

Consistent Distributed Cooperative Localization: A Coordinate Transformation Approach

Chungeng Tian[†], Ning Hao[†], Fenghua He^{*}, and Haodi Yao

Abstract—This paper addresses the consistency issue of multi-robot distributed cooperative localization. We introduce a consistent distributed cooperative localization algorithm conducting state estimation in a transformed coordinate. The core idea involves a linear time-varying coordinated transformation to render the propagation Jacobian independent of the state and make it suitable for a distributed manner. This transformation is seamlessly integrated into a server-based distributed cooperative localization framework, in which each robot estimates its own state while the server maintains the cross-correlations. The transformation ensures the correct observability property of the entire framework. Moreover, the algorithm accommodates various types of robot-to-robot relative measurements, broadening its applicability. Through simulations and real-world dataset experiments, the proposed algorithm has demonstrated better performance in terms of both consistency and accuracy compared to existing algorithms.

I. INTRODUCTION

Cooperative localization (CL) is crucial for autonomous mobile robots during collaborative missions [1], which collectively estimates the robot poses in a common reference frame by utilizing proprioceptive sensors (e.g., wheel encoders or odometry) and exteroceptive sensors (e.g., cameras or laser scanners). CL is resilient to environmental variations, such as scenarios with limited GPS signals or few landmarks. Nevertheless, in certain scenarios such as underwater or large-scale environments, communication may experience delays or be intermittently unavailable. Hence, the development of a distributed cooperative localization (DCL) framework that reduces communication costs is advantageous.

In DCL, two robot pose estimations are correlated after being updated with a relative measurement between them. If the pose estimations are continuously updated without considering these correlations, the problem of *double counting* arises, leading to overconfidence in pose estimations [2].

To address this issue, a widely used approach is the covariance intersection (CI) fusion technique [3]–[6]. However, CI is overly conservative because it assumes a maximal correlation between the estimations. To mitigate this conservatism, ellipsoidal intersection (EI) [7], split covariance intersection (SCI) [8], and inverse covariance intersection (ICI) [9, 10] techniques have been proposed. Nonetheless, these methods are not suitable for handling partial observation cases, where

[†]These authors contributed equally to this work. The authors are with the National Key Laboratory of Modeling and Simulation for Complex Systems, School of Astronautics, Harbin Institute of Technology, Harbin 150001, China. (e-mail: tcghit@outlook.com, haoning0082022@163.com, hefenghua@hit.edu.cn, 20B904013@stu.hit.edu.cn)

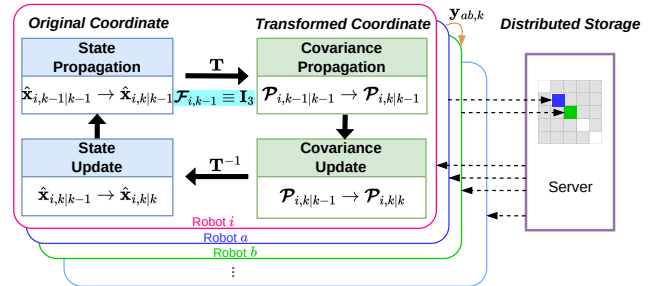


Fig. 1. Transformed server-based DCL framework. The covariance matrices corresponding to the transformed error states propagate and update in the transformed coordinates.

the dimension of robot-to-robot relative measurements is smaller than the robot state. To address partial relative measurements, decorrelated minimum variance (DMV) [11] integrate CI into the extended Kalman filter (EKF). Unlike CI class methods, a more flexible optimization framework was proposed by constructing the unknown cross-covariance matrix instead of the upper bound of the whole joint covariance matrix. The methods based on this framework, such as the game-theoretic approach [12] and the practical estimated minimum variance of cross-covariance (PECMV) [13], exhibit less conservatism but require higher computational effort.

Another method used to tackle the double-counting issue is the central-equivalent EKF-based DCL. It considers all robot poses as the system state and uses a single filter to estimate the system state, automatically recording the correlations among robot pose estimations through cross-covariance matrices. Although central-equivalent DCL achieves high localization accuracy, its naive distributed form [14] necessitates all-to-all communication for updating cross-covariance matrices. To alleviate communication costs, [15] proposed a method that only requires pairwise communication between robots that obtain relative measurements. However, this method sacrifices estimation consistency since it updates the cross-covariance matrices with block-diagonal approximation matrices. An alternative DCL algorithm was introduced in [16]. This algorithm incorporates a server responsible for managing and updating cross-covariance matrices. While this server-based framework enhances localization accuracy significantly, it also incurs substantial communication costs during the update steps. In a subsequent study [17], a technique equivalent to the Schmidt-Kalman filter is suggested to address issues related to unreliable communication links within the server-based framework. While central-equivalent

DCL and its distributed forms avoid double counting, they still suffer from inconsistency due to unjustified unobservable dimension reduction [18] [19]. Since the absolute measurement is absent, the CL system is partially observable. [20] has analytically proven that the EKF CL estimator exhibits an unobservable subspace of lower dimension than the original non-linear CL system. Consequently, the unjustified reduction of unobservable dimension leads to inconsistency and a decline in localization accuracy.

This paper focuses on enhancing the consistency of the central-equivalent DCL algorithms. We have devised a coordinate transformation that ensures the transformed propagation Jacobians are independent of the state. By incorporating this transformation into a server-based framework as shown in Figure 1, the correct observability property is guaranteed. With the assurance of correct observability, the estimator's consistency is enhanced. Furthermore, owing to the transformation, the transformed propagation Jacobian simplifies to an identity matrix, streamlining the propagation of cross-covariance matrices.

The main contributions of this paper are as follows.

- A coordinate transformation is designed in light of decomposing the propagation Jacobian. The transformation renders the propagation Jacobian independent of the state.
- A consistent DCL algorithm is presented by integrating the transformation into a server-based framework, in which the state estimation is performed based on the transformed error states.
- The proposed DCL algorithm is validated by Monte Carlo simulations and experiments. The results show that it outperforms state-of-the-art algorithms in terms of consistency and accuracy.

II. PROBLEM DESCRIPTION

Consider a team of N robots performing cooperative localization on a 2D plane in which each robot is equipped with proprioceptive sensors to sense its ego-motion information and exteroceptive sensors to obtain the robot-to-robot relative measurements.

