Distributed Invariant Extended Kalman Filter Using Lie Groups: Algorithm and Experiments

Jie Xu, Pengxiang Zhu, Yizhi Zhou, and Wei Ren, *Fellow, IEEE*

***Abstract*— Distributed Kalman filters have been widely studied in vector space and have been applied to 2-D target state estimation using sensor networks. In this article, we introduce a novel *distributed invariant extended Kalman filer* (DIEKF) that exploits matrix Lie groups and is suitable to track the target's 6-DOF motion in a 3-D environment. The DIEKF is based on the proposed extended covariance intersection (CI) algorithm that guarantees consistency in matrix Lie groups. The DIEKF is fully distributed as each agent only uses the information from itself and the one-hop communication neighbors, and it is robust to a time-varying communication topology and changing blind agents. In addition to assuming a known target model, we study the case where the target's true motion is unknown. To evaluate the performance, first, we apply the algorithm in a camera network to track a target pose. Extensive Monte-Carlo simulations have been performed to analyze the performance. More importantly, the performance is further verified with real data collected by using a quadrotor with multiple ultra-wideband (UWB) anchor receivers. Overall, the proposed algorithm is more accurate and more consistent in comparison with our recent work on the quaternion-based distributed extended Kalman filter (QDEKF).**

***Index Terms*— Distributed estimation, information fusion, invariant extended Kalman filtering, wireless sensor network.**

I. Introduction

SENSOR networks with the ability of communication and perception have a wide range of applications such as target tracking, area monitoring, and search and rescue. For the problem of target tracking, it is normally assumed that the target state evolves according to a known noisy model. Each agent can obtain observations of the target when the target is inside its sensing region. The objective is to use these measurements and the target dynamics to estimate its pose (i.e., orientation and position) on each agent. The problem can be solved in a centralized way where there exists a fusion center that collects all sensors' measurements and estimates the state using a *centralized* extended Kalman filter (EKF). Although this provides *optimal* accuracy, it requires communications from all agents to the center and expensive computational costs which makes the centralized algorithm not applicable for larger sensor networks. Therefore, distributed algorithms that use only each agent's own and communication neighbors' information draw more attention in both control and robotics societies.

In distributed algorithms, each agent maintains an estimator of the *same* target. To fuse the information from neighbors, there is a need to handle the unknown cross-covariances between different estimators on the agents. Naively fusing these estimators yields an inconsistent estimator that will diverge. The consensus [1] and the covariance intersection (CI) [2] algorithms have been widely used to design a consistent distributed EKF in the existing works. The consensus algorithm as a tool of information distributed averaging has been applied to the information pairs (i.e., information vectors and matrices) [3], the measurements [3], and the hybrid of the two in [4]. These approaches require multiple communication iterations at each timestamp. To be more efficient, the CI algorithm that computes a convex combination of the local information pairs from one-hop communication is used to design the DEKF. In [5], each agent first updates the estimator using its own measurements, and then the resulting information pairs are fused with the pairs from neighbors in CI. In [6], CI is first used to fuse the prior information pairs among neighborhoods and then the improved prior information is updated with the local and neighboring measurements. Note that all these algorithms work on *vector space* that has additive errors. Besides, the effectiveness is only evaluated on the tracking problem in 2-D cases. Although one can naively extend the vector space algorithm to the 3-D case by using the Euler angle representation for rotations, it suffers the well-known Gimbal lock problem.

To address this issue, our recent work [7] introduces a quaternion-based distributed EKF (QDEFK) algorithm where the 3-D orientation is represented as a unit quaternion. CI is for the first time extended to the 3-D space using "quaternion average" [8]. Good performance is shown by tracking a drone using a camera network. However, the filer is built upon the error-state EKF where the position, velocity, and orientation errors are decoupled [9], and the linearized error dynamics Jacobians and the measurement Jacobians are still functions of the estimated states. Then, the unobservable states can gain spurious information and become observable by the filter. This hurts the consistency and then the accuracy. Besides, we only study the case where the target motion model is known.

Recently, a new type of EKF is designed based on the invariant observer theory [10]. The estimation error is *invariant* under the action of matrix Lie groups and satisfies a log-linear autonomous differential equation with nice properties [11].

Manuscript received 11 March 2023; accepted 9 June 2023. Date of publication 28 August 2023; date of current version 23 October 2023. This work was supported by the National Science Foundation under Grant CMMI–2027139. Recommended by Associate Editor M. Abbaszadeh. *(Corresponding author: Wei Ren.)*

The authors are with the Department of Electrical and Computer Engineering, University of California at Riverside, Riverside, CA 92521 USA (e-mail: jxu150@ucr.edu; pzhu008@ucr.edu; yzhou280@ucr.edu; ren@ee.ucr.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCST.2023.3290299.

Digital Object Identifier 10.1109/TCST.2023.3290299

In particular, in the case of $\mathrm{SE}_K(3)$, the position, velocity, and orientation errors are coupled. This invariant EKF (IEKF) has been successfully applied to leg robot state estimation [12] and simultaneous localization and mapping (SLAM) [13], where the IEKF achieves promising performance, especially with poor initialization. It is proved in these articles that the observability of the linearized system is coincident with the original nonlinear system.

The main contribution of the article is that it introduced a novel distributed algorithm that utilizes the invariant error associated with the matrix Lie group to address the 3-D target estimation when the robots can only communicate with their neighbors and see the target from time to time. The proposed algorithm is fully distributed in that each agent estimates the 3-D motion of the target by using information from the neighborhood. To the best of our knowledge, there is no existing work in the literature that addresses 3-D motion estimation in a fully distributed manner. Furthermore, we show that the proposed algorithm can be extended to track the target even when the target motion is unknown, which is also a contribution.

Motivated by the IEKF and QDEKF algorithms, in this article, we design a novel distributed IEKF (DIEKF) that utilizes the invariant errors associated with the matrix Lie group for solving the problem of distributed state estimation using sensor networks in a 3-D environment. To design DIEKF, we extend the CI in a vector setting to matrix Lie groups for the first time. The proposed algorithm is fully distributed in that each agent only estimates the 3-D motion of the target by using the information among the neighborhood. Furthermore, we show that the proposed algorithm can be extended to track the target even when the target motion is unknown. The proposed algorithm is first applied in simulation to track target 3-D motion in a camera network. The algorithm is then implemented in experiments to track the 3-D pose of a quadrotor using multiple ultra-wideband (UWB) anchor receivers. As shown in both Monte-Carlo simulations and experiments, the accuracy and consistency of the DIEKF in both position and orientation are improved as compared against the QDEKF [7]. The current article expands on our preliminary result presented at [14] by further considering the case of the unknown target motion and providing experimental results.

II. Preliminaries

A. Notation and Definitions

We denote $\mathbf{0}_{m\times n}$ as a $m \times n$ zero matrix, and $\mathbf{I}_n$ ($\mathbf{0}_n$) as $n \times n$ square identity (zero) matrix. Given a 3×1 vector $\mathbf{q} = [q_1, q_2, q_3]^\top$, its skew symmetric matrix is defined as

$$(\mathbf{q})_\times = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}$$

and its projection function is defined as $\Pi(\mathbf{q}) = (1/q_3)[q_1, q_2]$. The Jacobian of the projection function is computed as

$$\mathbf{H}_p(\mathbf{q}) = \frac{1}{q_3}\begin{bmatrix} 1 & 0 & -\dfrac{q_1}{q_3} \\ 0 & 1 & -\dfrac{q_2}{q_3} \end{bmatrix}. \tag{1}$$

In a network of $\mathcal{M}$ agents, we define a directed graph $\mathfrak{G} = (\mathcal{V}, \mathcal{E})$ to represent the communication topology among agents, where $\mathcal{V}$ indicates the set of all the agents, and $\mathcal{E}$ stands for the set of communication links defined as $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$. Specifically, if $(j, i) \in \mathcal{E}$, agent j is a neighbor of agent i, and agent i can receive information from agent j. We assume that self communication always exists, i.e., $(i, i) \in \mathcal{E}$, $\forall i \in \mathcal{V}$. The set of all the communicating neighbors of agent i is defined as $\mathcal{N}_i = \{j | (j, i) \in \mathcal{E},\ j \in \mathcal{V}\}$.

B. Problem Formulation

Consider a network of agents in the 3-D environment with fixed and known positions aiming to cooperatively track a moving target's state. Each agent can communicate with its neighbors and is equipped with an on-board camera. Denote G, T, and C_i as the global frame, the target frame, and ith agent's camera frame, respectively. Let ${}^G_T R \in \mathrm{SO}(3)$ be the rotation matrix that describe the rotation from T to G. Let ${}^G v \in \mathbb{R}^3$ and ${}^G p \in \mathbb{R}^3$ be the target's velocity and position in the global frame. Let ${}^G p_{C_i} \in \mathbb{R}^3$ be the position of agent i's camera in the global frame. The target is modeled as a point cloud where one of the feature points is chosen as the representative point and the others as the nonrepresentative points. Computer vision techniques can be used to distinguish the representative point and the nonrepresentative points. The representative point is the origin of the target frame while the positions of the nonrepresentative points in the target frame are unknown but fixed. The nonrepresentative points provide additional measurements and constraints and can hence improve the estimation accuracy. For convenience, we assume that there is only one nonrepresentative feature point. However, the state can be augmented to include multiple nonrepresentative points. Let ${}^T p_f \in \mathbb{R}^3$ be the position of the nonrepresentative feature point in the target frame.

Here we consider two cases of state representation. First, we consider the case where the agents have access to the target's motion inputs. In this case, the state of the target is represented as

$$\mathsf{x} = \begin{pmatrix} {}^G_T R & {}^G v & {}^G p & {}^T p_f \end{pmatrix} \tag{2}$$

which includes the target's 6-DoF pose ${}^G_T R$ and ${}^G p$, the linear velocity ${}^G v$ in the global frame, and the 3-D position of a nonrepresentative point in the target frame ${}^T p_f$. The individual dynamics of the state are given as

$$\begin{aligned} {}^G_T \dot{R} &= {}^G_T R\,(\omega - n_\omega)_\times \\ {}^G \dot{v} &= {}^G_T R\,(a - n_a) + g \\ {}^G \dot{p} &= {}^G v \\ {}^T \dot{p}_f &= \mathbf{0}_{3\times 1} \end{aligned} \tag{3}$$

where ω and a are, respectively, the angular velocity and the linear acceleration of the target in the target frame, the corresponding n_ω and n_a are white Gaussian noises, and g is the gravity vector.

Second, we consider the case where the agents do not have access to the target's motion inputs. We adopt a generic model with constant angular velocity ω and linear acceleration a in

the target frame to propagate the target's state. The dynamics of the individual states are the same as (3) with n_ω and n_a removed with additional dynamics for ω and a as

$$\begin{aligned}\dot{\omega} &= n_\omega \\ \dot{a} &= n_a\end{aligned} \tag{4}$$

where ω and a are treated as random walks driven by zero-mean white Gaussian noises n_ω and n_a. These two parameters ω and a become extended states to estimate. The covariances of n_ω and n_a become tuning parameters in the filter design. The state of the target is represented as

$$\mathsf{x} = \begin{pmatrix} {}^G_T R & {}^G v & {}^G p & {}^T p_f & \omega & a \end{pmatrix}. \tag{5}$$

In both cases, the measurements of the representative feature at time t_k (specifying the target's position) and the nonrepresentative feature obtained by agent i's camera are given by, respectively,

$$\begin{aligned} z_i^k &= \Pi\left({}^{C_i}p^k\right) + n_i^k \\ z_{f_i}^k &= \Pi\left({}^{C_i}p_f^k\right) + n_{f_i}^k \end{aligned} \tag{6}$$

where n_i^k and $n_{f_i}^k$ are the measurement noises of agent i's camera at time t_k, assumed to be white Gaussian, and ${}^{C_i}p^k$ and ${}^{C_i}p_f^k$ denote, respectively, the representative feature's position (target's position) and the nonrepresentative feature's position in agent i's camera frame at time t_k. The objective of our work is to let each agent compute an accurate estimate of the target's state.

