

First Estimate Jacobian EKF for Multi-robot SLAM

Ranxi Liu, Yang Song, Shoudong Huang

Robotics Institute, University of Technology Sydney, Australia
ranxi.liu-1@uts.edu.au

Abstract

Addressing inconsistency issues, i.e. underestimation of uncertainty, is crucial for the performance of Extended Kalman Filter (EKF) based Simultaneous Localization and Mapping (SLAM). By using the first estimate Jacobians, this paper designs a consistent EKF for the point feature-based multi-robot SLAM. First, the standard EKF (Std-EKF) for the considered problems is presented. Then, through the observability analysis, we prove that Std-EKF has an observable subspace of dimension higher than the underlying system, leading to the inconsistency issue. Accordingly, we propose the first estimate Jacobian EKF (FEJ-EKF), which shares the same dimension of observable subspace with the underlying system, alleviating the inconsistency issue. Finally, the effectiveness of the proposed method is validated by simulations and a practical dataset. By making the MATLAB code for this research available online¹, we hope to facilitate collaboration and allow others to build upon and improve the methodology.

1 Introduction

Simultaneous localization and mapping (SLAM) algorithms typically use sensor data to estimate the pose of a robot and the environment structure in real time. As a single robot is not enough to complete more challenging, complex, and major tasks, such as large-scale emergency search and rescue missions, multi-robot collaboration has become a trend for performing SLAM in a large environment that requires high precision and speed. For instance, an online multi-robot SLAM system for 3D LiDARs is researched by [Dubé *et al.*, 2017] to permit rapid

exploration and higher redundancy than a single robot. [Thrun and Liu, 2005] presents a tree-based multi-robot SLAM algorithm and conducts an experiment based on a real-world dataset. An optimal map-merging algorithm for different robots is presented in [Zhou and Roumeliotis, 2006].

There is nonlinearity in robotic systems, which means that the relationship between the state change and the input of the robot during movement, perception, and control is not a simple linear relationship. The nonlinearity in robotic systems originates from several specific attributes related to dynamics, sensor characteristics, environmental interactions, control inputs, and inter-robot interactions in multi-robot systems. In the context of robot motion and observation models, filter consistency generally refers to the ability of a filter (e.g., Kalman Filter, Extended Kalman Filter, Particle Filter) to accurately represent the true uncertainty of the system state. Filter consistency ensures that the estimation remains reliable over time without gradually diverging from the actual state, which is essential for accurate and robust state estimation in mobile robots. This concept is critical in tasks such as localization, navigation, and mapping.

For the state estimations of nonlinear robotic systems, the EKF algorithms are one of the most popular approaches for decades. However, the conventional EKF algorithm (Std-EKF) typically suffers from the inconsistency issue, where they generate overconfident covariance about actual uncertainty, ultimately leading to poor performance. The inconsistency issue of EKF SLAM is first discovered by [Julier and Uhlmann, 2001]. By further investigating the fundamental reason for the issue [Huang and Dissanayake, 2007; Huang *et al.*, 2009; Huang *et al.*, 2010], it is concluded that a consistent EKF should share the same dimension of observability with the underlying system. To address this issue of Std-EKF, various consistent EKF algorithms such as first estimate Jacobian (FEJ)-EKF [Huang *et al.*, 2009], observability-constrained (OC)-EKF [Huang *et al.*, 2010], and right invariant error (RI)-EKF [Zhang *et al.*, 2017] have been de-

¹The MATLAB code is available at <https://github.com/Ranxisama/First-Estimate-Jacobian-EKF-for-Multi-robot-SLAM.git>

veloped for the point feature-based (single-robot) SLAM.

In this paper, we consider the point feature-based multi-robot SLAM problems. Through observability analysis, the corresponding Std-EKF still has the inconsistency issue for the considered problems. To address such issue, we apply the first estimate Jacobians to the multi-robot EKF SLAM. Finally, the experiments illustrate the effectiveness of the proposed algorithm. The multi-robot pose estimation algorithm proposed in this paper have great potential in various fields. Improved pose estimation accuracy can enable better task coordination, precise navigation, and efficient obstacle avoidance, which are essential for automation, search and rescue, and complex collaborative tasks in dynamic environments. In smart cities and autonomous driving fleets, these algorithms will facilitate synchronized navigation and reduce congestion, while in agriculture and industry, they can achieve efficient area coverage and precise object handling. In addition, optimized pose estimation promotes innovations in sensor fusion and uncertainty management, laying the foundation for more autonomous and intelligent multi-robot systems that can adapt to changing conditions and complete complex tasks. These developments will accelerate the deployment of multi-robot systems and improve productivity, safety, and efficiency in different industries.

2 Methodology

2.1 Problem Definition

In the considered 2D multi-robot SLAM problems, multiple robots navigate in a 2D environment by observing point features. At each step, robots will obtain the relative positions of surrounding point features as their observations. Without loss of generality, in Figure 1, we just assume that there are only two robots 1R and 2R , and two features f_1 and f_2 in the environment for simplifications. Both robots move one step. And these features have been observed by both 1R and 2R .

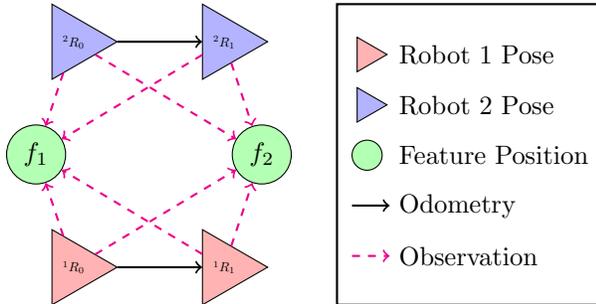


Figure 1: A simplified 2D multi-robot SLAM scenario: iR_k represents robot i at time-step k , and f_j represents feature j .

At step 0, the pose of 1R_0 can be regarded to be anchored in the global frame, whereas the pose of 2R_0 is unknown. Thus, the Gauss-Newton method is implemented to obtain the optimal estimates for the pose of 2R_0 and the positions of shared features with the corresponding covariance matrix.