A. Motion Model

Let $\mathbf{x}_{i,k} = [\mathbf{p}_{i,k}^\top \ \theta_{i,k}]^\top$ be the pose of robot i in a common fixed frame at time step k , where $\mathbf{p}_{i,k} \in \mathbb{R}^2$ and $\theta_{i,k} \in \mathbb{R}$ are the position and orientation of robot i , respectively, $i \in \{1, 2, \dots, N\}$. The motion of robot i is described by

$$\begin{aligned} \mathbf{x}_{i,k} &= \mathbf{x}_{i,k-1} + \begin{bmatrix} \mathbf{R}(\theta_{i,k-1}) & \mathbf{0}_{2 \times 1} \\ \mathbf{0}_{1 \times 2} & 1 \end{bmatrix} (\mathbf{u}_{i,k-1} \Delta t + \boldsymbol{\epsilon}_{i,k-1}) \\ &\triangleq \mathbf{f}_i(\mathbf{x}_{i,k-1}, \mathbf{u}_{i,k-1}, \boldsymbol{\epsilon}_{i,k-1}) \end{aligned} \quad (1)$$

where $\mathbf{u}_{i,k-1} = [\mathbf{v}_{i,k-1}^\top \ \omega_{i,k-1}]^\top$, $\mathbf{v}_{i,k-1} \in \mathbb{R}^2$ and $\omega_{i,k-1} \in \mathbb{R}$ are the linear and angular velocity of robot i , $\boldsymbol{\epsilon}_{i,k-1} \in \mathbb{R}^3$ denotes white Gaussian noise with $\mathbb{E}(\boldsymbol{\epsilon}_{i,k-1} \boldsymbol{\epsilon}_{i,k-1}^\top) = \mathbf{Q}_{i,k-1} > \mathbf{0}$. \mathbf{R} is the rotation matrix given by $\mathbf{R}(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$.

For the entire N robot system, the state vector, the motion input vector, and the process noise vector are defined as the stacked vectors, i.e., $\mathbf{x}_k = [\mathbf{x}_{1,k}^\top \ \dots \ \mathbf{x}_{N,k}^\top]^\top$, $\mathbf{u}_{k-1} = [\mathbf{u}_{1,k-1}^\top \ \dots \ \mathbf{u}_{N,k-1}^\top]^\top$ and $\boldsymbol{\epsilon}_{k-1} = [\boldsymbol{\epsilon}_{1,k-1}^\top \ \dots \ \boldsymbol{\epsilon}_{N,k-1}^\top]^\top$. Then the system motion equation is given by

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \boldsymbol{\epsilon}_{k-1}). \quad (2)$$

B. Measurement Model

At time step k , if robot $i, i \in \{1, 2, \dots, N\}$ obtains a robot-to-robot relative measurement of robot $j, j \in \{1, 2, \dots, N\}/i$, the relative measurement can be written as follows

$$\mathbf{y}_{ij,k} = \mathbf{h}_{ij}(\mathbf{x}_{i,k}, \mathbf{x}_{j,k}) + \boldsymbol{\epsilon}_{ij,k} \quad (3)$$

where $\boldsymbol{\epsilon}_{ij,k}$ is the measurement noise assumed to be the Gaussian white noise with $\mathbb{E}(\boldsymbol{\epsilon}_{ij,k} \boldsymbol{\epsilon}_{ij,k}^\top) = \mathbf{R}_{ij,k} > \mathbf{0}$. To be consistent with our simulation and experiment, we assume that the measurements are relative position measurements

$$\mathbf{h}_{ij}(\mathbf{x}_{i,k}, \mathbf{x}_{j,k}) = \mathbf{R}(\theta_{i,k})^\top (\mathbf{p}_{j,k} - \mathbf{p}_{i,k}). \quad (4)$$

Let $\{\dots, \mathbf{y}_{ij,k}, \dots\}$ denote all the robot-to-robot relative measurements at time step k . Note that we do not require an all-to-all relative measurement topology in this paper. The relative measurement vector of the system and the measurement noise vector are defined as the stacked vectors, i.e., $\mathbf{y}_k = [\dots, \mathbf{y}_{ij,k}^\top, \dots]^\top$ and $\boldsymbol{\epsilon}_k = [\dots, \boldsymbol{\epsilon}_{ij,k}^\top, \dots]^\top$. Then, the system measurement equation is given by

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \boldsymbol{\epsilon}_k. \quad (5)$$

C. Linearized Error State Model

Let $\hat{\mathbf{x}}_{k-1|k-1}$ denote the posterior estimate at time step $k-1$. Then the current time prior estimation $\hat{\mathbf{x}}_{k|k-1}$ is

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{f}(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}, \mathbf{0}). \quad (6)$$

Denoted the error state by $\tilde{\mathbf{x}}_k \triangleq \mathbf{x}_k - \hat{\mathbf{x}}_k$, the linearized error state propagation equation for (2) is

$$\tilde{\mathbf{x}}_{k|k-1} = \mathbf{F}_{k-1} \tilde{\mathbf{x}}_{k-1|k-1} + \mathbf{G}_{k-1} \boldsymbol{\epsilon}_{k-1} \quad (7)$$

where \mathbf{F}_{k-1} is the propagation Jacobian,

$$\mathbf{F}_{k-1} = \text{Diag}(\mathbf{F}_{1,k-1}, \dots, \mathbf{F}_{N,k-1}) \quad (8)$$

$$\mathbf{G}_{k-1} = \text{Diag}(\mathbf{G}_{1,k-1}, \dots, \mathbf{G}_{N,k-1}) \quad (9)$$

with

$$\mathbf{F}_{i,k-1} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}(\hat{\mathbf{p}}_{i,k|k-1} - \hat{\mathbf{p}}_{i,k-1|k-1}) \\ \mathbf{0}_{1 \times 2} & 1 \end{bmatrix} \quad (10)$$

$$\mathbf{G}_{i,k-1} = \begin{bmatrix} \mathbf{R}(\hat{\theta}_{i,k-1|k-1}) & \mathbf{0}_{2 \times 1} \\ \mathbf{0}_{1 \times 2} & 1 \end{bmatrix} \quad (11)$$

