# Invariant Extended Kalman Filtering for Underwater Navigation

Easton R. Potokar [ID], Kalin Norman [ID], and Joshua G. Mangelson [ID]

*Abstract*—Recent advances in the utilization of Lie Groups for robotic localization have led to dramatic increases in the accuracy of estimation and uncertainty characterization. One of the novel methods, the Invariant Extended Kalman Filter (InEKF) extends the Extended Kalman Filter (EKF) by leveraging the fact that some error dynamics defined on matrix Lie Groups satisfy a log-linear differential equation. Utilization of these observations result in linearization with minimal approximation error, no dependence on current state estimates, and excellent convergence and accuracy properties. In this letter we show that the primary sensors used for underwater localization, inertial measurement units (IMUs) and doppler velocity logs (DVLs) meet the requirements of the InEKF. Furthermore, we show that singleton measurements, such as depth, can also be used in the InEKF update with minor modifications, thus expanding the set of measurements usable in an InEKF. We compare convergence, accuracy and timing results of the InEKF to a quaternion-based EKF using a Monte Carlo simulation and show notable improvements in long-term localization and much faster convergence with negligible difference in computation time.

*Index Terms*—Unmanned underwater vehicles, filtering algorithms, state estimation, computational geometry.

## I. INTRODUCTION

AUTONOMOUS underwater vehicles (AUVs) have the potential to dramatically improve safety, quality of life and general scientific knowledge. Our coasts, lakes and rivers are filled with various forms of marine infrastructure including dams, bridges, ship hulls, communication lines, and oil rigs. Each of these structures require regular inspection and current methods utilize divers, which is dangerous, expensive, and time consuming. AUVs have the potential to alleviate these difficulties and enable more regular inspection of these structures. Furthermore, there are significant scientific discoveries in the fields of geology, marine biology and medicine that AUV exploration of our oceans will enable. However, successful AUV utilization is dependent on effective localization.

Underwater localization is a particularly difficult problem as many common land-based sensors such as GPS, LiDAR, etc. are unavailable underwater. This limits many potential options for use in a localization observer.

The authors are with the Department of Electrical and Computer Engineering, College of Physical and Mathematical Sciences, Brigham Young University (BYU), Provo, UT 84606 USA (e-mail: eastonpots@byu.edu; kalinnorman@byu.edu; joshua_mangelson@byu.edu).
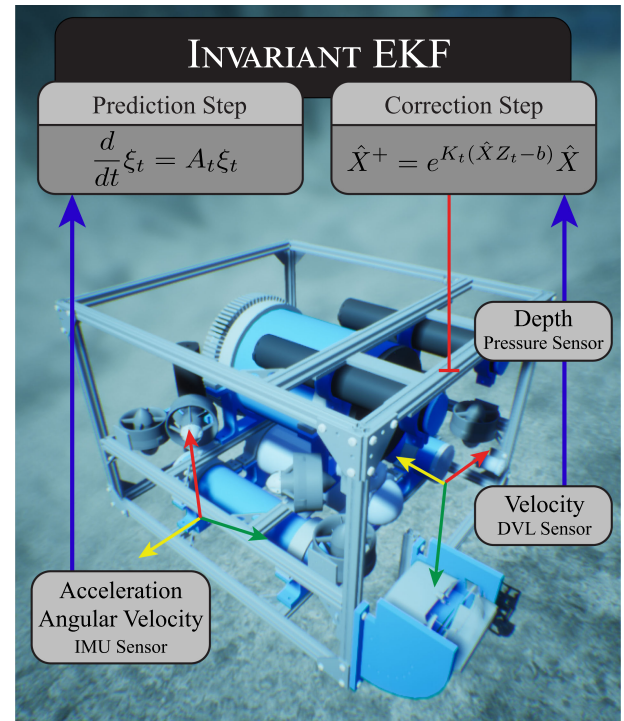
Fig. 1. The Invariant Extended Kalman Filter (InEKF) leverages Lie Group Theory to model robot state and error and results in improved convergence and localization. In this letter, we derive the InEKF for use with common underwater sensors such as an inertial measurement unit, doppler velocity log, and pressure sensor.

In this letter, we demonstrate how the Invariant Extended Kalman Filter (InEKF) can be used to improve upon current methods of underwater localization. The InEKF is derived by defining process models on matrix Lie Groups and showing the resulting error dynamics satisfy a log-linear property. This overcomes many of the shortcomings of Jacobian based linearization used in the Extended Kalman Filter (EKF). Specifically, we derive the InEKF for a vehicle equipped with common underwater sensors such as an inertial measurement unit (IMU), a doppler velocity log (DVL) and a pressure sensor as seen in Fig. 1. More specifically, our contributions are the following:

1) We show how to derive an InEKF for AUV localization that utilizes the IMU for the process model and DVL and a pressure sensor for correction.
2) We propose a novel method that enables the inclusion of non-standard singleton measurements (such as depth from a pressure sensor) into the InEKF via "pseudo" measurements composed of the current state estimate modeled with infinite covariance.

3) We evaluate our proposed method using simulated data and compare it to a Quaternion EKF (QEKF).

We also explicitly show the change of frames required to put DVL measurements in the IMU frame as well as release all our python source code as open source at https://bitbucket.org/frostlab/underwateriekf.

The letter is organized as follows. Section II reviews current methods and algorithms for underwater navigation and uses of the InEKF. In Section III, we give a brief background on Lie Groups and InEKFs. The InEKF for underwater localization is derived in Section IV with Subsection IV-B and IV-C reviewing the derivation of the process model, and Subsection IV-D and IV-E deriving the necessary measurement models. Simulation results and comparisons with the QEKF are shown in Section V. Finally, Section VI summarizes the article and proposes future research directions.

## II. RELATED WORK

Underwater localization poses many challenges and a wide range of solutions have been attempted in recent years, as summarized in [1]. These methods can be broken into three groups: filtering-based, direct-position measurement-based, and mapping-based.

Filtering-based methods attempt to fuse multiple noisy odometric measurements of the robot's motion. Typical inputs to a filter for underwater localization include an IMU, a DVL, and a pressure sensor. For linear systems with independent measurements and Gaussian noise, the Kalman Filter [2] provides an optimal fusion of these noisy measurements. The most common filter in use today is the EKF [3] which uses linearization to apply Kalman filtering techniques to systems with non-linear process and measurement models. Other methods such as the QEKF [4], try to improve on the EKF by modeling the non-linear rotation elements of a robot state using alternate representations.

While electromagnetic-based global positioning systems such as GPS do not function underwater, acoustic positioning techniques such as long baseline (LBL) or ultra-short baseline (USBL) [5], [6] are sometimes used. However, these methods generally require extensive setup and often fail in cluttered or shallow environments.

Mapping-based techniques such as simultaneous localization and mapping (SLAM) [7]–[9] or terrain-based navigation [10], [11] use features of the environment to overcome drift in the robot's odometry estimate. While these mapping-based techniques correct for drift, they often rely on underlying odometry solutions, such as the one we propose.

