

EECS 568 Group 12 Final Project:

Invariant Extended Kalman Filter for Localization in Underwater Caves

Samuel Ansaldo, AJ Bull, Xinyu Ma, Alyssa Scheske, Shane Storks

Abstract—As Autonomous Underwater Vehicles (AUVs) become more affordable and ubiquitous, their usage in complex underwater environments will become more desirable for inspecting, monitoring, etc. Due to the unavailable GPS signal in underwater condition, localization of the robot will be a challenge. Thus, a Right-Invariant Extended Kalman Filter (RI-EKF) is proposed to estimate the position, with the additional integration of multiple sensors into the correction step for a better model of the nonlinear uncertainty in robot motion.

Keywords: AUVs, RI-EKF, localization, integrated sensors

I. INTRODUCTION

AUV technology has been rapidly developing in recent years. Their widespread applications include search and rescue, scientific surveying, or even oil rig inspections [5]. As the environment in confined areas like caves may be dark and full of complex, unknown obstacles, localization is a challenge [4]. This particular dataset was recorded by an AUV in underwater caves off the southern coast of Spain in July 2013 and originally incorporated an algorithm by Mallios et al [4] using simultaneous localization and mapping (SLAM) to tackle this problem; however, they only use a simple Kalman filter for localization. While work toward robotic design [6] and motion planning [3] has recently advanced the line of work on this problem, the localization approach has not been revisited. This work explores a more powerful localization approach. An invariant extended Kalman filter (InEKF) could make use of the fact that geometry of robot motion is a manifold, so the linearized model becomes a better approximation to the nonlinear one within the set of trajectories. Additionally, this filter combines contact-inertial dynamics with forward kinematic corrections to estimate pose and velocity along with all current contact points. [8]. By using a right-invariant extended Kalman filter (RI-EKF), the error dynamic modeling uses long-linear differential equations for improved convergence. This method is novel to the underwater community and is able to be generalized for AUV exploration, especially in confined underwater environments where surfacing or predeployment of localization equipment is not feasible.

In this work, an RI-EKF incorporates several onboard sensor inputs to localize a AUV in an underwater cave system. The results are compared to a visual-odometry estimate and a pose-based graph SLAM approach.

The remainder of this report is organized as follows: project contributions are given in Section II, Section III provides material needed for understanding the underwater caves data set. Section IV describes the methodology of

the work, breaking down the process into preprocessing and RI-EKF derivations. In Section V, the RI-EKF results are presented and pitted against previous works. Lastly, the discussion of the work and future directions are elaborated in Section VI.

II. PROJECT CONTRIBUTIONS

The high-level contributions of this project are:

- 1) RI-EKF for localization in underwater caves. This required careful derivation of the propagation and correction of error dynamics. Plots and videos of the results are provided.
- 2) Comparison of the results with the onboard odometry estimate from the dataset, and a SLAM benchmark from [4] which provides a strong baseline for localization.

III. DATASET: UNDERWATER CAVES

The primary dataset for this project was the underwater caves sonar and vision dataset [5]. This data was taken with an AUV in an underwater cave complex in Costa Brava, Spain. The vehicle was guided through the cave by a diver during data collection. Data was collected from the following sensors:

- DS2806 HPS-A pressure sensor (which estimates AUV depth)
- LinkQuest NavQuest 600 Doppler velocity log (DVL)
- ADIS16480 inertial measurement unit (IMU), located on the bottom of the AUV's hull
- Xsens MTi IMU, located inside the AUV
- Tritech Micron DST sensor beam (mechanically scanned sonar)
- Tritech Super SeaKing DFP profile sensor beam (mechanically scanned sonar)
- Vertically mounted camera, calibrated in a pool offsite

It is difficult to obtain absolute ground truth in an underwater environment. The dataset includes six traffic cones as ground truth points for relative accuracy estimation. The distance between the cones was manually measured by divers when preparing the on-site localization objects (Figure 1). Placing the traffic cones at strategic locations along the AUV trajectory allowed the vehicle to pass over each cone twice, especially helpful for loop closures used in the original work's SLAM implementation. The cones' relative position can be obtained by extracting the timestamps from the video frames where the cones appear closest to the center of the image and compare them with the timestamps of the trajectory. The distance of the cone from the reference frame