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

Let $\tilde{\mathbf{y}}_k \triangleq \mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1})$ denote the measurement residual vector, then the linearized measurement equation for (5) is

$$\tilde{\mathbf{y}}_k = \mathbf{H}_k \tilde{\mathbf{x}}_{k|k-1} + \boldsymbol{\epsilon}_k, \quad (12)$$

where \mathbf{H}_k is the measurement Jacobian,

$$\mathbf{H}_k = [\dots \ \mathbf{H}_{ij,k}^\top \ \dots]^\top \quad (13)$$

with

$$\mathbf{H}_{ij,k} = [\dots \ \mathbf{H}_{ij,k}^i \ \dots \ \mathbf{H}_{ij,k}^j \ \dots] \quad (14)$$

$$\mathbf{H}_{ij,k}^i = \mathbf{R}(\hat{\theta}_{i,k|k-1})^\top [-\mathbf{I}_2 \quad -\mathbf{J}(\hat{\mathbf{p}}_{j,k|k-1} - \hat{\mathbf{p}}_{i,k|k-1})] \quad (15)$$

$$\mathbf{H}_{ij,k}^j = \mathbf{R}(\hat{\theta}_{i,k|k-1})^\top [\mathbf{I}_2 \quad \mathbf{0}_{2 \times 1}]. \quad (16)$$

$$\mathcal{O} = \mathbf{M} \begin{bmatrix} -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{2,1|0} - \hat{\mathbf{p}}_{1,1|0}) & \mathbf{I}_2 & \mathbf{0}_2 \\ -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{2,2|1} - \hat{\mathbf{p}}_{1,1|0}) + \delta\mathbf{p}_{1,1} & \mathbf{I}_2 & \mathbf{J}(\hat{\mathbf{p}}_{2,2|1} - \hat{\mathbf{p}}_{2,1|0}) - \delta\mathbf{p}_{2,1} \\ \vdots & \vdots & \vdots & \vdots \\ -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{2,k|k-1} - \hat{\mathbf{p}}_{1,1|0}) + \delta\mathbf{p}_{1,k-1} & \mathbf{I}_2 & \mathbf{J}(\hat{\mathbf{p}}_{2,k|k-1} - \hat{\mathbf{p}}_{2,1|0}) - \delta\mathbf{p}_{2,k-1} \end{bmatrix} \quad (18)$$

D. Observability

For estimators based on the linearized error state system (7) and (12), the *local observability matrix* can be employed to perform the observability analysis [21]. The local observability matrix over the time interval $[1, k]$ is given by

$$\mathcal{O} = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2\mathbf{F}_1 \\ \vdots \\ \mathbf{H}_k\mathbf{F}_{k-1} \cdots \mathbf{F}_1 \end{bmatrix}. \quad (17)$$

An observability matrix for N robots takes up too much space and is hard to read. Following [18], we consider a simple case in which the robot number is limited to 2 and only robot 1 can observe robot 2. Note that such a simple scenario is sufficient to illustrate the inconsistency problem. The local observability matrix for this case is written in (18) where

$$\mathbf{M} = \text{Diag}(\mathbf{R}(\hat{\theta}_{1,1|0}), \mathbf{R}(\hat{\theta}_{1,2|1}), \dots, \mathbf{R}(\hat{\theta}_{1,k|k-1})) \quad (19)$$

$$\delta\mathbf{p}_{i,k} = \sum_{l=1}^k \hat{\mathbf{p}}_{i,l|l} - \hat{\mathbf{p}}_{i,l|l-1}. \quad (20)$$

One can see that if $\delta\mathbf{p}_{i,k}$ is zero, there exist three unobservable dimensions in CL, s.t. $\mathcal{O}\mathbf{N} = \mathbf{0}$ with unobservable subspace

$$\mathbf{N} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{1,1|0} \\ \mathbf{0}_{1 \times 2} & 1 \\ \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{2,1|0} \\ \mathbf{0}_{1 \times 2} & 1 \end{bmatrix}. \quad (21)$$

The presence of three unobservable dimensions in the CL system arises from the inability to estimate the global position (x, y) and orientation (yaw) without absolute measurements. However, $\delta\mathbf{p}_{i,k} = \mathbf{0}$ is almost impossible to hold, resulting in an erroneous reduction of the unobservable subspace dimension to 2, as follows:

$$\mathbf{N} = [\mathbf{I}_2 \quad \mathbf{0}_{2 \times 1} \quad \mathbf{I}_2 \quad \mathbf{0}_{2 \times 1}]^\top. \quad (22)$$

Correspondingly, the global orientation is erroneously observable.

Since central-equivalent DCL algorithms are also based on (7) and (12), they will be inconsistent due to the reduction of the unobservable dimension, which in turn leads to a decrease in accuracy.

III. COORDINATE TRANSFORMATION BASED DCL

In this section, we design a coordinate transformation to ensure the correct observability property of the central-equivalent DCL algorithms.

A. Transformed Linearized Error State System

Section II-D reveals that the propagation Jacobian, which depends on the state estimates, introduces inconsistencies in central-equivalent DCL. More specifically, the difference between prior and posterior estimates results in $\delta\mathbf{p}_{i,k} \neq \mathbf{0}$, which subsequently leads to an unjustified reduction in unobservable dimensions.

An approach to address this inconsistency issue is to render the propagation Jacobian independent of the state. The state-independent propagation Jacobian can prevent the inconsistent terms like (20) appear in the local observability matrix, thus automatically guaranteeing the correct observability. Fortunately, employing state estimation in a transformed coordinate system [20] facilitates the achievement of this objective, in which the state variables in propagation Jacobian can be transmitted to the transformation matrices. Let the coordinate transformation \mathbf{T}_k be a time-varying matrix, and \mathbf{z}_k be the error state in the transformed coordinate, which is obtained by multiplying \mathbf{T}_k with the error state $\tilde{\mathbf{x}}_k$:

$$\mathbf{z}_{k|k-1} = \mathbf{T}_{k|k-1}\tilde{\mathbf{x}}_{k|k-1}, \quad \mathbf{z}_{k|k} = \mathbf{T}_{k|k}\tilde{\mathbf{x}}_{k|k}, \quad (23)$$