The InEKF is a recent extension of the EKF that is based on matrix Lie Groups [12]–[15], in contrast to the quaternion-based approach of the QEKF. Lie Groups have been shown to propagate uncertainty estimates more accurately [16]. It derives its properties from the estimation error being invariant under matrix multiplication, which is the action of a matrix Lie Group [17], [18]. The InEKF has given rise to many successful results and applications in SLAM [13], [19], guided navigation [13], [20]–[22], and for 3D bipedal robots [23].

The main advantage of the InEKF is due to the invariance of the estimation error, often referred to as the symmetries of the system, causing the error to satisfy a log-linear differential equation on the Lie algebra, resulting in a state independent trajectory in the error system dynamics. This leads to no linearization approximations based on current state estimates, and as a result

---

**Algorithm 1:** Right Invariant EKF

1   $\hat{\Sigma} = \Sigma_0$;
2   $\hat{X} = X_0$;
3   **while** *receiving data* **do**
4     **if** *Predict Step* **then**
5        $\dfrac{\mathrm{d}}{\mathrm{d}t}\hat{X} = f_{u_t}(\hat{X})$;
6        $\dfrac{\mathrm{d}}{\mathrm{d}t}\hat{\Sigma} = A_t\hat{\Sigma} + \hat{\Sigma}A_t^T + Ad_{\hat{X}}QAd_{\hat{X}}^T$;
7     **else if** *Update Step* **then**
8        $S^{-1} = (H\hat{\Sigma}H^T + \hat{R}M\hat{R}^T)^{-1}$;
9        $K = \hat{\Sigma}H^TS^{-1}$;
10       $\hat{X} = \exp(K\Pi\hat{X}z)\hat{X}$;
11       $\hat{\Sigma} = (I - KH)\hat{\Sigma}$;
12 **end**

Fig. 2. Outline of the Right Invariant EKF.

---

strong convergence properties. [15]. As far as we know, the InEKF has never before been applied to AUV localization.

## III. THEORETICAL BACKGROUND

In this section, we briefly explain the needed fundamentals of Lie Group theory and use it to briefly explain the derivation and important properties of the InEKF. This ends in an outline of the InEKF in Fig. 2.

### A. Lie Group Theory

We denote a matrix Lie Group by $\mathcal{G}$ [24], and its associated Lie algebra given by its tangent space as $\mathfrak{g}$, both of which have $n \times n$ elements. Common examples of Lie Groups are SO(3), which consists of 3D rotation matrices, and SE(3), which consists of 3D rigid body transformations.

Further, we map $\mathbb{R}^{\dim \mathfrak{g}}$ to the Lie algebra $\mathfrak{g}$ of $\mathcal{G}$ at the identity using the linear map

$$^{\wedge} : \mathbb{R}^{\dim \mathfrak{g}} \to \mathfrak{g}.$$

Thus, along with the exponential map $\exp : \mathfrak{g} \to \mathcal{G}$, we can map [24]

$$\exp(\cdot^{\wedge}) : \mathbb{R}^{\dim \mathfrak{g}} \to \mathcal{G}.$$

Throughout this letter, we also make use of the adjoint of an element of $\mathcal{G}$ as defined below.

*Definition 1:* [24] For any $X \in \mathcal{G}, \xi \in \mathbb{R}^{\dim \mathfrak{g}}$, the adjoint map $Ad_X : \mathfrak{g} \to \mathfrak{g}$ is given by $Ad_X(\xi^{\wedge}) = X\xi^{\wedge}X^{-1}$. The adjoint is linear, and is often given by its matrix representation as $Ad_X(\xi^{\wedge}) = (Ad_X\xi)^{\wedge}$.

### B. InEKF Process Model

The state process model evolving on the matrix Lie Group at time $t$ and state $X_t \in \mathcal{G}$ can be given by

$$\frac{\mathrm{d}}{\mathrm{d}t}X_t = f_{u_t}(X_t).$$

Further, letting $\hat{X}_t$ represent our state estimate, our state error on the matrix Lie Group can be defined as follows.

*Definition 2:* [15] The right and left invariant error between two trajectories $X_t$ and $\hat{X}_t$ is given by

$$\eta_t^r = \hat{X}_t X_t^{-1} \qquad \text{(Right Invariant)}, \qquad (1)$$

$$\eta_t^l = X_t^{-1} \hat{X}_t \qquad \text{(Left Invariant)}. \qquad (2)$$

Note in the case when $\hat{X}_t = X_t$, both errors reduce to the identity. Using the above definition, the following theorems are foundational in deriving the guarantees of the InEKF.

*Theorem 1:* [15] A system is said to be group affine if $f_{u_t}(\cdot)$ satisfies

$$f_{u_t}(X_1 X_2) = f_{u_t}(X_1)X_2 + X_1 f_{u_t}(X_2) - X_1 f_{u_t}(I)X_2 \quad (3)$$

for all time $t > 0$ and $X_1, X_2 \in \mathcal{G}$. If this condition is satisfied, the right and left invariant errors are trajectory independent and satisfy

$$\frac{\mathrm{d}}{\mathrm{d}t}\eta_t^r = g_{u_t}(\eta_t^r) \triangleq f_{u_t}(\eta_t^r) - \eta_t^r f_{u_t}(I), \qquad (4)$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\eta_t^l = g_{u_t}(\eta_t^l) \triangleq f_{u_t}(\eta_t^l) - f_{u_t}(I)\eta_t^l. \qquad (5)$$

Note in the above, $I \in \mathcal{G}$ is the group identity matrix. The above theorem results in a state independent differential equation, meaning any linearizations made to $g_{u_t}$ will have no dependence on the current state estimate. The right or left error differential equation can then be linearized by defining $A_t$ to satisfy

$$g_{u_t}(\exp(\xi^\wedge)) \triangleq (A_t \xi)^\wedge + \mathcal{O}(\|\xi\|^2). \qquad (6)$$

For $t > 0$, let $\xi_t$ be the solution of the differential equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\xi_t = A_t \xi_t. \qquad (7)$$

This results in linearized error dynamics with 2nd order error. However, the following theorem states that the true error can be recovered from $\xi_t$ with no approximation error.

*Theorem 2:* [15] Consider the right or left invariant error, $\eta_t$, between any two trajectories. For arbitrary initial error $\xi_0 \in R^{\dim \mathfrak{g}}$, if $\eta_0 = \exp(\xi_0^\wedge)$, then for all $t \geq 0$,

$$\eta_t = \exp(\xi_t^\wedge).$$

In other words, the nonlinear error $\eta_t$ can be exactly recovered from the time-varying linear differential eq. (7).

This implies the InEKF linearization of the invariant error dynamics is trajectory independent when the conditions of Theorem 1 are met, and thus by Theorem 2 introduces no approximation error since the invariant error can be recovered exactly, in contrast to that of the standard EKF. These properties lead to many of the same guarantees that follow the standard Kalman Filter, in particular local asymptotic stability [15, Theorem 4].

