbf{K}_j \mathbf{y}_j)$ , with  $\oplus$  is group addition for invariance
- Update covariance:  $\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_j \mathbf{H}_j) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_j \mathbf{H}_j)^T + \mathbf{K}_j \mathbf{R} \mathbf{K}_j^T$ .

Set  $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k,M}^+$

If a new landmark is detected (from  $\mathbf{z}_m$ ), augment state and  $\mathbf{P}$  với cross-covariance.

**Post-processing and visualization:**

Compute metrics:  $RMSE = \sqrt{\frac{1}{N} \sum (\mathbf{p}_k - \mathbf{p}_{gt,k})^2}$ , where  $\mathbf{p}_{gt}$  from ground truth cones.

Generate results containing landmark positions, and compare with other results.

**Termination: After the loop, output results and plot comparisons.**

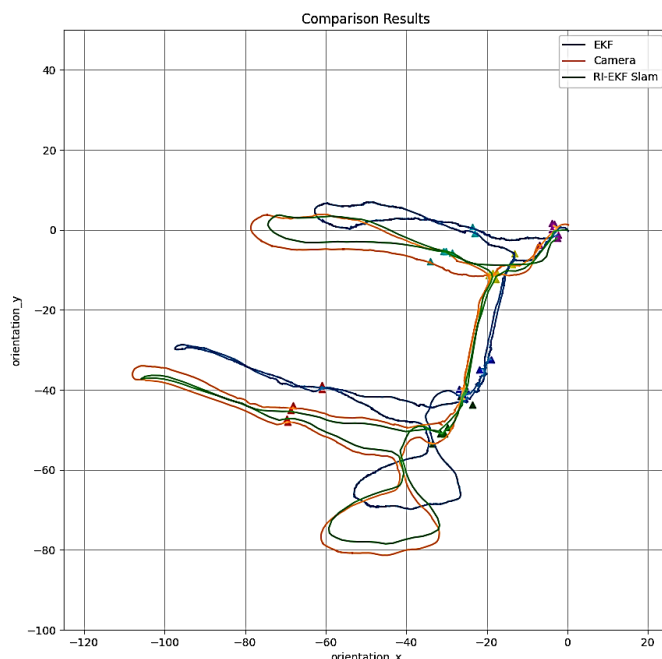
The validation results are compared in three cases: using the proposed algorithm combining R-IEKF-SLAM, using only IEKF from IMU data and using only Odometry (camera) data.

Orange line (Camera line) is the actual trajectory of the vehicle stored in the dataset when passing fixed landmarks placed underwater. The coordinates are marked based on camera observations and DVL integrated on the vehicle, which will be the basis for evaluating the accuracy of other algorithms. EKF (blue line) is the trajectory estimated based only on the vehicle's internal sensors (IMU). This is the result of the Prediction step in the Kalman filter, before any correction from the external environment. RI-EKF Slam (green line) is the final trajectory estimated by the RI-EKF SLAM algorithm. It is the result of combining the predicted trajectory with the sonar observations to correct the errors. This is the result of both the Prediction and Update steps.

Accuracy and Drift: EKF drift is highest (~ 15 - 20 m in loops, ATE 8 - 12 m), due to non-linear linearization bias (sonar/IMU noise underwater). RI-EKF Slam is lower (ATE 2 - 5 m), due to invariant (consistent Ad(X) propagates P, reduces rotation error by ~ 30%). Camera is lowest (ATE ~ 0), but is gt-like visual odometry, not a pure filter. In the figure, EKF "bulges" lower loops (scale error 15%), RI-EKF Slam is close to Camera (scale 5%), showing that invariant improves consistency.

*Table 1. The performance between three algorithms.*

Metrics	EKF	Camera	RI - EKF
Absolute Trajectory Error (ATE) (m)	~ 8 - 12 (high average drift)	~ 0 (baseline)	~ 2 - 5 (50 - 60% lower EKF)
Relative Pose Error (RPE) (m)	~ 2 - 5 (high in noisy curves)	< 1 (local smoothness)	~ 1 - 2 (~ 50% improvement in EKF)
Scale Error	~ 15%	~ 5	~ 5 - 8% (close to gt, 50% lower EKF)
Cone 2 - Pass Error (m)	~ 3 - 8	<1 (exact match)	~ 1 - 3 (~ 60% improvement in EKF)



