A Right Invariant Extended Kalman Filter for Object based SLAM

Report by*: Mayank Deshpande (120387333) and Tanmay Pancholi (120116711)

***Abstract*— Due to the recent advancements in deep learning-based object recognition and estimation, it is now feasible to implement object-level SLAM, which involves estimating the pose of each object during the SLAM process. This paper proposes a novel right-invariant extended Kalman filter (RI-EKF) for object-based SLAM, utilizing a unique Lie group structure. Observability analysis demonstrates that the proposed algorithm automatically preserves the proper unobservable subspace, unlike conventional EKF (Std-EKF) based SLAM algorithms. Consequently, the proposed algorithm exhibits superior consistency compared to Std-EKF. Simulations and real-world experiments validate not only the consistency and accuracy of the proposed algorithm but also its practicality for object-based SLAM problems.**

***Index Terms* - SLAM, Localization, Mapping**

I. INTRODUCTION

Over the past decade, visual sensors have gained widespread popularity due to their affordability and ability to capture rich information. Consequently, various frameworks have been developed for visual SLAM. However, most of these efforts focus solely on low-level features such as points, lines, or planes, overlooking the potential of high-level features like objects, which offer strong geometric constraints . High-level features possess several advantages over low-level features, including the ability to facilitate semantic loop closure, High level features are more robust and hence have longer tracking distance, and incorporate the intrinsic relationships among low-level features like points, lines and planes.

Early SLAM works that considered object features include. These works focused on recognizing object features using traditional methods. Recent works have utilized neural networks to detect object features and optimization methods to estimate the poses of both the objects and the robot. However, optimization-based methods require much more computation than filter-based methods when the robot trajectory is very long. Thus, they are not suitable for deployment on lightweight platforms. Therefore, it is important to develop filter-based methods.Nevertheless, to the best of our knowledge, works focused on the filter based SLAM frameworks that consider the object features are still blank. One of the reasons is that the conventional filter based SLAM (like standard EKF) usually suffers from the problem of inconsistency. And a consistent filter based estimator requires elaborate modelling and algorithmic design. When a SLAM system relies on an inconsistent filter, it tends to underestimate the uncertainty associated with its estimated state. This gradual underestimation can eventually lead to poor performance and even cause the algorithm to diverge.

The reason for these inconsistencies is that filter-based SLAM methods like EKF, linearizes the complex system model which breaks the unobservable space which results in the inclusion of random information into the system which makes the estimate inconsistent over the period of time. The methods like first Jacobian estimate (FEJ) and observability constraint (OC) add additional constraints into the system to tackle the problem of inconsistency.

Rather than imposing artificial constraints on the estimator, algorithms based on Lie group theory have demonstrated the ability to naturally handle invariances, including observability constraints, in point feature-based SLAM algorithms. Leveraging the properties of the invariant tangent vector field, Lie group theory gives rise to the invariant-EKF methodology. To mitigate the inconsistency issue and enhance the accuracy of state estimates, a right invariant EKF (RI-EKF) algorithm is proposed for 2D point feature-based SLAM.

Existing filter-based SLAM methods focus exclusively on point features, leaving the question of how to consistently handle object features (poses) unaddressed. This paper addresses this gap by proposing a consistent EKF algorithm for object-based SLAM. The key contributions of this paper include:

- Designing an invariant EKF based on a new Lie group for SLAM with object features.
- Demonstrating, through observability analysis, that the proposed algorithm inherently maintains

*This work was originally authored by Yang Song, Zhuqing Zhangy, Jun Wuy, Yue Wangy, Liang Zhao, Shoudong Huang

the correct unobservable subspace.
- Validating the effectiveness of the proposed algorithm through simulations and real-data experiments.

Notations:
- Bold lowercase and uppercase letters represent vectors and matrices/elements in the Lie group, respectively.
- SO(3) denotes the 3D special rotation group, comprising all rotation transformations in R³.
- so(3) represents the Lie algebra of SO(3), consisting of all 3 × 3 skew-symmetric matrices.
- $(\cdot)^\wedge$ denotes the skew-symmetric operator that converts a 3-dimensional vector into a skew-symmetric matrix.
- expG represents the exponential map on a Lie group G.
- logG is the inverse of the exponential map on a Lie group G.
- N(0, P) represents a zero-mean Gaussian distribution with covariance P.

II. OBJECT BASED SLAM PROBLEM

In this work, we represent object features as their 3D poses within the environment. As a robot navigates an uncharted 3D environment, it observes various object features. Object-based SLAM aims to estimate the robot's current pose and the poses of all observed object features using both process and observation models.

1) State Space: The object features are denoted as:

$$(\mathbf{R}^f, \mathbf{p}^f), \tag{1}$$

And the state space can be defined as a set of all states having robot pose and the *K* observed features, where

$$\mathcal{G}_K = \Big\{(\mathbf{R}^r, \mathbf{R}^{f_1}, \ldots, \mathbf{R}^{f_K}, \mathbf{p}^r, \mathbf{p}^{f_1}, \ldots, \mathbf{p}^{f_K}) \\ |\mathbf{R}^r, \mathbf{R}^{f_j} \in SO(3),\ \mathbf{p}^r, \mathbf{p}^{f_j} \in R^3\}, \tag{2}$$

where $\mathbf{R}^r, \mathbf{R}^{fj} \in SO(3)$ represent the robots and features rotation respectively and $\mathbf{p}^r, \mathbf{p}^{fj} \in R^3$ represent the robots and features positions respectively, all in global coordinate system.

2) Process Model: The process model describes how the systems states evolve over time. The Process model for object based SLAM problem is defined as follows:

The lie algebra $so(3) \cong R^3$, keeping this in mind we can define the exponential map (Converts Lie-Algebra to corresponding Lie-group) of $SO(3)$ can be described as follows: For $\boldsymbol{\xi} \in R^3$,

$$\begin{array}{rccc} \exp^{SO(3)}: & R^3 & \rightarrow & SO(3) \\ & \boldsymbol{\xi} & \rightarrow & \sum_{k=0}^{\infty} \frac{(\boldsymbol{\xi}^\wedge)^k}{k!}. \end{array} \tag{3}$$

We will use the first-order integration scheme of discrete noisy process model to find the estimated state at time *n+1* ($\mathbf{X}_{n+1}$), based on the state at time step *n* ($\mathbf{X}_n$), Odometry $\mathbf{U}_n$ and odometry noise $\mathbf{w}_n$:

$$\mathbf{X}_{n+1} = f(\mathbf{X}_n, \mathbf{U}_n, \mathbf{w}_n)$$

In first order integration scheme for discrete noisy process model we first descritize using euler integration:

$$x_{n+1} = x_n + \Delta t \cdot v_n \cos(\theta_n) + w_{x,n}$$

where $\mathbf{X}_n$ and $\mathbf{X}_{n+1}$ represent the state at time steps *n* and *n+1*, and Δt is the time step.Then we factor in the noisy control inputs:

$$v_n = \text{actual}_v + \epsilon_v$$
$$\omega_n = \text{actual}_\omega + \epsilon_\omega$$

where $\epsilon_v \sim \mathcal{N}(0, \sigma_v^2)$ and $\epsilon_\omega \sim \mathcal{N}(0, \sigma_\omega^2)$. Using these notations the discrete time update equations for the robots position orientation can be represented as follows :

$$\mathbf{R}^r_{n+1} = \mathbf{R}^r_n \exp^{SO(3)}(\mathbf{w}R_n)\mathbf{R}^u_n$$
$$\mathbf{p}^r_{n+1} = \mathbf{p}^r_n + \mathbf{R}^r_n(\mathbf{p}^u + \mathbf{w}p_n)$$

where $\mathbf{R}^u_n$ is the rotation matrix obtained from the control input $\mathbf{U}_n = (\mathbf{R}^n_u, \mathbf{p}^n_u)$ and $\mathbf{w}R_n$ is the skew symmetric matrix obtained from the noise vector $\mathbf{w}_n$ using skew symmetric operator. Substituting this in equation (2) we get,

$$\mathbf{X}_{n+1} = (\mathbf{R}^r_n \exp^{SO(3)}(\mathbf{w}R_n)\mathbf{R}^u_n, \mathbf{R}^f_n, \mathbf{p}^r_n \\ + \mathbf{R}^r_n(\mathbf{p}^u + \mathbf{w}p_n), \mathbf{p}^f_n), \tag{4}$$