Finally, noise can also be introduced into the deterministic process model via

$$\frac{\mathrm{d}}{\mathrm{d}t}X_t = f_{u_t}(X_t) - X_t w_t^\wedge \qquad w_t \sim \mathcal{N}(0, Q). \qquad (8)$$

### C. InEKF Measurement Model

Furthermore, the InEKF requires measurement models to fit the following form.

*Definition 3:* [15] Right and left invariant observations are of the form

$$z_t^r = X_t^{-1} b + W_t \qquad \text{(Right Invariant)} \qquad (9)$$

$$z_t^l = X_t b + W_t \qquad \text{(Left Invariant)} \qquad (10)$$

where $b$ is some known vector and $W_t$ is zero-mean Gaussian additive noise with covariance $M$. They have corresponding innovations given by

$$V_t^r = \hat{X}_t(z_t^r - \hat{z}_t^r) \qquad \text{(Right Invariant)} \qquad (11)$$

$$V_t^l = \hat{X}_t^{-1}(z_t^l - \hat{z}_t^l) \qquad \text{(Left Invariant)} \qquad (12)$$

where $\hat{z}_t$ is the measurement estimate using the current state estimate.

These innovations make linearization simple using the first order approximation $\eta_t = \exp(\xi_t) \approx I + \xi_t^\wedge$ as follows [14].

$$V_t^r = \hat{X}_t(z_t^r - \hat{z}_t^r) = \hat{X}_t(X_t^{-1}b + W_t - \hat{X}_t^{-1}b)$$

$$= \eta_t^r b + \hat{X}_t W_t - b \approx (I + \xi_t^{r\wedge})b + \hat{X}_t W_t - b$$

$$= \xi_t^{r\wedge} b + \hat{X}_t W_t \triangleq -H\xi_t^r + \hat{X}_t W_t. \qquad (13)$$

The linearization for the left invariant innovation is near identical. Further, since in most cases the last rows of these innovations are identically zero, an auxiliary matrix $\Pi = \begin{bmatrix} I & 0 \end{bmatrix}$ is used to remove them. The corresponding rows of $H$ are removed accordingly as well.

The resulting Right InEKF equations are seen in Fig. 2.

## IV. UNDERWATER LOCALIZATION USING InEKF

In this section, we derive a Right Invariant Extended Kalman Filter (RInEKF) for general applications in underwater navigation. It uses an IMU motion model with bias tracking for the prediction step, and the following sensors for the predict step,

- DVL measuring velocity in the body frame, see eq. (33).
- Pressure sensor measuring depth, see eq. (36).

The pressure sensor model in particular doesn't fit the standard right invariant observation model, and we show the required modifications for use in the InEKF.

### A. State Representation

We seek to track the orientation, velocity and position of the IMU (body) represented in the world frame, as is common in aided inertial navigation. This can be represented respectively by $R_{WB}, {}_W v_{WB}, {}_W p_{WB}$, but for conciseness we abbreviate these as $R_t, v_t, p_t$. For our purposes, we have chosen the world frame to be an arbitrary point at the surface, with a right handed system with the gravity direction representing the negative z-axis, and x and y-axes parallel to the water surface. Together, these state variables form the group of double direct isometries, or the matrix Lie Group $SE_2(3)$ [13]. An element $X_t \in SE_2(3)$ is a $5 \times 5$ matrix in the form of

$$X_t \triangleq \begin{bmatrix} R_t & v_t & p_t \\ 0_{1 \times 3} & 1 & 0 \\ 0_{1 \times 3} & 0 & 1 \end{bmatrix}. \qquad (14)$$

Further, this Lie Group has an associated Lie algebra $\mathfrak{se}_2(3)$ with an associated map $^\wedge : \mathbb{R}^9 \to \mathfrak{se}_2(3)$ defined as follows.

Given $\xi \in \mathbb{R}^9$

$$\xi^\wedge = \begin{bmatrix} \xi_R \\ \xi_v \\ \xi_p \end{bmatrix}^\wedge = \begin{bmatrix} (\xi_R)_\times & \xi_v & \xi_p \\ 0_{1x\times3} & 0 & 0 \\ 0_{1x\times3} & 0 & 0 \end{bmatrix} \quad (15)$$

where $(\cdot)_\times$ denotes a $3\times3$ skew-symmetric matrix such as,

$$\left( \begin{bmatrix} a \\ b \\ c \end{bmatrix} \right)_\times = \begin{bmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{bmatrix}. \quad (16)$$

The adjoint is given by [23]

$$Ad_{X_t}\xi = \begin{bmatrix} R_t & 0 & 0 \\ (v_t)_\times R_t & R_t & 0 \\ (p_t)_\times R_t & 0 & R_t \end{bmatrix} \xi. \quad (17)$$

### B. IMU Motion Model

The IMU measurements of angular velocity and acceleration are modeled as being corrupted by zero-mean Gaussian noise as

$$\tilde{\omega}_t = \omega_t + \mathrm{w}_t^\omega, \qquad \mathrm{w}_t^\omega \sim \mathcal{N}(0, \Sigma^\omega), \quad (18)$$
$$\tilde{a}_t = a_t + \mathrm{w}_t^a, \qquad \mathrm{w}_t^a \sim \mathcal{N}(0, \Sigma^a). \quad (19)$$

Using these measurements, our continuous system dynamics are then [23]

$$\dot{R}_t = R_t(\tilde{\omega}_t - \mathrm{w}_t^\omega)_\times$$
$$\dot{v}_t = R_t(\tilde{a}_t - \mathrm{w}_t^a) + g$$
$$\dot{p}_t = v_t \quad (20)$$

where $g$ is the gravity vector. These continuous dynamics can be written as elements of $SE_2(3)$ as

$$\frac{\mathrm{d}}{\mathrm{d}t} X_t = \begin{bmatrix} R_t(\tilde{\omega}_t)_\times & R_t\tilde{a}_t + g & v_t \\ 0_{1\times3} & 0 & 0 \\ 0_{1\times3} & 0 & 0 \end{bmatrix}$$
$$- \begin{bmatrix} R_t & v_t & p_t \\ 0_{1\times3} & 1 & 0 \\ 0_{1\times3} & 0 & 1 \end{bmatrix} \begin{bmatrix} (\mathrm{w}_t^\omega)_\times & \mathrm{w}_t^a & 0 \\ 0_{1\times3} & 0 & 0 \\ 0_{1\times3} & 0 & 0 \end{bmatrix} \quad (21)$$
$$\triangleq f_{u_t}(X_t) - X_t \mathrm{w}_t^\wedge$$

where we have defined $\mathrm{w}_t \triangleq \begin{bmatrix} \mathrm{w}_t^\omega & \mathrm{w}_t^a & 0_3 \end{bmatrix}^T$ with covariance $Q = block\_diag(\Sigma^\omega, \Sigma^a, 0_{3\times3})$. The deterministic $f_{u_t}(\cdot)$ can be shown to follow the group affine property (3), and thus by Theorem 1, both the right and left invariant error trajectories are state independent. As shown by Hartley, *et al.* [23], using the first order approximation $\eta_t^r = \exp(\xi_t^r) \approx I + \xi_t^{r\wedge}$ in $g_{u_t}(\cdot)$ of eq. (4) results in

$$g_{u_t}(I + \xi_t^{r\wedge}) = \left( \begin{bmatrix} 0 & 0 & 0 \\ (g)_\times & 0 & 0 \\ 0 & I & 0 \end{bmatrix} \xi_t^r \right)^\wedge \triangleq (A_t\xi_t^r)^\wedge. \quad (22)$$