2.2 Multi-robot EKF SLAM

In our considered problems, the state vector comprises the poses of two robots and the positions of two features in the global frame, which at time step k is given by

$$\mathbf{x}_k = [\mathbf{x}_{r_k}^T, \mathbf{x}_{f_k}^T]^T, \quad (1)$$

where $\mathbf{x}_{r_k} = [{}^1\mathbf{x}_{r_k}^T, {}^2\mathbf{x}_{r_k}^T]^T$, $\mathbf{x}_{f_k} = [{}^1\mathbf{x}_{f_k}^T, {}^2\mathbf{x}_{f_k}^T]^T$. ${}^i\mathbf{x}_{r_k} = [{}^i\mathbf{x}_{p_k}^T, {}^i\phi_k]^T$ ($i = 1, 2$) denotes the true robot pose (position and orientation) of R_i , and ${}^j\mathbf{x}_{f_k}$ ($j = 1, 2$) is the true feature position of f_j . An EKF is based on a discrete-time process and measurement model, evolving recursively in two steps: propagation and update. In the following, we add two steps (step-0 optimization and feature initialization) to clearly describe the presented algorithm, starting with the Std-EKF.

In the following formulations, the components of a state vector without any script denote true values, with “ $\hat{\cdot}$ ” as estimated values. While $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$ is the error in this estimate. The subscript $i|j$ refers to the estimate of a quantity at time-step i after all measurements up to time-step j have been processed. $\mathbf{0}_{m \times n}$ denotes $m \times n$ zero matrix, while \mathbf{I}_n is the $n \times n$ identity matrix.

EKF Propagation

In this step, we process the robot’s odometry measurements between two consecutive time steps, obtain an estimate of the pose transition and then employ it in the EKF to propagate the robot state estimate. In this case, the underlying nonlinear propagation equations for R_i (from time-step k to $k+1$) are given by

$$\begin{aligned} {}^i\hat{\mathbf{x}}_{r_{k+1}|k} &= \begin{bmatrix} {}^i\hat{\mathbf{x}}_{p_{k+1}|k} \\ {}^i\hat{\phi}_{k+1|k} \end{bmatrix} = \begin{bmatrix} {}^i\hat{\mathbf{x}}_{p_{k|k}} + \mathbf{C}({}^i\hat{\phi}_{k|k}) {}^i\mathbf{u}_k \\ {}^i\hat{\phi}_{k|k} + {}^i\mathbf{v}_k \end{bmatrix} \\ &= \mathbf{f}_i({}^i\hat{\mathbf{x}}_{r_{k|k}}, {}^i\mathbf{u}_k, {}^i\mathbf{v}_k), \end{aligned} \quad (2)$$

$${}^j\hat{\mathbf{x}}_{f_{k+1}|k} = {}^j\hat{\mathbf{x}}_{f_{k|k}}, \quad (3)$$

where $\mathbf{C}(\phi)$ is the 2×2 rotation matrix of the angle ϕ . ${}^i\mathbf{u}_k$ is the odometry-based position estimate, which is corrupted by process noise ${}^i\mathbf{v}_k$, ${}^i\mathbf{v}_k \sim \mathcal{N}(0, {}^i\mathbf{Q})$. ${}^i\mathbf{v}_k$ is the odometry-based heading estimate, which is corrupted by process noise ${}^i\nu_k$, ${}^i\nu_k \sim \mathcal{N}(0, {}^i\mathbf{\Omega})$. The linearized error-state propagation equation for the whole

state is given by:

$$\begin{aligned}\tilde{\mathbf{x}}_{k+1|k} &= \begin{bmatrix} \Phi_{r_k} & \mathbf{0}_{6 \times 4} \\ \mathbf{0}_{4 \times 6} & \mathbf{I}_4 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_{r_k|k} \\ \tilde{\mathbf{x}}_{f_k|k} \\ \tilde{\mathbf{x}}_{f_k|k} \end{bmatrix} + \begin{bmatrix} \mathbf{G}_{r_k} \\ \mathbf{0}_{4 \times 6} \end{bmatrix} \begin{bmatrix} \mathbf{w}_k \\ \mathbf{w}_k \end{bmatrix} \\ &= \Phi_k \tilde{\mathbf{x}}_{k|k} + \mathbf{G}_k \mathbf{w}_k,\end{aligned}\quad (4)$$

where ${}^i \mathbf{w}_k = \begin{bmatrix} {}^i \nu_k \\ {}^i \nu_k \end{bmatrix}$ is the process noise, $\Phi_{r_k} = \begin{bmatrix} {}^1 \Phi_{r_k} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & {}^2 \Phi_{r_k} \end{bmatrix}$ and $\mathbf{G}_{r_k} = \begin{bmatrix} {}^1 \mathbf{G}_{r_k} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & {}^2 \mathbf{G}_{r_k} \end{bmatrix}$. ${}^i \Phi_{r_k}$ and ${}^i \mathbf{G}_{r_k}$ are obtained from the state propagation equations ((2) and (3)):

$$\begin{aligned}{}^i \Phi_{r_k} &= \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\mathbf{C}({}^i \hat{\phi}_{k|k}) {}^i \mathbf{u}_k \\ \mathbf{0}_{1 \times 2} & 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}({}^i \hat{\mathbf{x}}_{p_{k+1|k}} - {}^i \hat{\mathbf{x}}_{p_{k|k}}) \\ \mathbf{0}_{1 \times 2} & 1 \end{bmatrix},\end{aligned}\quad (5)$$

$${}^i \mathbf{G}_{r_k} = \begin{bmatrix} \mathbf{C}({}^i \hat{\phi}_{k|k}) & \mathbf{0}_{2 \times 1} \\ \mathbf{0}_{1 \times 2} & 1 \end{bmatrix},\quad (6)$$

where $\mathbf{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$.