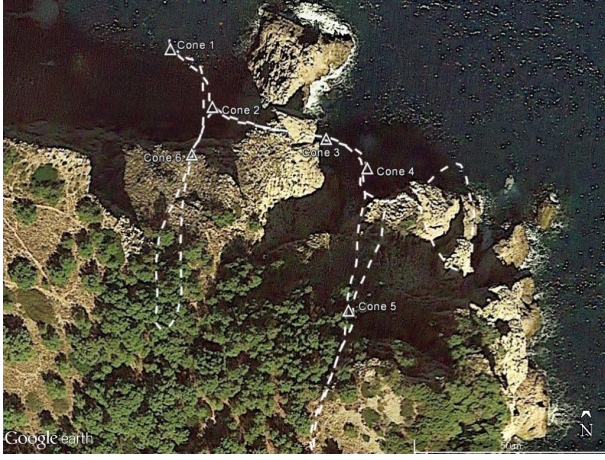


Fig. 1. Overhead view of cone positions used to generate the ground truth visual-odometry route data. Copied from <https://cirs.udg.edu/caves-dataset/>.

of the robot can be obtained by using an existing camera calibration.

The IMU data includes accelerometer, gyroscope, magnetometer, and temperature data. Three dimensional bias on the gyroscope is also reported at each time stamp. The IMU provides an on-board calculation of the AUV's estimated angular velocity and linear acceleration in three dimensions.

As the dataset contains a variety of sensor data verified by human observation, it makes a robust set to validate this project's RI-EKF. Other projects have tried different localization methods on this set and provide decent benchmarks.

IV. METHODS

The code behind this filter is published in a public GitHub repository [1].

A. Dataset Preprocessing

The dataset is provided in ROS bag files and CSV files. The data loader implements code for both versions, but the CSV version is used in the final implementation of the algorithm, which runs as a post-process. The dataset contains both raw sensor data and the preprocessed optometry data used for bench-marking. Figure 2 shows the available sensors on the AUV, and the frames of reference that their measurements come in on.

All of the sensor data used were transformed to the IMU frame, which was designated the AUV frame. An Euler angle transform of $[90^\circ \ 0 \ 180^\circ]$ in $[Z \ Y \ X]$ was applied to the DVL measurement vector. Depth and magnetometer were transformed within the lie algebra calculations and are discussed further in their respective sections.

Gyroscope bias was already incorporated into the data set and a small manual IMU z -axis rotation bias was added in fine tuning.

All sensor data was collected aboard the AUV while running a synchronized ROS system; each data point has a corresponding system time stamp in ROS time. ROS time can be converted to clock time by dividing by nanoseconds.

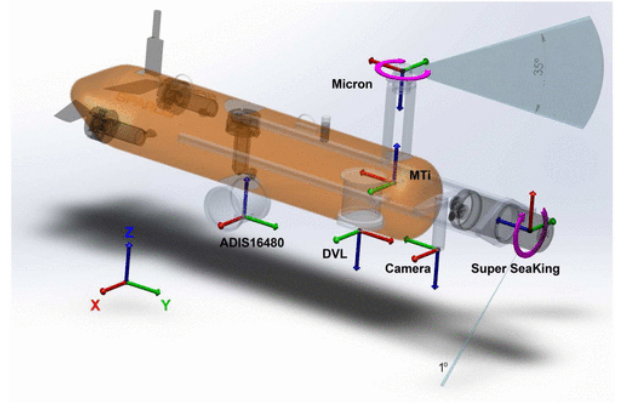


Fig. 2. Diagram of sensors and their frames of reference on the AUV used to collect the dataset for this work. Copied from [5].

It was important to capture the difference in time, or the "delta t ", between the last and current sensor readings in a global sense to be used in the calculations. Each individual sensor recorded at a regular frequency, but they did not consistently align, thus the global time stamp had to be utilized. Processing the data in chronological order is crucial to yield the best state estimation. Thus, a prediction computation was only executed if the succeeding time stamp's data point was classified as an accelerometer or gyroscope data point from the IMU. Likewise, a correction step was completed only if the next data point was from the DVL, depth sensor or magnetometer. Note that since the magnetometer was integrated into the IMU, each had correlating time stamps and would run complimenting prediction and correction calculations.