Using the above derivation, our update step will be computed using the following differential equations for state estimate $\hat{X}_t$ and state covariance $\hat{\Sigma}_t$

$$\frac{\mathrm{d}}{\mathrm{d}t} \hat{X}_t = f_{u_t}(\hat{X}_t) \quad (23)$$

$$\frac{\mathrm{d}}{\mathrm{d}t} \hat{\Sigma}_t = A_t \hat{\Sigma}_t + \hat{\Sigma}_t A_t^T + Ad_{\hat{X}_t} Q Ad_{\hat{X}_t}^T \quad (24)$$

where $A_t$ is defined as in eq. (22). Note the left invariant error $\eta_t^l$ follows a similar derivation, but doesn't result in a constant $A_t$ as the right invariant error does.

### C. Tracking IMU Biases

Generally speaking, when using measurements from an IMU sensor, it is also necessary to estimate the IMU bias for accurate tracking. While the bias doesn't fit into a Lie Group, an "Imperfect InEKF" can be designed as in [13], that still outperforms the standard EKF, even though it doesn't have the same guarantees as the standard InEKF.

The IMU biases can be modeled as slowly varying signals, often done using Brownian Motion. Our models are as follows,

$$\tilde{\omega}_t = \omega_t + b_t^\omega + \mathrm{w}_t^\omega, \qquad \mathrm{w}_t^\omega \sim \mathcal{N}(0, \Sigma^\omega),$$
$$\tilde{a}_t = a_t + b_t^a + \mathrm{w}_t^a, \qquad \mathrm{w}_t^a \sim \mathcal{N}(0, \Sigma^a),$$
$$\dot{b}_t^\omega = \mathrm{w}_t^{b\omega}, \qquad \mathrm{w}_t^{b\omega} \sim \mathcal{N}(0, \Sigma^{b\omega}),$$
$$\dot{b}_t^a = \mathrm{w}_t^{ba}, \qquad \mathrm{w}_t^{ba} \sim \mathcal{N}(0, \Sigma^{ba}). \quad (25)$$

Along with the right invariant error that has been used, we define our bias error as

$$\zeta_t = \begin{bmatrix} \hat{b}_t^\omega - b_t^\omega \\ \hat{b}_t^a - b_t^a \end{bmatrix}. \quad (26)$$

By expanding the differential of the right invariant error as done in [23], an augmented $A_t$ and $Ad_{\hat{X}_t}$ can be found.

$$A_t \triangleq \begin{bmatrix} 0 & 0 & 0 & -\hat{R}_t & 0 \\ (g)_\times & 0 & 0 & -(\hat{v}_t)_\times \hat{R}_t & -\hat{R}_t \\ 0 & I & 0 & -(\hat{p}_t)_\times \hat{R}_t & -\hat{R}_t \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (27)$$

$$Ad_{\hat{X}_t, \hat{b}_t} \triangleq \begin{bmatrix} Ad_{\hat{X}} & 0_{12\times6} \\ 0_{6\times12} & I \end{bmatrix} \quad (28)$$

We augment our original noise vector to be $\mathrm{w}_t \triangleq \begin{bmatrix} \mathrm{w}_t^\omega & \mathrm{w}_t^a & 0_3 & \mathrm{w}_t^{b\omega} & \mathrm{w}_t^{ba} \end{bmatrix}^T$ with covariance $Q = block\_diag(\Sigma^\omega, \Sigma^a, 0_{3\times3}, \Sigma^{b\omega}, \Sigma^{ba})$. The resulting Kalman gain will be split in two as $K = \begin{bmatrix} K^\xi & K^\zeta \end{bmatrix}^T$. $K^\xi$ will be used to update the state estimate $\hat{X}$ via the matrix exponential, while $K^\zeta$ will be for updating the bias estimate $\hat{b}$ via vector addition. By assuming constant IMU measurements between sampling, the deterministic dynamics can be discretized using Euler Integration with resulting equations [23]

$$\hat{R}_{t+1} = \hat{R}_t \exp((\tilde{\omega}_t - \hat{b}_t^\omega)\Delta t)$$
$$\hat{v}_{t+1} = \hat{v}_t + \hat{R}_t(\tilde{a} - \hat{b}_t^a)\Delta t + g\Delta t$$
$$\hat{p}_{t+1} = \hat{p}_t + \hat{v}_t\Delta t + \frac{1}{2}\hat{R}_t(\tilde{a} - \hat{b}_t^a)\Delta t^2 + \frac{1}{2}g\Delta t^2$$
$$\hat{b}_{t+1}^\omega = \hat{b}_t^\omega, \quad \hat{b}_{t+1}^a = \hat{b}_t^a \quad (29)$$

---

**Algorithm 2:** RInEKF for Underwater Navigation

1   $H_1 := \begin{bmatrix} 0 & I & 0 & 0 & 0 \end{bmatrix}$;

2   $H_2 := \begin{bmatrix} 0 & 0 & I & 0 & 0 \end{bmatrix}$;

3   $\hat{\Sigma} = \Sigma_0$;

4   $\hat{X} = X_0$;

5   **while** *receiving data* **do**

6     **if** *IMU measurement* **then**

7       $\hat{X}, \hat{b} = \hat{f}_{u_t}(\hat{X}, \hat{b})$;

8       $\Phi =$
$$\exp\left(\begin{bmatrix} 0 & 0 & 0 & -\hat{R}_t & 0 \\ (g)_\times & 0 & 0 & -(\hat{v}_t)_\times \hat{R}_t & -\hat{R}_t \\ 0 & I & 0 & -(\hat{p}_t)_\times \hat{R}_t & -\hat{R}_t \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \Delta t\right);$$

9       $\hat{\Sigma} = \Phi\hat{\Sigma}\Phi^T + \Phi Ad_{\hat{X},\hat{b}} Q Ad_{\hat{X},\hat{b}}^T \Phi^T \Delta t$;

10    **else if** *z = DVL Measurement* **then**

11      $z = R_{BD}z + (_Bp_{BD})_\times(\tilde{\omega}_t - \hat{b}_t^\omega)$;

12      $S^{-1} = (H_1\hat{\Sigma}H_1^T + \hat{R}M^v\hat{R}^T)^{-1}$;

13      $K^\xi, K^\zeta = \hat{\Sigma}H_1^T S^{-1}$;

14      $\hat{X} = \exp(K^\xi\Pi\hat{X}z)\hat{X}$;

15      $\hat{b} = \hat{b} + K^\zeta\Pi\hat{X}z$;

16      $\hat{\Sigma} = (I - KH_1)\hat{\Sigma}$;