EKF Update

In the considered SLAM problems, the robots observe the relative position of the point feature with respect to the robots. Thus, the observation equation at step k is given by:

$$\begin{aligned}{}_j \mathbf{z}_{k+1} &= \mathbf{C}({}^i \hat{\phi}_{k+1})^T ({}_j \mathbf{x}_{f_{k+1}} - {}^i \mathbf{x}_{p_{k+1}}) + {}^j \mathbf{e}_{k+1} \\ &= \mathbf{h}_{i,j}({}^i \mathbf{x}_{k+1}) + {}^j \mathbf{e}_{k+1},\end{aligned}\quad (7)$$

where ${}_j \mathbf{z}_{k+1}$ is the observation of f_j from R_i and ${}^j \mathbf{e}_{k+1}$ is the corresponding observation noise, ${}^j \mathbf{e}_{k+1} \sim \mathcal{N}(0, {}^j \mathbf{R})$. The linearized measurement-error equation is given by:

$$\begin{aligned}\tilde{\mathbf{z}}_{k+1} &\approx \begin{bmatrix} \mathbf{H}_{r_{k+1}} & \mathbf{H}_{f_{k+1}} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_{r_{k+1|k}} \\ \tilde{\mathbf{x}}_{f_{k+1|k}} \end{bmatrix} + \mathbf{e}_{k+1} \\ &= \mathbf{H}_{k+1} \tilde{\mathbf{x}}_{k+1|k} + \mathbf{e}_{k+1},\end{aligned}\quad (8)$$

where

$$\mathbf{H}_{r_{k+1}} = \begin{bmatrix} {}^1 \mathbf{H}_{r_{k+1}} & \mathbf{0}_{2 \times 3} \\ {}^2 \mathbf{H}_{r_{k+1}} & \mathbf{0}_{2 \times 3} \\ \mathbf{0}_{2 \times 3} & {}^1 \mathbf{H}_{r_{k+1}} \\ \mathbf{0}_{2 \times 3} & {}^2 \mathbf{H}_{r_{k+1}} \end{bmatrix}, \mathbf{H}_{f_{k+1}} = \begin{bmatrix} {}^1 \mathbf{H}_{f_{k+1}} & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & {}^1 \mathbf{H}_{f_{k+1}} \\ {}^2 \mathbf{H}_{f_{k+1}} & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & {}^2 \mathbf{H}_{f_{k+1}} \end{bmatrix},\quad (9)$$

are the Jacobians of $\mathbf{h} = (\mathbf{h}_{1,1}, \mathbf{h}_{1,2}, \mathbf{h}_{2,1}, \mathbf{h}_{2,2})^T$ with respect to the robot poses and the shared feature positions, respectively, evaluated at the state estimate $\hat{\mathbf{x}}_{k+1|k}$,

$${}^i \mathbf{H}_{f_{k+1}} = \mathbf{C}^T({}^i \hat{\phi}_{k+1|k}),\quad (10)$$

$${}^i \mathbf{H}_{r_{k+1}} = {}^i \mathbf{H}_{f_{k+1}} \begin{bmatrix} -\mathbf{I}_2 & -\mathbf{J}({}_j \hat{\mathbf{x}}_{f_{k+1|k}} - {}^i \hat{\mathbf{x}}_{p_{k+1|k}}) \end{bmatrix}.\quad (11)$$

Initialization

As described in Section 2.1, we have an unknown pose estimate of ${}^2 R$, and both robots observe f_1 and f_2 . In other words, we have the unknown state vector $\hat{\mathbf{x}}_s = \begin{bmatrix} {}^2 \hat{\mathbf{x}}_{r_0,1}^T & {}^1 \hat{\mathbf{x}}_{f_0,2}^T & {}^2 \hat{\mathbf{x}}_{f_0}^T \end{bmatrix}^T$. To find the optimal $\hat{\mathbf{x}}_s$, this can be solved as a nonlinear least squares problem [Huang *et al.*, 2024] by implementing Gauss-Newton method at step 0 to minimize the **Cost Function**:

$$\sum_{i=1}^2 \sum_{j=1}^2 \|{}_j \mathbf{z}_s - \mathbf{C}({}^i \hat{\phi}_0)^T ({}_j \hat{\mathbf{x}}_{f_0} - {}^i \hat{\mathbf{x}}_{p_0})\|_{{}^j \mathbf{R}},\quad (12)$$

where ${}_j \mathbf{z}_s$ is the observation of f_j from R_i , ${}^i \mathbf{R}$ is the covariance matrix of observation noise ${}^j \mathbf{e}_s$ for ${}_j \mathbf{z}_s$, ${}^j \mathbf{e}_s \sim \mathcal{N}(0, {}^j \mathbf{R})$. In the simulation, the initial values of ${}^2 \hat{\mathbf{x}}_{r_0}$ and ${}_j \hat{\mathbf{x}}_f$ are random.

When the Gauss-Newton method converges, we get the optimized state estimate $\hat{\mathbf{x}}_s$ and its covariance matrix \mathbf{P}_s . The shared state at step 0 is $\hat{\mathbf{x}}_0 = \begin{bmatrix} {}^1 \mathbf{x}_{r_0}^T & \hat{\mathbf{x}}_s^T \end{bmatrix}^T$, with ${}^1 \mathbf{x}_{r_0} = [0, 0, 0]^T$. And the covariance matrix of $\hat{\mathbf{x}}_0$ is $\mathbf{P}_0 = \mathbf{Diag}(\mathbf{0}_{3 \times 3}, \mathbf{P}_s)$.

3 Observability Analysis

For single-robot SLAM, the observability property has been analysed in [Huang *et al.*, 2010] that the nonlinear SLAM system for a single robot is unobservable, with three unobservable degrees of freedom (DoF) corresponding to the global position and orientation of the initial robot pose. Later, the observability properties of the EKF linearized error-state system are examined and analysed in [Huang *et al.*, 2009]. Further inconsistency analysis of unobservable subspace is conducted in [Huang *et al.*, 2010]. Similarly, we can show that our two robots' nonlinear SLAM system has three unobservable DoF. In this section, we analyze the observability properties of three EKF algorithms, ideal-EKF, Std-EKF and FEJ-EKF.

Since the EKF model is time-varying ((4) and (8)), we employ the *local observability matrix* to perform the observability analysis. In this paper, the local observability matrix for the time interval between time steps k and $k+m$ is:

$$\begin{aligned}\mathbf{M} &= \begin{bmatrix} \mathbf{H}_k \\ \mathbf{H}_{k+1} \Phi_k \\ \vdots \\ \mathbf{H}_{k+m} \Phi_{k+m-1} \cdots \Phi_k \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{H}_{r_k} & \mathbf{H}_{f_k} \\ \mathbf{H}_{r_{k+1}} \Phi_{r_k} & \mathbf{H}_{f_{k+1}} \\ \vdots & \vdots \\ \mathbf{H}_{r_{k+m}} \Phi_{r_{k+m-1}} \cdots \Phi_{r_k} & \mathbf{H}_{f_{k+m}} \end{bmatrix}.\end{aligned}\quad (13)$$

In the following sections, the matrices (such as \mathbf{M}) without any script are Std-EKF based, with “ \sim ” are ideal-EKF based, with “ $\bar{\cdot}$ ” are FEJ-EKF based.

