Robust Cooperative Localization With Failed Communication and Biased Measurements

Ronghai He, Yunxiao Shan, and Kai Huang, *Member, IEEE*

***Abstract*—Cooperative Localization (CL) plays a crucial role in achieving precise localization without relying on localization sensors. However, the performance of CL can be significantly affected by failed communication and biased measurements. This letter presents a robust decentralized CL method that addresses these challenges effectively. To tackle the issue of communication failures, the proposed method adopts a multi-centralized framework that separates the measurement and communication processes. This decoupling allows each robot to utilize measurement information even in the absence of communication. Additionally, an reasonable state estimation method for other robots is proposed by approximating the actual input velocity model of unknown states and then propagating them using the motion model. To handle biased measurements, the method incorporates the M-estimation technique into the measurement update process. This technique weights the received measurements according to their reliability, mitigating the impact of biased measurements on the estimation accuracy. Simulation experiments have been conducted to validate the effectiveness of the proposed method in challenging scenarios.**

***Index Terms*—Cooperative localization, failed communication, biased measurements.**

I. Introduction

Cooperative Localization (CL) is a crucial element in the robotics community, particularly in GNSS (Global Navigation Satellite System)-denied scenarios. By relying on intercommunication and measurement between robots, CL can achieve precise localization without depending on localization sensors. As a result, there is a growing interest in developing state-of-the-art CL methods to improve performance and meet the requirements of these applications.

CL methods can be classified into two categories: centralized and decentralized methods based on the communication flow direction. Centralized methods treat the states of the entire robot team as a single vector and perform computation in a central unit. Centralized methods achieve optimal performance when communication and measurements are perfect. However, in real-world scenarios, communication is not guaranteed. Decentralized methods, on the other hand, are widely used in practical applications. In decentralized methods, each robot calculates its pose based on active measurements and received information. This approach is not dependent on specific communication with a fixed central unit, thus making it adaptable to complex environments. Despite this advantage, decentralized methods may also suffer from the adverse effects of communication, such as low bandwidth [1], [2] and failed communication, and measurements like biased measurement between robots, all of which can significantly impact their performance.

Developing a robust decentralized CL (DCL) method that can guide multirobot systems in scenarios with failed communication and biased measurements is a complex task. The main challenge lies in the fact that failed communication prevents robots from receiving the state information of other robots, making measurement updates difficult, especially in frameworks that integrate communication and measurement. One potential solution is to adopt a multi-centralized framework [1], [3] that separates measurement from communication. This approach allows each robot to estimate not only its own state but also the states of other robots, even in the absence of communication, thereby enabling measurement updates. However, reasonably estimating the unknown states of other robots still presents a significant challenge. On the other hand, improving state estimation with measurement updates has a high requirement on the quality of the measurements. If the measurements are biased due to sensor errors or other factors and used for state estimation, the accuracy of state estimation cannot be guaranteed [4].

After careful consideration of the challenges mentioned earlier, we propose a robust decentralized CL method that addresses the issues of failed communication and biased measurements. Our method utilizes a multi-centralized framework, which allows each robot to estimate its own state and the states of other robots independently. By decoupling the measurement and communication processes, our framework ensures that measurement information is fully utilized even in the absence of communication. To reasonably estimate the unknown states of other robots using the motion model, we approximate the actual

Manuscript received 16 September 2023; accepted 16 January 2024. Date of publication 6 February 2024; date of current version 19 February 2024. This letter was recommended for publication by Associate Editor A. Dutta and Editor M. A. Hsieh upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant 62232008, in part by Shenzhen Science and Technology Program under Grant JCYJ20210324122203009, in part by GuangDong Basic and Applied Basic Research Foundation under Grant 2020A1515110199, and in part by the Key Laboratory of MIIT for Intelligent Products Testing and Reliability 2023 Key Laboratory Open Project Fund under Grant CEPREI2023-02. *(Corresponding author: Yunxiao Shan.)*

Ronghai He is with the School of Artificial Intelligence, Sun Yat-sen University, Guangzhou 510275, China, and also with the Southern Marine Science and Engineering Guangdong Laboratory, Zhuhai, Guangdong 519000, China (e-mail: herh5@mail2.sysu.edu.cn).

Yunxiao Shan is with the School of Artificial Intelligence, Sun Yat-sen University, Guangzhou 510275, China, also with the Southern Marine Science and Engineering Guangdong Laboratory, Zhuhai, Guangdong 519000, China, and also with the Shenzhen Institute, Sun Yat-sen University, Shenzhen, Guangdong 518057, China (e-mail: shanyx@mail.sysu.edu.cn).

Kai Huang is with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, Guangdong 510275, China (e-mail: huangk36@mail.sysu.edu.cn).

The source code has been made accessible to the public via https://github.com/RonghaiHe/RobustCL.