17    **else if** *z = Depth Measurement* **then**

18      $\tilde{\Sigma} = (H_2 Ad_{\hat{X}_t,\hat{b}_t} \hat{\Sigma} Ad_{\hat{X}_t,\hat{b}_t}^T H_2^T)^{-1}$;

19      $S^{-1} = \tilde{\Sigma} - \tilde{\Sigma}\left(\hat{R}^T M^z \hat{R} + \tilde{\Sigma}\right)\tilde{\Sigma}$;

20      $K^\xi, K^\zeta = \hat{\Sigma}Ad_{\hat{X}_t,\hat{b}_t}^T H_2^T S^{-1}$;

21      $\hat{X} = \exp(K^\xi\Pi\hat{X}^{-1}z)\hat{X}$;

22      $\hat{b} = \hat{b} + K^\zeta\Pi\hat{X}^{-1}z$;

23      $\hat{\Sigma} = (I - KH_2 Ad_{\hat{X}_t,\hat{b}_t})\hat{\Sigma}$;

24   **end**

---

Fig. 3. Outline of RInEKF for underwater navigation. Included are (1) the prediction step using IMU measurements, (2) the update step for DVL measurements, and (3) the update step for depth measurements. $H_1$ and $H_2$ are as defined in eqs. (35) and (39), respectively.

We use the shorthand $\hat{X}_{t+1}, \hat{b}_{t+1} = \hat{f}_{u_t}(\hat{X}_t, \hat{b}_t)$ to represent this discretized system. The resulting predict step can be seen in lines 6–9 of Fig. 3.

### D. Velocity Measurements

A common sensor in underwater navigation, the DVL uses acoustic waves to determine the velocity of the vehicle in the DVL frame. We assume that the returned values are corrupted by zero mean Gaussian noise as follows.

$$_D\tilde{v}_{WD} = {}_Dv_{WD} + w_t^{v_D}, \quad w_t^{v_D} \sim \mathcal{N}(0_3, \Sigma^v) \qquad (30)$$

While it can be assumed the DVL and IMU reside in the same frame for conciseness, in reality this is never the case. For completeness, we show the transform here, since it is not a standard rigid body transformation. Note there exists a rigid body transformation $R_{BD}, {}_Bp_{BD}$ between the body (IMU) frame and the DVL frame. Given this, the transformation of

$_D\tilde{v}_{WD}$ to $_B\tilde{v}_{WB}$ is given by [25]

$$_B\tilde{v}_{WB} = R_{BD} \cdot {}_D\tilde{v}_{WD} + (_Bp_{BD})_\times(\tilde{\omega}_t - \hat{b}_t^\omega). \qquad (31)$$

The resulting covariance of $_B\tilde{v}_{WB}$ propagated through this transform is given by

$$M^v = R_{BD}\Sigma^v R_{BD}^T + (_Bp_{BD})_\times(\Sigma^\omega + \Sigma^{b\omega})(_Bp_{BD})_\times^T. \qquad (32)$$

We do this calculation before introducing the measurement to the RInEKF and thus model the DVL in the IMU frame,

$$_B\tilde{v}_{WB} = R_t^T v_t + w_t^v, \quad w_t^v \sim \mathcal{N}(0_3, M^v) \qquad (33)$$

Put into the Lie Group, this has right invariant observation structure as in eq. (9),

$$\begin{bmatrix} _B\tilde{v}_{WB} \\ -1 \\ 0 \end{bmatrix} = \begin{bmatrix} R_t^T & -R_t^T v_t & -R_t^T p_t \\ 0_{1\times3} & 1 & 0 \\ 0_{1\times3} & 0 & 1 \end{bmatrix} \begin{bmatrix} 0_{3\times1} \\ -1 \\ 0 \end{bmatrix} + \begin{bmatrix} w_t^v \\ 0 \\ 0 \end{bmatrix}. \qquad (34)$$

We then linearize the right invariant innovation as in eq. (13).

$$\begin{aligned} \Pi V_t^r &= \Pi\hat{X}_t(z_t^r - \hat{z}_t^r) \\ &= \Pi\begin{bmatrix} (\xi_t^R)_\times & \xi_t^v & \xi_t^p \\ 0_{1\times3} & 0 & 0 \\ 0_{1\times3} & 0 & 0 \end{bmatrix}\begin{bmatrix} 0_{3\times1} \\ -1 \\ 0 \end{bmatrix} + \Pi\hat{X}_t W_t \\ &= -\begin{bmatrix} 0 & I & 0 \end{bmatrix}\xi_t^r + \hat{R}_t w_t^v \triangleq -H_1\xi_t^r + \hat{R}_t w_t^v \end{aligned} \qquad (35)$$

With a linearized innovation, we can use conventional Kalman theory to derive the gain $K_t$ and the rest of the update step [14]. The resulting steps can be seen in lines 10–16 of Fig. 3.

### E. Depth & Singleton Measurements

The pressure sensor is another common underwater sensor that is particularly useful as it returns a measurement that is directly proportional to the z component of $p_t = \begin{bmatrix} p_t^x & p_t^y & p_t^z \end{bmatrix}^T$. Note in nearly all cases, the pressure sensor is not located at the body frame, but can be trivially transformed between frames using a rigid body transformation. For conciseness, we assume it's in the body frame. We again model this with zero mean Gaussian noise as

$$_W\tilde{p}_{WB}^z = {}_Wp_{WB}^z + w_t^z, \quad w_t^z \sim \mathcal{N}(0_1, \sigma_z). \qquad (36)$$

In Lie Group form, this is similar, but not identical, to a left invariant observation, as shown in eq. (10)

$$_W\tilde{z}_{WB} = \begin{bmatrix} 0_{1\times2} & 1 & 0_{1\times2} \end{bmatrix}\begin{bmatrix} R_t & v_t & p_t \\ 0_{1\times3} & 1 & 0 \\ 0_{1\times3} & 0 & 1 \end{bmatrix}\begin{bmatrix} 0_{3\times1} \\ 0 \\ 1 \end{bmatrix} + w_t^z. \qquad (37)$$