3) Observation model: The goal is to obtain the relative rotation $\mathbf{R}_z$ and the relative position $\mathbf{p}_z$ of the object feature $(\mathbf{R}_f, \mathbf{p}_f)$ in the $(\mathbf{n}+\mathbf{1})^{th}$ robot

frame $(\mathbf{R}^r_{n+1}, \mathbf{p}^r_{n+1})$ considering the observation noise $\mathbf{n+1}$. The corresponding observation model can be described as:

$$\mathbf{Z} = h(\mathbf{X}_{n+1}, \mathbf{v}_{n+1}) = (\mathbf{R}^z, \mathbf{p}^z), \tag{5}$$

We know that composing rotations involves multiplying the rotation matrices. The exponential map from $SO(3)$ to 3D rotations provide a mathematical way to convert rotation vectors into rotation matrices:

$$\mathbf{R}^z \quad = \exp^{SO(3)}(\mathbf{v}^R_{n+1})(\mathbf{R}^r_{n+1})^\top \mathbf{R}^f_{n+1}$$

In the above equation, the exponential map $\exp^{SO(3)}$ converts $\mathbf{v}^R_{n+1}$ into rotation matrix. In the context of lie group this operation ensures that small rotations are correctly incorporated into the overall rotation composition.The term $(\mathbf{R}^r_{n+1})^\top$ brings rotation from global frame to robots local frame. Whereas $\mathbf{R}^f_{n+1}$ transforms the object features local frame to global frame.

By combining these operations, the relative rotation $\mathbf{R}_z$ is computed in a way that correctly transforms the object features local frame (given by $\mathbf{R}^f_{n+1}$) to robots local frame at time step $\mathbf{n+1}$.

Similarly, the relative position $\mathbf{p}_z$ can be calculated as the difference between the object features position in the robots frame and the robots position:

$$\mathbf{p}^z \quad = (\mathbf{R}^r_{n+1})^\top(\mathbf{p}^f_{n+1} - \mathbf{p}^r_{n+1}) + \mathbf{v}^p_{n+1}$$

where $\mathbf{v}_{n+1} = \begin{pmatrix} \mathbf{v}^R_{n+1} \\ \mathbf{v}^p_{n+1} \end{pmatrix} \in R^6 \sim \mathcal{N}(\mathbf{0}, \mathbf{\Omega}_{n+1})$ is the observation noise. In this equation $(\mathbf{p}^f_{n+1} - \mathbf{p}^r_{n+1})$ is the position of object feature relative to the robot in the object features local frame. The term $(\mathbf{R}^r_{n+1})^\top$ is used to transform the difference vector from object features local coordinate frame to the global frame.

III. RI-EKF FOR OBJECT BASED SLAM

A. RI-EKF framework for object based SLAM

1) A novel Lie group structure on state space: An operator $\oplus$ has been defined for all the elements belonging to $(\mathbf{R}_i, \mathbf{R}^i_f, \mathbf{p}^i_r, \mathbf{p}^i_f) \in G$ *i = 1,2*:

$$\begin{aligned}(\mathbf{R}^r_1, \mathbf{R}^f_1, \mathbf{p}^r_1, \mathbf{p}^f_1) \oplus (\mathbf{R}^r_2, \mathbf{R}^f_2, \mathbf{p}^r_2, \mathbf{p}^f_2) \\ = (\mathbf{R}^r_1\mathbf{R}^r_2, \mathbf{R}^f_1\mathbf{R}^f_2, \mathbf{R}^r_1\mathbf{p}^r_2 + \mathbf{p}^r_1, \mathbf{R}^r_1\mathbf{p}^f_2 + \mathbf{p}^f_1).\end{aligned} \tag{6}$$

Equipped with $\oplus$ the state space G becomes a Lie group and is isomorphic to $SE_{K+1}(3) \times (SO(3))^K$, where K is the number of observed features. The notation $\ominus$, is defined as $\mathbf{X_a} \ominus \mathbf{X_b} = \mathbf{X_a} \oplus \mathbf{X_b^{-1}}$ and $\mathbf{X_b} \oplus \mathbf{X_b^{-1}} = \mathbf{X_b^{-1}} \oplus \mathbf{X_b} = (\mathbf{I}, \mathbf{I}, \mathbf{0}, \mathbf{0})$.The Lie algebra $g \cong se_{K+1}(3) \times (so(3))^K \cong R^{6+6K}$. An element ξ in lie algebra g can be constructed as:

$$\boldsymbol{\xi}^\top = ((\boldsymbol{\xi}^{R^r})^\top, (\boldsymbol{\xi}^{R^f})^\top, (\boldsymbol{\xi}^{p^r})^\top, (\boldsymbol{\xi}^{p^f})^\top), \tag{7}$$

The exponential map $\exp^G$ on this lie group can be defined by:

$$\begin{aligned}\exp^{\mathcal{G}}(\boldsymbol{\xi}) =& (\exp^{SO(3)}(\boldsymbol{\xi}^{R^r}), \exp^{SO(3)}(\boldsymbol{\xi}^{R^f}) \\ & J_l(\boldsymbol{\xi}^{R^r})\boldsymbol{\xi}^{p^r}, J_l(\boldsymbol{\xi}^{R^r})\boldsymbol{\xi}^f),\end{aligned} \tag{8}$$

where, the jacobian ensures that the translational part of $\exp^G(\xi)$ is consistent with the lie group structure. It basically scales and transforms vector to ensures consistence.

$$J_l(\boldsymbol{\xi}^{R^r}) = \sum_{k=0}^{\infty} \frac{((\boldsymbol{\xi}^{R^r})^\wedge)^k}{(k+1)!}. \tag{9}$$

The error state ξ is obtained using the exponential map $\exp^G$ and is the difference between the true state and the estimated state:

$$\xi = \exp_G^{-1}(\mathbf{X} \ominus \hat{\mathbf{X}})$$

The Lie group operation $\oplus$ combines the estimated state $\hat{\mathbf{X}}$ with the error state ξ to obtain the true state X:

$$\mathbf{X} = \exp^{\mathcal{G}}(\boldsymbol{\xi}) \oplus \hat{\mathbf{X}}, \tag{10}$$

2) Propagation: The propagation step in EKF involves predicting the state of the system at the next time step based on the current state and the system dynamics. On the basis of the Lie group structure defined in (6) the process model (4) can be re written as:

$$\begin{aligned}\mathbf{X}_{n+1} &= (\mathbf{R}^r_n \exp^{SO(3)}(\mathbf{w}R_n)\mathbf{R}^u_n, \mathbf{R}^f_n, \mathbf{p}^r_n \\ &\quad + \mathbf{R}^r_n(\mathbf{p}^u + \mathbf{w}p_n), \mathbf{p}^f_n) \\ &= (\mathbf{R}^r_n, \mathbf{R}^f_n, \mathbf{p}^r_n, \mathbf{p}^f_n) \oplus (\exp^{SO(3)}(\mathbf{w}^R_n)\mathbf{R}^u_n, \mathbf{I}_3, \mathbf{p}^u_n \\ &\quad + \mathbf{w}^p_n, \mathbf{0}_{3\times1}) \\ \mathbf{X}_{n+1} &= \mathbf{X}_n \oplus (\exp^{SO(3)}(\mathbf{w}^R_n)\mathbf{R}^u_n, \mathbf{I}_3, \mathbf{p}^u_n + \mathbf{w}^p_n, \mathbf{0}_{3\times1}).\end{aligned} \tag{11}$$