where $\mathbf{T}_{k|k-1}$ and $\mathbf{T}_{k|k}$ are evaluated at $\hat{\mathbf{x}}_{k|k}$ and $\hat{\mathbf{x}}_{k|k-1}$, respectively. Substituting (23) into (7) and (12) yields the linearized error state system in the transformed coordinate

$$\begin{cases} \mathbf{z}_{k|k-1} = \mathcal{F}_{k-1}\mathbf{z}_{k-1|k-1} + \mathcal{G}_{k-1}\epsilon_{k-1} \\ \tilde{\mathbf{y}}_k = \mathcal{H}_k\mathbf{z}_{k|k-1} + \epsilon_k \end{cases} \quad (24)$$

where

$$\mathcal{F}_{k-1} = \mathbf{T}_{k|k-1}\mathbf{F}_{k-1}\mathbf{T}_{k-1|k-1}^{-1}, \quad (25)$$

$$\mathcal{G}_{k-1} = \mathbf{T}_{k|k-1}\mathbf{G}_{k-1}, \quad (26)$$

$$\mathcal{H}_k = \mathbf{H}_k\mathbf{T}_{k|k-1}^{-1}. \quad (27)$$

Not all transformations can render the transformed propagation Jacobian independent of the state. For example, the trivial transformation $\mathbf{T}_k = \mathbf{I}_{3N}$ does not take any effect. There are two requirements for designing the transformation:

- The transformation can render the transformed propagation Jacobian independent of the state to ensure the correct observability.
- The transformation matrix is a diagonal block matrix. Otherwise, the state estimates of different robots would be coupled to each other, which is unfavorable for DCL.

Consequently, it is a challenging work to design such a suitable transformation. Our design is inspired by the decomposition of the original propagation Jacobian as

$$\mathbf{F}_{i,k-1} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{i,k|k-1} \\ \mathbf{0}_{1 \times 2} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & -\mathbf{J}\hat{\mathbf{p}}_{i,k-1|k-1} \\ \mathbf{0}_{1 \times 2} & 1 \end{bmatrix}. \quad (28)$$

By comparing (28) with (25), we can construct a proper transformation matrix:

$$\mathbf{T}_k = \text{Diag}(\mathbf{T}_{1,k}, \dots, \mathbf{T}_{N,k}) \quad (29)$$

with

$$\mathbf{T}_{i,k} = \begin{bmatrix} \mathbf{I}_2 & -\mathbf{J}\hat{\mathbf{p}}_{i,k} \\ \mathbf{0}_{1 \times 2} & 1 \end{bmatrix}, \mathbf{T}_{i,k}^{-1} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{i,k} \\ \mathbf{0}_{1 \times 2} & 1 \end{bmatrix}. \quad (30)$$

Then, we have

$$\mathcal{F}_{k-1} = \mathbf{I}_{3N} \quad (31)$$

$$\mathcal{G}_{k-1} = \text{Diag}(\mathcal{G}_{1,k-1}, \dots, \mathcal{G}_{N,k-1}) \quad (32)$$

$$\mathcal{H}_k = [\dots \ \mathcal{H}_{ij,k}^\top \ \dots]^\top \quad (33)$$

with

$$\mathcal{G}_{i,k-1} = \begin{bmatrix} \mathbf{R}(\hat{\theta}_{i,k-1|k-1}) & -\mathbf{J}\hat{\mathbf{p}}_{i,k|k-1} \\ \mathbf{0}_{1 \times 2} & 1 \end{bmatrix} \quad (34)$$

$$\mathcal{H}_{ij,k} = [\dots \ \mathcal{H}_{ij,k}^i \ \dots \ \mathcal{H}_{ij,k}^j \ \dots] \quad (35)$$

$$\mathcal{H}_{ij,k}^i = \mathbf{R}(\hat{\theta}_{i,k|k-1})^\top [-\mathbf{I}_2 \quad -\mathbf{J}\hat{\mathbf{p}}_{j,k|k-1}] \quad (36)$$

$$\mathcal{H}_{ij,k}^j = \mathbf{R}(\hat{\theta}_{i,k|k-1})^\top [\mathbf{I}_2 \quad \mathbf{J}\hat{\mathbf{p}}_{j,k|k-1}]. \quad (37)$$

Now we get a state-independent transformed propagation Jacobian \mathcal{F}_{k-1} . The more interesting property is that it's also an identity matrix, which is more suitable for DCL than the original propagation Jacobian as shown in Section III-B.

B. Transformed Server-based DCL

We deploy the transformation on a server-based framework proposed in [16, 17]. The server-based DCL is a central-equivalent method, in which each robot propagates its estimations while the server maintains the cross-correlations. Note that besides the server-based framework, the transformation can be also integrated into other central-equivalent DCL algorithms. We choose the server-based framework for its better performance. The steps of the proposed transformed server-based DCL are shown in Algorithm 1.

Algorithm 1: Transformed Server-based DCL

Propagation:

Server:

$$\mathcal{P}_{ij,k|k-1} \leftarrow \mathcal{P}_{ij,k-1|k-1} \quad i \neq j \in \{1, \dots, N\}$$

Robot i:

propagate $\hat{\mathbf{x}}_{i,k|k-1}$ using (40)

propagate $\mathcal{P}_{i,k|k-1}$ using (41)

Update:

// measurement $\mathbf{y}_{ab,k}$

Server:

receive $(\hat{\mathbf{x}}_{a,k|k-1}, \mathcal{P}_{a,k|k-1}, \mathbf{y}_{ab,k})$ from robot a

receive $(\hat{\mathbf{x}}_{b,k|k-1}, \mathcal{P}_{b,k|k-1})$ from robot b

calculate $(\mathbf{r}_i, \mathbf{\Gamma}_i)$ for each robot using (42) and (43)

update $\mathcal{P}_{ij,k|k}$ using (48)

Robot i:

receive $(\mathbf{r}_i, \mathbf{\Gamma}_i)$ from Server

update $\hat{\mathbf{x}}_{i,k|k}$ using (46)

update $\mathcal{P}_{i,k|k}$ using (47)