C. Lie Group and Lie Algebra

Here we briefly introduce the matrix Lie group theory that we will use to derive our algorithm. The material is adopted from [10]. A matrix Lie group $\mathcal{G}$ is a subset of square invertible $N \times N$ matrices satisfying

$$\begin{aligned} &\mathbf{I}_N \in \mathcal{G} \\ &\forall a \in \mathcal{G}, \quad a^{-1} \in \mathcal{G} \\ &\forall a, b \in \mathcal{G}, \quad ab \in \mathcal{G}. \end{aligned}$$

Its Lie algebra is denoted as $\mathfrak{g}$, which is a vector space with the same dimension as $\mathcal{G}$. For convenience, let $(\cdot)^\wedge : \mathbb{R}^{\dim \mathfrak{g}} \to \mathfrak{g}$ be the linear map that transforms the elements in the Lie algebra to the corresponding matrix representation. The exponential map is further defined as $\exp(\xi) = \exp_m(\xi^\wedge) \in \mathcal{G}$, where $\xi \in \mathbb{R}^{\dim \mathfrak{g}}$ is an element in $\mathfrak{g}$, and $\exp_m$ is the matrix exponential. The logarithm map, which is the inverse function of the exponential map, is denoted by $\log(\cdot)$, and satisfies $\log(\cdot) = (\log_m(\cdot))^\vee : \mathcal{G} \to \mathbb{R}^{\dim \mathfrak{g}}$, where $\log_m$ is the matrix logarithm, and $(\cdot)^\vee$ is the inverse operator of $(\cdot)^\wedge$.

Let $\mathbf{X}_t \in \mathcal{G}$ be the state of a system at time t. The dynamics of the system are denoted as

$$\frac{d}{dt}\mathbf{X}_t = f_{u_t}(\mathbf{X}_t) \tag{7}$$

where u_t is the input. Let $\mathbf{X}_t$ and $\bar{\mathbf{X}}_t$ be two distinct trajectories of (7). The right invariant error is then defined as

$$\eta_t = \mathbf{X}_t \left(\bar{\mathbf{X}}_t\right)^{-1}. \tag{8}$$

The error (8) is invariant to the right multiplication of any element $\Upsilon \in \mathcal{G}$.

Let $\mathbf{I}_d \in \mathcal{G}$ be the identity element of $\mathcal{G}$. If the dynamics of the system satisfy

$$f_{u_t}\left(\mathbf{X}_t \bar{\mathbf{X}}_t\right) = f_{u_t}(\mathbf{X}_t)\bar{\mathbf{X}}_t + \mathbf{X}_t f_{u_t}\left(\bar{\mathbf{X}}_t\right) - \mathbf{X}_t f_{u_t}(\mathbf{I}_d)\bar{\mathbf{X}}_t$$

the system is group affine. Then, the right invariant error dynamics satisfy $(d/dt)\eta_t = g_{u_t}(\eta_t)$, where $g_{u_t}(\eta_t) = f_{u_t}(\eta_t) - \eta_t f_{u_t}(\mathbf{I}_d)$. Define A_t as a matrix satisfying $g_{u_t}(\exp(\xi_t)) \triangleq (A_t\xi_t)^\wedge + \mathcal{O}(||\xi_t||^2)$, and let ξ_t be the solution of $(d/dt)\xi_t = A_t\xi_t$. From the log-linear property of the error η_t, given $\eta_0 = \exp(\xi_0)$, for $t \geq 0$, the error η_t can be computed from ξ_t by

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

D. Matrix Lie Group Representation

As shown in [10] and [12], the state collection shown in (2) forms a matrix Lie group $\mathrm{SE}_{2+L}(3)$, where $L = 1$ denotes the number of nonrepresentative points, represented as

$$\mathsf{X} = \begin{bmatrix} {}^G_T R & {}^G v & {}^G p & {}^T p_f \\ \mathbf{0}_{1\times 3} & 1 & 0 & 0 \\ \mathbf{0}_{1\times 3} & 0 & 1 & 0 \\ \mathbf{0}_{1\times 3} & 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{6\times 6}.$$

In the case that there are more than one nonrepresentative point (i.e., $L \geq 1$), it is straightforward to expand X to include the other nonrepresentative point(s). Let X_t be the state representation at time t, and $\bar{\mathsf{X}}_t$ be the state estimate. Define the right invariant estimation error η_t given as

$$\eta_t = \mathsf{X}_t\left(\bar{\mathsf{X}}_t\right)^{-1} = \begin{bmatrix} \theta_{11}, & \theta_{12}, & \theta_{13}, & \theta_{14} \\ \mathbf{0}_{1\times 3}, & 1, & 0, & 0 \\ \mathbf{0}_{1\times 3}, & 0, & 1, & 0 \\ \mathbf{0}_{1\times 3}, & 0, & 0, & 1 \end{bmatrix}$$

where the individual terms are calculated as

$$\begin{aligned} \theta_{11} &= {}^G_T R_t \left({}^G_T \bar{R}_t\right)^\top \in \mathrm{SO}(3) \\ \theta_{12} &= {}^G v_t - \theta_{11}{}^G \bar{v}_t \in \mathbb{R}^3 \\ \theta_{13} &= {}^G p_t - \theta_{11}{}^G \bar{p}_t \in \mathbb{R}^3 \\ \theta_{14} &= {}^T p_{f_t} - \theta_{11}{}^T \bar{p}_{f_t} \in \mathbb{R}^3. \end{aligned}$$

Note that the orientation estimation error θ_{11} is coupled in the velocity estimation error θ_{12}, the position estimation error θ_{13}, and the nonrepresentative point estimation error θ_{14}. It is worth mentioning that there are closed-loop formulas to compute the inverse and log function of an element in $\mathrm{SE}_{2+L}(3)$ as well as the exp function of a vector (see [10], [12]). Hence the computation complexity for these quantities is mild. The error vector ξ_t, defined in the Lie algebra of $\mathrm{SE}_3(3)$, denoted by $\mathfrak{se}_3(3)$, is given by

$$\xi_t = \begin{bmatrix} \left(\xi_{R_t}\right)^\top & \left(\xi_{v_t}\right)^\top & \left(\xi_{p_t}\right)^\top & \left(\xi_{p_{f_t}}\right)^\top \end{bmatrix}^\top \in \mathbb{R}^{12}$$

where ξ_{R_t}, ξ_{v_t}, ξ_{p_t}, and $\xi_{p_{f_t}} \in \mathbb{R}^3$. Here, $\eta_t = \exp(\xi_t) = \exp_m(\xi_t^\wedge)$, with $\xi_t^\wedge$ given as

$$\xi_t^\wedge = \begin{bmatrix} (\xi_{R_t})_\times & \xi_{v_t} & \xi_{p_t} & \xi_{p_{f_t}} \\ \mathbf{0}_{1\times 3} & 0 & 0 & 0 \\ \mathbf{0}_{1\times 3} & 0 & 0 & 0 \\ \mathbf{0}_{1\times 3} & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{6\times 6}. \tag{9}$$

It is shown in [12] that (3) without noise is group affine, and the dynamics of ξ_t are given as

$$\frac{d}{dt}\xi_t = A_t\xi_t - \mathrm{Ad}_{\bar{\mathsf{X}}_t}\mathsf{U}_t \tag{10}$$

where

$$A_t = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ (g)_\times & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}$$

and $\mathrm{Ad}_{\bar{\mathsf{X}}_t}$ denotes the adjoint of $\mathrm{SE}_3(3)$ at $\bar{\mathsf{X}}_t$ given as

$$\begin{bmatrix} {}^G_T\bar{R}_t & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \\ ({}^G\bar{v}_t)_\times {}^G_T\bar{R}_t & {}^G_T\bar{R}_t & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \\ ({}^G\bar{p}_t)_\times {}^G_T\bar{R}_t & \mathbf{0}_{3\times 3} & {}^G_T\bar{R}_t & \mathbf{0}_{3\times 3} \\ ({}^T\bar{p}_{f_t})_\times {}^G_T\bar{R}_t & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & {}^G_T\bar{R}_t \end{bmatrix}$$

and $\mathsf{U}_t = [n_{\omega_t}^\top, n_{a_t}^\top, \mathbf{0}_{1\times 6}]^\top$. Note that here A_t does not depend on the estimated state, which improves the estimation consistency and accuracy.

For the state shown in (5), it is no longer possible to find a matrix Lie group representation that satisfies the group affine property. Instead, we represent the state in $\mathrm{SE}_3(3) \times \mathbb{R}^6$ as a combination of X_t and $\{\omega_t, a_t\}$. Let $\bar{\omega}_t$ and $\bar{a}_t$ denote the state estimates of, respectively, ω_t and a_t. The estimation error in $\mathrm{SE}_3(3)\times\mathbb{R}^6$ is given by a combination of η_t, $\tilde{\omega}_t$, and $\tilde{a}_t$, where $\tilde{\omega}_t = \omega_t - \bar{\omega}_t$ and $\tilde{a}_t = a_t - \bar{a}_t$. The corresponding error in $\mathrm{se}_3(3) \times \mathbb{R}^6$ is given by $\xi_{\mathrm{ext},t} = [\xi_t^\top, \tilde{\omega}_t, \tilde{a}_t]^\top$. The dynamics of the vector error are shown as

$$\frac{d}{dt}\xi_{\mathrm{ext},t} = A_t\xi_{\mathrm{ext},t} - \mathsf{U}_t \tag{11}$$

where

$$A_t = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & {}^G_T\bar{R}_t & \mathbf{0}_3 \\ (g)_\times & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \left(({}^G\bar{v}_t)_\times\right) {}^G_T\bar{R}_t & {}^G_T\bar{R}_t \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \left(({}^G\bar{p}_t)_\times\right) {}^G_T\bar{R}_t & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \left(({}^T\bar{p}_{f_t})_\times\right) {}^G_T\bar{R}_t & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}$$

and $\mathsf{U}_t = [\mathbf{0}_{1\times 12}, n_{\omega_t}^\top, n_{a_t}^\top]^\top$. Note that compared with [12], the nonzero entries in the last two columns of A_t have opposite signs in our setting.

III. Proposed Algorithm

A. Known Target Motion

Suppose that each agent knows the target motion. For notation simplicity, let X^k denote the target state at time t_k. Also let $\bar{\mathsf{X}}_i^k$ and $\hat{\mathsf{X}}_i^k$ denote, respectively, agent i's prior and posterior estimates of the target state at time t_k. Let $\eta_i^{k|k-1} = \mathsf{X}^k(\bar{\mathsf{X}}_i^k)^{-1} = \exp(\xi_i^{k|k-1})$, and $\eta_i^{k|k} = \mathsf{X}^k(\hat{\mathsf{X}}_i^k)^{-1} = \exp(\xi_i^{k|k})$ denote, respectively, agent i's prior and posterior estimation errors in $\mathrm{SE}_3(3)$. Here $\xi_i^{k|k-1}$ and $\xi_i^{k|k}$ denote, respectively, the prior and posterior estimation errors in $\mathrm{se}_3(3)$. The covariances associated with $\xi_i^{k|k-1}$ and $\xi_i^{k|k}$ are denoted, repectively, as $\bar{P}_i^k$ and $\hat{P}_i^k$.