The predicted state, $\mathbf{X_{n+1|n}}$, by propagation is computed by

$$\mathbf{X}_{n+1|n} = (\mathbf{R}^r_{n|n}\mathbf{R}^u_n, \mathbf{R}^f_{n|n}, \mathbf{R}^r_{n|n}\mathbf{p}^u_n + \mathbf{p}^r_{n|n}, \mathbf{p}^f_{n|n}). \tag{12}$$

where $\mathbf{X_{n|n}} = (\mathbf{R^r_{n|n}}, \mathbf{R^f_{n|n}}, \mathbf{p^r_{n|n}}, \mathbf{p^f_{n|n}})$ is the updated state at time step n. The estimated error $\xi_{\mathbf{n+1|n}}$ by propagation is

$$\boldsymbol{\xi}_{n+1|n} \doteq \log(\mathbf{X}_{n+1} \ominus \mathbf{X}_{n+1|n})$$

Applying first order taylor series expansion

$$\xi_{n+1|n} \approx \xi_{n|n} + J(\xi_{n+1|n})\Delta X$$

where $J(\xi_{n+1|n})$ is the Jacobian of error propagation model. Substituting this in error propagation model we get

$$\boldsymbol{\xi}_{n+1|n} \approx \mathbf{F}_n\boldsymbol{\xi}_{n|n} + \mathbf{G}_n\mathbf{w}_n, \tag{13}$$

where $\xi_{n|n} \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_n)$ is the estimation error for $X_{n|n}$, $\mathbf{w_n} = ((\mathbf{w^R_n})^\mathbf{T}, (\mathbf{w^p_n})^\mathbf{T})$ is the odometry noise, the coefficient matrix associated with error propagation model $\mathbf{F_n}$ and the coefficient matrix associated with noise in the system $\mathbf{G_n}$ in RI-EKF are

$$\begin{aligned}\mathbf{F}_n =& \mathbf{I}_{6+6\,K},\\ \mathbf{G}_n =& \begin{bmatrix} \mathbf{R^r_{n|n}} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ (\mathbf{p}^r_{n|n} + \mathbf{R}^r_{n|n}\mathbf{p}^u_n)^\wedge\mathbf{R}^r_{n|n} & \mathbf{R}^r_{n|n} \\ (\mathbf{p}^f_{n|n})^\wedge\mathbf{R}^r_{n|n} & \mathbf{0}_{3\times3} \end{bmatrix}.\end{aligned} \tag{14}$$

The equation representing how the covariance of the state evolves from time n to n+1 based on linearized error propagation:

$$\mathbf{P}_{n+1|n} = \mathbf{F}_n\mathbf{P}_n\mathbf{F}_n^\top + \mathbf{G}_n\boldsymbol{\Sigma}_n\mathbf{G}_n^\top. \tag{15}$$

where $\mathbf{F}_n\mathbf{P}_n\mathbf{F}_n^\top$ represents the covariance propagated through system dynamics and $\mathbf{G}_n\boldsymbol{\Sigma}_n\mathbf{G}_n^\top$ represents the covariance introduced by noise in the system.

Basically the equation (15) is the result of linearizing the error propagation model and considering the covariance introduced by the system dynamics and the noise. To linearize the error propagation model we use first order taylor series expansion. It basically involves finding the jacobians F_n and G_n that capture the partial derivatives of the error propagation model w.r.t current state and noise.

3) Update: The goal of the update steps is to use the innovation to correct the state estimate improve the accuracy of predictions. We introduce a new minus operator $\times$ for the observation. The innovation $\mathbf{y}$ is defined as

$$\mathbf{y} = \begin{bmatrix} \mathbf{y^R} \\ \mathbf{y}^p \end{bmatrix} = \mathbf{Z} \times [(\hat{\mathbf{R}}^r)^\top\hat{\mathbf{R}}^f, (\hat{\mathbf{R}}^r)^\top(\hat{\mathbf{p}}^f - \hat{\mathbf{p}}^r)]$$

Here,

$$\begin{aligned}\mathbf{Z} &= (\mathbf{R}_z, \mathbf{p}_z)\\ &= (\exp^{SO(3)}(\mathbf{v}^R_{n+1})(\mathbf{R}^r_{n+1})^\top\mathbf{R}^f_{n+1},\\ &\quad (\mathbf{R}^r_{n+1})^\top(\mathbf{p}^f_{n+1} - \mathbf{p}^r_{n+1}) + \mathbf{v}^p_{n+1})\end{aligned}$$

On substituting, the innovation $\mathbf{y}$ is defined as

$$\mathbf{y} \doteq \begin{bmatrix} \log^{SO(3)}(\exp^{SO(3)}((\mathbf{v}^R)^\wedge)(\mathbf{R}^r)^\top\mathbf{R}^f(\hat{\mathbf{R}}^f)^\top\hat{\mathbf{R}}^r) \\ (\mathbf{R}^r)^\top(\mathbf{p}^f - \mathbf{p}^r) + \mathbf{v}^p - [(\hat{\mathbf{R}}^r)^\top(\hat{\mathbf{p}}^f - \hat{\mathbf{p}}^r)] \end{bmatrix} \tag{16}$$

We then perform linearization of rotation innovation $\mathbf{y}_R$ using exponential map $\mathbf{SO}(3)$. The linearization involves expressing the exponential map in terms of error vector ξ^{R^r} and ξ^{R^f}, and omitting second order small quantities

$$\begin{aligned}\mathbf{I}_3 + (\mathbf{y}^R)^\wedge \approx& \exp^{SO(3)}((\mathbf{y}^R)^\wedge)\\ =& \exp^{SO(3)}((\mathbf{v}^R)^\wedge)(\mathbf{R}^r)^\top\mathbf{R}^f(\hat{\mathbf{R}}^f)^\top\hat{\mathbf{R}}^r\\ \approx& \mathbf{I}_3 - ((\hat{\mathbf{R}}^r)^\top\boldsymbol{\xi}^{R^r})^\wedge\\ &+ ((\hat{\mathbf{R}}^r)^\top\boldsymbol{\xi}^{R^f})^\wedge + (\mathbf{v}^R)^\wedge.\end{aligned} \tag{17}$$

After omitting second-order terms, the linearization of $\mathbf{y}^R$ is given by

$$\mathbf{y}^R = -(\hat{\mathbf{R}}^r)^\top\boldsymbol{\xi}^{R^r} + (\hat{\mathbf{R}}^r)^\top\boldsymbol{\xi}^{R^f} + \mathbf{v}^R. \tag{18}$$

Similarly, to perform the linearization of $\mathbf{y^p}$ we know

that

$$\mathbf{y}^p = (\mathbf{R}^r)^\top(\mathbf{p}^f - \mathbf{p}^r) + \mathbf{v}^p - [(\hat{\mathbf{R}}^r)^\top(\hat{\mathbf{p}}^f - \hat{\mathbf{p}}^r)]$$
$$= \frac{\partial}{\partial \mathbf{p}_r}\left[(\mathbf{R}^r)^\top(\mathbf{p}^f - \mathbf{p}^r) + (\hat{\mathbf{R}}^r)^\top(\hat{\mathbf{p}}^f - \hat{\mathbf{p}}^r)\right]\delta\mathbf{p}_r$$

Omitting the second-order terms the linearizarion of $\mathbf{y^p}$ is given by

$$\mathbf{y^p} = -(\mathbf{R^r})^\top\delta\mathbf{p_r} + (\hat{\mathbf{R}}^\mathbf{r})^\top\delta\mathbf{p_r} + \mathbf{v^p}$$