1) *Propagation in the transformed system:* Different from the original server-based DCL, the proposed algorithm estimates the robot states based on the transformed error state. Robot i maintains $\hat{\mathbf{x}}_{i,k}$ and its transformed covariance $\mathcal{P}_{i,k}$,

while the server retains the transformed cross-covariances $\{\mathcal{P}_{ij}\}_{i,j \in \{1, \dots, N\}, i \neq j}$, where

$$\mathcal{P}_{i,k} \triangleq \mathbf{T}_{i,k} \mathbf{P}_{i,k} \mathbf{T}_{i,k}^\top \quad (38)$$

$$\mathcal{P}_{ij,k} \triangleq \mathbf{T}_{i,k} \mathbf{P}_{ij,k} \mathbf{T}_{j,k}^\top. \quad (39)$$

The state estimates and the transformed covariance matrix propagates as

$$\hat{\mathbf{x}}_{i,k|k-1} = \mathbf{f}_i(\hat{\mathbf{x}}_{i,k-1|k-1}, \mathbf{u}_{k-1}, \mathbf{0}) \quad (40)$$

$$\mathcal{P}_{i,k|k-1} = \mathcal{F}_{i,k-1} \mathcal{P}_{i,k-1|k-1} \mathcal{F}_{i,k-1}^\top + \mathcal{G}_{i,k-1} \mathbf{Q}_{i,k-1} \mathcal{G}_{i,k-1}^\top. \quad (41)$$

As the transformed propagation Jacobian $\mathcal{F}_{i,k-1}$ is an identity matrix, the transformed cross-covariances stored in the server do not need to change in the propagation steps.

Remark 1: Compared to the original server-based DCL, the cross-covariances do not need to propagate. This transformation allows us not to design additional mechanisms for propagation. The transformation can be also utilized in other DCL algorithms to reduce the complexity of the cross-covariance propagation.

2) *Update in the transformed system:* Assume that robot a obtains a relative measurement $\mathbf{y}_{ab,k}$ on robot b at time step k . The update steps are detailed below.

Firstly, the server receives a message $(\hat{\mathbf{x}}_{a,k|k-1}, \mathcal{P}_{a,k|k-1}, \mathbf{y}_{ab,k})$ from robot a and a message $(\hat{\mathbf{x}}_{b,k|k-1}, \mathcal{P}_{b,k|k-1})$ from robot b .

Subsequently, the server sends the correction message to each robot. The correction message $(\mathbf{r}_i, \mathbf{\Gamma}_i)$ for robot i is

$$\mathbf{r}_i = \mathcal{K}_i(\mathbf{y}_{ab,k} - \mathbf{h}_{ab}(\hat{\mathbf{x}}_{a,k|k-1}, \hat{\mathbf{x}}_{b,k|k-1})) \quad (42)$$

$$\mathbf{\Gamma}_i = \mathcal{K}_i \mathcal{S} \mathcal{K}_i^\top \quad (43)$$

where

$$\mathcal{K}_i = (\mathcal{P}_{ia,k|k-1} \mathcal{H}_{ab,k}^a{}^\top + \mathcal{P}_{ib,k|k-1} \mathcal{H}_{ab,k}^b{}^\top) \mathcal{S}^{-1} \quad (44)$$

is the transformed Kalman gain obtained by minimizing the trace of the transformed covariance,

$$\begin{aligned} \mathcal{S} = & \mathcal{H}_{ab,k}^a \mathcal{P}_{a,k|k-1} \mathcal{H}_{ab,k}^a{}^\top + \mathcal{H}_{ab,k}^a \mathcal{P}_{ab,k|k-1} \mathcal{H}_{ab,k}^b{}^\top \\ & + \mathcal{H}_{ab,k}^b \mathcal{P}_{ba,k|k-1} \mathcal{H}_{ab,k}^a{}^\top + \mathcal{H}_{ab,k}^b \mathcal{P}_{b,k|k-1} \mathcal{H}_{ab,k}^b{}^\top \\ & + \mathbf{R}_{ab,k}. \end{aligned} \quad (45)$$

Lastly, each robot updates its estimates with the correction message.

$$\hat{\mathbf{x}}_{i,k|k} = \hat{\mathbf{x}}_{i,k|k-1} + \mathbf{T}_{i,k|k-1}^{-1} \mathbf{r}_i \quad (46)$$

$$\mathcal{P}_{i,k|k} = \mathcal{P}_{i,k|k-1} - \mathbf{\Gamma}_i. \quad (47)$$

As the transformed Kalman gain minimizes the trace of the transformed covariance, the estimate correction \mathbf{r}_i is also expressed in the transformed coordinate system. The reverse transformation of \mathbf{r}_i into the original coordinate system is required, as indicated in (46). Simultaneously, the transformed cross-covariances stored in the server are updated as:

$$\mathcal{P}_{ij,k|k} = \mathcal{P}_{ij,k|k-1} - \mathcal{K}_i \mathcal{S} \mathcal{K}_j^\top, i \neq j \in \{1, \dots, N\}. \quad (48)$$

Remark 2: As the robot number increases, the communication cost of updates can be significant. In [17], a method equivalent to the Schmidt-Kalman filter [22] is developed to deal with the message dropouts. Furthermore, we can use this method to actively reduce the communication cost. For

example, instead of all robots, only robot a and robot b update their estimates given relative measurement $\mathbf{y}_{ab,k}$. Correspondingly, the server only updates the cross-covariances related to robot a and robot b . For more information, the reader is recommended to refer to [17].

C. Observability Property

The local observability matrix for the transformed system (24) over the time interval $[1, k]$ is given by

$$\mathcal{O}' = \begin{bmatrix} \mathcal{H}_1 \\ \mathcal{H}_2 \mathcal{F}_1 \\ \vdots \\ \mathcal{H}_k \mathcal{F}_{k-1} \cdots \mathcal{F}_1 \end{bmatrix}. \quad (49)$$

As the same as II-D, we consider a simple case with only two robots. The observability matrix for the simple case is written as

$$\mathcal{O}' = \mathbf{M} \begin{bmatrix} -\mathbf{I}_2 & -\mathbf{J}\hat{\mathbf{p}}_{2,1|0} & \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{2,1|0} \\ -\mathbf{I}_2 & -\mathbf{J}\hat{\mathbf{p}}_{2,2|1} & \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{2,2|1} \\ \vdots & \vdots & \vdots & \vdots \\ -\mathbf{I}_2 & -\mathbf{J}\hat{\mathbf{p}}_{2,k|k-1} & \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{2,k|k-1} \end{bmatrix}. \quad (50)$$

There exists three unobservable dimensions for the transformed system, s.t. $\mathcal{O}'\mathbf{N}' = \mathbf{0}$, with unobservable subspace

$$\mathbf{N}' = [\mathbf{I}_3 \quad \mathbf{I}_3]^\top. \quad (51)$$

Thus, the proposed transformed server-based DCL has the correct observability property.