B. RI-EKF Derivation

Derivations from Hartley, Ghaffari, Eustice, and Grizzle inform the filter dynamics, corrections, Jacobians, and adjoint equations [2]. The robot state is represented by \mathbf{X}_k , a modified version of the standard homogeneous representation in $SE(3)$. Any timestep k is expressed as a 5×5 matrix:

$$\mathbf{X}_k = \begin{bmatrix} \mathbf{R}_{k3 \times 3} & \mathbf{v}_{k3 \times 1} & \mathbf{p}_{k3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 & 0 \\ \mathbf{0}_{1 \times 3} & 0 & 1 \end{bmatrix} \quad (1)$$

Here, \mathbf{R}_k is the 3×3 rotation matrix, \mathbf{v}_k is the 3×1 robot velocity vector, and \mathbf{p}_k is the 3×1 robot position vector. The lie algebra is defined based on each of these components, forming the skew matrix.

$$\mathbf{u}_k^\wedge = \begin{bmatrix} \mathbf{0} & -x_2 & x_1 & x_3 & x_6 \\ x_2 & \mathbf{0} & -x_0 & x_4 & x_7 \\ -x_1 & x_0 & \mathbf{0} & x_5 & x_8 \\ 0 & 0 & 0 & \mathbf{0} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{0} \end{bmatrix} \quad (2)$$

Where x_0, x_1, x_2 represent rotational velocity, x_3, x_4, x_5 linear acceleration, and x_6, x_7, x_8 linear velocity. Using the variety of AUV sensor data, a correction can be made using RI-EKF predicted propagation and correction steps to estimate \mathbf{X}_k .

1) *Initialization*: The filter is initialized with the earliest depth sensor reading for $p_{0,z}$ and earliest orientation estimate from the IMU for \mathbf{R}_0 . The chosen world coordinate frame is the right-handed frame with x true north and z normal to the surface of the earth, with the origin at the surface of the water at the robot's starting position. There were no significant gains in performance when tuning the initial state covariance.

2) *Propagation*: The prediction step is performed using gyroscope and accelerometer measurements from the IMU. At every timestamp, these sensors read sensor measurements for the angular velocity $\boldsymbol{\omega}_k = [\omega_x \ \omega_y \ \omega_z]^T$ and the linear acceleration $\mathbf{a}_k = [a_x \ a_y \ a_z]^T$. Let the concatenation $\mathbf{u}_k = [\boldsymbol{\omega}_k \ \mathbf{a}_k \ \mathbf{0}_{1 \times 3}]^T$ be the full control inputs for the filter. As the data IMU data is relatively noisy it is first passed through a floating-mean low-pass filter.

The log-linear right-invariant dynamics A are defined as:

$$A = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{g}^\wedge & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix} \quad (3)$$

where $\mathbf{g} = [0 \ 0 \ -9.80665]^T$ is the acceleration due to gravity. The matrix representing the adjoint is given by Ad_{X_k} .

$$Ad_{X_k} = \begin{bmatrix} \mathbf{R}_k & \mathbf{0} & \mathbf{0} \\ \mathbf{v}_k^\wedge \mathbf{R}_k & \mathbf{R}_k & \mathbf{0} \\ \mathbf{p}_k^\wedge \mathbf{R}_k & \mathbf{0} & \mathbf{R}_k \end{bmatrix} \quad (4)$$

Given these inputs and the time difference Δt between the current measurement $k+1$ and the previous measurement k , the prediction step uses the following equations to estimate the state $\bar{\mathbf{X}}_{k+1}$ and its covariance P_{k+1} :

$$\bar{\mathbf{X}}_{k+1} = f(\bar{\mathbf{X}}_k, \mathbf{u}_k^\wedge) \quad (5)$$

$$\Phi = \exp(\mathbf{A}\Delta t) \quad (6)$$

$$\mathbf{Q}_d \approx \Phi \mathbf{Q} \Phi^T \Delta t \quad (7)$$

$$\mathbf{P}_{k+1} = \Phi \mathbf{P}_k \Phi^T + \text{Ad}(\bar{\mathbf{X}}_k) \mathbf{Q}_d \text{Ad}(\bar{\mathbf{X}}_k)^T \quad (8)$$

where f is the discrete-time IMU dynamic from IMU readings \mathbf{u}_k detailed in [2]. Due to small Δt , the gamma functions are approximated as their first-order terms. \mathbf{Q}_d is the discrete-time process model noise covariance which is added to the estimated state covariance. There were no significant gains in performance from tuning the IMU or linear velocity covariances.