Using the linearized results of $\mathbf{y^R}$ and $\mathbf{y^p}$ we can define innovation at time step $n+1$ as

$$\mathbf{y}_{n+1} = \mathbf{H}_{n+1}\boldsymbol{\xi}_{n+1|n} + \mathbf{v}_{n+1}, \tag{19}$$

where

$$\mathbf{H}_{n+1} = \begin{bmatrix} \mathbf{H}_{\mathbf{n+1}}^{\mathbf{R},\mathbf{R^r}} & \mathbf{H}_{n+1}^{R,R^f} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{H}_{n+1}^{p,p^r} & \mathbf{H}_{n+1}^{p,p^f} \end{bmatrix},$$
$$\mathbf{H}_{n+1}^{R,R^r} = \mathbf{H}_{n+1}^{p,p^r} = -(\mathbf{R}_{n+1|n}^r)^\top,$$
$$\mathbf{H}_{n+1}^{R,R^f} = \mathbf{H}_{n+1}^{p,p^f} = (\mathbf{R}_{n+1|n}^r)^\top, \tag{20}$$

and $v_{n+1} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Omega}_{n+1})$ is the observation noise. Then the state is updated by

$$\mathbf{X}_{n+1|n+1} = \exp^{\mathcal{G}}(\boldsymbol{\xi}_{n+1|n+1}) \oplus \mathbf{X}_{n+1|n}, \tag{21}$$

where $\xi_{n+1|n+1} = \mathbf{K}_{n+1}\mathbf{y}_{n+1}$ is the update state error vector, and

$$\mathbf{K}_{n+1} = \mathbf{P}_{n+1|n}\mathbf{H}_{n+1}^\top(\mathbf{H}_{n+1}\mathbf{P}_{n+1|n}\mathbf{H}_{n+1}^\top + \mathbf{\Omega}_{n+1})^{-1}.$$

Its covariance is updated as

$$\mathbf{P}_{n+1} = (\mathbf{I} - \mathbf{K}_{n+1}\mathbf{H}_{n+1})\mathbf{P}_{n+1|n}. \tag{22}$$

Algorithm 1 gives the entire process of RI-EKF SLAM with object features.

B. New Feature Initialization

Object feature initialization is a crucial step in simultaneous localization and mapping (SLAM) systems, where a robot explores an unknown environment, simultaneously estimating its own pose and creating a map of the environment. Object feature initialization specifically refers to the process of incorporating information about new observed features (landmarks or objects) into the SLAM system.When the robot observes a new object feature with the observation

Algorithm 1 RI-EKF for Object-based SLAM

$\mathbf{F}_n$, $\mathbf{G}_n$, $\mathbf{H}_n + 1$ are given in (14) and (20)
Input: $\mathbf{X}_{n|n}, \mathbf{P}_n, \mathbf{U}_n, \mathbf{Z}_{n+1}$
Output: $\mathbf{X}_{n+1|n+1}, \mathbf{P}_{n+1}$
Propagation:

$$\mathbf{X}_{n+1|n} \leftarrow \{(\mathbf{X}_{n|n}, \mathbf{U}_n, \mathbf{0})$$
$$\mathbf{P}_{n+1|n} \leftarrow \mathbf{F}_n\mathbf{P}_n\mathbf{F}_n^T + \mathbf{G}_{n+1}\mathbf{\Sigma}_n\mathbf{G}_{n+1}^T$$

Update:

$$\mathbf{K}_{n+1} \leftarrow \mathbf{P}_{n+1|n}\mathbf{H}_{n+1}^T\left(\mathbf{H}_{n+1}\mathbf{P}_{n+1|n}\mathbf{H}_{n+1}^T + \mathbf{\Omega}_{n+1}\right)^{-1}$$
$$\mathbf{y}_{n+1} \leftarrow \mathbf{Z}_{n+1} \times \mathbf{h}_{n+1}(\mathbf{X}_{n+1|n}, \mathbf{0})$$
$$\mathbf{X}_{n+1|n+1} \leftarrow \exp_G(\mathbf{K}_{n+1}\mathbf{y}_{n+1}) \oplus \mathbf{X}_{n+1|n}$$
$$\mathbf{P}_{n+1} \leftarrow (\mathbf{I} - \mathbf{K}_{n+1}\mathbf{H}_{n+1})\mathbf{P}_{n+1|n}$$

$\mathbf{Z} = (\mathbf{R}^z, \mathbf{p}^z) \in SO(3) \times R^3$ the goal is to augment the estimated state X and covariance matrix P
Given a state error $\xi \in g \cong R^{6+6K}$ is of the form

$$\xi = \begin{bmatrix}\xi_R \\ \xi_p\end{bmatrix} = log^{\mathcal{G}}(\mathbf{X} \oplus \hat{\mathbf{X}})$$

Here, ξ_R represents the rotation part and ξ_p represents the translational part

$$\xi^R = ((\xi^{R^r})^\top, (\xi^{R^f})^\top)^\top$$
$$\xi^p = ((\xi^{p^r})^\top, (\xi^{p^f})^\top)^\top$$

Suppose the error ξ follows a Gaussian distribution $N(\mathbf{0}, \mathbf{P})$, where P is the covariance matrix. The new object feature, denoted by $(\hat{R}^{f_{new}}, \hat{p}^{f_{new}})$, related to the observation $\mathbf{Z} = (\mathbf{R}_z, \mathbf{R}_z) \in SO(3) \times \mathbf{R}^3$ is given by:

$$\hat{R}_{f_{new}} = \hat{R}^r R^z$$
$$\hat{p}_{f_{new}} = \hat{p}^r + \hat{R}^r p^z$$

This corresponds to transforming the observed rotation R_z and translational p_z into the reference frame of the robot.The augmented state error vector is denoted as $\xi = (\delta_{R_r}, \delta_{p_r}, \delta_{R_{f_{new}}}, \delta_{p_{f_{new}}})$. The augmented covariance $\mathbf{P}_{aug}$ is given as

$$\mathbf{P}_{\text{aug}} = \begin{bmatrix} \mathbf{P}^{R,R} & \mathbf{P}^{R,R}\mathbf{M}_1^\top & \mathbf{P}^{R,p} & \mathbf{P}^{R,p}\mathbf{M}_2^\top \\ \mathbf{M}_1\mathbf{P}^{R,R} & \mathbf{P}_f^{R,R} & \mathbf{M}_1\mathbf{P}^{R,p} & \mathbf{P}_f^{R,p} \\ \mathbf{P}^{p,R} & \mathbf{P}^{p,R}\mathbf{M}_1^\top & \mathbf{P}^{p,p} & \mathbf{P}^{p,p}\mathbf{M}_2^\top \\ \mathbf{M}_2\mathbf{P}^{p,R} & (\mathbf{P}_f^{R,p})^\top & \mathbf{M}_2\mathbf{P}^{p,p} & \mathbf{P}_f^{p,p} \end{bmatrix}$$

where

$$\begin{aligned}\mathbf{M}_1 =&[\mathbf{I}_3\ \mathbf{0}_{3,3\,K}],\\ \mathbf{M}_2 =&[\mathbf{I}_3\ \mathbf{0}_{3,3\,K}],\\ \mathbf{P}_f^{R,R} =&\mathbf{M}_1\mathbf{P}^{R,R}\mathbf{M}_1^\top + \hat{\mathbf{R}}^r\mathbf{\Omega}^{R,R}(\hat{\mathbf{R}}^r)^\top,\\ \mathbf{P}_f^{R,p} =&\mathbf{M}_1\mathbf{P}^{R,p}\mathbf{M}_2^\top + \hat{\mathbf{R}}^r\mathbf{\Omega}^{R,p}(\hat{\mathbf{R}}^r)^\top,\\ \mathbf{P}_f^{p,p} =&\mathbf{M}_2\mathbf{P}^{p,p}\mathbf{M}_2^\top + \hat{\mathbf{R}}^r\mathbf{\Omega}^{p,p}(\hat{\mathbf{R}}^r)^\top. \end{aligned} \tag{23}$$

The whole process to augment the state is summarized in **Algorithm 2**.