The first step is to propagate the posterior estimation pair $(\hat{\mathsf{X}}_i^{k-1}, \hat{P}_i^{k-1})$ to obtain the prior estimation pair $(\bar{\mathsf{X}}_i^k, \bar{P}_i^k)$. As this step is standard, we include it in Appendix A.

In the second step, agent i aims to update its local estimate by communicating with its neighbors. The goal is to fuse all the prior estimates among the neighbors, i.e., $(\bar{\mathsf{X}}_j^k, \bar{P}_j^k)$, $j \in \mathcal{N}_i^k$, to obtain an intermediate estimation pair $(\check{\mathsf{X}}_i^k, \check{P}_i^k)$. By doing so, the agents with better estimates would help those with poor estimates. For example, blind agents, which refer to the agents that themselves and their one-hop communication neighbors cannot see the target, would have poor estimates. Because of the communication from previous timesteps, and the fact that the agents are estimating the same target, $\bar{\mathsf{X}}_i^k$ and $\bar{\mathsf{X}}_j^k$ would be correlated. However, in the distributed setting, it is not possible to keep tracking the cross-correlations and hence they are unknown. The CI algorithm [15] can be used to fuse estimates with unknown correlations. However, the CI algorithm is applicable in the vector space and it is not clear how to fuse the Lie group elements $\bar{\mathsf{X}}_i^k$ and $\bar{\mathsf{X}}_j^k$, which are in $\mathrm{SE}_3(3)$. Next, we extend the CI algorithm in the context of error-state EKF in Lie groups to estimate the error states represented in $\mathrm{se}_3(3)$ and map back to $\mathrm{SE}_3(3)$.

Recall that for each $j \in \mathcal{N}_i^k$, $\eta_j^{k|k-1} = \mathsf{X}^k(\bar{\mathsf{X}}_j^k)^{-1}$ is the prior estimation error in $\mathrm{SE}_3(3)$, and $\xi_j^{k|k-1} = \log(\eta_j^{k|k-1})$ is the corresponding prior estimation error in $\mathrm{se}_3(3)$. We will make use of $\bar{\mathsf{X}}_j^k$, $j \in \mathcal{N}_i^k$, to identify the estimates of $\xi_i^{k|k-1}$, and then fuse these estimates. Note that for each $j \in \mathcal{N}_i^k$

$$\begin{aligned} \exp\left(\xi_j^{k|k-1}\right) &= \mathsf{X}^k\left(\bar{\mathsf{X}}_j^k\right)^{-1} = \mathsf{X}^k\left(\bar{\mathsf{X}}_i^k\right)^{-1}\left(\bar{\mathsf{X}}_i^k\right)\left(\bar{\mathsf{X}}_j^k\right)^{-1} \\ &= \exp\left(\xi_i^{k|k-1}\right)\left(\bar{\mathsf{X}}_i^k\right)\left(\bar{\mathsf{X}}_j^k\right)^{-1}. \end{aligned}$$

It follows that

$$\bar{\mathsf{X}}_j^k\left(\mathsf{X}_i^k\right)^{-1} = \exp\left(-\xi_j^{k|k-1}\right)\exp\left(\xi_i^{k|k-1}\right). \tag{12}$$

According to the Baker–Campbell–Hausdorff (BCH) formula [16], the exponential map satisfies the property $\exp(\xi_1)\exp(\xi_2) \approx \exp(\xi_1+\xi_2)$, if ξ_1 and ξ_2 are small by ignoring higher-order terms involving ξ_1 and ξ_2. In the remainder of the article, the approximation symbol denotes the fact that the higher-order terms of small quantities are ignored. Then, (12) can be rewritten as $\bar{\mathsf{X}}_j^k(\bar{\mathsf{X}}_i^k)^{-1} \approx \exp(\xi_i^{k|k-1} - \xi_j^{k|k-1})$, which implies that $\xi_i^{k|k-1} - \xi_j^{k|k-1} \approx \log(\bar{\mathsf{X}}_j^k(\bar{\mathsf{X}}_i^k)^{-1})$, or equivalently

$\xi_i^{k|k-1} - \log(\bar{\mathsf{X}}_j^k(\bar{\mathsf{X}}_i^k)^{-1}) \approx \xi_j^{k|k-1}$, for each $j \in \mathcal{N}_i^k$. As a result, each $\log(\bar{\mathsf{X}}_j^k(\bar{\mathsf{X}}_i^k)^{-1})$, $j \in \mathcal{N}_i^k$, can be treated as one prior estimate of $\xi_i^{k|k-1}$, and the corresponding estimation error is given by $\xi_j^{k|k-1}$ with covariance $\bar{P}_j^k$. We can use the CI algorithm to fuse all estimation pairs $(\log(\bar{\mathsf{X}}_j^k(\bar{\mathsf{X}}_i^k)^{-1}), \bar{P}_j^k)$ to get an intermediate estimation pair $(\check{\xi}_i^{k|k-1}, \check{P}_i^k)$ for $\xi_i^{k|k-1}$ by

$$\check{P}_i^k = \left[\sum_{j\in\mathcal{N}_i^k} \pi_j^k \left(\bar{P}_j^k\right)^{-1}\right]^{-1}$$

$$\check{\xi}_i^{k|k-1} = \check{P}_i^k \left[\sum_{j\in\mathcal{N}_i^k} \pi_j^k \left(\bar{P}_j^k\right)^{-1} \log\left(\left(\bar{\mathsf{X}}_j^k\right)\left(\bar{\mathsf{X}}_i^k\right)^{-1}\right)\right]$$

where $\pi_j^k \in [0,1]$ and $\sum_{j\in\mathcal{N}_i} \pi_j^k = 1$. Note that as $i \in \mathcal{N}_i^k$, the prior estimate of $\xi_i^{k|k-1}$ by agent i is simply $\log(\bar{\mathsf{X}}_i^k(\bar{\mathsf{X}}_i^k)^{-1}) = \mathbf{0}_{12\times 1}$ with covariance $\bar{P}_i^k$, which is consistent with the definition of $\xi_i^{k|k-1}$. While π_j^k can be solved from an optimization problem, a simplified algorithm can be used to compute π_j^k according to [15]

$$\pi_j^k = \frac{1/\mathrm{Tr}\{\bar{P}_j^k\}}{\sum_{j\in\mathcal{N}_i} 1/\mathrm{Tr}\{\bar{P}_j^k\}}.$$

The intermediate state estimate of X^k in $\mathrm{SE}_3(3)$, denoted by $\check{\mathsf{X}}_i^k$, can be recovered from $\check{\xi}_i^{k|k-1}$ by

$$\check{\mathsf{X}}_i^k = \exp\left(\check{\xi}_i^{k|k-1}\right)\bar{\mathsf{X}}_i^k. \tag{13}$$

Now define $\varepsilon_i^{k|k-1}$ as the new error vector in $\mathrm{se}_3(3)$ with $\check{\mathsf{X}}_i^k$ being the prior estimate of X^k satisfying

$$\exp\left(\varepsilon_i^{k|k-1}\right) = \mathsf{X}^k\left(\check{\mathsf{X}}_i^k\right)^{-1}. \tag{14}$$

Because we are going to apply the error-state EKF on the intermediate estimation error $\varepsilon_i^{k|k-1}$, we need the corresponding covariance. Note that

$$\begin{aligned}
\mathsf{X}^k &= \exp\left(\varepsilon_i^{k|k-1}\right)\check{\mathsf{X}}_i^k = \exp\left(\xi_i^{k|k-1}\right)\bar{\mathsf{X}}_i^k \\
&\Rightarrow \exp\left(\varepsilon_i^{k|k-1}\right)\exp\left(\check{\xi}_i^{k|k-1}\right)\bar{\mathsf{X}}_i^k = \exp\left(\xi_i^{k|k-1}\right)\bar{\mathsf{X}}_i^k \\
&\Rightarrow \exp\left(\varepsilon_i^{k|k-1}\right)\exp\left(\check{\xi}_i^{k|k-1}\right) = \exp\left(\xi_i^{k|k-1}\right) \\
&\Rightarrow \exp\left(\varepsilon_i^{k|k-1}\right) = \exp\left(\xi_i^{k|k-1}\right)\exp\left(-\check{\xi}_i^{k|k-1}\right) \\
&\Rightarrow \exp\left(\varepsilon_i^{k|k-1}\right) \approx \exp\left(\xi_i^{k|k-1} - \check{\xi}_i^{k|k-1}\right) \\
&\Rightarrow \varepsilon_i^{k|k-1} \approx \xi_i^{k|k-1} - \check{\xi}_i^{k|k-1}
\end{aligned} \tag{15}$$

and $\check{P}_i^k$ is the estimated covariance for $\xi_i^{k|k-1} - \check{\xi}_i^{k|k-1}$. As a result, $\check{P}_i^k$ can be directly used as the estimated covariance for the new error $\varepsilon_i^{k|k-1}$.

The third step is to fuse agent i's intermediate estimation pair $(\check{\mathsf{X}}_i^k, \check{P}_i^k)$ with all the measurements from itself and its neighbors, i.e., representative feature measurements z_j^k, $\forall j \in \mathcal{N}_i^k$, and nonrepresentative feature measurements $z_{f_j}^k$, $\forall j \in \mathcal{N}_i^k$, defined by (6). To calculate the Jacobian associated with z_j^k, denoted by H_j^k, first define

$${}^{C_j}\check{p}_i^k = {}_G^{C_j}R\left({}^G\check{p}_i^k - {}^Gp_{C_j}\right) \tag{16}$$

as the linearization point in agent j's camera frame, where ${}_G^{C_j}R$ and ${}^Gp_{C_j}$ together denote the 3-D pose of agent j's camera that is fixed and known. Then, the difference between the true target's position and the linearization point in agent j's camera frame is calculated as

$$\begin{aligned}
{}^{C_j}p_i^k - {}^{C_j}\check{p}_i^k &= {}_G^{C_j}R\left({}^Gp^k - {}^Gp_{C_j}\right) - {}_G^{C_j}R\left({}^G\check{p}_i^k - {}^Gp_{C_j}\right) \\
&= {}_G^{C_j}R\left({}^Gp^k - {}^G\check{p}_i^k\right).
\end{aligned}$$