3.1 Standard EKF

We now study the observability properties of the Std-EKF for the considered multi-robot SLAM in (13), in which the Jacobians are evaluated at the estimated state. Starting by noting that,

$$\begin{aligned} & {}^i\Phi_{r_{k+t-1}} \ {}^i\Phi_{r_{k+t-2}} \ \cdots \ {}^i\Phi_{r_k} = \\ & \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}({}^i\hat{\mathbf{x}}_{p_{k+t}|k+t-1} - {}^i\hat{\mathbf{x}}_{p_{k+t}|k}) - \sum_{d=k+1}^{k+t-1} \Delta^i \mathbf{x}_{p_d} \\ \mathbf{0}_{1 \times 2} & 1 \end{bmatrix}, \end{aligned} \quad (14)$$

where $\Delta^i \mathbf{x}_{p_t} = {}^i\hat{\mathbf{x}}_{p_t|t} - {}^i\hat{\mathbf{x}}_{p_t|t-1}$ is the correction in the robot position due to the update at time-step t when $t = 1, \dots, m$. Therefore,

$${}^i\mathbf{H}_{r_{k+t}} \ {}^i\Phi_{r_{k+t-1}} \ \cdots \ {}^i\Phi_{r_k} = {}^i\mathbf{H}_{f_{k+t}} \times [-\mathbf{I}_2 \ {}^i\Psi_{k+t}], \quad (15)$$

$${}^i\Psi_{k+t} = -\mathbf{J}({}_j\hat{\mathbf{x}}_{f_{k+t}|k+t-1} - {}^i\hat{\mathbf{x}}_{p_{k+t}|k}) + \sum_{d=k+1}^{k+t-1} \Delta^i \mathbf{x}_{p_d}, \quad (16)$$

where $\sum_{d=k+1}^{k+t-1} \Delta^i \mathbf{x}_{p_d} = 0$ when $t = 0, 1$. Using this result,

$$\mathbf{M} = \underbrace{\text{Diag}(\mathbf{H}'_{f_k}, \dots, \mathbf{H}'_{f_{k+m}})}_{\mathbf{D}} \quad (17)$$

$$\times \underbrace{\begin{bmatrix} -\mathbf{I}_2 & {}^1\Psi_k & \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 1} & \mathbf{I}_2 & \mathbf{0}_{2 \times 2} \\ -\mathbf{I}_2 & {}^2\Psi_k & \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 1} & \mathbf{0}_{2 \times 2} & \mathbf{I}_2 \\ \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 1} & -\mathbf{I}_2 & {}^2\Psi_k & \mathbf{I}_2 & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 1} & -\mathbf{I}_2 & {}^2\Psi_k & \mathbf{0}_{2 \times 2} & \mathbf{I}_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\mathbf{I}_2 & {}^1\Psi_{k+m} & \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 1} & \mathbf{I}_2 & \mathbf{0}_{2 \times 2} \\ -\mathbf{I}_2 & {}^2\Psi_{k+m} & \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 1} & \mathbf{0}_{2 \times 2} & \mathbf{I}_2 \\ \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 1} & -\mathbf{I}_2 & {}^2\Psi_{k+m} & \mathbf{I}_2 & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 1} & -\mathbf{I}_2 & {}^2\Psi_{k+m} & \mathbf{0}_{2 \times 2} & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{N}}$$

where

$$\mathbf{H}'_{f_{k+t}} = \text{Diag}({}^1\mathbf{H}_{f_{k+t}}, {}^1\mathbf{H}_{f_{k+t}}, {}^2\mathbf{H}_{f_{k+t}}, {}^2\mathbf{H}_{f_{k+t}}). \quad (18)$$

Lemma 1. *The rank of the observability matrix of the system model of the Std-EKF, \mathbf{M} , is equal to 8.*

Proof. We start by analysing the rank of \mathbf{N} defined in (17). Since the estimates of any given state variable at different time instants are generally different in the standard EKF SLAM, $\forall i, j, t, k, l$ ($t \neq l$):

$${}^i\hat{\mathbf{x}}_{p_{k+t}|k+t-1} \neq {}^i\hat{\mathbf{x}}_{p_{k+t}|k+t}, \quad (19)$$

$${}_j\hat{\mathbf{x}}_{f_{k+t}|k+t-1} \neq {}_j\hat{\mathbf{x}}_{f_{k+t}|k+t-1}. \quad (20)$$

With (16) and (17), it implies that ${}^i\Psi_{k+t}$ in the third and the sixth columns of \mathbf{N} are vectors with unequal elements, thus $\text{rank}(\mathbf{N}) = 8$. We can easily prove that \mathbf{D} is invertible, thus $\text{rank}(\mathbf{M}) = \text{rank}(\mathbf{N}) = 8$. The dimension of the unobservable subspace for the Std-EKF linearized model is 2. \square

3.2 Ideal-EKF

In the linearizations in ideal-EKF, the state estimates are replaced with the ground truth when evaluating the Jacobians of the filter. In other words, the inequality signs in (19) and (20) become equal signs. We start by noting that ((10) and (16)):

$${}^i\check{\mathbf{H}}_{f_k} = \mathbf{C}^T({}^i\phi_k), \quad (21)$$

$${}^i_j\check{\Psi}_{k+t} = -\mathbf{J}({}_j\mathbf{x}_{f_k} - {}^i\mathbf{x}_{p_k}). \quad (22)$$

Similarly, we can get $\check{\mathbf{M}} = \check{\mathbf{D}} \times \check{\mathbf{N}}$ in the ideal-EKF ((17) and (18)). However, given i and j , as ${}^i_j\check{\Psi}_{k+t}$ in the third and the sixth columns of $\check{\mathbf{N}}$ are vectors with equal elements, $\text{rank}(\check{\mathbf{M}}) = 7$. Thus, the dimension of the unobservable subspace for the underlying system should be 3. As the unobservable subspace of the Std-EKF linearized model has 2 dimensions, it implies that the Std-EKF gains “spurious information” along the unobservable directions of the underlying nonlinear SLAM system, which is the main cause of the inconsistency between the physical (nonlinear) and the linearized model [Huang *et al.*, 2009].