IV. MONTE CARLO SIMULATIONS

A series of Monte Carlo simulations were conducted to compare the proposed algorithm with other algorithms under various conditions, aiming to validate the preceding observability property analysis and demonstrate the capabilities of the proposed algorithm.

A. Setup

In the simulation scenario, robots moved 360 seconds on a square field. The robot trajectories are shown in Figure 2. The linear velocities \mathbf{v}_i of the robots vary between 0.628 m/s and 1.256 m/s, with angular velocities ω_i ranging from 0.157 rad/s to 0.314 rad/s. The odometry noise ϵ_i is [0.02 m, 0 m, 0.005 rad], and the time interval for odometry calculations Δt is fixed at 0.1 seconds. Inter-robot relative position information is obtained through a combination of relative bearing and distance measurements. The noise levels associated with bearing and distance measurements are 0.01 rad and 0.2 m, respectively, with relative position measurements being conducted at a frequency of 2 Hz.

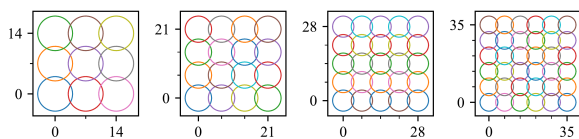


Fig. 2. Simulated trajectories of 9, 16, 25, and 36 robots. The unit is meter. The robots move in a circle with a radius of four meters. The period of one circle of motion varies from 20 to 40 seconds.

We compared the following methods in the Monte Carlo simulations:

- Central: A centralized Cooperative Learning (CL) algorithm that utilizes a joint Extended Kalman Filter (EKF) [14]. However, it calculates the Jacobians using ground truth instead of state estimates to ensure correct observability [19]. This method serves as the baseline.
- OSB: The original server-based DCL algorithm [17].
- TSB: The proposed transformed server-based DCL algorithm.

Although this paper does not specifically investigate the impact of communication failures on the performance of DCL algorithms, for the reader's reference, we assess the performance of the DCL algorithms under three communication conditions: a 50% success rate (OSB-50 and TSB-50), a 75% success rate (OSB-75 and TSB-75) and a 99% success rate (OSB-99 and TSB-99). Each algorithm undergoes 100 Monte Carlo runs.

B. Metrics

The evaluation metrics used in this paper comprise the root mean square error (RMSE) for accuracy assessment and the normalized estimation error squared (NEES) for consistency evaluation. A lower RMSE indicates higher precision, while a NEES value closer to 1 signifies better consistency.

C. Results

1) *Test on a fixed number of robots and various sensor ranges:* In the first test, the robot count was fixed at $N = 16$, and the sensor range was limited to 5, 10, 15, or 20 meters. Figure 3 reports the position and orientation RMSE of 10 robots across 100 Monte Carlo simulations. Table I presents the average NEES of these algorithms. As the range

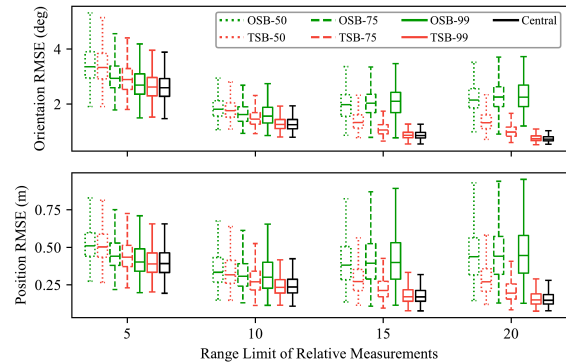


Fig. 3. Orientation and position RMSE with $N = 16$ in 100 Monte Carlo runs with different sensor range limits.

limit increases, both Central and the proposed method OSB demonstrate better accuracy. However, when the range limit exceeds 10 meters, the performance of the OSB methods deteriorates. We attribute the inferior performance of OSB to overconfidence in orientation estimation as discussed in Section II-D. As the range limit and communication success rate rise, more relative measurements become accessible for updating state estimates. This leads to a significant decrease

TABLE I

AVERAGE ORIENTATION / POSITION NEES OF 100 MONTE CARLO SIMULATIONS WITH $N = 16$

Sensor range	5 m	10 m	15 m	20 m
Central	1.02 / 1.02	1.08 / 1.47	1.13 / 1.56	1.09 / 1.53
OSB-50	1.07 / 1.06	1.30 / 1.31	2.99 / 2.18	5.01 / 3.05
TSB-50	1.02 / 1.01	1.15 / 1.21	1.44 / 1.41	1.89 / 1.63
OSB-75	1.13 / 1.08	1.56 / 1.61	4.81 / 3.37	7.84 / 4.83
TSB-75	1.03 / 1.00	1.12 / 1.33	1.34 / 1.50	1.55 / 1.57
OSB-99	1.18 / 1.12	1.95 / 2.02	6.68 / 4.90	10.7 / 7.21
TSB-99	1.04 / 1.00	1.08 / 1.42	1.14 / 1.56	1.14 / 1.52

in the consistency of OSB, as demonstrated in Table I. Over time, the consistency of OSB declines further compared to TSB, as depicted in Figure 4.

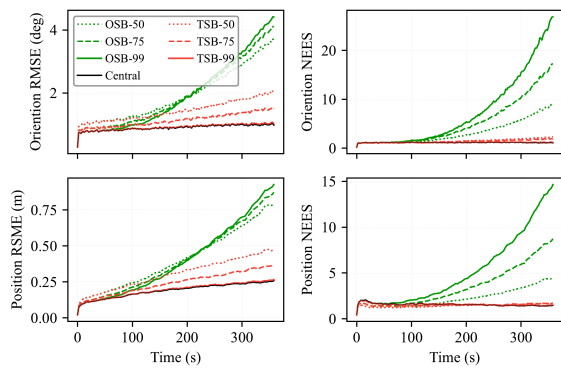


Fig. 4. Average RMSE and NEES of 100 Monte Carlo simulations over time with $N = 16$ and sensor range limited to 15 m.