Note that $\varepsilon_i^{k|k-1}$ is a column stack vector of $\varepsilon_{R_i}^{k|k-1}$, $\varepsilon_{v_i}^{k|k-1}$, $\varepsilon_{p_i}^{k|k-1}$, and $\varepsilon_{p_{f_i}}^{k|k-1}$. Recall from (14) that $\mathsf{X}^k = \exp(\varepsilon_i^{k|k-1})\check{\mathsf{X}}_i^k \approx (\mathbf{I}_6 + (\varepsilon_i^{k|k-1})^\wedge)\check{\mathsf{X}}_i^k$, where $(\cdot)^\wedge$ is defined by (9). The position of the target can be calculated as ${}^Gp^k \approx [\mathbf{I}_3 + (\varepsilon_{R_i}^{k|k-1})_\times]{}^G\check{p}_i^k + \varepsilon_{p_i}^{k|k-1} = -({}^G\check{p}_i^k)_\times\varepsilon_{R_i}^{k|k-1} + \varepsilon_{p_i}^{k|k-1} + {}^G\check{p}_i^k$. Hence the Jacobian H_j^k is calculated as

$$H_j^k = \mathbf{H}_p\left({}^{C_j}\check{p}_i^k\right)\left({}_G^{C_j}R\right)\left[-\left({}^G\check{p}_i^k\right)_\times,\ \mathbf{0}_3,\ \mathbf{I}_3,\ \mathbf{0}_3\right]$$

where $\mathbf{H}_p(\cdot)$ is defined by (1). To calculate the Jacobian associated with the nonrepresentative feature, denoted by $H_{f_j}^k$, define

$${}^{C_j}\check{p}_{f_i}^k = {}_G^{C_j}R\left({}_T^G\check{R}_i^k\left({}^T\check{p}_{f_i}^k\right) + {}^G\check{p}_i^k - {}^Gp_{C_j}\right) \tag{17}$$

as its linearization point. Then, the difference between the true position of the nonrepresentative feature and the linearization point is given by

$${}^{C_j}p_{f_i}^k - {}^{C_j}\check{p}_{f_i}^k = M_i^k\varepsilon_{R_i}^{k|k-1} + \varepsilon_{p_i}^{k|k-1} + {}_T^G\check{R}_i^k\varepsilon_{p_{f_i}}^{k|k-1}$$

where M_i^k is given by

$$M_i^k = -{}_T^G\check{R}_i^k\left({}^T\check{p}_{f_i}^k\right)_\times - \left({}_T^G\check{R}_i^k{}^T\check{p}_{f_i}^k\right)_\times - \left({}^G\check{p}_i^k\right)_\times. \tag{18}$$

Hence the Jacobian $H_{f_j}^k$ is calculated as

$$H_{f_j}^k = \mathbf{H}_p\left({}^{C_j}\check{p}_{f_i}^k\right)\left({}_G^{C_j}R\right)\left[M_i^k,\ \mathbf{0}_3,\ \mathbf{I}_3,\ {}_T^G\check{R}_i^k\right].$$

Note that at time t_k, it is possible that some or even all neighbors do not see the representative or nonrepresentative feature. Let $\bar{\mathcal{N}}_i^k \subset \mathcal{N}_i^k$ (respectively, $\bar{\mathcal{N}}_{f_i}^k \subset \mathcal{N}_i^k$) be the subset of agent i's neighbors that can see the representative feature (respectively, nonrepresentative feature) at time t_k. Define $\tilde{H}_i^k$ by stacking all H_j^k, $j \in \bar{\mathcal{N}}_i^k$, and $H_{f_j}^k$, $j \in \bar{\mathcal{N}}_{f_i}^k$. Let $\tilde{C}_i^k$ be a block diagonal matrix with the measurement noise covariances C_j^k, $j \in \bar{\mathcal{N}}_i^k$, and $C_{f_j}^k$, $j \in \bar{\mathcal{N}}_{f_i}^k$. Let $S_i^k = \tilde{H}_i^k\check{P}_i^k(\tilde{H}_i^k)^\top + \tilde{C}_i^k$. The Kalman gain K_i^k is calculated as

$$K_i^k = \check{P}_i^k\left(\tilde{H}_i^k\right)^\top\left(S_i^k\right)^{-1} \tag{19}$$

and the measurement residual y_i^k is given by stacking $z_j^k - \Pi({}^{C_j}\check{p}_i^k)$, $j \in \bar{\mathcal{N}}_i^k$, and $z_{f_j}^k - \Pi({}^{C_j}\check{p}_{f_i}^k)$, $j \in \bar{\mathcal{N}}_{f_i}^k$. Then, the

TABLE I
DIEKF ALGORITHM WITH KNOWN TARGET MOTION

Propagation:

obtain $\bar{\mathsf{X}}_i^k$ using (23), calculate $\bar{P}_i^k$ using (24)

Compute the intermediate update:

obtain$\breve{\mathsf{X}}_i^k$and $\breve{P}_i^k$ using (12) and (13)

Compute the measurement update:

obtain$\hat{\mathsf{X}}_i^k$ and $\hat{P}_i^k$ using (20) and (21)

posterior estimate of $\varepsilon_i^{k|k-1}$, denoted as $\hat{\varepsilon}_i^k$, and its covariance $\hat{P}_i^k$ are calculated as

$$\begin{aligned}\hat{\varepsilon}_i^k &= K_i^k y_i^k\\ \hat{P}_i^k &= \left(\mathbf{I}_{12} - K_i^k \tilde{H}_i^k\right)\breve{P}_i^k.\end{aligned} \tag{20}$$

In the last step, we recover the estimated state $\hat{\mathsf{X}}_i^k$ from $\hat{\varepsilon}_i^k$ and $\breve{\mathsf{X}}_i^k$ analogously to (13) as

$$\hat{\mathsf{X}}_i^k = \exp\left(\hat{\varepsilon}_i^k\right)\breve{\mathsf{X}}_i^k. \tag{21}$$

Note that a similar procedure to (15) can be used to show that $\xi_i^{k|k} \approx \varepsilon_i^{k|k-1} - \hat{\varepsilon}_i^k$. Hence $\hat{P}_i^k$ can be used as the estimated covariance for $\xi_i^{k|k}$. The proposed DIEKF algorithm with known target motion is summarized in Tabel I.

B. Unknown Target Motion

Suppose that we do not know the motion of the target. Then, the target's angular velocity ω^k and linear acceleration a^k will be additional states to be estimated. For agent i, its prior and posterior estimates of the target state at time t_k is given by, respectively, $\bar{\mathsf{X}}_{\text{ext},i}^k = (\bar{\mathsf{X}}^k, \bar{\omega}^k, \bar{a}^k)$ and $\hat{\mathsf{X}}_{\text{ext},i}^k = (\hat{\mathsf{X}}^k, \hat{\omega}^k, \hat{a}^k)$. The prior estimate's extended error state in $\mathrm{se}_3(3) \times \mathbb{R}^6$ is defined as $\xi_{\text{ext},i}^{k|k-1} = [(\xi_i^{k|k-1})^\top, (\omega^k - \bar{\omega}_i^k)^\top, (a^k - \bar{a}_i^k)^\top]^\top$, where $\exp(\xi_i^{k|k-1}) = \mathsf{X}^k(\bar{\mathsf{X}}_i^k)^{-1}$. The posterior estimate's extended error state in $\mathrm{se}_3(3) \times \mathbb{R}^6$ is defined similarly as $\xi_{\text{ext},i}^{k|k} = [(\xi_i^{k|k})^\top, (\omega^k - \hat{\omega}_i^k)^\top, (a^k - \hat{a}_i^k)^\top]^\top$, where $\exp(\xi_i^{k|k}) = \mathsf{X}^k(\hat{\mathsf{X}}_i^k)^{-1}$. Let the covariances associated with $\xi_{\text{ext},i}^{k|k-1}$ and $\xi_{\text{ext},i}^{k|k}$ be denoted as, respectively, $\bar{P}_i^k$ and $\hat{P}_i^k$.

The first step is to propagate the target state. The standard propagation steps are included in Appendix A.

The second step is to compute the intermediate estimation pair $(\breve{\mathsf{X}}_{\text{ext},i}^k, \breve{P}_i^k)$. By following a similar procedure in Section III-A, we have

$$\begin{aligned}\breve{P}_i^k &= \left[\sum_{j\in\mathcal{N}_i^k} \pi_j^k \left(\bar{P}_j^k\right)^{-1}\right]^{-1}\\ e_i^k &= \left[\left(\log\left(\left(\bar{\mathsf{X}}_j^k\right)\left(\bar{\mathsf{X}}_i^k\right)^{-1}\right)\right)^\top, \left(\bar{\omega}_j^k - \bar{\omega}_i^k\right)^\top, \right.\\ &\qquad \left.\left(\bar{a}_j^k - \bar{a}_i^k\right)^\top\right]^\top\\ \xi_{\text{ext},i}^{k|k-1} &= \breve{P}_i^k\left[\sum_{j\in\mathcal{N}_i^k} \pi_j^k \left(\bar{P}_j^k\right)^{-1} e_i^k\right]\\ \breve{\mathsf{X}}_i^k &= \exp\left(\xi_{\text{ext},i,l}^{k|k-1}\right)\bar{\mathsf{X}}_i^k\\ \breve{\omega}_i^k &= \bar{\omega}_i^k + \xi_{\text{ext},i,w}^{k|k-1}\\ \breve{a}_i^k &= \bar{a}_i^k + \xi_{\text{ext},i,a}^{k|k-1}\end{aligned}$$

where $\xi_{\text{ext},i}^{k|k-1}$ can be written as $[(\xi_{\text{ext},i,l}^{k|k-1})^\top, (\xi_{\text{ext},i,w}^{k|k-1})^\top, (\xi_{\text{ext},i,a}^{k|k-1})^\top]^\top$ with $\xi_{\text{ext},i,l}^{k|k-1} \in \mathbb{R}^{12}$, $\xi_{\text{ext},i,w}^{k|k-1} \in \mathbb{R}^3$, and $\xi_{\text{ext},i,a}^{k|k-1} \in \mathbb{R}^3$.

The third step is to fuse the intermediate estimate with measurements. We define ${}^{C_j}\breve{p}_i^k$ and ${}^{C_j}\breve{p}_{f_i}^k$ the same as (16) and (17). The corresponding Jacobian for the representative feature H_j^k is calculated as

$$H_j^k = \mathbf{H}_p\left({}^{C_j}\breve{p}_i^k\right)\left({}_G^{C_j}R\right)\left[-\left({}^G\breve{p}_i^k\right)_\times, \mathbf{0}_3, \mathbf{I}_3, \mathbf{0}_3, \mathbf{0}_3, \mathbf{0}_3\right]$$

and the Jacobian for the nonrepresentative feature $H_{f_j}^k$ is calculated as

$$H_{f_j}^k = \mathbf{H}_p\left({}^{C_j}\breve{p}_{f_i}^k\right)\left({}_G^{C_j}R\right)\left[M_i^k, \mathbf{0}_3, \mathbf{I}_3, {}_T^G\breve{R}_i^k, \mathbf{0}_3, \mathbf{0}_3\right]$$

where M_i^k is defined in (18).

In the last step, the same equations from (19) and (20) with $\mathbf{I}_{12}$ replaced with $\mathbf{I}_{18}$ are used to calculate K_i^k, $\hat{\varepsilon}_i^k$, and $\hat{P}_i^k$. Here $\hat{\varepsilon}_i^k = [(\hat{\varepsilon}_{i,l}^k)^\top, (\hat{\varepsilon}_{i,w}^k)^\top, (\hat{\varepsilon}_{i,a}^k)^\top]^\top$ with $\hat{\varepsilon}_{i,l}^k \in \mathbb{R}^{12}$, $\hat{\varepsilon}_{i,w}^k \in \mathbb{R}^3$, and $\hat{\varepsilon}_{i,a}^k \in \mathbb{R}^3$. To calculate the posterior estimate $\hat{\mathsf{X}}_{\text{ext},i}^k = (\hat{\mathsf{X}}_i^k, \hat{\omega}_i^k, \hat{a}_i^k)$, the equations from (21) are adjusted as

$$\begin{aligned}\hat{\mathsf{X}}_i^k &= \exp\left(\hat{\varepsilon}_{i,l}^k\right)\breve{\mathsf{X}}_i^k\\ \hat{\omega}_i^k &= \breve{\omega}_i^k + \hat{\varepsilon}_{i,w}^k\\ \hat{a}_i^k &= \breve{a}_i^k + \hat{\varepsilon}_{i,a}^k.\end{aligned}$$

The DIEKF algorithm with unknown target motion is adapted from that with known target motion and can be summarized analogously by adapting Table I (hence omitted here).