3) *DVL Correction*: The prediction is corrected using measurements from the DVL, which gives a linear velocity measurement $\mathbf{v}_k = [v_x \ v_y \ v_z]^T$. Due to the difference in location of the DVL and IMU [5], the velocity is subtracted due to rotation $\mathbf{d} \times \boldsymbol{\omega}_k$, where \mathbf{d} is the vector from the IMU to DVL, to get $\mathbf{v}_k - \mathbf{d} \times \boldsymbol{\omega}_k = \tilde{\mathbf{v}}_k = [\tilde{v}_x \ \tilde{v}_y \ \tilde{v}_z]^T$.

a) *Measurement Model*: The right-invariant measurement model used is defined as $\tilde{\mathbf{v}} = \bar{\mathbf{X}}_{k+1}^{-1} \mathbf{b} + \mathbf{W}_{DVL,k}$, where $\mathbf{W}_{DVL,k}$ is the measurement noise covariance. First, the inverse of the state matrix is derived using the fact that the inverse of a rotation matrix is its transpose.

$$\begin{aligned} \mathbf{X} \mathbf{X}^{-1} &= \mathbf{I} \\ \begin{bmatrix} \mathbf{R} & \mathbf{v} & \mathbf{p} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{X}^{-1} &= \mathbf{I} \\ \begin{bmatrix} \mathbf{R} & \mathbf{v} & \mathbf{p} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}^T & [\mathbf{a}]_{3 \times 1} & [\mathbf{d}]_{3 \times 1} \\ 0 & b & e \\ 0 & c & f \end{bmatrix} &= \mathbf{I} \end{aligned}$$

It is straightforward to solve this system; it is found that $\mathbf{a} = -\mathbf{R}^T \mathbf{v}$, $\mathbf{d} = -\mathbf{R}^T \mathbf{p}$, $b = f = 1$, and $c = e = 0$. Thus:

$$\mathbf{X}_k^{-1} = \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T \mathbf{v} & -\mathbf{R}^T \mathbf{p} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (9)$$

Next, it is trivial to solve for \mathbf{b} in $\tilde{\mathbf{v}} = \bar{\mathbf{X}}_{k+1}^{-1} \mathbf{b} + \mathbf{W}_k$:

$$\begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T \mathbf{v} & -\mathbf{R}^T \mathbf{p} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{b}_{DVL} + \mathbf{W} = \begin{bmatrix} \tilde{v}_x \\ \tilde{v}_y \\ \tilde{v}_z \\ 1 \\ 0 \end{bmatrix} \quad (10)$$

$$\mathbf{b}_{DVL} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad (11)$$

Additional values are added to the end of $\tilde{\mathbf{v}}$ in order to match the velocity part of the state matrix and make the matrix dimensions match up properly.

b) *Measurement Model Jacobian*: To perform the correction based on this measurement model, its Jacobian is needed \mathbf{H} . \mathbf{H} needs to satisfy $\mathbf{H} \boldsymbol{\xi} = -\boldsymbol{\xi}^\wedge \mathbf{b}$, where $\boldsymbol{\xi}$ is the concatenation of 3D angular velocity $\boldsymbol{\xi}_\omega$, linear velocity $\boldsymbol{\xi}_v$, and position $\boldsymbol{\xi}_p$, and \mathbf{b}_{DVL} is the vector derived above. The derivation is straightforward:

$$\mathbf{H}_{DVL} \boldsymbol{\xi} = -\boldsymbol{\xi}^\wedge \mathbf{b}_{DVL} \quad (12)$$

$$\mathbf{H}_{DVL} \begin{bmatrix} \boldsymbol{\xi}_\omega \\ \boldsymbol{\xi}_v \\ \boldsymbol{\xi}_p \end{bmatrix} = - \begin{bmatrix} \boldsymbol{\xi}_\omega^\wedge & \boldsymbol{\xi}_v & \boldsymbol{\xi}_p \\ \mathbf{0} & 0 & 0 \\ \mathbf{0} & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad (13)$$

$$\mathbf{H}_{DVL} \begin{bmatrix} \boldsymbol{\xi}_\omega \\ \boldsymbol{\xi}_v \\ \boldsymbol{\xi}_p \end{bmatrix} = - \begin{bmatrix} \boldsymbol{\xi}_v \\ 0 \\ 0 \end{bmatrix} \quad (14)$$

$$\mathbf{H}_{DVL} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & -\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (15)$$