However, the right and left multiplication of the state makes linearization of the innovation as done in eq. (13) impossible. Instead, we create "pseudo" measurements for $p_x$ and $p_y$ to force the measurement into left invariant observation form. We do this by giving these "pseudo" measurements infinite covariance and measurements equal to our current state estimate. These measurements immediately cancel out in the left invariant
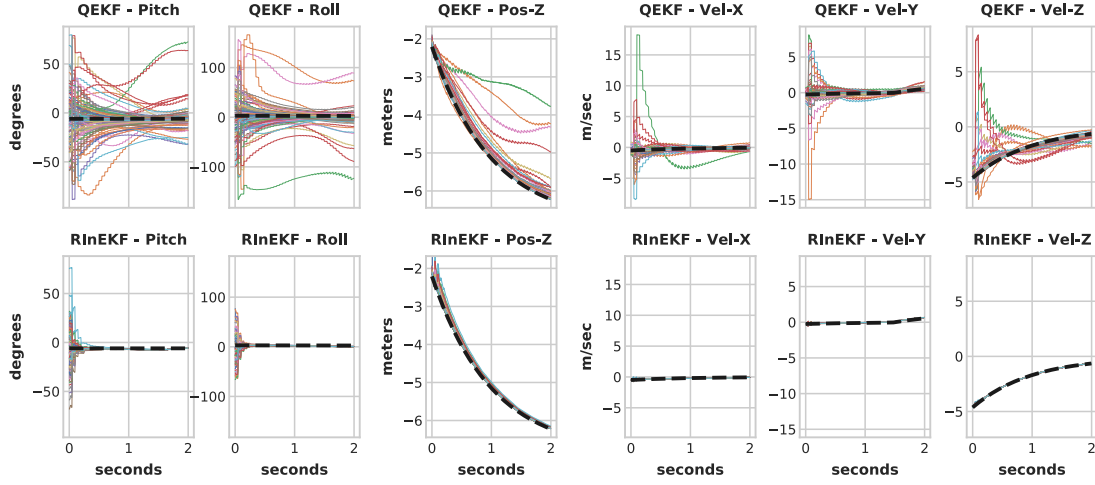
Fig. 4. Both the QEKF and InEKF were run 100 times (offline) on the first 2 seconds of simulation data using the same noise statistics, initial covariance, and initial starting point. Simulation data was generated by our in-house simulator based on Holodeck [26]. The initial starting point was randomly chosen from a Gaussian centered around the true mean and with covariances shown in Table II. The above plots show the resulting trajectories, with the dashed black line representing the true state. The RInEKF converged faster than the QEKF for all observable states and trajectories. Velocity is shown in the body (IMU) frame, since yaw is unobservable [23]. Note that pitch and roll are Euler angles derived from the change of orientation between the local and world frame, where we first rolled around the fixed x-axis, then pitched about the fixed y-axis, and finally yawed about the fixed z-axis.

innovation

$$V_t^l = \hat{X}_t^{-1} \left( \begin{bmatrix} \hat{p}_t^x \\ \hat{p}_t^y \\ \tilde{p}_t^z \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} \hat{p}_t^x \\ \hat{p}_t^y \\ \hat{p}_t^z \\ 0 \\ 1 \end{bmatrix} \right) = \hat{X}_t^{-1} \begin{bmatrix} 0 \\ 0 \\ \tilde{p}_t^z - \hat{p}_t^z \\ 0 \\ 0 \end{bmatrix}. \quad (38)$$

Note measurements are given by ˜ and estimates by ˆ. To linearize $\Pi V_t^l$, a similar linearization as done in eq. (13) is performed

$$\Pi V_t^l = \Pi \hat{X}_t^{-1}(z_t^l - \hat{z}_t^l)$$

$$= - \begin{bmatrix} 0 & 0 & I \end{bmatrix} \xi_t^l + \hat{R}^T w_t^p \triangleq -H_2 \xi_t^l + \hat{R}^T w_t^p \quad (39)$$

where $w_t^p$ is the noise of our true measurement and "pseudo" measurements.

However, note this linearization is about the left invariant $\xi_t^l$ instead of $\xi_t^r$. A simple transformation exists between the two as follows

$$\eta^l = X_t^{-1} \hat{X}_t = \hat{X}_t^{-1} \hat{X}_t X_t^{-1} \hat{X}_t = \hat{X}_t^{-1} \eta^r \hat{X}_t$$

$$\Rightarrow \exp(\xi^{l\wedge}) = \hat{X}_t^{-1} \exp(\xi^{r\wedge}) \hat{X}_t$$

$$= \exp(\hat{X}_t^{-1} \xi^{r\wedge} \hat{X}_t)$$

$$= \exp((Ad_{\hat{X}_t^{-1}} \xi^r)^\wedge) \quad (40)$$

$$\Rightarrow \xi^l = Ad_{\hat{X}_t^{-1}} \xi^r$$

Using this results in a linearized innovation as

$$\Pi V_t^l = -H_2 Ad_{\hat{X}_t^{-1}} \xi_t^r + \hat{R}^T w_t^p. \quad (41)$$

The infinite covariance of the "pseudo" measurements may be approximated with a very large value. However, this can cause problems if not made sufficiently large. Instead, it can be determined analytically as follows. Note the only location this infinite covariance, $L$, will be used is in the calculation of

$Cov(\Pi V_t^l)^{-1} = S^{-1}$ as follows

$$S^{-1} = \lim_{L \to \infty} \left( H_2 Ad_{\hat{X}_t^{-1}} \Sigma_t Ad_{\hat{X}_t^{-1}}^T H_2^T \right.$$

$$\left. + \hat{R}_t^T \begin{bmatrix} L & 0 & 0 \\ 0 & L & 0 \\ 0 & 0 & \sigma_z \end{bmatrix} \hat{R}_t \right)^{-1} \quad (42)$$

Note that limits may be passed into operations given they are continuous and given the limit is finite after doing so. The above equation satisfies the former since matrix multiplication and inversion are both continuous operations, but fails to satisfy the latter. However, by leveraging the Woodbury matrix identity [27], which states given $n \times n$ matrices $A, B$, we have

$$(A + B)^{-1} = A^{-1} - A^{-1}(B^{-1} + A^{-1})A^{-1} \quad (43)$$

Using this, and defining $\tilde{\Sigma}_t \triangleq (H_2 Ad_{\hat{X}_t^{-1}} \hat{\Sigma}_t Ad_{\hat{X}_t^{-1}}^T H_2^T)^{-1}$, we arrive at

$$S^{-1} = \lim_{L \to \infty} \left( \tilde{\Sigma}_t - \tilde{\Sigma}_t \left( \hat{R}_t^T \begin{bmatrix} L & 0 & 0 \\ 0 & L & 0 \\ 0 & 0 & \sigma_z \end{bmatrix}^{-1} \hat{R}_t + \tilde{\Sigma}_t \right)^{-1} \tilde{\Sigma}_t \right)$$

$$= \tilde{\Sigma}_t - \tilde{\Sigma}_t \left( \hat{R}_t^T \lim_{L \to \infty} \begin{bmatrix} \frac{1}{L} & 0 & 0 \\ 0 & \frac{1}{L} & 0 \\ 0 & 0 & \frac{1}{\sigma_z} \end{bmatrix} \hat{R}_t + \tilde{\Sigma}_t \right)^{-1} \tilde{\Sigma}_t$$

$$= \tilde{\Sigma}_t - \tilde{\Sigma}_t \left( \hat{R}_t^T \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sigma_z} \end{bmatrix} \hat{R}_t + \tilde{\Sigma}_t \right)^{-1} \tilde{\Sigma}_t$$

$$\triangleq \tilde{\Sigma}_t - \tilde{\Sigma}_t \left( \hat{R}_t^T M^z \hat{R}_t + \tilde{\Sigma}_t \right)^{-1} \tilde{\Sigma}_t \quad (44)$$