Algorithm 2 New Feature Initialization

Input: The state and its covariance before augmentation:

$$\hat{X} = \begin{bmatrix}\hat{R}^r & \hat{R}^f & \hat{p}^r & \hat{p}^f\end{bmatrix}, P = \begin{bmatrix}P^{R,R} & P^{R,p}\\ P^{p,R} & P^{p,p}\end{bmatrix}$$

The observation of the new feature: $Z = (R_z, p_z) \in \mathrm{SO}(3) \times R^3$

The covariance of observation noise:

$$\Omega = \begin{bmatrix}\Omega^{R,R} & \Omega^{R,p}\\ \Omega^{p,R} & \Omega^{p,p}\end{bmatrix}$$

Output: The augmented state and its covariance:

$$\hat{X}_{\text{aug}} = \begin{bmatrix}\hat{R}^r & \hat{R}^f & \hat{R}^r R^z & \hat{p}^r & \hat{p}^f & \hat{p}^r + \hat{R}^r p^z\end{bmatrix}$$

P_{aug} is obtained by (25)

IV. OBSERVABILITY ANALYSIS

Drawing from prior investigations into inconsistency it has been established that the inconsistency in EKF SLAM primarily arises due to the violation of observability constraints. A robust EKF SLAM estimator must adhere to specific observability constraints, ensuring that the unobservable subspace of the estimator's system model aligns with that of the actual system (an ideal scenario where Jacobians are computed at the true state). This section presents a demonstration that our RIEKF for object-based SLAM is inherently capable of preserving observability constraints automatically. In contrast, the standard EKF for object-based SLAM, briefly introduced here, lacks the capability to sustain observability constraints. These findings contribute to the improved consistency exhibited by our algorithms in subsequent experiments.

Definition 1: The unobservable subspace $\hat{\mathcal{N}}$, derived from the state estimates, is defined as the null space of the corresponding observability matrix $\hat{\mathcal{O}}$, where

$$\hat{\mathbf{O}} = \begin{bmatrix}\hat{\mathbf{H}}_0\\ \hat{\mathbf{H}}_1\hat{\mathbf{F}}_{0,0}\\ \vdots\\ \hat{\mathbf{H}}_{n+1}\hat{\mathbf{F}}_{n,0}\end{bmatrix}, \tag{24}$$

with $\hat{\mathbf{H}}_i$ being the Jacobian matrix for the i-th step observation model evaluated at the state estimate $\hat{\mathbf{X}}_i$, and $\hat{\mathbf{F}}_{i,0} = \hat{\mathbf{F}}_i\hat{\mathbf{F}}_{i-1}....\hat{\mathbf{F}}_0$, where $\hat{\mathbf{F}}_j$ is the Jacobian matrix for the j-th step propagation model of estimator evaluated at the state $\hat{\mathbf{X}}_j$ for $j = 0, \ldots, i$. If the models are linearized at the ground truth, the unobservable subspace based on the true states is represented by $\breve{\mathcal{N}}$, and the corresponding observability matrix is denoted as $\breve{\mathcal{O}}$.

A. Observability Analysis for RI-EKF

Theorem 1: The unobservable subspace $\hat{\mathcal{N}}^{RI}$ and $\breve{\mathcal{N}}^{RI}$ are same for RI-EKF, where

$$\hat{\mathcal{N}}^{RI} = \breve{\mathcal{N}}^{RI} = \underset{col.}{\text{span}}\begin{bmatrix}\mathbf{I_3} & \mathbf{0}_{3\times3}\\ \mathbf{I}_3 & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times3} & \mathbf{I}_3\\ \mathbf{0}_{3\times3} & \mathbf{I}_3\end{bmatrix}, \tag{25}$$

and $dim(\hat{\mathcal{N}}^{RI}) = dim(\breve{\mathcal{N}}^{RI}) = 6$

Proof: The ideal case system model for RI-EKF can be described as follows:

$$\begin{aligned}\breve{\mathcal{O}}^{RI} &= \begin{bmatrix}\breve{\mathbf{H}}_0^{RI}\\ \breve{\mathbf{H}}_1^{RI}\breve{\mathbf{F}}_{0,0}^{RI}\\ \vdots\\ \breve{\mathbf{H}}_{n+1}^{RI}\breve{\mathbf{F}}_{n,0}^{RI}\end{bmatrix}\\ &= \begin{bmatrix}\breve{\mathbf{H}}_0^{R,R^r} & \breve{\mathbf{H}}_0^{R,R^f} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \breve{\mathbf{H}}_0^{p,p^r} & \breve{\mathbf{H}}_0^{p,p^f}\\ \vdots & \vdots & \vdots & \vdots\\ \breve{\mathbf{H}}_{n+1}^{R,R^r} & \breve{\mathbf{H}}_{n+R^f}^{R} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \breve{\mathbf{H}}_{n+1}^{p,p^r} & \breve{\mathbf{H}}_{n+1}^{p,p^f}\end{bmatrix},\end{aligned}$$

where

$$\begin{aligned}&\breve{\mathbf{H}}_0^{RI} = \begin{bmatrix}-(\mathbf{R}_0^r)^T & (\mathbf{R}_0^r)^T & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -(\mathbf{R}_0^r)^T & (\mathbf{R}_0^r)^T\end{bmatrix},\\ &\breve{\mathbf{H}}_{i+1}^{R,R^r} = \breve{\mathbf{H}}_{i+1}^{p,p^r} = -\left(\mathbf{R}_{i+1}^r\right)^T, \forall i = 0, \cdots, n,\\ &\breve{\mathbf{H}}_{i+1}^{R,R^f} = \breve{\mathbf{H}}_{i+1}^{p,p^f} = \left(\mathbf{R}_{i+1}^r\right)^T, \forall i = 0, \cdots, n,\end{aligned}$$

$\breve{\mathbf{H}}_{i+1}^{RI}$ corresponds to the Jacobian matrix for the observation model at the (i + 1)-th step, computed based on the true state $\breve{\mathbf{X}}_{i+1}^{RI}$, and $\breve{\mathbf{F}}_{i,0}^{RI} = \breve{\mathbf{F}}_{i}^{RI}\breve{\mathbf{F}}_{i-1}^{RI}....\breve{\mathbf{F}}_{0}^{RI}$ where $\breve{\mathbf{F}}_{0}^{RI}$ is an identity matrix. Additionally, $\breve{\mathbf{F}}_{j}^{RI}$ represents the Jacobian matrix for the j-th step propagation model, evaluated at the true state $\breve{\mathbf{X}}_{j}^{RI}, j = 0, ..., i$. Employing the linearization method introduced in the preceding description of the RI-EKF approach, we can derive the unobservable subspace $\breve{\mathcal{N}}^{RI}$ through this procedure

$$\breve{\mathcal{N}}^{RI} = \operatorname{span}_{\text{col.}} \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3,3} \\ \mathbf{I}_3 & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & \mathbf{I}_3 \\ \mathbf{0}_{3,3} & \mathbf{I}_3 \end{bmatrix}$$

The observability matrix derived from the RI-EKF $\hat{\mathcal{O}}$ is

$$\begin{aligned} \hat{\mathcal{O}}^{RI} &= \begin{bmatrix} \hat{\mathbf{H}}_0^{RI} \\ \hat{\mathbf{H}}_1^{RI}\hat{\mathbf{F}}_{0,0}^{RI} \\ \vdots \\ \hat{\mathbf{H}}_{n+1}^{RI}\hat{\mathbf{F}}_{n,0}^{RI} \end{bmatrix} \\ &= \begin{bmatrix} \hat{\mathbf{H}}_0^{R,R^r} & \hat{\mathbf{H}}_0^{R,R^f} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \hat{\mathbf{H}}_0^{p,p^r} & \hat{\mathbf{H}}_0^{p,p^f} \\ \vdots & \vdots & \vdots & \vdots \\ \hat{\mathbf{H}}_{n+1}^{R,R^r} & \hat{\mathbf{H}}_{n+1}^{R,R^f} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \hat{\mathbf{H}}_{n+1}^{p,p^r} & \hat{\mathbf{H}}_{n+1}^{p,p^f} \end{bmatrix}, \end{aligned}$$

where

$$\begin{aligned} \hat{\mathbf{H}}_{i+1}^{R,R^r} &= \hat{\mathbf{H}}_{i+1}^{p,p^r} = -\left(\hat{\mathbf{R}}_{i+1|i}^{r}\right)^T, \forall i = 0, \cdots, n, \\ \hat{\mathbf{H}}_{i+1}^{R,R^f} &= \hat{\mathbf{H}}_{i+1}^{p,p^f} = \left(\hat{\mathbf{R}}_{i+1|i}^{r}\right)^T, \forall i = 0, \cdots, n, \end{aligned}$$

$\hat{\mathbf{H}}_{i+1}^{RI}$ is the Jacobian matrix for the $(i + 1) - th$ step observation model evaluated at the estimated state $\hat{X_{i+1|i}^{RI}}$.