4) *Depth Sensor Correction*: Without further data past IMU and DVL, the z axis predictions were quite off, so measurements from the depth sensor were added to correct the predictions. The depth sensor reads a position measurement \tilde{z}_k relative to the water surface. As the depth sensor only

displays one coordinate of the position, the full measurement is considered to be $\tilde{z}_k = -R_k^T [p_x \ p_y \ \tilde{z}_k]^T$, where R_k , p_x and p_y are the rotation matrix and x and y coordinates from p_k in the estimate \bar{X}_k from the propagation step.

a) *Measurement Model*: To incorporate depth data, a transformation to a robot-centric measurement is necessary. The right-invariant measurement model is formed as $\tilde{z} = \bar{X}_{k+1}^{-1} b + W_{D,k}$, where $W_{D,k}$ is the measurement noise covariance. The inverse of this is the same as the previously derived inverse of the state matrix above. From here, solve for b in the measurement model to ensure the correction is performed on the position portion of the state estimation:

$$\begin{bmatrix} R^T & -R^T v & -R^T p \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} b_d + W = -R^T \begin{bmatrix} p_x \\ p_y \\ \tilde{z}_k \\ 0 \\ 1 \end{bmatrix} \quad (16)$$

$$b_d = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \quad (17)$$

Performing this transformation turns a normally left-invariant measurement to a right-invariant approach. Despite this, it is quite effective in practice. Due to the correction using the current state estimate, depth measurements had higher variance. There were no significant gains in performance from transforming the depth measurement to compensate for the spatial difference between the IMU and depth sensor.

b) *Measurement Model Jacobian*: To perform the correction based on this measurement model, the Jacobian H_d is required.

$$H_d \begin{bmatrix} \xi_\omega \\ \xi_v \\ \xi_p \end{bmatrix} = - \begin{bmatrix} \xi_\omega^\wedge & \xi_v & \xi_p \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (18)$$

$$H_d \begin{bmatrix} \xi_\omega \\ \xi_v \\ \xi_p \end{bmatrix} = - \begin{bmatrix} \xi_p \\ 0 \\ 0 \end{bmatrix} \quad (19)$$

$$H_d = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} & -I_{3 \times 3} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (20)$$

5) *Magnetometer Correction*: The magnetometer gives the magnetic field data \tilde{m}_k , which is used to correct orientation. The dataset provides $\tilde{m}_k = [\tilde{m}_x \ \tilde{m}_y \ \tilde{m}_z]^T$ in units of gauss (G), which is converted to nanoteslas (nT) for the model. Magnetometers are highly variable depending time and place. The exact location was (Lat: 42.103883N, Lon: 3.182550E) and taken during July 2013. Historical geomagnetic field data provided the magnetic field at the location for July 10, 2013 [7].

a) *Measurement Model*: Using the stated coordinate conventions, the magnetic vector was found to be $m = [m_x \ m_y \ m_z]^T = [0.24494 \ -0.002385 \ -0.38615]^T$.

$$\begin{bmatrix} R^T & -R^T v & -R^T p \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} b_m + W = \begin{bmatrix} \tilde{m}_x \\ \tilde{m}_y \\ \tilde{m}_z \\ 0 \\ 0 \end{bmatrix} \quad (21)$$

$$b_m = \begin{bmatrix} m_x \\ m_y \\ m_z \\ 0 \\ 0 \end{bmatrix} \quad (22)$$

b) *Measurement Model Jacobian*: Following the derivations above, the Jacobian will be

$$H_m \begin{bmatrix} \xi_\omega \\ \xi_v \\ \xi_p \end{bmatrix} = - \begin{bmatrix} \xi_\omega^\wedge & \xi_v & \xi_p \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} m_x \\ m_y \\ m_z \\ 0 \\ 0 \end{bmatrix} \quad (23)$$

$$H_m \begin{bmatrix} \xi_\omega \\ \xi_v \\ \xi_p \end{bmatrix} = - \begin{bmatrix} \xi_\omega^\wedge m \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} m^\wedge \xi_\omega \\ 0 \\ 0 \end{bmatrix} \quad (24)$$

$$H_m = \begin{bmatrix} m^\wedge & 0_{3 \times 3} & 0_{3 \times 3} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (25)$$