TABLE I
SIMULATION NOISE STATISTICS

| Measurement Type | Noise std. |
|---|---|
| Angular Velocity | $.00009$ rad / sec / $\sqrt{\text{Hz}}$ |
| Linear Acceleration | $.0002$ m / sec$^2$ / $\sqrt{\text{Hz}}$ |
| Gyroscope Bias | $.0003$ rad / sec$^2$ / $\sqrt{\text{Hz}}$ |
| Accelerometer Bias | $.0001$ m / sec$^3$ / $\sqrt{\text{Hz}}$ |
| DVL Sensor | $.02626$ m / sec |
| Pressure Sensor | $.255$ m |

TABLE II
INITIAL NOISE COVARIANCE

| State | Initial std. |
|---|---|
| Orientation | 30 deg |
| Velocity | 2.0 m/sec |
| Position | 1.0 m |
| Gyroscope Bias | $.005$ rad / sec |
| Accelerometer Bias | $0.05$ m / sec$^2$ |

By using this closed-form solution of our limit, our "pseudo" measurements are modeled as being infinitely unreliable, and as such will be completely ignored by the InEKF. Using this allows use of the depth measurement in the correction step, even though it originally doesn't fit the structure of an invariant measurement. Leveraging this form opens up the type of measurements that can be used by the InEKF to include any form of singleton measurements, such as the pressure sensor.

After applying all this, as before, general Kalman Filter theory can be applied to derive the rest of the update step. The results can be seen in lines 17–23 of Fig. 3.

Note since both of the sensor models have no dependence upon the IMU biases, both $H_1$ and $H_2$ are appended with zeros, as seen in Fig. 3, and the augmented adjoint $Ad_{\hat{X}_t, \hat{b}_t}$ is used.

## V. RESULTS

To evaluate the RInEKF for underwater navigation, experiments were done in our in-house simulator built upon Holodeck [26] and Unreal Engine 4 with the vehicle as seen in Fig 1. The vehicle was equipped with an IMU, DVL and pressure sensor with noise statistics as shown in Table I. The IMU was sampled at 200 Hz, DVL at 20 Hz and pressure sensor at 100 Hz. All results were compared to that of a standard Quaternion-based EKF (QEKF) [4]. Note both yaw and various bias states are unobservable, and are thus not shown in these plots [23]. Since yaw is unobservable, velocity is also shown in the body frame.

### A. RInEKF Convergence

We tested convergence via a Monte Carlo method. A simulation was run consisting of a simple descent of the vehicle with thrusters slightly pushing forward. Then, each filter was run 100 times for the first 2 seconds of simulation with varying initial starting points. The starting points were chosen by sampling from a normal distribution with mean zero and standard deviation as seen in Table II. This sample was then combined with the true starting mean via right multiplication, $\exp(\xi^\wedge)X$. The results for pitch, roll, velocity (in body frame), and z-component of position can be seen in Fig. 4. Every trajectory converged faster than that of the QEKF for each of the observable states.

### B. RInEKF Localization

Similarly, we evaluated long term convergence under high initial uncertainty and initialization error. This was done by scaling the standard deviation of the sampled initialization as shown in Table II by 0.5, 1.0, 1.5, and 2.0. The RInEKF and QEKF were run 400 times, 100 for each standard deviation scale, on the same 100 seconds of simulation data with random initial starting points, chosen as in Subsection V-A. The mean absolute error (MAE) was then calculated over the last 5 seconds of filter estimates to evaluate performance. Results are shown in Fig. 5. Note for the lower covariance scales, namely 0.5 and 1.0, the QEKF and RInEKF MAE routinely outperformed each other depending on which simulation data was used. To counteract this, a simulation run was chosen where they performed nearly identically in these low covariance settings. Ideally, a Monte Carlo simulation spanning all possible trajectories would be ran; however, this space is much too big to sample sufficiently. However, regardless of the trajectory, the RInEKF routinely outperfomed the QEKF under high initial uncertainty, with very similar results to that in Fig. 5 for all tested trajectories.

Notice how as the initial uncertainty increases, the QEKF increasingly struggles to converge, even after 100 seconds of run time. This is likely due to poor linearization points at the beginning, as well as the QEKF struggling to handle large initial uncertainty. On the other hand, the RInEKF performance hardly shows a difference between poor and perfect initial starting conditions, localizing extremely well in both scenarios. This suggests that not only does the RInEKF converge faster, but also more robustly from nearly any initial position and covariance.

### C. RInEKF Timing

Further, due to the extra computation needed for the update depth step as seen in Fig. 3, it remained to be seen if this would cause a significant delay in actual implementation as compared to the QEKF. To test this, we ran each step of each filter 3500 times in a Python implementation and plotted the computation time distributions in Fig. 6 as a violin plot, which plots the estimated probability density function.

Notice that while both the predict and the update DVL step had slight increases in time, these are rather negligible and likely won't impact performance significantly.

The update depth step did have an increase of about 0.15 ms. This is likely due to the extra steps required in the Woodbury matrix identity along with the larger matrix multiplications. The QEKF update depth step has a $1 \times 15$ $H$ matrix, along with a $1 \times 1$ $S$ matrix as compared to a $3 \times 15$ $H$ and $3 \times 3$ $S$ in the RInEKF, resulting in faster computations than all the other update steps. However, the 0.15 ms gap is small enough to be calculated between the sample rates of our sensors and is likely to be even smaller in a C++ implementation. Thus, the 0.15 ms delay is a small price to pay for the increased convergence and localization results shown above.

## VI. CONCLUSION

Using the recently developed InEKF, we derived an observer for underwater navigation using common sensors in the underwater regime such as an IMU, DVL and pressure sensor. This observer seeks to track rotation, position, velocity and IMU biases. Further, we derived an extension to the usual invariant measurement models to allow measurements that only include
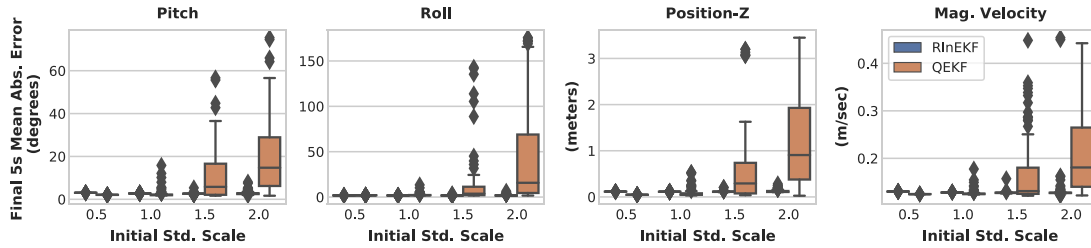
Fig. 5. To test localization under increasingly uncertain initial positions, the QEKF and RInEKF were run 400 times on 100 seconds of simulation data with initial standard deviations (std.) scaled by 0.5, 1, 1.5 and 2.0 times that of Table II. The initial starting point was randomly chosen from a Gaussian centered around the true mean and with standard deviation as stated before. The resulting mean absolute error (MAE) of the last 5 seconds of estimates was plotted vs the scale of the standard deviations. Notice how the RInEKF keeps a tight distribution of results even under high uncertainty, while the QEKF increasingly fails to localize beginning even at standard deviation scale 1.0. Pitch and roll were defined as in Fig. 4.
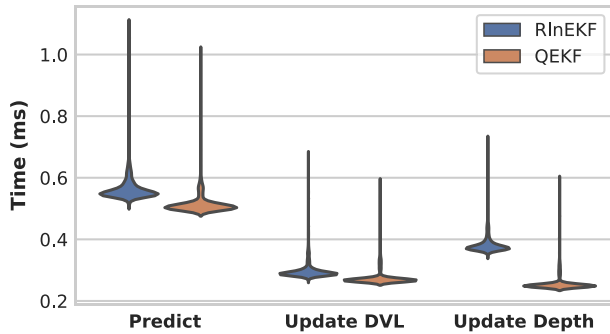