3.3 First Estimate Jacobian EKF

In the linearized FEJ-EKF SLAM, the state estimates are replaced with the first estimates when evaluating the Jacobians of the filter. We start by noting that ((10) and (16)):

$${}^i\bar{\mathbf{H}}_{f_k} = \mathbf{C}^T({}^i\hat{\phi}_{k|k-1}), \quad (23)$$

$${}^i_j\bar{\Psi}_{k+t} = -\mathbf{J}({}_j\hat{\mathbf{x}}_{f_\tau|_\tau} - {}^i\hat{\mathbf{x}}_{p_{k+t}|k-1}), \quad (24)$$

where ${}^i\hat{\mathbf{x}}_{p_{k+t}|k-1}$ is the position estimate of R_i prior to updating, and ${}_j\hat{\mathbf{x}}_{f_\tau|_\tau}$ is the first estimated position of f_j at time-step τ . Then, we can get $\bar{\mathbf{M}} = \bar{\mathbf{D}} \times \bar{\mathbf{N}}$ in the FEJ-EKF. Similar to the ideal-EKF, given i and j , as ${}^i_j\bar{\Psi}_{k+t}$ in the third and the sixth columns of $\bar{\mathbf{N}}$ are vectors with equal elements, $\text{rank}(\bar{\mathbf{M}}) = 7$. Thus, the dimension of the unobservable subspace for the linearized FEJ-EKF SLAM system is 3, which is the same with the underlying system. This indicates that, unlike Std-EKF, the proposed FEJ-EKF satisfies the observability constraint for the considered problems.

4 Experiments

In this section, a series of Monte Carlo simulation tests and an experiment on the real-world Victoria Park

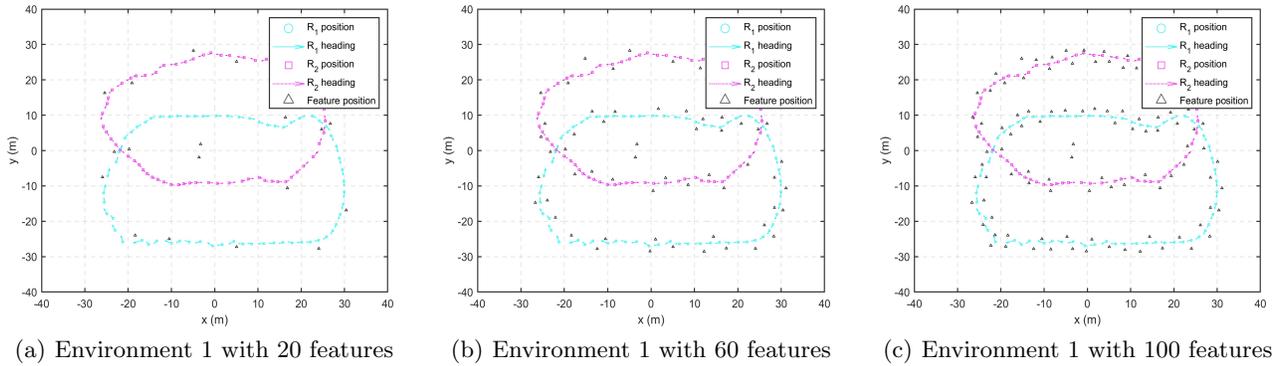


Figure 2: Simulation environment 1 with 20, 60 and 100 features

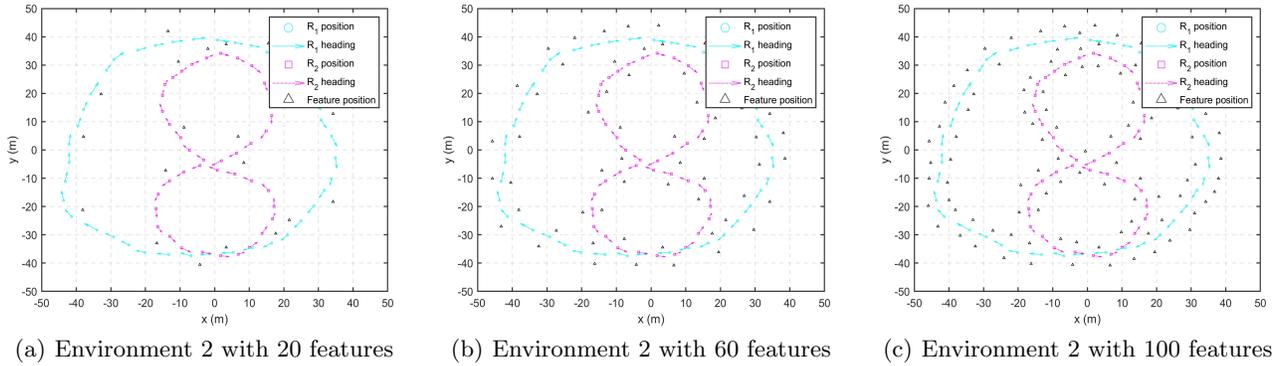


Figure 3: Simulation environment 2 with 20, 60 and 100 features

dataset [Guivant and Nebot, 2001] are conducted to demonstrate the effectiveness of the proposed FEJ-EKF for the considered multi-robot SLAM problems.

We use the metrics of the root mean square error (RMSE) and the average normalized (state) estimation error squared (NEES) to evaluate the accuracy and consistency, respectively. Specifically, the NEES indicator is defined by [Song *et al.*, 2022; Zhang *et al.*, 2023]

$$\text{NEES} = \frac{1}{m \times d} \sum_{i=1}^m \mathbf{e}_i^T \mathbf{P}_i^{-1} \mathbf{e}_i, \quad (25)$$

where m is the number of Monte Carlo runs, and \mathbf{e}_i is a d dimensional error sample vector with estimated covariance matrix \mathbf{P}_i . NEES should approximately equal to 1 for large m , if the estimator is consistent [Song *et al.*, 2022].