6) *Batch Update*: The sensor data is combined in a batch update to get informative corrections for the filter.

a) *Stacked Measurement Model*: When stacking the magnetometer reading to the measurement model the N' uses only the significant rows, or the rows that affect each element of the robot's pose or motion, from the covariances from each sensor and aligns them in block diagonal form. These can be written as

$$\tilde{z}'_k = \begin{bmatrix} \tilde{v}_k \\ \tilde{z}_k \\ \tilde{m}_k \end{bmatrix} \quad b' = \begin{bmatrix} b_{DVL} \\ b_d \\ b_m \end{bmatrix} \quad (26)$$

$$H' = \begin{bmatrix} 0_{3 \times 3} & -I_{3 \times 3} & 0_{3 \times 3} \\ 0 & 0 & -1 \\ m^\wedge_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix}_{7 \times 9} \quad (27)$$

$$N' = \begin{bmatrix} N_{DVL} & 0 & 0 \\ 0 & N_d & 0 \\ 0 & 0 & N_m \end{bmatrix} \quad (28)$$

b) *Update Equations*: For any measurement \tilde{z}_k , measurement noise covariance N , and the above derivations, the correction step is preformed with the following update equations:

$$\bar{N} = \bar{X}_{k+1} N \bar{X}_{k+1}^T \quad (29)$$

$$S = H \bar{P}_{k+1} H^T + \bar{N} \quad (30)$$

$$L = \bar{P}_{k+1} H^T S^{-1} \quad (31)$$

$$\Delta = L(\bar{X}_{k+1} \tilde{v}_k - b) \quad (32)$$

$$\bar{X}_{k+1} = \exp(\Delta^\wedge) \bar{X}_{k+1} \quad (33)$$

$$P_{k+1} = (I - LH) \bar{P}_{k+1} (I - LH)^T + L \bar{N} L^T \quad (34)$$

Where \bar{N} is the discrete-time measurement model noise covariance. The filter was more effective when the DVL covariance was much lower than the depth sensor or magnetometer. Some rows are removed in H for convenience, i.e., the zero rows in the Jacobians and the rows corresponding to x and y coordinates in the depth sensor correction.

V. RI-EKF RESULTS

A. Evaluation Metrics

The empirical results from the RI-EKF model on this data set were evaluated in similar fashion to the previous work by Mallios et al. [4] who created the data set. Results relied upon the position of the six ground truth cones placed in the underwater cave, where the AUV passed each cone twice along its route. The authors of [4] generously shared the times that the AUV passed each of these cones during data collection, so at each time a cone is passed, it is possible to extract the predicted AUV position from any approach. For a more exact estimate of the cone positions, they also shared the distance between the AUV and each cone at these times, calculated using the calibration of the onboard camera.

Based on the available information about these cones, two metrics are considered. To understand the localization error, the first metric was the difference in the predicted Euclidean position (plus the known offsets between the AUV and the cone) of the two passes over the same cone. To appreciate the error, the second metric was the total distance between each consecutive pair of cones visited. This could be contrasted to the ground truth measurement taken by the divers when placing the cones as landmarks on the seafloor, provided in [4]. The route of the AUV through the cones was $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 6 \rightarrow 6 \rightarrow 1$. To reduce the impact of sensor noise on the second metric, only every tenth data point is sampled when calculating the predicted distance traveled between consecutive cones.

B. Comparison Approaches

The proposed RI-EKF is compared against a visual-odometry (powered by the onboard camera images) localization approach included in the dataset, as well as a pose-based graph SLAM approach using sonar data from [4].

C. Results Discussion

The RI-EKF system is compared against the visual-odometry and the previous work's SLAM approach, as shown in Figure 4 and Figure 3. The triangles represent the placement of each cone and each Cone 1-6 has a different