2) *Test on a fixed sensor range and various number of robots:* In the second test, the sensor range is restricted to 10 meters. The algorithms are evaluated with various numbers of robots. Figure 5 illustrates the average RMSE of all robot estimates in 100 Monte Carlo simulations as the number of robots is raised from 9 to 36. Table II reports the average

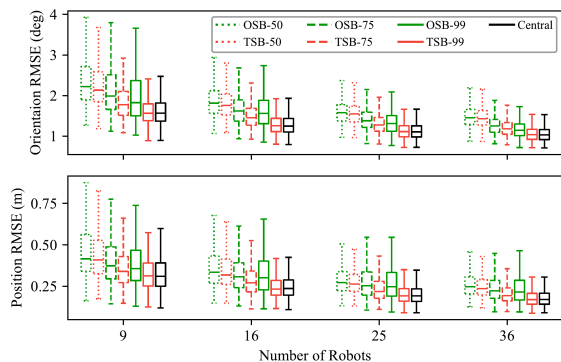


Fig. 5. Orientation and position RMSE of N robots in 100 Monte Carlo runs with sensor range limited to 10 m.

NEES of these algorithms. As the number of robots increases, both OSB and TSB demonstrate enhanced accuracy. However, the proposed algorithm TSB still outperforms OSB in terms of accuracy and consistency.

TABLE II

AVERAGE ORIENTATION / POSITION NEES OF 100 MONTE CARLO SIMULATIONS WITH SENSOR RANGE LIMITED TO 10 M

Robot number	9	16	25	36
Central	1.07 / 1.31	1.08 / 1.47	1.08 / 1.39	1.08 / 1.46
OSB-50	1.38 / 1.26	1.30 / 1.31	1.12 / 1.18	1.10 / 1.27
TSB-50	1.11 / 1.16	1.15 / 1.21	1.05 / 1.11	1.05 / 1.19
OSB-75	1.62 / 1.37	1.56 / 1.61	1.34 / 1.59	1.21 / 1.56
TSB-75	1.05 / 1.15	1.12 / 1.33	1.08 / 1.29	1.06 / 1.26
OSB-99	1.87 / 1.63	1.95 / 2.02	1.63 / 2.06	1.37 / 2.10
TSB-99	1.04 / 1.30	1.08 / 1.42	1.07 / 1.37	1.07 / 1.43

V. EXPERIMENTS

The experiments were conducted on the publicly available UTIAS dataset [23]. This dataset comprises nine subsets, each lasting from 15 to 70 minutes. Each subset records the odometry, measurements, and ground truth pose of five robots. In addition to inter-robot measurements, the measurements also include relative bearing and distance with respect to known position landmarks. Subset 9 poses a challenge due to obstacles obstructing the view, as illustrated in Figure 6. In addition to Central and OSB, we compare the proposed method TSB with the following noncentral-equivalent DCL algorithms:

- BDA: employ a block-diagonal approximation to update covariance matrices [15].
- DMV: be capable of processing various types of robot-to-robot relative measurements [11].
- Naive: disregard cross-correlations among robot pose estimations.

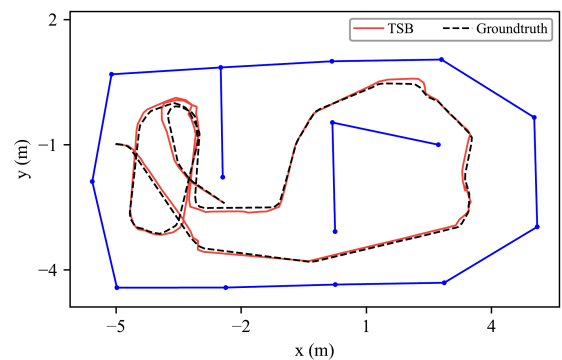


Fig. 6. Ground truth and estimated trajectories of Robot 3 during the initial 300 seconds on UTIAS Subset 9. The blue lines represent obstacles obstructing the view.

These algorithms are evaluated using relative bearing and range measurements as relative position measurements, with odometry utilized as the motion input. Five percent of landmark measurements are utilized to mitigate trajectory drift, while all landmark measurements in Subset 9 are used. Among these algorithms, Central is regarded with the best performance in terms of accuracy and consistency [19]. Thus, Central was tested with over 100 different noise parameter

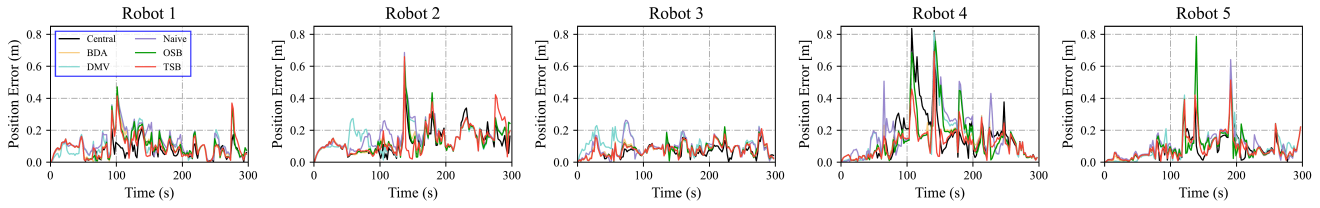


Fig. 7. 5 robot position estimation errors during the first 300s on UTIAS Subset 9

sets for each subset, and the one with the best position accuracy was selected as the noise parameter for that subset.

Figure 7 illustrates the estimation errors of the positions of 5 robots during the initial 300 seconds of UTIAS Subset 9. Table III reports the RMSE values of the nine subsets. It is evident that the proposed algorithm, TSB, outperforms the other four DCL algorithms in terms of accuracy.