4.1 Simulations

To compare the accuracy and consistency of Std-EKF, ideal-EKF and the proposed FEJ-EKF, we conduct 100 Monte Carlo experiments on 2 major categories of simulated environments for the considered problems. Each category of simulated environment has 3 feature configurations. During each run, 1R and 2R execute 5 loops on

the trajectories with 20 (Figure 2(a) and 3(a)), 60 (Figure 2(b) and 3(b)) and 100 features (Figure 2(c) and 3(c)), respectively, shown in Figure 2 and Figure 3. To provide enough observation information for the pose initialization of 2R , 1R and 2R are set up to observe at least 2 shared features at step 0 in the simulation.

In Figure 2 and Figure 3, the circles (\circ) and the arrows are respectively the true positions and headings of 1R , the squares (\square) and the dash-dotted arrows are the true positions and headings of 2R , the triangles (\triangle) are the true positions of the features.

The standard deviations of the noise level for the simulations in the environment 1 and 2 are respectively set as: (i) $({}^i\sigma_v, {}^i\sigma_\nu) = (0.3\text{m}, 2^\circ)$ and $({}^i\sigma_v, {}^i\sigma_\nu) = (0.2\text{m}, 3^\circ)$ for the position and heading process noises of the i th robot; (ii) ${}^1\sigma_e = 0.5\text{m}$ and ${}^2\sigma_e = 0.6\text{m}$ for the observation noises; 1R and 2R observe features' position within their sensing range of 15 m.

The RMSE and NEES for each step of the corresponding Monte Carlo simulations with 20, 60 and 100 features for environment 1 are presented in Figure 4. Specifically, for the simulation environment with (i) 20 features, Figure 4(a) - 4(c) describe the pose NEES, position RMSE and heading RMSE of 1R , respectively, with Figure 4(d)

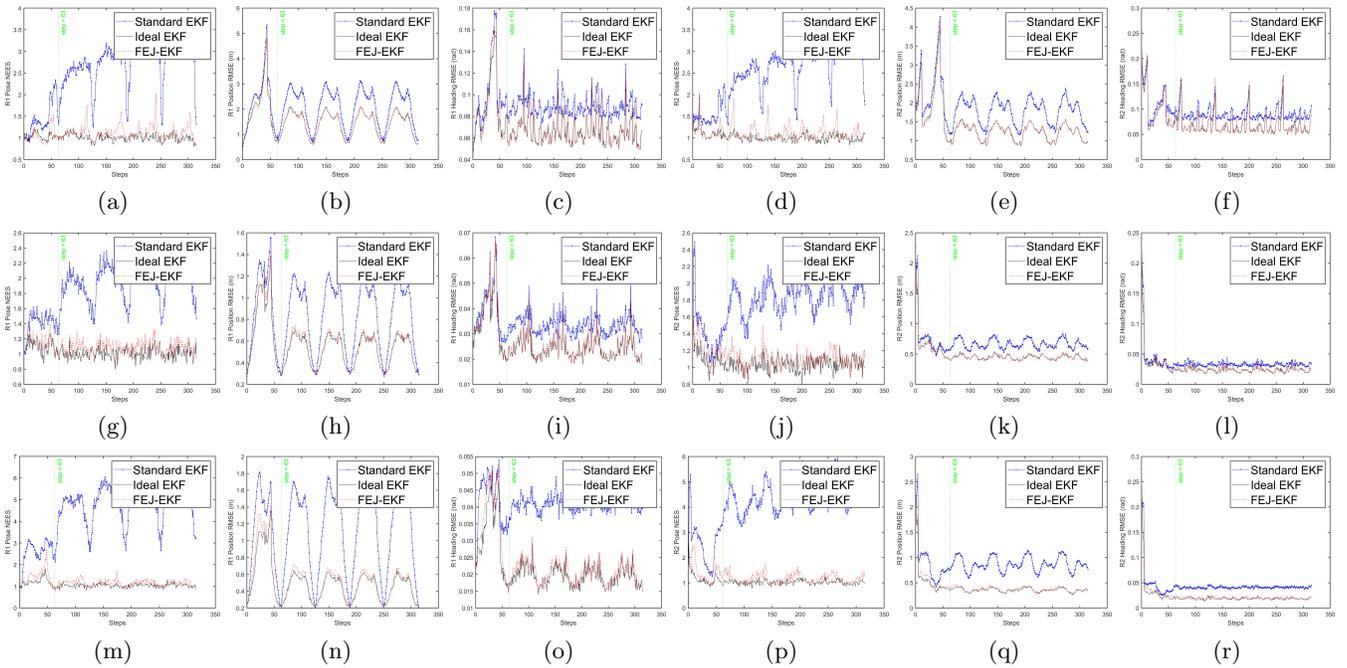


Figure 4: The RMSE and NEES results from the Monte Carlo simulations for environment 1 with different numbers of features: the upper six figures are for the 20 features, the middle six figures are for the 60 features, the lower six figures are for the 100 features