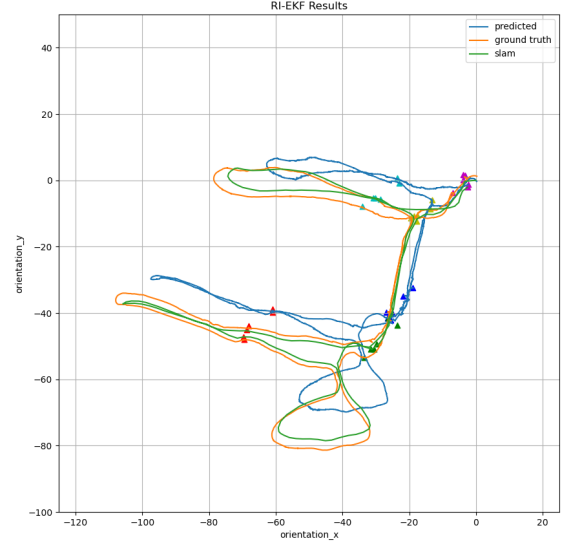


Fig. 3. Predicted 2D positions of comparison approaches from overhead. Predicted cone positions marked as colored triangles, where color indicates unique cones.

color. The RI-EKF result has a similar shape to the visual-odometry and SLAM approaches. At a glance, the proposed approaches matches the shape of the comparison approaches quite strongly, while being slightly translated from them in the x and y dimensions.

Table I compares all approaches based on the first metric, i.e., the difference between the predicted AUV position at the first and second pass of each cone, with the best result in each row bold. The proposed RI-EKF outperforms visual-odometry for 5 out of the 6 cones, while it outperforms SLAM in 2 out of the 6 cones. On average, RI-EKF markedly outperforms visual-odometry, and approaches the performance of SLAM. For these results, it is worth noting that the largest difference in predicted cone position could be expected between passes on Cone 1 because it sees Cone 1 first and last. The AUV travels the whole route and has the largest opportunity to drift over time with Cone 1. In theory, it is expected that the same exact position is reported above the same cone, as the cone does not move, but the results show slight drift over each cone due to model assumptions and sensor drift. This also may explain the relatively high difference for Cone 1 under visual-odometry, which is especially susceptible to this drift over time. Nonetheless, the proposed RI-EKF is fairly robust under this evaluation.

Table II compares all approaches based on the second metric for all consecutive pairs of cones for which [4] reports the ground truth distance between cones, with the best result in each row bolded. The proposed RI-EKF outperforms visual-odometry on three pairs of these cones, rivaling other cones, and outperforms SLAM nearly across the board. It seems that the RI-EKF is generally a more robust approach

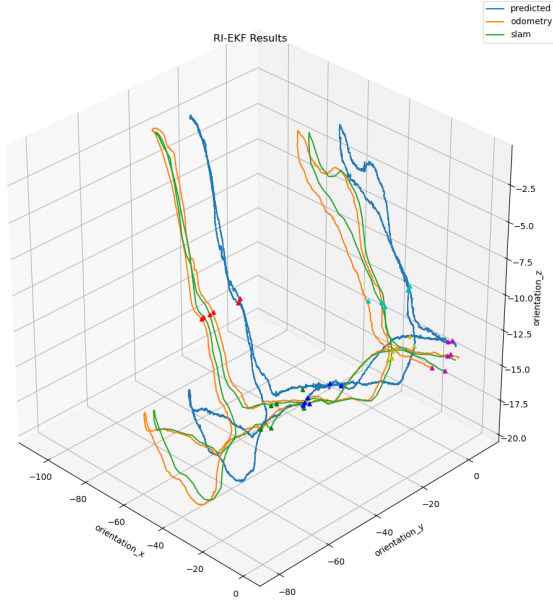


Fig. 4. Predicted 3D positions of comparison approaches. Predicted cone positions marked as colored triangles, where color indicates unique cones.

Cone	RI-EKF	Visual-Odometry	SLAM
1	2.30	6.49	1.96
2	2.36	2.95	1.77
3	3.03	2.20	1.64
4	3.66	6.17	1.79
5	0.70	4.03	2.10
6	2.02	4.73	2.31
Avg.	2.35	4.43	1.93

TABLE I

DISTANCE BETWEEN PREDICTED POSITIONS IN FIRST AND SECOND PASS OF CONES (M).

for preserving distances in the environment than SLAM.