IV. SIMULATION RESULTS

We apply the DIEKF algorithm to solve the distributed state estimation problem where ten fixed agents equipped with cameras are employed to track the 3-D motion of a drone. The positions of the agents are known and the target's state is to be estimated. A nonrepresentative point is created in addition to the center feature point of the target. The position of the nonrepresentative point in the target frame is unknown but is fixed. In this simulation, each camera has a resolution of $[752, 480]$ and is assumed to have a noise of 1 pixel. The maximum sensing depth of the cameras is all set to be 5 m. Based on the positions of the cameras and the drone, the status of which the agent can sense the drone is shown in Fig. 1. It is clear that each one of the agents is not able to see the target for a long period of time. In our framework, the agents communicate every time when they receive a measurement (e.g., 10 Hz in this camera measurement case), but the communication frequency can

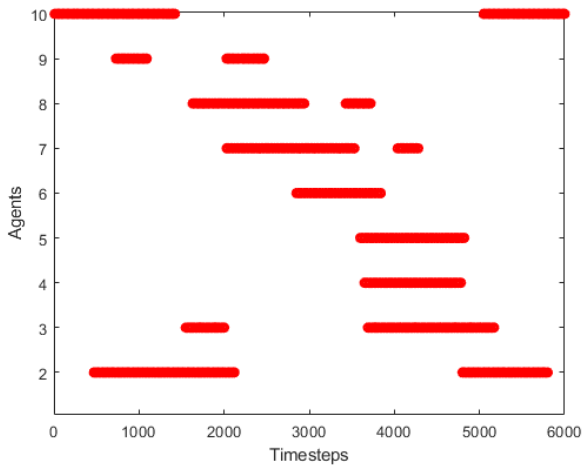


Fig. 1. Target's visibility to the agents. The red lines indicate the timesteps when the agent can directly observe the target's representative and nonrepresentative features.

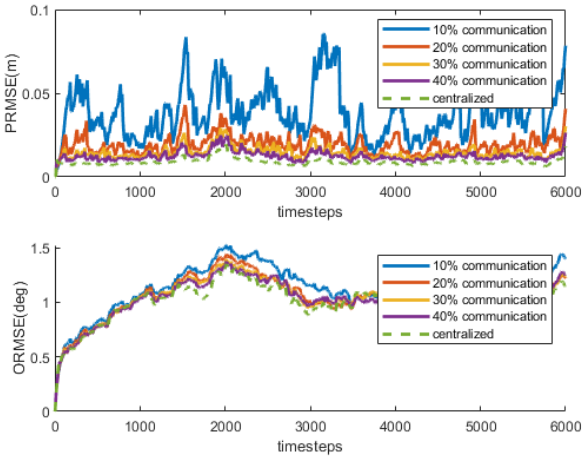


Fig. 2. Average PRMSE and ORMSE with known target motion using DIEKF at different communication rates without initialization errors.

certainly be reduced as the communication graph is allowed to be time-varying. Extensive Monte-Carlo simulations are performed to validate the algorithm. The results are quantified by rooted mean square error (RMSE) which evaluates the accuracy, and normalized estimation error squared (NEES) which evaluates the consistency. The comparison to the results of our previous QDEKF algorithm [7] is also included. See Appendix B for a brief introduction to the QDEKF. For a fair comparison, the initial covariance of the error state in DIEKF $\bar{P}_i^0$ is converted from the initial covariance of QDEKF $\bar{P}_{i,q}^0$ by $\bar{P}_i^0 \approx B\bar{P}_{i,q}^0 B^\top$, where

$$B = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \left({}^G v_i^0\right)_\times & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \left({}^G p_i^0\right)_\times & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \left({}^T p_{f_i}^0\right)_\times & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix}. \tag{22}$$

In addition, centralized implementations of both DIEKF and QDEKF, where a central station collects all data from each agent, are included as baselines.

A. Results for Known Target Motion

We set the noise of the linear acceleration ω and angular velocity a to be white Gaussian noise with standard deviations of 6.6968×10^{-3} rad/(s$\sqrt{\text{Hz}}$) and 2×10^{-2} m/(s$^2\sqrt{\text{Hz}}$), respectively. To show the result of cooperative tracking, we define a communication rate, which means that each agent has a certain probability to communicate with other agents. For instance, 20% communication means that each agent has 20% probability to communicate with each one of the other agents. Hence the set of communication neighbors are randomly determined at every time step.

In the first test, we assume that the agents have knowledge of the target's ground truth state at the first timestep. So we initialize our estimator with the ground truth and give it a very small initial covariance. We conduct 50 Monte-Carlo simulations and calculate the average result of the 50 runs among agents. Fig. 2 shows the averaged position RMSE (PRMSE) and averaged orientation RMSE (ORMSE) for different percentages of the communication as well as the centralized scenario. A comparison with the QDEKF algorithm in our previous work [7] is shown in Table II. The last column shows the centralized results. It is clear that DIEKF outperforms the QDEKF in estimating both position and orientation. In addition, the estimation accuracy improves as the communication rate increases and the centralized case performs better than the distributed cases as expected.

TABLE II

AVERAGE RMSE FOR THE ESTIMATED TARGET POSE OVER 50 MONTE-CARLO RUNS AND ALL TIMESTEPS

communication rate		10 %	20 %	30 %	40%	cen
QDEKF	PRMSE (m)	0.088	0.035	0.021	0.016	0.009
	ORMSE (deg)	1.410	1.229	1.168	1.131	1.029
DIEKF	PRMSE (m)	0.039	0.019	0.014	0.012	0.009
	ORMSE (deg)	1.152	1.077	1.080	1.061	1.060

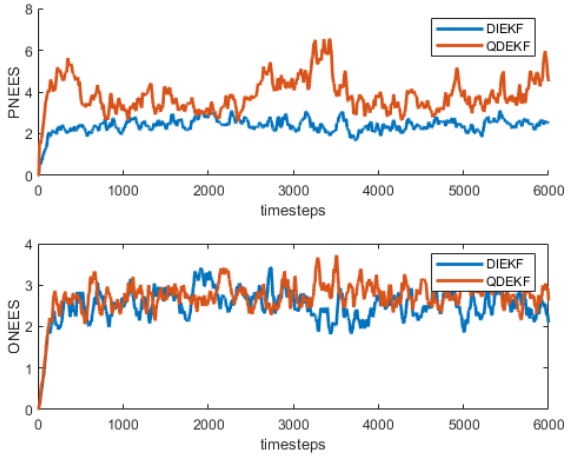


Fig. 3. Average PNEES and ONEES with known target motion using DIEKF and QDEKF at 40% communication rate without initialization errors.

We further show the NEES result at 40% communication rate. As shown in Fig. 3, while the orientation NEES (ONEES) appears to be similar, the position NEES (PNEES) for DIEKF is closer to 3 as compared to the QDEKF algorithm. This indicates the improvement of the consistency [17].

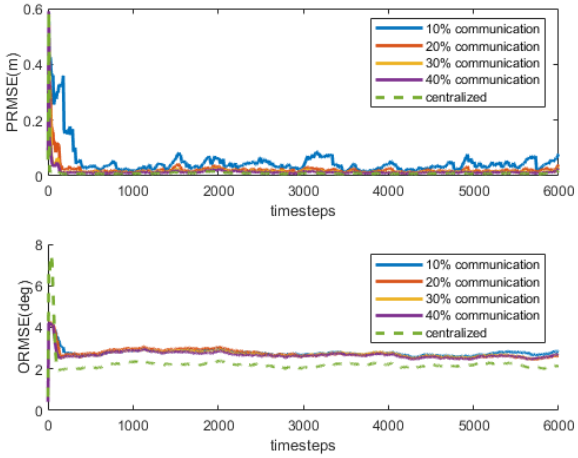


Fig. 4. Average PRMSE and ORMSE with known target motion using DIEKF at different communication rates with initialization errors.

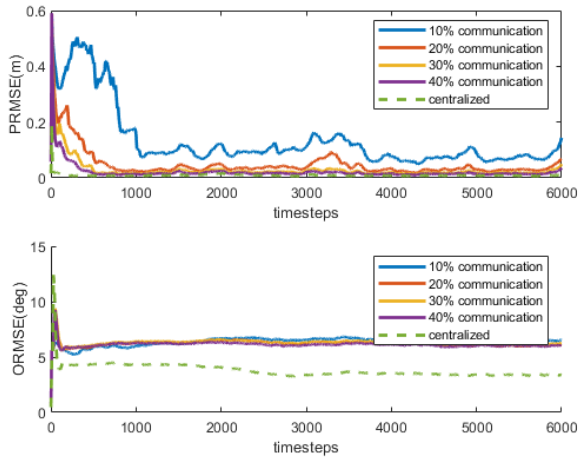


Fig. 5. Average PRMSE and ORMSE with known target motion using QDEKF at different communication rates with initialization errors.

In the second test, we assume that the agents do not have perfect knowledge of the target's ground truth state at the first timestep. We initialize our state estimator to a value near the ground truth and give it a larger covariance. Similarly, we conduct 50 Monte-Carlo simulations on the DIEFK and QDEKF with the same initialized state estimate and equivalent covariance. Figs. 4 and 5 show the result for both DIEKF and QDEKF using the same initialization, measurements, and noise. Compared with the results of QDEKF, our DIEKF algorithm obviously converge faster in the position estimation, and is more accurate in general.

Overall, if the target motion is known, our algorithm can accurately track the trajectory of the target in the 3-D space and can maintain consistency.

B. Results for Unknown Target Motion

In the case where the target motion is unknown, we select, respectively, 1×10^{-2} rad/(s$\sqrt{\text{Hz}}$) and 1.2×10^{-1} m/(s$^2\sqrt{\text{Hz}}$), as the standard deviations associated with n_ω and n_a in (4). It is clear from Figs. 6 and 7 that even if the target motion is unknown, both algorithms can still track the target position well, while the orientation error is larger. The DIEKF has

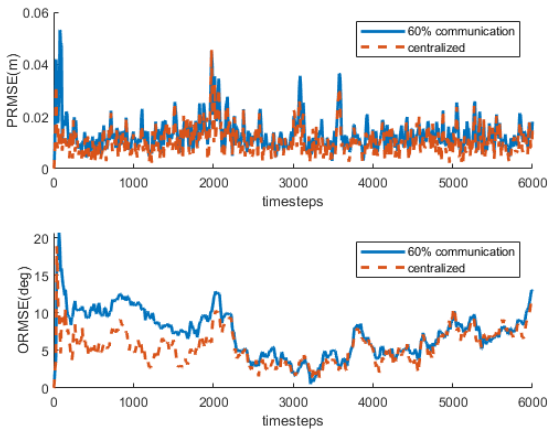


Fig. 6. Average PRMSE and ORMSE using DIEKF with unknown target motion without initialization errors.