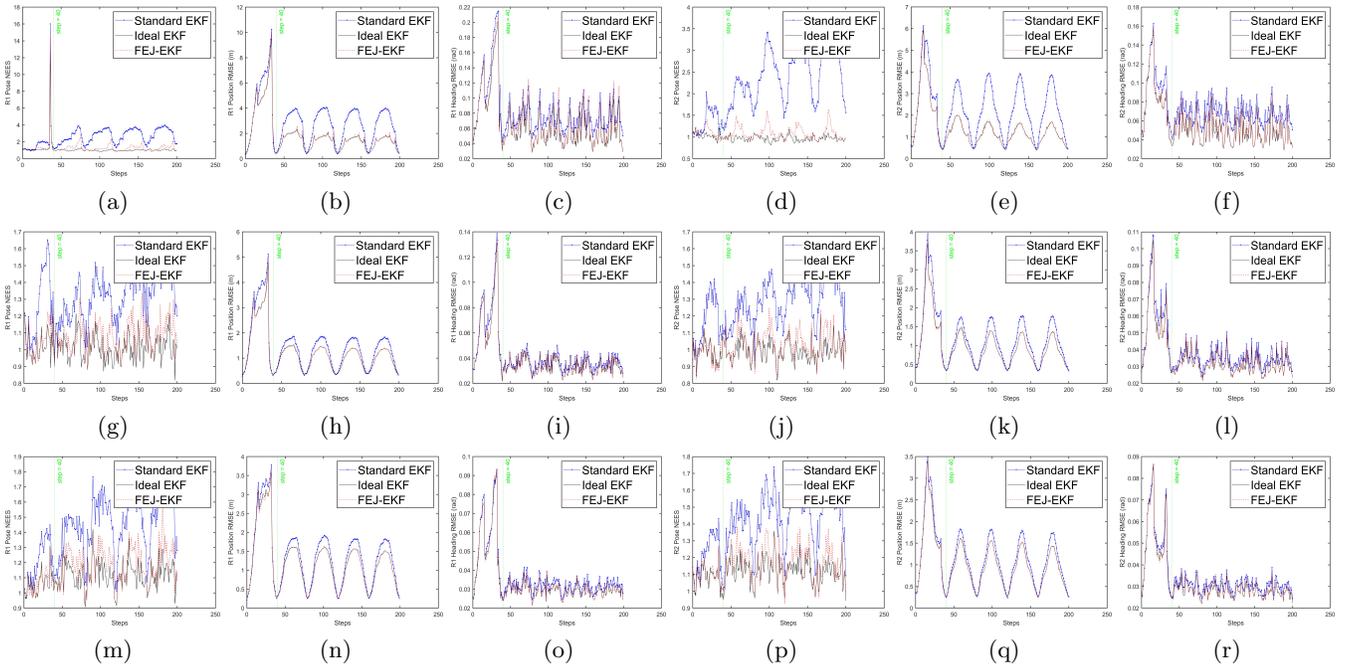


Figure 5: The RMSE and NEES results from the Monte Carlo simulations for environment 2 with different numbers of features: the upper six figures are for the 20 features, the middle six figures are for the 60 features, the lower six figures are for the 100 features

Metric		Robot 1			Robot 2			Feature	
		Pose NEES	Position RMSE	Heading RMSE	Pose NEES	Position RMSE	Heading RMSE	Position NEES	Position RMSE
20 features	Standard	2.1436	1.7601	0.0739	2.0672	1.5915	0.0775	3.6847	1.6984
	Ideal	0.9919	1.2915	0.0567	0.9992	1.1831	0.0594	0.9735	1.0658
	FEJ	1.0399	1.3085	0.0574	1.0370	1.1941	0.0603	1.0276	1.0767
60 features	Standard	3.8475	1.2282	0.0459	3.4688	0.9802	0.0484	6.4923	1.1229
	Ideal	1.0830	0.5786	0.0252	1.0827	0.4942	0.0265	1.0867	0.4312
	FEJ	1.1143	0.5794	0.0255	1.1235	0.4913	0.0269	1.0872	0.4297
100 features	Standard	2.4322	0.8085	0.0305	2.3029	0.6501	0.0325	3.6766	0.6975
	Ideal	1.0830	0.4762	0.0204	1.0660	0.3985	0.0213	1.0958	0.3513
	FEJ	1.1297	0.4904	0.0210	1.1178	0.4046	0.0218	1.1097	0.3534

Red and **Blue** for RMSE (the smaller the better) generally indicate the best and second best results, respectively.

Bold for NEES generally indicates the smallest and second smallest values (NEES values are incomparable if they are around 1).

Table 1: Performance of different EKF algorithms for three feature configuration of environment 1

Metric		Robot 1			Robot 2			Feature	
		Pose NEES	Position RMSE	Heading RMSE	Pose NEES	Position RMSE	Heading RMSE	Position NEES	Position RMSE
20 features	Standard	2.6083	3.0072	0.0819	2.2871	2.3172	0.0724	4.2160	2.2272
	Ideal	1.0729	1.8633	0.0663	1.0070	1.4300	0.0565	1.0183	1.0575
	FEJ	1.2683	1.9153	0.0702	1.0967	1.4349	0.0583	1.0853	1.0642
60 features	Standard	1.3032	1.4797	0.0432	1.2585	1.1988	0.0389	1.6542	1.1069
	Ideal	0.9814	1.2093	0.0394	0.9621	0.9849	0.0351	0.9440	0.7802
	FEJ	1.0552	1.2359	0.0406	0.9902	0.9905	0.0358	0.9358	0.7889
100 features	Standard	1.2834	1.2466	0.0344	1.2879	1.0441	0.0323	1.6020	0.9713
	Ideal	0.9704	1.0349	0.0310	0.9918	0.8766	0.0289	0.9658	0.7386
	FEJ	1.0374	1.0444	0.0317	1.0246	0.8792	0.0293	0.9965	0.7363

Red and **Blue** for RMSE (the smaller the better) generally indicate the best and second best results, respectively.

Bold for NEES generally indicates the smallest and second smallest values (NEES values are incomparable if they are around 1).

Table 2: Performance of different EKF algorithms for three feature configuration of environment 2

- 4(f) showing the corresponding results of 2R ; (ii) 60 features, Figure 4(g) - 4(i) describe the pose NEES, position RMSE and heading RMSE of 1R , respectively, with Figure 4(j) - 4(l) showing the corresponding results of 2R ; (iii) 100 features, Figure 4(m) - 4(o) describe the pose NEES, position RMSE and heading RMSE of 1R , respectively, with Figure 4(p) - 4(r) showing the corresponding results of 2R . The vertical dash line in Figure 4 denotes the step for the first loop closure. Similarly, Figure 5 shows the Monte Carlo results for the simulation environment 2 (Figure 3) with different numbers of features. In addition, Table 1 and Table 2 also presents the comparative results of corresponding Monte Carlo simulations.

In summary, by comparing the RMSE and NEES in all the figures and the corresponding average ones in all the tables, it can be generally revealed that for both 1R and 2R , FEJ-EKF has a better performance in terms of accuracy and consistency than Std-EKF.