TABLE III

ORIENTATION (DEG) AND POSITION (M) RMSE ON UTIAS DATASET

Set	Central	BDA	DMV	Naive	OSB	TSB
1	16.6 / 0.20	18.1 / 0.31	15.9 / 0.39	26.2 / 1.11	19.2 / 0.30	17.4 / 0.24
2	9.28 / 0.14	9.66 / 0.17	12.2 / 0.29	11.1 / 0.26	9.62 / 0.16	9.55 / 0.14
3	8.81 / 0.13	9.27 / 0.16	9.50 / 0.20	9.29 / 0.18	9.20 / 0.15	10.2 / 0.13
4	11.2 / 0.12	11.9 / 0.19	13.2 / 0.27	12.7 / 0.28	11.6 / 0.18	11.5 / 0.15
5	7.98 / 0.18	8.31 / 0.21	10.1 / 0.29	10.2 / 0.39	8.14 / 0.19	8.23 / 0.19
6	6.83 / 0.12	7.23 / 0.14	8.93 / 0.20	10.5 / 0.22	7.00 / 0.13	6.93 / 0.12
7	7.76 / 0.14	8.24 / 0.16	8.99 / 0.25	13.3 / 0.58	7.96 / 0.15	7.95 / 0.14
8	12.9 / 0.16	18.3 / 0.32	27.7 / 0.82	17.9 / 0.46	17.1 / 0.27	13.7 / 0.20
9	16.9 / 0.24	19.1 / 0.31	19.4 / 0.31	19.3 / 0.30	19.1 / 0.30	17.8 / 0.25

VI. CONCLUSION

In this paper, we propose a method to improve the consistency and accuracy of the central-equivalent DCL by performing state estimation based on the transformed error state. We first design a coordinate transformation for DCL. This transformation ensures the correct observability property by rendering the propagation Jacobian independent of states and simplifies the propagation of cross-covariance matrices in DCL. Then we integrate the transformation into the server-based framework. Finally, the simulations and the real-world data experiments show that the proposed algorithm outperforms other DCL algorithms in terms of consistency and accuracy.

REFERENCES

- [1] A. Borgese, D. C. Guastella, G. Sutera, and G. Muscato, "Tether-based localization for cooperative ground and aerial vehicles," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8162–8169, 2022.
- [2] A. Bahr, M. R. Walter, and J. J. Leonard, "Consistent cooperative localization," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2009, pp. 3415–3422.
- [3] S. Julier and J. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," in *Proceedings of the 1997 American Control Conference (Cat. No.97CH36041)*, vol. 4, Jun. 1997, pp. 2369–2373 vol.4.
- [4] L. C. Carrillo-Arce, E. D. Nerurkar, J. L. Gordillo, and S. I. Roumeliotis, "Decentralized multi-robot cooperative localization using covariance intersection," in *IEEE International Conference on Intelligent Robots and Systems*, 2013, pp. 1412–1417.
- [5] T. K. Chang, K. Chen, and A. Mehta, "Resilient and Consistent Multirobot Cooperative Localization with Covariance Intersection," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 197–208, Feb. 2022.
- [6] R. He, Y. Shan, and K. Huang, "Robust cooperative localization with failed communication and biased measurements," *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2997–3004, 2024.
- [7] J. Sijts and M. Lazar, "State fusion with unknown correlation: Ellipsoidal intersection," *Automatica*, vol. 48, no. 8, pp. 1874–1878, Aug. 2012.
- [8] H. Li, F. Nashashibi, and M. Yang, "Split Covariance Intersection Filter: Theory and Its Application to Vehicle Localization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1860–1871, Dec. 2013.
- [9] B. Noack, J. Sijts, M. Reinhardt, and U. D. Hanebeck, "Decentralized data fusion with inverse covariance intersection," *Automatica*, vol. 79, pp. 35–41, May 2017.
- [10] P. Zhu and W. Ren, "Fully distributed joint localization and target tracking with mobile robot networks," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 4, pp. 1519–1532, 2021.
- [11] J. Zhu and S. S. Kia, "Cooperative Localization under Limited Connectivity," *IEEE Transactions on Robotics*, vol. 35, no. 6, pp. 1523–1530, Dec. 2019.
- [12] S. Leonardos and K. Daniilidis, "A game-theoretic approach to Robust Fusion and Kalman filtering under unknown correlations," in *2017 American Control Conference (ACC)*, May 2017, pp. 2568–2573.
- [13] C. Chen and S. S. Kia, "Cooperative Localization Using Learning-Based Constrained Optimization," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7052–7058, Jul. 2022.
- [14] S. Roumeliotis and G. Bekey, "Distributed multirobot localization," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, Oct. 2002.
- [15] L. Luft, T. Schubert, S. I. Roumeliotis, and W. Burgard, "Recursive decentralized localization for multi-robot systems with asynchronous pairwise communication," *The International Journal of Robotics Research*, vol. 37, no. 10, pp. 1152–1167, Sep. 2018.
- [16] S. S. Kia, S. Rounds, and S. Martínez, "Cooperative localization under message dropouts via a partially decentralized EKF scheme," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 5977–5982.
- [17] S. S. Kia, J. Hechtbauer, D. Gogokhiya, and S. Martínez, "Server-Assisted Distributed Cooperative Localization Over Unreliable Communication Links," *IEEE Transactions on Robotics*, vol. 34, no. 5, pp. 1392–1399, Oct. 2018.
- [18] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Analysis and improvement of the consistency of extended Kalman filter based SLAM," in *2008 IEEE International Conference on Robotics and Automation*, May 2008, pp. 473–479.
- [19] G. P. Huang, N. Trawny, A. I. Mourikis, and S. I. Roumeliotis, "Observability-based consistent EKF estimators for multi-robot cooperative localization," *Autonomous Robots*, vol. 30, no. 1, pp. 99–122, Jan. 2011.
- [20] N. Hao, F. He, C. Tian, Y. Yao, and W. Xia, "Kd-ekf: A consistent cooperative localization estimator based on kalman decomposition," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 11 064–11 070.
- [21] Z. Chen, K. Jiang, and J. Hung, "Local observability matrix and its application to observability analyses," in *[Proceedings] IECON '90: 16th Annual Conference of IEEE Industrial Electronics Society*, 1990, pp. 100–103 vol.1.
- [22] S. F. Schmidt, "Application of state-space methods to navigation problems," in *Advances in control systems*. Elsevier, 1966, vol. 3, pp. 293–340.
- [23] K. Y. Leung, Y. Halpern, T. D. Barfoot, and H. H. Liu, "The utias multi-robot cooperative localization and mapping dataset," *The International Journal of Robotics Research*, vol. 30, no. 8, pp. 969–974, 2011.