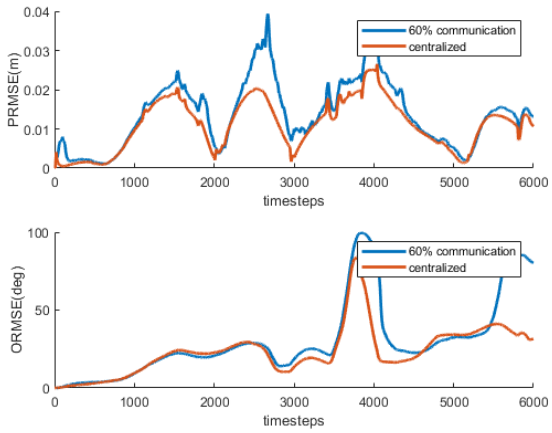


Fig. 7. Average PRMSE and ORMSE using QDEKF with unknown target motion without initialization errors.

a better performance, especially for the orientation estimate. The larger estimation errors compared to the case where the target motion is known might be due to the inaccurate model applied.

V. Experiments

In this section, both the QDEKF algorithm and the DIEKF algorithm are tested to track a quadrotor by using multiple UWB anchor receivers. Infrared markers are attached to the quadrotor for obtaining the ground truth data through the VICON system. The effectiveness of the two algorithms is verified with the measurement data obtained from the UWB receivers. The performances of the two algorithms are also compared. Centralized implementations of both DIEKF and QDEKF are included as baselines. As in Section IV, both the cases that the target motion is known or unknown are considered.

The experiment is conducted within a 4 × 3 × 2 m indoor space as shown in Fig. 8. A Crazyflie 2.1 quadrotor is used as the target of interest. The quadrotor flies following a predesigned figure "8" trajectory in addition to taking off and landing. Eight UWB anchor receivers, named as agents 1–8, are placed in the environment. The positions of the UWB receivers in the world coordinate frame are shown in Table III.

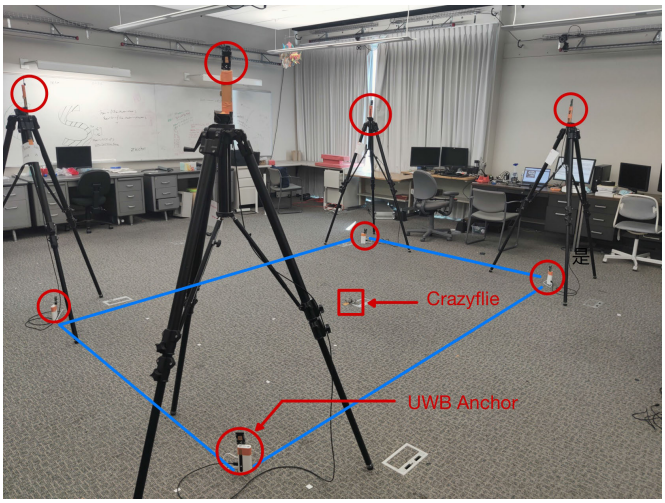


Fig. 8. Experimental environment.

TABLE III
UWB ANCHOR POSITIONS

ID	x [m]	y [m]	z [m]
0	-1.83	-1.34	0.20
1	-1.88	1.10	1.73
2	2.02	1.10	0.20
3	1.98	-1.32	1.73
4	-1.83	1.34	1.73
5	-1.87	1.11	0.20
6	2.02	1.10	1.73
7	1.98	-1.32	0.20

Each UWB receiver is set to two-way ranging (TWR) mode to obtain the distance measurements between itself and the target. The measurements for every agent are tested to be subject to a zero mean white Gaussian process noise with a standard deviation of 0.15 m. The communications among agents are randomly assigned according to the communication rates.

In our experiment, only the representative feature on the target is used and there is no nonrepresentative feature point. The initial state estimate of both DIEKF and QDEKF is set according to the ground truth. The initial covariance corresponding to the error state in QDEKF is set to be

$$\bar{P}_{i,q}^{0} = \operatorname{diag}\Big(0.15^2, 0.15^2, 0.20^2, 10^{-4}, 10^{-4}, 10^{-4}, 10^{-4}, 10^{-4}, 10^{-4}\Big).$$

For fair comparisons, the initial covariance of the error state $\bar{P}_i^0$ in DIEKF is converted according to $\bar{P}_i^0 \approx B_s \bar{P}_{i,q}^0 B_s^\top$, where B_s is changed from B in (22) by removing the last row and column. The calculated Jocabians in Section III are also changed accordingly by removing the corresponding line and column which corresponds to the nonrepresentative feature point.

In the case that the target's motion is known, the built-in IMU in the quadrotor is used to obtain the motion of the target. The standard deviation of the process noise for the accelerometer n_a and the gyroscope n_ω of the x–y–z axes are, respectively, $n_a = [0.02, 0.02, 0.05]^\top$ m/s$^2/\sqrt{\text{Hz}}$ and $n_\omega = [0.005, 0.005, 0.015]^\top$ deg/s$/\sqrt{\text{Hz}}$. All the data, including IMU, UWB measurements, and ground truth are collected at the frequency of 100 Hz. Figs. 9 and 10 show the ground truth trajectory of the target and the estimated trajectory of the target by agent 1 for, respectively, QDEKF and DIEKF at 40% communication rate. To further illustrate the result, the PRMSE and ORMSE of agents 1–3 for the two algorithms at 40% communication rate are shown in Figs. 11 and 12. It can be observed from Figs. 9 to 12 that the trajectory of the target can be well estimated by both algorithms with DIEKF achieving better performance. Although the PRMSE of DIEKF has a little bit larger overshoot than that of QDEKF at the very beginning, DIEKF has a better performance over time as seen in the overall trajectory. For further comparison, the resulting average RMSE for QDEKF and DIEKF in different communication rates are shown in Table IV. The last column shows the centralized results for both algorithms. It is obvious that DIEKF outperforms QDEKF in all the cases in our experiment.

We further apply the DIEKF to the case that the target's motion is unknown. According to the generic model (4), the angular velocity ω and linear acceleration a are assumed to progress as random walks driven by Gaussian noises. We select, respectively, $[0.005, 0.005, 0.01]^\top$ deg/s$/\sqrt{\text{Hz}}$ and $[0.4, 0.4, 0.6]^\top$ m/s$^2/\sqrt{\text{Hz}}$, as the standard deviations for the x–y–z axes associated with n_ω and n_a in (4). The initial

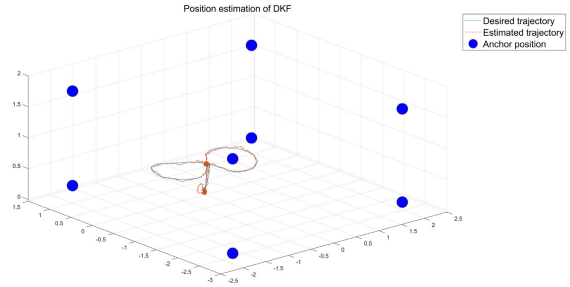


Fig. 9. Estimated trajectory using QDEKF and ground truth with known target motion in the experiment.

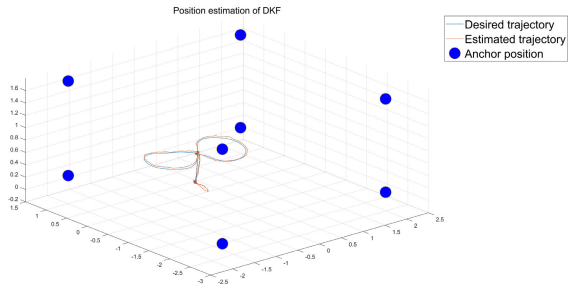


Fig. 10. Estimated trajectory using DIEKF and ground truth with known target motion in the experiment.

TABLE IV
AVERAGE RMSES FOR THE POSE ESTIMATION AMONG EIGHT AGENTS IN EXPERIMENT

Communication rate		10%	20%	30%	40%	cen
QDEKF	PRMSE (m)	0.063	0.050	0.044	0.039	0.030
	ORMSE (deg)	12.597	11.897	9.918	9.764	9.720
DIEKF	PRMSE (m)	0.058	0.043	0.037	0.034	0.028
	ORMSE (deg)	6.901	7.200	6.963	6.515	6.340

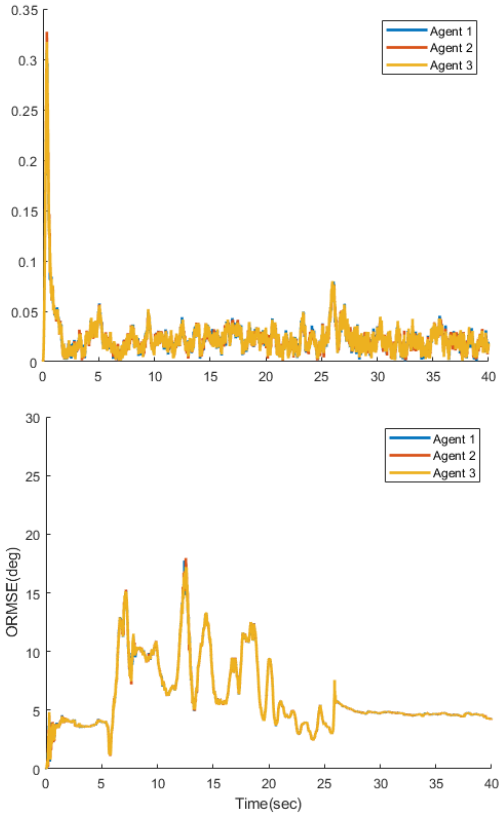


Fig. 11. PRMSE and ORMSE using DIEKF with known target motion in the experiment.

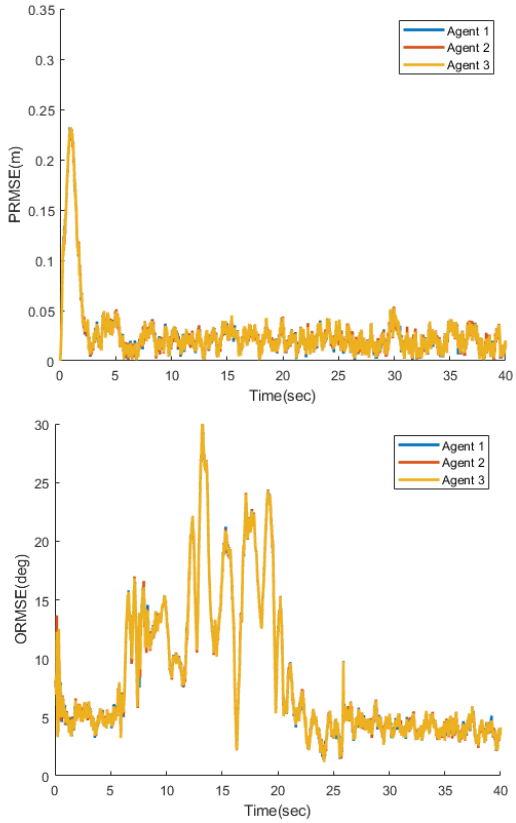


Fig. 12. PRMSE and ORMSE using QDEKF with known target motion in the experiment.