4.2 Real-world Experiment

The Victoria Park dataset is used to test and validate the experimental performance of the FEJ-EKF. In the dataset [Guivant and Nebot, 2001], a single robot runs 6897 steps and observes tree trunks as features. As the dataset is collected by a single robot, we tend to find a suitable step to split the trajectory into two parts for two robots 1R and 2R . And we assume that these two part of corresponding data are collected separately by such two robots. Eventually, we select the following two parts of trajectory: the part of 1R starts from the 719th step and ends at the 3264th step; the part of 2R starts from the 3265th step and ends at the 5810th step. We chose these parts because, from the Gauss-Newton optimization results, the poses of robot at the 719th step and the 3265th step are relatively similar. Besides, 1R at the 719th step and 2R at the 3265th step have more than 8 shared features. We used the Gauss-Newton method to optimize the robot pose at the 719th step as the initial

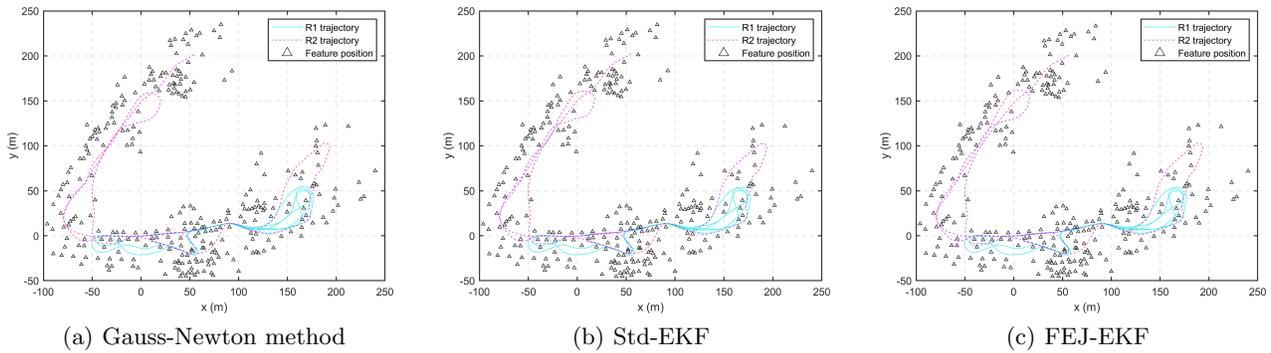


Figure 6: The estimated robot trajectories and estimated feature positions for Victoria Park Dataset

pose 0 of 1R .

In this experiment, as the ground truth for the poses of the robots and the positions of the features are unknown, the ideal-EKF is not available. We only show the estimates of (i) Gauss-Newton method, (ii) Std-EKF and (iii) FEJ-EKF in Figure 6(a) - 6(c), respectively. In these figures, the solid lines are the trajectories of 1R ; the dashed lines are the trajectories of 2R ; the triangles (Δ) are the positions of the features.

As implied in Figure 6, both Std-EKF and FEJ-EKF have good performances on the Victoria Park dataset as compared with the results from full non-linear least squares by Gauss-Newton method. The reason why Std-EKF and FEJ-EKF perform similarly is probably due to the small process and sensor noises involved in this practical dataset.

5 Conclusions and Future Works

In this paper, the first estimate Jacobian (FEJ)-EKF is applied to the point feature-based multi-robot SLAM problems. Based on the observability analysis, the proposed method maintains the correct observability property, addressing the issue appeared in the standard (Std)-EKF. The simulation results demonstrate that our proposed FEJ-EKF outperforms Std-EKF in terms of both accuracy and consistency, validating the effectiveness of our method. In the future, we will focus on developing consistent EKFs to the 3D cases. Also, the proposed algorithms will be applied to more real-world datasets.

References

- [Dubé *et al.*, 2017] Renaud Dubé, Abel Gawel, Hannes Sommer, Juan Nieto, Roland Siegwart, and Cesar Cadena. An online multi-robot slam system for 3d lidars. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 1004–1011, 2017.
- [Guivant and Nebot, 2001] J.E. Guivant and E.M. Nebot. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Trans. on Robotics and Automation*, 17(3):242–257, 2001.
- [Huang and Dissanayake, 2007] Shoudong Huang and Gamini Dissanayake. Convergence and Consistency Analysis for Extended Kalman Filter Based SLAM. *IEEE Trans. on Robotics*, 23(5):1036–1049, 2007.
- [Huang *et al.*, 2009] Guoquan P Huang, Anastasios I Mourikis, and Stergios I Roumeliotis. A first-estimates jacobian ekf for improving slam consistency. In *Experimental Robotics: The Eleventh International Symposium*, pages 373–382. Springer, 2009.
- [Huang *et al.*, 2010] Guoquan P Huang, Anastasios I Mourikis, and Stergios I Roumeliotis. Observability-based rules for designing consistent ekf slam estimators. *The international journal of Robotics Research*, 29(5):502–528, 2010.
- [Huang *et al.*, 2024] Shoudong Huang, Kai Pan, and Gamini Dissanayake. Perception in complex and unstructured infrastructure environments. *Infrastructure Robotics: Methodologies, Robotic Systems and Applications*, pages 33–58, 2024.
- [Julier and Uhlmann, 2001] Simon J Julier and Jeffrey K Uhlmann. A counter example to the theory of simultaneous localization and map building. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, volume 4, pages 4238–4243. IEEE, 2001.
- [Song *et al.*, 2022] Yang Song, Zhuqing Zhang, Jun Wu, Yue Wang, Liang Zhao, and Shoudong Huang. A Right Invariant Extended Kalman Filter for Object Based SLAM. *IEEE Robotics and Automation Letters*, 7(2):1316–1323, 2022.
- [Thrun and Liu, 2005] Sebastian Thrun and Yufeng Liu. Multi-robot slam with sparse extended information filters. In *Robotics Research. The Eleventh International Symposium: With 303 Figures*, pages 254–266. Springer, 2005.

- [Zhang *et al.*, 2017] Teng Zhang, Kanzhi Wu, Jingwei Song, Shoudong Huang, and Gamini Dissanayake. Convergence and consistency analysis for a 3-d invariant-ekf slam. *IEEE robotics and automation letters*, 2(2):733–740, 2017.
- [Zhang *et al.*, 2023] Zhuqing Zhang, Yang Song, Shoudong Huang, Rong Xiong, and Yue Wang. Toward consistent and efficient map-based visual-inertial localization: Theory framework and filter design. *IEEE Trans. on Robotics*, 39(4):2892–2911, 2023.
- [Zhou and Roumeliotis, 2006] Xun S Zhou and Stergios I Roumeliotis. Multi-robot slam with unknown initial correspondence: The robot rendezvous case. In *2006 IEEE/RSJ international conference on intelligent robots and systems*, pages 1785–1792. IEEE, 2006.