VI. DISCUSSION AND FUTURE DIRECTIONS

In this work, an RI-EKF approach was proposed for localizing an AUV in an underwater cave system, incorporating sensor measurements from Doppler velocity log (DVL), water pressure-based depth sensor, and an inertial measurement unit (IMU), including a gyroscope, accelerometer, and magnetometer. The results, while running in minutes, approach the performance of a SLAM-based approach which takes much longer to run. This work can serve as a proof of concept that InEKF could provide a faster and more accurate localisation than convectional methods. The code provided

Cones	Ground Truth	RI-EKF	Visual-Odometry	SLAM
1→2	19	17.8	20.0	24.6
2→3	32	30.8	32.8	46.7
3→4	16	14.2	13.1	20.0
4→3	16	17.7	18.5	19.5
3→2	32	32.0	33.9	38.6
6→1	30	25.2	33.8	30.8

TABLE II

DISTANCE BETWEEN SUCCESSIVE CONES (M).

from this project is divided in such a way to provide a framework for future work in this area. The data-loader, riekf class, and visualiser can all be used independently of the main application.

A. Sensor Biases

Most of the sensors in the dataset did not offer biases to offset known sensor inaccuracies, especially for the IMU bias. For the gyro bias, it was presented inside `imu_audis.csv` file, but the processed file named `imu_audis_ros.csv` already incorporated the gyro bias. Without access to the system further sensor bias is hard to predict. As such, experimentation with tuning manually specified bias parameters occurred. As IMU data frequently has inaccuracy in rotational velocity measurement due to temperature fluctuations and initialization conditions, a calibratable bias was added to the z rotation axis. Though the bias was small, it the results showed improvement to most of the result metrics in relation of the ground truth cones. It should be noted that this may cause the system to be over-fitted and preform worse on alternate applications. It's recommended that all bias parameters be tuned to the system using testing and initial state conditions.

B. Accuracy Improvement

The ground truth values for this dataset provide a way to measure the effectiveness of the proposed method, but do not provide a good foundation for more detailed tuning of the algorithm. The cone distance measurements were taken with tape measure and are acknowledged to be vague in accuracy. The cone photos were not shot from directly overhead, leading to an unknown offset from the AUV path. Overall further work should be conducted in measured conditions to better tune bias and contribution of different sensors. It can be seen in the plots, that globally the algorithm works for localization, but on a smaller local scale there is a lot of noise in the AUV movement. More work into putting physical constraints on the devise or smoothing trajectory could result in a more precise localization.

C. ROS Package

A framework is provided for translating dataset loading and preprocessing into a ROS environment. The code listens for the data playing from a ROS bag file rather than loading it directly from CSVs offline, which enables online use by the AUV during future cave exploration. For online use of the algorithms it would be straightforward to connect the data processing code to the offline algorithm with some small modifications. This is left for future work.

REFERENCES

- [1] Samuel Ansaldi, AJ Bull, Xinyu Ma, Alyssa Scheske, and Shane Storks. Invariant extended kalman filter for localization in underwater caves. https://github.com/sansaldi/IEKF_AUV_Cave_Navigation, 2021. Accessed: 2021-04-20.
- [2] Ross Hartley, Maani Ghaffari, Ryan M. Eustice, and Jessy W. Grizzle. Contact-aided invariant extended kalman filtering for robot state estimation, 2019.

- [3] Juan David Hernández, Eduard Vidal, Mark Moll, Narcís Palomeras, Marc Carreras, and Lydia E. Kavraki. Online motion planning for unexplored underwater environments using autonomous underwater vehicles. *Journal of Field Robotics*, 36(2):370–396, 2019.
- [4] Angelos Mallios, Pere Ridao, David Ribas, Marc Carreras, and Richard Camilli. Toward autonomous exploration in confined underwater environments. *Journal of Field Robotics*, 33(7):994–1012, 2016.
- [5] Angelos Mallios, Eduard Vidal, Ricard Campos, and Marc Carreras. Underwater caves sonar data set. *The International Journal of Robotics Research*, 36(12):1247–1251, 2017. <https://cirs.udg.edu/caves-dataset/>.
- [6] A. Martins, J. Almeida, C. Almeida, A. Dias, N. Dias, J. Aaltonen, A. Heininen, K. T. Koskinen, C. Rossi, S. Dominguez, C. Vörös, S. Henley, M. McLoughlin, H. van Moerkerk, J. Tweedie, B. Bodo, N. Zajzon, and E. Silva. Ux 1 system design - a robotic system for underwater mining exploration. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1494–1500, 2018.
- [7] National Centers For Environmental Information. Magnetic field calculators. <https://www.ngdc.noaa.gov/geomag/calculators/magcalc.shtml#igrfwmm>. Accessed: 2021-04-16.
- [8] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. Probabilistic robotics.[sl], 2005.