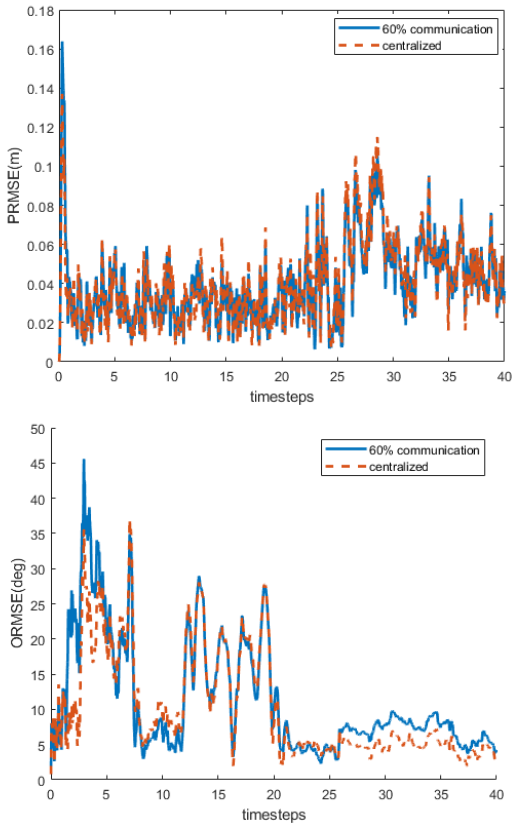


Fig. 13. PRMSE and ORMSE using DIEKF with unknown target motion in the experiment.

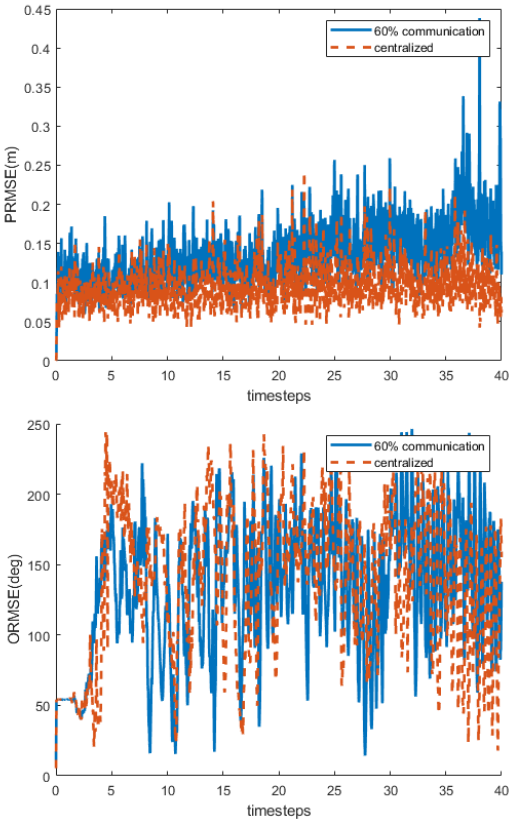


Fig. 14. PRMSE and ORMSE using QDEKF with unknown target motion in the experiment.

estimates of ω and a are both set to $\mathbf{0}_{3\times1}$ at the initialization. Figs. 13 and 14 show, respectively, the RMSE of the estimated pose of agent 1 using DIEKF and QDEKF. The results for the other agents are similar to that of agent 1 and hence not

shown here. As observed in Figs. 13 and 14, even though the target's motion is unknown, the position of the target can be well tracked. While DIEKF can still track the target's orientation with larger errors than the known target motion case, QDEKF fails to track the target's orientation. DIEKF achieves comparable performance compared to its centralized counterpart, and the performance of DIEKF is better than QDEKF.

VI. Conclusion and Future Work

In this article, by using the matrix Lie group representation of the state, we introduced a new DIEKF algorithm that yields consistent and accurate estimates of the target in the 3-D space over the sensor networks given the target's motion. We also include the case that the target's motion is unknown, the algorithm can still track the position of the target accurately. The proposed algorithm requires only one communication iteration with its communication neighbors at every time instant. Furthermore, the algorithm is shown to be robust to changing communication topology and blind agents. These properties ensure that our approach can have a wide application in multiagent scenarios. The performance of the proposed algorithm is tested via both Monte-Carlo simulations and experiments. Some key performance measures are compared with the algorithm that did not use the Lie group representation for the state. In this article, all derivations of the DIEKF are based on first-order approximations for error propagation on the matrix Lie group by ignoring higher-order terms. In future work, an interesting direction is to consider second-order terms for error propagation to further improve the estimation accuracy.

Appendix A
Discretized Propagation

In the case where the target motion is known, we can discretize (3) without process noises to obtain the state estimate at time t_k, denoted as $\bar{\mathrm{X}}_i^k$. According to [12], the individual states in $\bar{\mathrm{X}}_i^k$ can be propagated from $\hat{\mathrm{X}}_i^{k-1}$ as

$$\begin{aligned}
{}^G_T\bar{R}_i^k &= {}^G_T\hat{R}_i^{k-1}\Gamma_0\left(\omega^{k-1}\Delta t\right)\\
{}^G\bar{v}_i^k &= {}^G\hat{v}_i^{k-1} + {}^G_T\hat{R}_i^{k-1}\Gamma_1\left(\omega^{k-1}\Delta t\right)a^{k-1}\Delta t + g\Delta t\\
{}^G\bar{p}_i^k &= {}^G\hat{p}_i^{k-1} + {}^G\hat{v}_i^{k-1}\Delta t\\
&\quad + {}^G_T\hat{R}_i^{k-1}\Gamma_2\left(\omega^{k-1}\Delta t\right)a^{k-1}(\Delta t)^2 + \frac{1}{2}g(\Delta t)^2\\
{}^T\bar{p}_{f_i}{}^k &= {}^T\hat{p}_{f_i}{}^{k-1}
\end{aligned} \tag{23}$$

where $\Gamma_m(\phi)$ is defined as [18]

$$\Gamma_m(\phi) = \left(\sum_{n=0}^{\infty}\frac{1}{(n+m)!}(\phi)_\times^n\right)$$

ω^{k-1} and a^{k-1} denote, respectively, the target's angular velocity and acceleration at time t_{k-1}, and $\Delta t = t_k - t_{k-1}$. To propagate the covariance, the state transition matrix $\Phi(t_k, t_{k-1})$, associated with A_t defined in (10), is computed as

$$\Phi(t_k, t_{k-1}) = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3\\ (g)_\times\Delta t & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3\\ \mathbf{0}_3 & \mathbf{I}_3\Delta t & \mathbf{I}_3 & \mathbf{0}_3\\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix}.$$

Then, the covariance $\bar{P}_i^k$ associated with $\xi_i^{k|k-1}$ can be propagated from $\hat{P}_i^{k-1}$ as

$$\begin{aligned}
Q_i^{k-1} &= \mathrm{Ad}_{\hat{\mathrm{X}}_i^{k-1}}\mathrm{cov}\left(\mathsf{U}^{k-1}\right)\left(\mathrm{Ad}_{\hat{\mathrm{X}}_i^{k-1}}\right)^\top\\
Q_d^{k-1} &= \Phi(t_k, t_{k-1})\,Q_i^{k-1}\Phi(t_k, t_{k-1})\,\Delta t\\
\bar{P}_i^k &= \Phi(t_k, t_{k-1})\,\hat{P}_i^{k-1}\Phi(t_k, t_{k-1}) + Q_d^{k-1}
\end{aligned} \tag{24}$$

where U^{k-1} denotes U_t defined after (10) at time t_{k-1}, and $\mathrm{cov}(\cdot)$ denotes the covariance.

In the case where the target motion is unknown, in addition to (23) with ω^{k-1} and a^{k-1} replaced with $\hat{\omega}^{k-1}$ and $\hat{a}^{k-1}$, the posterior estimates of ω^{k-1} and a^{k-1} from the previous time t_{k-1}, new equations are introduced as

$$\begin{aligned}
\bar{\omega}_i^k &= \hat{\omega}_i^{k-1}\\
\bar{a}_i^k &= \hat{a}_i^{k-1}.
\end{aligned}$$

According to [12], the state transition matrix for agent i associated with A_t defined in (11) is given as

$$\Phi_i(t_k, t_{k-1}) = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & -\beta_{15} & \mathbf{0}_3\\ (g)_\times\Delta t & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & -\beta_{25} & -\beta_{26}\\ \mathbf{0}_3 & \mathbf{I}_3\Delta t & \mathbf{I}_3 & \mathbf{0}_3 & -\beta_{35} & -\beta_{36}\\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & -\beta_{45} & \mathbf{0}_3\\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3\\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix}$$

where the individual terms are computed as

$$\begin{aligned}
\Psi_1 &= \int_{t_{k-1}}^{t_k}\left(\Gamma_0\left(\hat{\omega}_i^{k-1}t\right)\hat{a}_i^{k-1}\right)_\times\Gamma_1\left(\hat{\omega}_i^{k-1}t\right)t\,dt\\
\phi_l &= \Gamma_0^\top\left(\hat{\omega}_i^{k-1}\Delta t\right)\Psi_1\\
\Psi_2 &= \int_{t_{k-1}}^{t_k}\Gamma_0\left(\hat{\omega}_i^{k-1}t\right)\phi_l\,dt\\
\beta_{15} &= -\left({}^G_T\hat{R}_i^{k-1}\right)\Gamma_1\left(\hat{\omega}_i^{k-1}\Delta t\right)\Delta t\\
\beta_{25} &= -\left({}^G\bar{v}_i^k\right)_\times\left({}^G_T\hat{R}_i^{k-1}\right)\Gamma_1\left(\hat{\omega}_i^{k-1}\Delta t\right)\Delta t + \left({}^G_T\hat{R}_i^{k-1}\right)\Psi_1\\
\beta_{35} &= -\left({}^G\bar{p}_i^k\right)_\times\left({}^G_T\hat{R}_i^{k-1}\right)\Gamma_1\left(\hat{\omega}_i^{k-1}\Delta t\right)\Delta t + \left({}^G_T\hat{R}_i^{k-1}\right)\Psi_2\\
\beta_{45} &= -\left({}^T\bar{p}_{f_i}^k\right)_\times\left({}^G_T\hat{R}_i^{k-1}\right)\Gamma_1\left(\hat{\omega}_i^{k-1}\Delta t\right)\Delta t\\
\beta_{26} &= -\left({}^G_T\hat{R}_i^{k-1}\right)\Gamma_1\left(\hat{\omega}_i^{k-1}\Delta t\right)\Delta t\\
\beta_{36} &= -\left({}^G_T\hat{R}_i^{k-1}\right)\Gamma_2\left(\hat{\omega}_i^{k-1}\Delta t\right)\Delta t^2.
\end{aligned}$$

Then, the covariance $\bar{P}_i^k$ can be propagated according to (24) with $\mathrm{Ad}_{\hat{\mathrm{X}}_i^{k-1}}$ removed and U^{k-1} denoting U_t defined after (11).