Therefore, the theorem proves the fact that RI-EKF is able to automatically maintain the correct unobservable subspace, which inturn significantly improves the overall consistency.

B. Standard EKF for Object based SLAM

The conventional EKF (Std-EKF) designed for object-based SLAM is referred to as the SO(3)-EKF, with its state space being isomorphic to $(SO(3))^{K+1} \times (R^3)^{K+1}$. Assuming a state vector with only one feature, an error state $\eta \in (SO(3))^{K+1} \times (R^3)^{K+1}$ for $K = 1$ can be obtained within the standard EKF

$$\begin{aligned} \boldsymbol{\eta} =& (\boldsymbol{\eta}^{R^r}, \boldsymbol{\eta}^{R^f}, \boldsymbol{\eta}^{p^r}, \boldsymbol{\eta}^{p^f}) \\ =& (\log^{SO(3)}(\mathbf{R}^r(\hat{\mathbf{R}}^r)^\top), \log^{SO(3)}(\mathbf{R}^f(\hat{\mathbf{R}}^f)^\top), \\ & \mathbf{p}^r - \hat{\mathbf{p}}^r, \mathbf{p}^f - \hat{\mathbf{p}}^f), \end{aligned} \tag{26}$$

In this context, (R^r, p^r) and $(\hat{R^r}, \hat{p^r})$ epresent the actual and estimated poses of the robot, while (R^f, p^f) and $(\hat{R^f}, \hat{p^f})$ correspond to the true and estimated features of the object. Using this approach to linearization, the Jacobians of the considered system are derived

$$\begin{aligned} \mathbf{F}_n^{Std} &= \begin{bmatrix} \mathbf{I_3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ -(\mathbf{R}_{n|n}^r\mathbf{p}^u)^\wedge & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 \end{bmatrix}, \\ \mathbf{G}_n^{Std} &= \begin{bmatrix} \mathbf{R_{n|n}^r} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{R}_{n|n}^r \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}, \\ \mathbf{H}_{n+1}^{Std} &= \begin{bmatrix} \mathbf{H}_{\mathbf{n+1}}^{\mathbf{R},\mathbf{R^r}} & \mathbf{H}_{n+1}^{R,R^f} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{H}_{n+1}^{p,R^r} & \mathbf{0}_{3\times3} & \mathbf{H}_{n+1}^{p,p^r} & \mathbf{H}_{n+1}^{p,p^f} \end{bmatrix}, \end{aligned} \tag{27}$$

where

$$\begin{aligned} \mathbf{H}_{n+1}^{R,R^r} &= \mathbf{H}_{n+1}^{p,p^r} = -(\mathbf{R}_{n+1|n}^r)^\top, \\ \mathbf{H}_{n+1}^{R,R^f} &= \mathbf{H}_{n+1}^{p,p^f} = (\mathbf{R}_{n+1|n}^r)^\top, \\ \mathbf{H}_{n+1}^{p,R^r} &= (\mathbf{R}_{n+1|n}^r)^\top(\mathbf{p}_{n+1|n}^f - \mathbf{p}_{n+1|n}^r)^\wedge, \end{aligned}$$

F_n and G_n represent the Jacobians of the process model for the state error $\eta_{n|n}$ and the odometry noise $\mathbf{w}_n$, respectively. Additionally, $\mathbf{H}_{n+1}^{Std}$, evaluated at $\mathbf{X}_{n+1|n}$, signifies the Jacobian of the innovation $\mathbf{y}$ as defined in equation (16).

C. Observability Analysis for Standard EKF

Theorem 2: In the case of Std-EKF, the unobservable subspace $\hat{\mathcal{N}}^{Std}$, is a proper subspace of $\breve{\mathcal{N}}^{Std}$, where

$$\hat{\mathcal{N}}^{Std} = \operatorname*{span}_{col.}[\mathbf{0}_{3\times3}, \mathbf{0}_{3\times3}, \mathbf{I}_3, \mathbf{I}_3]^\top, \tag{28}$$

and

$$\breve{\mathcal{N}}^{Std} = \operatorname*{span}_{col.} \begin{bmatrix} \mathbf{0_{3\times3}} & \mathbf{I}_3 \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 \\ \mathbf{I}_3 & (\mathbf{p}_0^r)^\wedge \\ \mathbf{I}_3 & (\mathbf{p}^f)^\wedge \end{bmatrix}, \tag{29}$$

p_0^r and p^f are the true positions of initial robot and object feature respectively. The dimensions of $\hat{\mathcal{N}}^{Std}$ and $\breve{\mathcal{N}}^{Std}$ are 3 and 6 respectively.

Proof: The observability matrix of standard EKF based on state estimates is constructed as

$$\hat{\mathcal{O}}^{Std} = \begin{bmatrix} \hat{\mathbf{H}}_0^{Std} \\ \hat{\mathbf{H}}_1^{Std}\hat{\mathbf{F}}_{0,0}^{Std} \\ \vdots \\ \hat{\mathbf{H}}_{n+1}^{Std}\hat{\mathbf{F}}_{n,0}^{Std} \end{bmatrix}$$

where $\hat{\mathbf{H}}_{i+1}^{Std}$ is the jacobian matrix evaluated at the prediction $\mathbf{X}_{i+1|i}$ for the $(i+1)-th$ step observation model. Ideally, the jacobians are evaluated at the true state $\breve{X}_i$ which gives us

$$\begin{aligned} \mathbf{p}_{i+1|i+1}^r &= \mathbf{p}_{i+1|i}^r \\ \mathbf{p}_{i+1|i}^f &= \mathbf{p}^f, \forall i. \end{aligned}$$

And thus $\hat{\mathbf{H}}_i^{Std}\hat{\mathbf{F}}_{i-1,0}^{Std}$ in $\breve{\mathcal{O}}^{Std}$ in $\breve{\mathcal{O}}_{Std}$ for ideal case becomes

$$\hat{\mathbf{H}}_i^{Std}\hat{\mathbf{F}}_{i-1,0}^{Std} = \begin{bmatrix} -\left(\mathbf{R}_i^r\right)^T & \left(\mathbf{R}_i^r\right)^T & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ -\left(\mathbf{R}_i^r\right)^T\left(\mathbf{p}^f - \mathbf{p}_0^r\right)^\wedge & \mathbf{0}_{3\times3} & -\left(\mathbf{R}_i^r\right)^T & \left(\mathbf{R}_i^r\right)^T \end{bmatrix}$$

Then unobservable subspace based on the ground truth is obtained as

$$\breve{\mathcal{N}}^{Std} = \operatorname{span}_{\text{col.}} \begin{bmatrix} \mathbf{0}_{3,3} & \mathbf{I}_3 \\ \mathbf{0}_{3,3} & \mathbf{I}_3 \\ \mathbf{I}_3 & \left(\mathbf{p}_0^r\right)^\wedge \\ \mathbf{I}_3 & \left(\mathbf{p}^f\right)^\wedge \end{bmatrix}$$

Therefore, the dimension of unobservable subspace of system is 6 . The same conclusion is for the case of multiple features.