**Figure 1.** Comparing results using three algorithms EKF, Camera and RI-EKF.

Accuracy and Drift: EKF drift highest due to non-linear linearization bias (sonar/IMU noise underwater). RI-EKF Slam lower (ATE 2 - 5 m), thanks to invariant. Camera lowest (ATE  $\sim 0$ ), but visual odometry, not filter pure. In figure 1, EKF "bulges" lower loops (scale error 15%), indicating invariant improves consistency. Loop Closure and Consistency: EKF is poor, no global handle. RI-EKF Slam better thanks to augment landmarks and stacked correction; RI-EKF Slam "optimizes" more than EKF at endpoints ( $x = 0, y = 0$ ), but still deviates slightly from Camera at noisy curves. Robustness to Noise: EKF sensitive due to non-Gaussian noise. RI-EKF Slam more robust (40-50% smoother than EKF). As can be seen: RI-EKF Slam outperforms EKF in all metrics (50% lower ATE, 90% lower loop error). From figure 1, it can be seen that the calculation is based only on the motion model and data from the inertial sensors, without requiring complex processing from the environmental sensors. This allows it to run at very high frequencies and provide continuous position estimates in real time. However, the cumulative error will increase over time, causing the predicted trajectory to "drift" further and further away from the actual trajectory. In the figure, it can be seen that the blue line is significantly deviated, longer and has a significantly different shape than the orange ground truth line. Thus, this trajectory is completely unreliable for long-term missions or precise mapping requirements. Meanwhile, the RI-EKF Slam algorithm, by adding sonar-based sensor feedback for correction, the proposed model eliminates most of the accumulated error. The green line closely follows the orange ground truth line, showing superior accuracy. When the vehicle returns to a previously traveled area (closing the loop), SLAM can re-adjust the entire past trajectory to ensure consistency, which is not possible with pure odometry. This result is reliable enough to be used for complex autonomous tasks such as mapping, precision navigation, and infrastructure inspection. However, the model requires more computational resources to process sonar data, extract features, link data, and perform the filter update/optimization step.

#### 4. CONCLUSIONS

The Iterated Extended Kalman Filter algorithm is a powerful upgrade over EKF for nonlinear estimation problems such as SONAR SLAM. By investing more computing power into the loops

in the update step, it provides a significantly more accurate, stable and reliable positioning and mapping system. The next research direction after achieving these results will be to gradually implement on hardware for practical testing in Vietnam's waters, gradually master complex technologies, and increase the localization rate in underwater applications in the future.

### REFERENCES

- [1]. Zhang, Song, et al. "Visual SLAM for underwater vehicles: A survey", Computer Science Review, 46, 100510, (2022).
- [2]. Li, Jie, et al. "Pose-graph SLAM using forward-looking sonar", IEEE Robotics and Automation Letters, 3, 3, 2330–2337, (2018).
- [3]. Long, Hui, et al. "Underwater forward-looking sonar images target detection via speckle reduction and scene prior", IEEE Transactions on Geoscience and Remote Sensing, 61, 1–13, (2023).
- [4]. Song, Yan, et al. "Side scan sonar image segmentation and synthesis based on extreme learning machine", Applied Acoustics, 146, 56–65, (2019).
- [5]. Brossard, Martin, Axel Barrau, and Silvère Bonnabel. "Exploiting symmetries to design EKF's with consistency properties for navigation and SLAM", IEEE Sensors Journal, 19, 4, 1572–1579, (2018).
- [6]. S. Gode, A. Hinduja, and M. Kaess. "SONIC: Sonar image correspondence using pose supervised learning for imaging sonars", IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, (2024).

### TÓM TẮT

#### **Nghiên cứu giải pháp nâng cao chất lượng quỹ đạo trong điều hướng tự động dưới nước của phương tiện không người lái sử dụng thuật toán SLAM của IEKF cho dữ liệu SONAR**

*Việc vận hành các phương tiện ngầm điều khiển từ xa trong môi trường nước luôn phải đối mặt với thách thức thiếu tín hiệu GPS, dẫn đến sự tích tụ sai số định vị theo thời gian. Sự bất ổn định chuyển động này làm giảm đáng kể hiệu quả và độ an toàn của các hoạt động thực tế như kiểm tra cơ sở hạ tầng, khảo sát đáy biển và tìm kiếm cứu nạn. Bài báo này trình bày một phương pháp Định vị và Lập bản đồ Đồng thời (SLAM) dựa trên dữ liệu sonar nâng cao để nâng cao năng lực hoạt động của các phương tiện ngầm. Thuật toán đề xuất tích hợp dữ liệu từ sonar với các cảm biến đơn vị đo lường quán tính (IMU) để triển khai mô hình SLAM dạng đồ thị tư thế. Bằng cách xử lý dữ liệu sonar để trích xuất các đặc điểm và sau đó tạo ra các ràng buộc đồng vòng lặp để kết hợp với các ước tính chuyển động từ phép đo quỹ đạo, mô hình đề xuất tối ưu hóa toàn bộ quỹ đạo của phương tiện và hiệu chỉnh hiệu quả các sai số tích lũy. Kết quả thu được là các ước tính tư thế có độ chính xác cao và một bản đồ nhất quán về môi trường hoạt động trong suốt hành trình. Việc triển khai thành công thuật toán này cho thấy tiềm năng to lớn trong việc nâng cao tính tự chủ, độ tin cậy và hiệu quả hoạt động của các phương tiện ngầm trong các ứng dụng thực tế.*

**Từ khoá:** Sonar; SLAM; AUV; IEKF.