APPENDIX B
QDEKF

For completeness, we include the QDEKF from our previous work [7]. Let $\mathsf{X}_q^k = [{}_G^T\bar{q}^{k\top}\ {}^G v^{k\top}\ {}^G p^{k\top}\ {}^T p_f^{k\top}]^\top$ denote the state of the target at time t_k in the QDEKF algorithm which uses quaternion as the representation of the orientation. The QDEKF represents the orientation, velocity, position, and nonrepresentative point separately ($\mathrm{SO}(3) \times \mathbb{R}^9$). The orientation, velocity, position, and nonrepresentative point estimation errors are decoupled. The orientation error vector is defined in the Lie algebra of SO(3) while the velocity error, position error, and nonrepresentative point error are defined as the arithmetic difference between the true and estimated values. In contrast to DIEKF, where the error vector is defined in the Lie algebra of $\mathrm{SE}_3(3)$, the resulting matrix A_t in the error dynamics in the QDEKF depends on the estimated state (linearization point). Let $\bar{\mathsf{X}}_{i,q}^k$ and $\hat{\mathsf{X}}_{i,q}^k$ denote, respectively, the prior and posterior estimate of X_q^k by agent i. Let $\bar{P}_{i,q}^k$ and $\hat{P}_{i,q}^k$ denote, respectively, the corresponding prior and posterior covariance. Let $\mathsf{X}_{v,q}^k = [{}^G v^{k\top}\ {}^G p^{k\top}\ {}^T p_f^{k\top}]^\top$ denote the vector quantities in X_q^k excluding the quaternion. Let $\bar{\mathsf{X}}_{v,i,q}^k$ and $\hat{\mathsf{X}}_{v,i,q}$ denote the prior and posterior estimates of $X_{v,q}^k$ by agent i. Suppose that at time t_k, each agent maintains a prior estimator $(\bar{\mathsf{X}}_{i,q}^k, \bar{P}_{i,q}^k)$ after standard propagation. Now agent i aims to update its local estimator $(\bar{\mathsf{X}}_{i,q}^k, \bar{P}_{i,q}^k)$ by using information from it one-hop communication neighbors. The first step is to fuse all the prior estimation pairs. We synchronize the prior estimation pairs to reduce their uncertainty in a weighted manner. The weight $\pi_{j,q}$ satisfies $\pi_{j,q} \in [0, 1]$ and $\sum_{j\in\mathcal{N}_i^k} \pi_{j,q} = 1$. For the estimation of $\mathsf{X}_{v,q}^k$, we compute

$$\check{\mathsf{X}}_{v,i,q} = \sum_{j\in\mathcal{N}_i^k} \pi_{j,q}^k \bar{\mathsf{X}}_{v,j,q}^k.$$

To average the orientations represented by the quaternions, we employ the method in [8] to provide a closed-form solution of averaged quaternion ${}_G^T\check{\bar{q}}_i^k$ by the following maximization procedure:

$$\begin{aligned} {}_G^T\check{\bar{q}}_i^k &= \arg\max_{\bar{q}\in\mathcal{S}^3} \bar{q}^\top \mathbf{Q}\bar{q} \\ \mathbf{Q} &= \sum_{j\in\mathcal{N}_i^k} \pi_{j,q}^k \left({}_G^T\bar{\bar{q}}_j^k\right)^\top {}_G^T\bar{\bar{q}}_j^k \end{aligned} \tag{25}$$

where ${}_G^T\bar{\bar{q}}_j^k$ is agent j's prior estimation of ${}_G^T\bar{q}^k$, and $\mathcal{S}^3$ denotes the unit 3-sphere. Solving (25) in fact gives a quaternion that minimizes the weighted sum of the orientation errors. We define a compatible symbol $\boxtimes$ for computing the weighted average and then we obtain

$$\check{\mathsf{X}}_{i,q}^k = \begin{bmatrix} {}_G^T\check{\bar{q}}_i^k \\ \check{\mathsf{X}}_{v,i,q}^k \end{bmatrix} = \sum_{j\in\mathcal{N}_i^k} \pi_{j,q}^k \boxtimes \bar{\mathsf{X}}_{j,q}^k.$$

For the covariance, it can be directly computed as $\check{P}_{i,q}^k = \sum_{j\in\mathcal{N}_i^k} \pi_j^k \bar{P}_{j,q}^k$, since the quaternion error is represented by the error of the rotational angle that is a vector quantity. The weight π_j^k is chosen to minimize the determinant or the trace of $\bar{P}_{i,q}^k$.

The second step is to fuse the intermediate estimation pair $(\check{\mathsf{X}}_{i,q}^k, \check{P}_{i,q}^k)$ with all the local measurements z_j^k, $\forall j \in \mathcal{N}_i^k$. Let $z_i^k = h_i(\mathsf{X}_q^k) + w_i^k$ be the measurement model, and let V_i^k denote the covariance of the measurement noise w_i^k. After linearization of z_i^k about the current estimated state, we compute

$$\begin{aligned} s_i^k &= \left(H_{i,q}^k\right)^\top \left(V_i^k\right)^{-1} H_{i,q}^k \\ y_i^k &= \left(H_{i,q}^k\right)^\top \left(V_i^k\right)^{-1} \tilde{z}_i^k \end{aligned}$$

where $\tilde{z}_i^k = z_i^k - h_i(\check{\mathsf{X}}_{i,q}^k)$ and $H_{i,q}^k = (\partial h_i/\partial \mathsf{X}_i)(\check{\mathsf{X}}_{i,q}^k)$. Then, the updated covariance $\hat{P}_{i,q}^k$ and the state correction $\delta\mathsf{X}_{i,q}^k$ is calculated as

$$\begin{aligned} \hat{P}_{i,q}^k &= \left[\left(\check{P}_{i,q}^k\right)^{-1} + \sum_{j\in\mathcal{N}_i^k} s_j^k\right]^{-1} \\ \delta\mathsf{X}_{i,q}^k &= \begin{bmatrix} \delta\theta_i^k \\ \delta\mathsf{X}_{v,i,q}^k \end{bmatrix} = \hat{P}_{i,q}^k \sum_{j\in\mathcal{N}_i^k} y_j^k \end{aligned}$$

where $\delta\theta_i^k$ is the correction for the orientation, and $\delta\mathsf{X}_{v,i,q}^k$ is the correction for the vector quantities. Next, we update $\hat{\mathsf{X}}_{i,q}$ by using $\delta\mathsf{X}_{i,q}$. For the vector quantities $\mathsf{X}_{v,i,q}$, it is calculated by $\hat{\mathsf{X}}_{v,i,q}^k = \check{\mathsf{X}}_{v,i,q}^k + \delta\mathsf{X}_{v,i,q}^k$. For the quaternion, it is updated according to

$${}_G^T\hat{\bar{q}}_i^k = {}_G^T\check{\bar{q}}_i^k \otimes \delta\bar{q}_i$$

where $\otimes$ represents the quaternion multiplication, and

$$\delta\bar{q}_i = \frac{1}{\sqrt{1 + \frac{1}{4}\delta\theta_i^{k\top}\delta\theta_i^k}} \begin{bmatrix} \frac{1}{2}\delta\theta_i^k \\ 1 \end{bmatrix}$$

is a unit quaternion. A compatible symbol $\boxplus$ is defined for updating all the states as

$$\hat{\mathsf{X}}_{i,q}^k = \begin{bmatrix} {}_G^T\hat{\bar{q}}_i^k \\ \hat{\mathsf{X}}_{v,i,q}^k \end{bmatrix} = \check{\mathsf{X}}_{i,q}^k \boxplus \delta\mathsf{X}_{i,q}^k.$$

REFERENCES

[1] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Syst. Mag.*, vol. 27, no. 2, pp. 71–82, Apr. 2007.

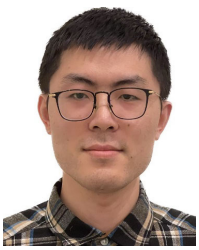
[2] S. J. Julier and J. K. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," in *Proc. Amer. Control Conf.*, Jun. 1997, pp. 2369–2373.

[3] G. Battistelli, L. Chisci, G. Mugnai, A. Farina, and A. Graziano, "Consensus-based linear and nonlinear filtering," *IEEE Trans. Autom. Control*, vol. 60, no. 5, pp. 1410–1415, May 2015.

[4] G. Battistelli and L. Chisci, "Kullback–Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability," *Automatica*, vol. 50, no. 3, pp. 707–718, Mar. 2014.

[5] X. He, W. Xue, and H. Fang, "Consistent distributed state estimation with global observability over sensor network," *Automatica*, vol. 92, pp. 162–172, Jun. 2018.

[6] S. Wang and W. Ren, "On the convergence conditions of distributed dynamic state estimation using sensor networks: A unified framework," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 4, pp. 1300–1316, Jul. 2018.
[7] P. Zhu and W. Ren, "Distributed Kalman filter for 3-D moving object tracking over sensor networks," in *Proc. 59th IEEE Conf. Decis. Control (CDC)*, Dec. 2020, pp. 2418–2423.
[8] F. L. Markley, Y. Cheng, J. L. Crassidis, and Y. Oshman, "Averaging quaternions," *J. Guid., Control, Dyn.*, vol. 30, no. 4, pp. 1193–1197, Jul. 2007.
[9] N. Trawny and S. I. Roumeliotis, "Indirect Kalman filter for 3D attitude estimation," Dept. Comput. Sci. Eng., Univ. Minnesota, Minneapolis, MN, USA, Tech. Rep., 2005.
[10] 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.
[11] 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.
[12] 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, Mar. 2020.
[13] T. Zhang, K. Wu, J. Song, S. Huang, and G. Dissanayake, "Convergence and consistency analysis for a 3-D invariant-EKF SLAM," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 733–740, Apr. 2017.
[14] J. Xu, P. Zhu, and W. Ren, "Distributed invariant extended Kalman filter for 3-D dynamic state estimation using lie groups," in *Proc. Amer. Control Conf. (ACC)*, Jun. 2022, pp. 2367–2372.
[15] W. Niehsen, "Information fusion based on fast covariance intersection filtering," in *Proc. 5th Int. Conf. Inf. Fusion (FUSION)*, Jul. 2002, pp. 901–904.
[16] T. D. Barfoot, *State Estimation for Robotics*. Cambridge, U.K.: Cambridge Univ. Press, 2017.
[17] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation With Applications to Tracking and Navigation: Theory, Algorithms and Software*. Hoboken, NJ, USA: Wiley, 2004.
[18] M. Bloesch et al., "State estimation for legged robots consistent fusion of leg kinematics and IMU," *Robotics*, vol. 17, pp. 17–24, Jul. 2013.



Jie Xu received the B.S. degree in mechanical design, manufacturing, and automation from the Huazhong University of Science and Technology, Wuhan, Hubei, China, in 2018, and the M.S. degree in mechanical engineering from the University of California at Riverside, Riverside, CA, USA, in 2020, where he is currently pursuing the Ph.D. degree in electrical engineering.

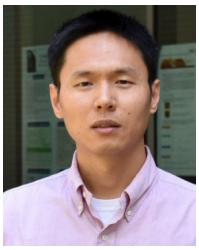
His research interests include sensor fusion, motion planning, and SLAM.

Pengxiang Zhu received the B.Eng. and M.S. degrees in control science and engineering from Harbin Engineering University, Harbin, China, in 2013 and 2016, respectively, and the Ph.D. degree in electrical engineering from the University of California at Riverside, Riverside, CA, USA, in 2022.

His current research interests include state estimation, sensor fusion, and visual-inertial navigation for autonomous robots.

Yizhi Zhou received the B.S. degree in automation from Hangzhou Dianzi University, Hangzhou, China, in 2020, and the M.S. degree in electrical engineering from the University of California Riverside, Riverside, CA, USA, in 2022. He is currently pursuing the Ph.D. degree in electrical engineering with the ECE Department, George Mason University, Fairfax, VA, USA.

His research interests include control and robotics.



Wei Ren (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Brigham Young University, Provo, UT, USA, in 2004.

He is currently a Professor with the Department of Electrical and Computer Engineering, University of California at Riverside, Riverside, CA, USA. His research focuses on distributed control of multiagent systems.

Dr. Ren is an IEEE Control Systems Society Distinguished Lecturer. He was a recipient of the IEEE Control Systems Society Antonio Ruberti Young Researcher Prize in 2017 and the National Science Foundation CAREER Award in 2008.