However, in practice, generally

$$\begin{aligned} \mathbf{p}_{i+1|i+1}^r &\neq \mathbf{p}_{i+1|i}^r, \\ \mathbf{p}_{i+1|i}^f &\neq \mathbf{p}_{j+1|j}^f, \forall i \neq j \end{aligned}$$

and thus the unobservable subspace of standard EKF evaluated by the estimates becomes

$$\hat{\mathcal{N}}^{Std} = \underset{\text{col.}}{\operatorname{span}} \left[\mathbf{0}_{3\times3}, \mathbf{0}_{3\times3}, \mathbf{I}_3, \mathbf{I}_3\right]^T$$

whose dimension is 3.

As per Theorem 2, because of this flawed linearization in object-based SLAM, the standard EKF fails to uphold the accurate observability constraints. As a result, the standard EKF erroneously incorporates misleading information into the estimation process, resulting in an overly confident estimate, thereby introducing inconsistency.

V. SIMULATIONS

This section compares our proposed RI-EKF to the standard EKF (Std-EKF) and an ideal EKF (Ideal-EKF), which is a variant of the Std-EKF where Jacobians are evaluated at the ground truth. Note that the Ideal-EKF is impractical to apply in real scenarios because the ground truth is unavailable. It is only used to illustrate the impact of observability constraints on inconsistency. To evaluate the consistency of an estimation method, we use the Normalized Estimation Error Squared (NEES) indicator.

$$\text{NEES} = \frac{1}{m \times d} \sum_{i=1}^{m} \mathbf{e}_i^\top \mathbf{P}_i^{-1} \mathbf{e}_i, \tag{30}$$

where m is the number of samples, and e_i is a d dimensional error sample vector, which is estimated to be a zero mean Gaussian with a d×d covariance matrix $\mathbf{P}_i$. For a large number of samples, the Normalized Estimation Error Squared (NEES) indicator should approximately equal 1 if the estimator is consistent. In addition to NEES, the root mean squared error (RMSE) is used to evaluate the accuracy of each estimator.

To compute NEES for our proposed RI-EKF, the estimated covariance corresponds to the nonlinear error, rather than the standard error in the vector space. However, for a fair comparison, we still use the standard error to compute the RMSE of RI-EKF.

A. Settings

Figure 1 shows the simulation environment and robot trajectory. The environment contains 6 object features, represented by red-green-blue arrows. The robot moves in a circle 25 times (total length: 200 m) with a constant linear velocity of 0.1 m/s and angular velocity of π/40 rad/s. The robot can measure the relative poses of object features within a range of 0.5 to 2 m. Rotation and position are measured in radians and meters, respectively. The covariance matrices of the odometry and observation noise in (4) and (5) are set to Σ_n = diag$(0.1^2, 0.1^2, 0.1^2, 0.1^2, 0.1^2, 0.1^2) and \Omega_n$ = diag$(0.1^2, 0.1^2, 0.1^2, 0.1^2, 0.1^2, 0.1^2), respectively.$

B. Results and Analysis

Here is a paraphrase of the given text in IEEE format:

We conducted 50 Monte-Carlo simulations, i.e., m = 50. The NEES and RMSE results are shown in Figure 2 and Table 1. Figure 2 shows the RMSE and NEES results for robot pose and feature pose every 50 steps. Table 1 lists the average RMSE and NEES for rotation

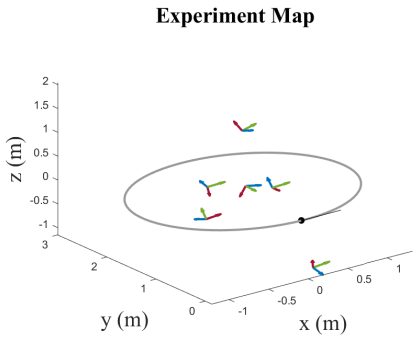

Fig. 1. Simulation environment: 6 object features (the poses are shown as red-green-blue arrows) in a 3D environment, robot moves on the circle (the yellow arrow shows the initial heading of the robot).

	Std-EKF	RI-EKF	Ideal-EKF
	RMSE		
Robot Rotation (rad)	.0239	.0230	.0199
Robot Position (m)	.0039	.0038	.0035
Feature Rotation (rad)	.0072	.0066	.0046
Feature Position (m)	.0007	.0007	.0006
	NEES		
Robot Rotation	1.215	0.973	0.903
Robot Position	1.155	1.090	0.959
Robot Pose	1.216	1.016	0.973
Feature Rotation	1.531	0.953	0.735
Feature Position	1.135	1.113	1.058
Feature Pose	1.306	1.007	0.909

Fig. 2. SIMULATION RESULTS OF RI-EKF, STD-EKF AND IDEAL-EKF

and position error (in rad and m, respectively) in the last time step.

The results show that in this experiment, RI-EKF and Ideal-EKF outperform Std-EKF in terms of both accuracy and consistency. The inconsistency of Std-EKF mainly comes from the inaccuracy of linearization points. These linearization points break the observability constraints, leading Std-EKF to obtain spurious information from the unobservable subspace. As a result, its estimation is more inaccurate and its estimated covariance is smaller than the actual uncertainty, and it becomes more and more inconsistent over time.

In contrast, RI-EKF remains consistent (NEES 1) for a longer duration, behaving like Ideal-EKF. The analysis of RI-EKF in Section IV indicates that RI-EKF naturally maintains the observability constraints, as does Ideal-EKF, while the Jacobians are evaluated at the latest estimate. These factors make the results of RI-EKF more reliable than those of Std-EKF.

VI. REAL DATA EXPERIMENTS

This section evaluates our algorithm on the YCB-Video real-world dataset and compares it to Std-EKF, DVO, ORB-SLAM3, and pose graph optimization (PGO) to demonstrate its effectiveness. All of these algorithms are fed RGB-D images from YCB-Video. Four sequences (0019, 0036, 0041, and 0049) with relatively long trajectories are selected for the experiments (Figure 3).

Unlike simulations, real-world data may contain numerous outliers. The point cloud matches in Sequence 0019 are highly accurate, but in other sequences, some object observations are highly inaccurate, as seen in the last three images of the bottom row of Figure 3. To make the algorithms robust, outliers must be detected and removed. Additionally, these data sequences lack odometry information. We assume constant velocity for these data sequences, which were obtained at low camera motion speeds, to apply our algorithms.

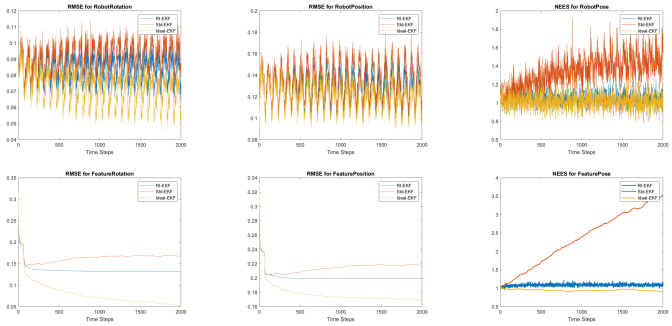

Fig. 3. Accuracy (RMSE) and consistency (NEES) of Std-EKF, RI-EKF, and Ideal-EKF in simulations.

A. Dataset and Object Detection

The YCB-Video dataset contains 21 YCB objects with a variety of textures. It includes 92 RGB-D videos used for training and testing object detection, of which 80 videos are used for training and 2,949 keyframes from the remaining 12 videos are used for testing. Additionally, 80,000 synthetic images are released for training. The dataset contains many scenes of partially occluded stacked objects, as shown in Figure 3.

B. Observation of Object Features

To obtain object feature observations, we use the REDE algorithm from [30], an end-to-end object pose estimator that takes RGB-D data as input. On the YCB dataset, the pose estimator achieves a 98.9% recall under the average ADD metric [31]. The outputs of REDE are used directly as observations in (5).

C. Constant Velocity Assumption

Because the data is collected from a slow-moving camera, we assume that the camera is moving at a constant velocity. We use a simple method to obtain the odometry: we assume that the angular velocity is zero

with noise, and we compute the expected linear velocity by averaging the linear velocities calculated from the previous 6 estimations. The variances are computed based on these 6 estimations.