Fig. 6. Computation time comparison of RInEKF and QEKF, both implemented in Python. Each filter step was run 3500 times and the resulting computation time distributions can be seen above as violin plots. A violin plot displays the estimated probability density function. Note the negligible difference in computation time of the predict and update DVL steps. The update depth step for the RInEKF has added complexity due to leveraging the Woodbury matrix identity, but this complexity only contributes about 0.15 ms per iteration, small enough to not be a concern.

part of a 3D state, such as a depth measurement, opening the avenue to many sensor models to be used in the InEKF. We then compared the RInEKF to a Quaternion-based EKF with favorable results in convergence and accuracy and comparable results in computation time. Future work includes integration of other common underwater sensors such as sonar and camera-based sensors along with comparison on real world data.

## REFERENCES

[1] L. Paull, S. Saeedi, M. Seto, and H. Li, "AUV navigation and localization: A review," *IEEE J. Ocean. Eng.*, vol. 39, no. 1, pp. 131–149, Jan. 2014.

[2] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME-J. Basic Eng.*, vol. 82, no. Series D., pp. 35–45, 1960.

[3] M. T. Sabet, H. Mohammadi Daniali, A. Fathi, and E. Alizadeh, "A low-cost dead reckoning navigation system for an AUV using a robust AHRS: Design and experimental analysis," *IEEE J. Ocean. Eng.*, vol. 43, no. 4, pp. 927–939, Oct. 2018.

[4] J. Sola, "Quaternion kinematics for the error-state kalman filter," 2015, *arXiv:1711.02508*.

[5] I. Ullah, J. Chen, X. Su, C. Esposito, and C. Choi, "Localization and detection of targets in underwater wireless sensor using distance and angle based algorithms," *IEEE Access*, vol. 7, pp. 45693–45704, 2019.

[6] E. Olson, J. J. Leonard, and S. Teller, "Robust range-only beacon localization," *IEEE J. Ocean. Eng.*, vol. 31, no. 4, pp. 949–958, Oct. 2006.

[7] A. Mallios, P. Ridao, D. Ribas, F. Maurelli, and Y. Petillot, "EKF-SLAM for AUV navigation under probabilistic sonar scan-matching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 4404–4411.

[8] J. Li, M. Kaess, R. M. Eustice, and M. Johnson-Roberson, "Pose-graph SLAM using forward-looking sonar," *IEEE Robot. Automat. Lett.*, vol. 3, no. 3, pp. 2330–2337, Jul. 2018.

[9] A. Kim and R. M. Eustice, "Real-time visual SLAM for autonomous underwater hull inspection using visual saliency," *IEEE Trans. Robot.*, vol. 29, no. 3, pp. 719–733, Jun. 2013.

[10] G. T. Donovan, "Position error correction for an autonomous underwater vehicle inertial navigation system (INS) using a particle filter," *IEEE J. Ocean. Eng.*, vol. 37, no. 3, pp. 431–445, Jul. 2012.

[11] S. Carreño, P. Wilson, P. Ridao, and Y. Petillot, "A survey on terrain based navigation for AUVs," in *Proc. IEEE/MTS OCEANS Conf. Exhib.*, Seattle, WA, USA, Sep. 2010, pp. 1–7.

[12] S. Bonnabel, "Left-invariant extended kalman filter and attitude estimation," in *Proc. IEEE Conf. Decis. Control*, New Orleans, LA, USA, Dec. 2007, pp. 1027–1032.

[13] A. Barrau, "Non-linear state error based extended kalman filters with applications to navigation," Ph.D. dissertation, *Mines Paristech*, 2015.

[14] A. Barrau and S. Bonnabel, "Invariant kalman filtering," *Annu. Rev. Control*, *Robot., Auton. Syst.*, vol. 1, no. 1, pp. 237–257, 2018.

[15] A. Barrau and S. Bonnabel, "The invariant extended kalman filter as a stable observer," *IEEE Trans. Autom. Control*, vol. 62, no. 4, pp. 1797–1812, Apr. 2017.

[16] J. G. Mangelson, M. Ghaffari, R. Vasudevan, and R. M. Eustice, "Characterizing the uncertainty of jointly distributed poses in the lie algebra," *IEEE Trans. Robot.*, vol. 36, no. 5, pp. 1371–1388, Oct. 2020.

[17] N. Aghannan and P. Rouchon, "On invariant asymptotic observers," in *Proc. the IEEE Conf. Decis. Control*, vol. 2, Las Vegas, NV, USA, Dec. 2002, pp. 1479–1484.

[18] S. Bonnabel, P. Martin, and P. Rouchon, "Non-linear symmetry-preserving observers on lie groups," *IEEE Trans. Autom. Control*, vol. 54, no. 7, pp. 1709–1713, Jul. 2009.

[19] M. Brossard, S. Bonnabel, and A. Barrau, "Invariant kalman filtering for visual inertial SLAM," in *Proc. Int. Conf. Inf. Fusion*, Cambridge, United Kingdom, Jul. 2018, pp. 2021–2028.

[20] K. Wu, T. Zhang, D. Su, S. Huang, and G. Dissanayake, "An invariant-EKF VINS algorithm for improving consistency," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Vancouver, Canada, Sep. 2017, pp. 1578–1585.

[21] M. Barczyk and A. F. Lynch, "Invariant observer design for a helicopter UAV aided inertial navigation system," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 3, pp. 791–806, May 2013.

[22] M. Barczyk and A. F. Lynch, "Invariant extended kalman filter design for a magnetometer-plus-GPS aided inertial navigation system," in *Proc. IEEE Conf. Decis. Control*, Orlando, FL, USA, Dec. 2011, pp. 5389–5394.

[23] R. Hartley, M. Ghaffari, R. M. Eustice, and J. W. Grizzle, "Contact-aided invariant extended kalman filtering for robot state estimation," *Int. J. Robot. Res.*, vol. 39, no. 4, pp. 402–430, 2020.

[24] B. C. Hall, *Lie Groups, Lie Algebras, and Representations an Elementary Introduction*. Berlin, Germany: Springer, 2015.

[25] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*. 1st ed., USA: CRC Press, Inc., 1994.

[26] J. Greaves *et al.*, "Holodeck: A High Fidelity Simulator," GitHub 2018.

[27] M. A. Woodbury, Inverting Modified Matrices, ser. Memorandum Report / Statistical Research Group, no. 42, Princeton. Statistical Research Group, 1950.