D. Outlier Removal

Suppose there is an object feature observed in the $n+1^{\text{th}}$ step, and $Z \in SO(3) \times R^3$ is its observation. Before updating the state vector, we will first compute $y = Z \times h(X_{n+1|n}, 0)$.

$$\mathbf{y} \approx \mathbf{H}_{n+1}\boldsymbol{\xi}_{n+1|n} + \mathbf{v}_{n+1} \tag{31}$$

The Jacobian of the observation function evaluated at $\mathbf{X}_{n+1|n}$ is denoted by $\mathbf{H}_{n+1}$, and the noise is represented by $\mathbf{v}_{n+1} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Omega}_{n+1})$. If the estimation is accurate, then $\mathbf{y}_n$ should follow a zero-mean Gaussian distribution with covariance $\boldsymbol{\Omega}_{n+1}$.

$$\mathbf{P}_y = \mathbf{H}_{n+1}\mathbf{P}_{n+1|n}(\mathbf{H}_{n+1})^\top + \boldsymbol{\Omega}_{n+1}. \tag{32}$$

Hence, If all of the elements of the vector y are within 3 standard deviations of their mean, then...

$$|\mathbf{y}(k)| < 3\sqrt{\mathbf{P}_y(k,k)}, \;\; k = 1, \cdots 6,$$

If all of the elements of the observation vector $\mathbf{y}$ are within 3 standard deviations of their mean, then we will use $\mathbf{y}$ to update the state vector. Otherwise, we will discard $\mathbf{y}$ as an outlier. This outlier removal method is similar to the Mahalanobis distance method. EKF methods that use this outlier removal method are called Robust EKF methods.

E. Results and Analysis

The measurement noise levels for sequences 0019, 0036, and 0041 are around 0.001 meters and 0.04 radians for position and rotation, respectively. The measurement noise level for sequence 0049 is much larger, at 0.04 meters and 0.6 radians for position and rotation, respectively. In order to remove outliers, we typically set the measurement uncertainties in the estimators slightly lower than the actual noise levels. In these experiments, we set the standard deviations of the measurement noises in both the Std-EKF and RI-EKF to be around 0.001 meters and 0.03 radians for all four data sequences.

The measurement noise levels for sequences 0019, 0036, and 0041 are around 0.001 meters and 0.04 radians for position and rotation, respectively. The measurement noise level for sequence 0049 is much larger, at 0.04 meters and 0.6 radians for position and rotation, respectively. In order to remove outliers, we typically set the measurement uncertainties in the estimators slightly lower than the actual noise levels. In these experiments, we set the standard deviations of the measurement noises in both the Std-EKF and RI-EKF to be around 0.001 meters and 0.03 radians for all four data sequences.

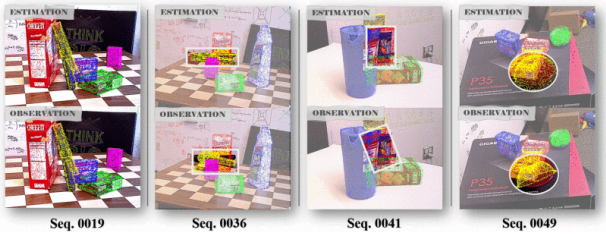

Fig. 4. Sample images from the four sequences in YCB-Video Dataset used in the experiments. For each column, the lower row shows an image with feature observations while the upper row shows the final estimate from our proposed method.

VII. Comparison with other methods

We compared our algorithm to DVO, ORB-SLAM3, and Std-EKF on four different data sequences. We also tested PGO, a common back-end approach in object SLAM systems, using the same information as in Robust RI-EKF. As expected, PGO performed the best on all four sequences. Robust RI-EKF performed the closest to PGO in terms of accuracy. However, EKF methods are more efficient than PGO when the number of objects is much smaller than the number of time steps. This is because the computational complexity of EKF methods is $O(T \cdot N^3)$, while the computational complexity of PGO is $O(T^3 + T^2 \cdot N)$, where T is the number of time steps and N is the number of objects.

Although DVO and ORB-SLAM3 use all of the features in the trajectory, they only exploit low-level point features. In contrast, object features have broader perspective loop closures, longer feature tracks, and more intrinsic constraints information between low-level features. Additionally, the constant velocity model provides extra information. Therefore, DVO and ORB-SLAM3 are not fairly comparable to Robust RI-EKF. These factors could explain why Robust RI-EKF, a filter-based method, outperforms these two optimization-based methods.

The results on Sequence 0049 show that robust methods, such as Robust RI-EKF and Robust Std-EKF, perform much better than non-robust methods, such as RI-EKF and Std-EKF, on inaccurate data sequences. However, we also noticed that some of the RMSE

values for Robust Std-EKF are higher than those for Std-EKF, especially on Sequence 0036. This is likely due to the fact that Std-EKF is inconsistent and can mistakenly delete correct data in Robust Std-EKF.

Figure 4 shows the errors in the robot pose estimates and the 3σ bounds for each component for Robust RI-EKF and Robust Std-EKF on Sequence 0036. Robust RI-EKF consistently produces accurate estimates, while Robust Std-EKF underestimates the uncertainty of the state in the latter half of the sequence. This inconsistency of Robust Std-EKF can lead to larger errors. In contrast, the estimates produced by Robust RI-EKF are more reliable, making our outlier removal method more effective.

Figure 5 shows the root mean squared error (RMSE) of the Pitcher Base object in Sequence 0041 every 100 steps. The RMSE of each object in Sequence 0041 is shown in the full version of this paper. The results show that Robust RI-EKF also produces more accurate estimates of the object poses than Robust Std-EKF.

Overall, the real-world experiments show that Robust RI-EKF can generate accurate estimates. In Figure 3, the top row of images shows the objects estimated by Robust RI-EKF, which are significantly better than the corresponding observations shown in the bottom row of images for sequences 0036, 0041, and 0049.

VIII. CONCLUSION

In this paper, we propose a new EKF algorithm called RI-EKF for object-based SLAM. In RI-EKF, object features are represented by 3D poses and are estimated together with the robot pose. We theoretically prove that our RI-EKF algorithm maintains the correct observability properties, which is not the case with standard EKF algorithms used for object-based SLAM. Simulation and real-world experiments show that the proposed RI-EKF algorithm performs well.

Like all EKF-based methods, RI-EKF assumes that the noise in the system is Gaussian white noise and only considers first-order errors. Therefore, RI-EKF may have limitations in problems with non-Gaussian noise or large noise levels.

This paper focuses on the back-end of SLAM, where we assume that the objects observed are within a given database so that they can be easily detected and matched from the front-end of SLAM. In the future, we plan to investigate the more challenging problem of object-based SLAM, where the objects in the environment are more general and may not belong to a known database.

References are important to the reader; therefore, each citation must be complete and correct. If at all possible, references should be commonly available publications.

REFERENCES

[1] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, et al., "SLAM++: Simultaneous localisation and mapping at the level of objects," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2013, pp. 1352–1359

[2] Xiang Gao and Tao Zhang and Yi Liu and Qinrui Yan, 14 Lectures on Visual SLAM: From Theory to Practice, Publishing House of Electronics Industry,2017

[3] S. Yang and S. Scherer, "CubeSLAM: Monocular 3-D Object SLAM," in IEEE Transactions on Robotics, vol. 35, no. 4, pp. 925-938, Aug. 2019.

[4] G. P. Huang, A.I. Mourikis, and S.I. Roumeliotis. "A first-estimates Jacobian EKF for improving SLAM consistency," In 11th International Symposium on Experimental Robotics (ISER'08), Athens, Greece July 2008.

[5] G. P. Huang, A. I. Mourikis, S. I. Roumeliotis, "Observability-based rules for designing consistent EKF SLAM estimators," The International Journal of Robotics Research, 2010, 29(5): 502-528.

[6] R. Mahony and T. Hamel, "A geometric nonlinear observer for simultaneous localisation and mapping," 2017 IEEE 56th Annual Conference on Decision and Control (CDC), 2017, pp. 2408-2415.