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2024.3362682, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3362682

input velocity model of other robots and validate the accuracy of this approximation. Additionally, to handle biased sensor data, our approach incorporates a robust estimation technique called M-estimation (generalized maximum likelihood estimation) [5], [6]. This technique can weight the received measurements according to the degree of reliability during the estimation process, mitigating the impact of biased measurements. We have conducted simulation experiments to validate our approach, and the results demonstrate its effectiveness in challenging scenarios.

II. Related Work

A. Approaches for Dealing With Failed Communication

Failed communication can lead to message dropouts [7], [8] and impact the EKF-based DCL approach [9] because the update of the cross-covariance relies heavily on communication. There are two methods for achieving DCL based on how they deal with cross-covariance: ignoring cross-covariance and approximating cross-covariance. Classical methods like Covariance Intersection (CI) [10] fuse two estimated state-covariances without considering their cross-covariance, but they tend to be conservative. Split CI [11] attempts to divide the covariance terms into correlated and independent terms to reduce conservatism, but complete separation is challenging and can affect estimation accuracy.

Recognizing the limitations of CI-based methods, Luft et al. [12] proposed a Block-Diagonal Approximation (BDA) technique to approximate the cross-covariance, so the communication is needed only when one robot measures another. However, this approach may compromise estimation consistency and result in overconfident pose estimation. Zhu et al. [13] proposed the Discorrelated Minimum Variance (DMV) approach to replace the original covariance of two robots, eliminating the need to track cross-covariance at each time step. Chen et al. [14] further proposed a learning-based DMV to reduce computation time. Based on [15], Zhu et al. [16] proposed a consistent algorithm for DCL and target tracking with the caveat that the storage cost grows exponentially. To further improve the robustness, referring to the multi-centralized framework, Chang et al. [3] proposed a DCL framework where one robot estimates its own state-covariance as well as the state-covariance of all other robots. By estimating the state of others, the relative measurement can be decoupled from communication. Nevertheless, they did not consider the biased measurements.

B. Approaches for Dealing With Biased Measurements

Besides of communication, the performance of DCL also relies on measurements. However, if the measurements are biased, the DCL performance will degrade. To mitigate the impact of biased measurements, fault detection and exclusion (FDE) methods are commonly employed [17]. One such method proposed by Al Hage et al. [18] is based on the Kullback-Leibler divergence (KL-divergence). This method requires pre-experiments to obtain the distribution in abnormal situations, making it complex. Similarly, El Mawa et al. [19] proposed a Jensen Shannon divergence-based FDE method that also necessitates pre-experiments. Another approach to handling biased measurements is to fuse them in a more conservative manner, known as Covariance Union (CU). Wang et al. [4] proposed a method called BDA-CU. In this method, if biased measurements occur, both robots first fuse using the BDA as usual, and the CU is employed as the final step to cover both estimations. Nevertheless, it may fail when both measurements are biased.

In conclusion, while some methods claim to handle failed communication or biased measurements individually, in real-world scenarios, both failed communication and biased measurements often occur simultaneously. Therefore, in alignment with the multi-centralized framework, we propose a robust decentralized CL method that can accommodate both failed communication and biased measurements.

III. Multi-Robot System Model

Consider N homogeneous robots moving in a 2D scenario. Set $\boldsymbol{x}_{i,t} = [x_{i,t}, y_{i,t}, \theta_{i,t}]^\top$ as the state of the robot i in a global coordinate at time t, where $[x_{i,t}, y_{i,t}]^\top$ stands for the Cartesian position, $\theta_{i,t}$ means the orientation and $i \in \{1, \ldots, N\}$. Set $\boldsymbol{X}_t$ as states of all N robots at time t. The robot i moves with forward velocity $v_{i,t}$ and angular velocity $\omega_{i,t}$ as the actual input velocity vector $\boldsymbol{u}_{i,t} = [v_{i,t}, \omega_{i,t}]^\top$. Besides, each robot equips with 2 devices:

1) an exteroceptive sensor such as a camera to identify other robots and obtain relative measurements.
2) a communication device to send and receive messages among robots.

The proposed algorithm utilizes a multi-centralized framework, where each robot not only estimates its own state but also estimates the states of all other robots [1]. However, traditional methods with a multi-centralized framework typically treat each robot as a fusion center and rely on high-quality communication to receive information from other robots for more accurate state estimation. To address the challenge of failed communication, the proposed algorithm learns from the multi-centralized framework but does not require communication. The robot i can estimate the states of both itself and other robots without relying on communication. This adaptation allows the algorithm to handle scenarios with failed communication. With the multi-centralized framework, each robot estimates states of the whole system based on the motion model and the measurement model at time t denoted as $\hat{\boldsymbol{X}}_t$:

$$\hat{\boldsymbol{X}}_t = \left[\hat{\boldsymbol{x}}_{1,t}^\top, \ldots, \hat{\boldsymbol{x}}_{i,t}^\top, \ldots, \hat{\boldsymbol{x}}_{N,t}^\top\right]^\top \tag{1}$$

A. Motion Model

With $\boldsymbol{u}_{i,t}$, the state of the robot i at time $t+1$ can be recursively obtained using the motion model with a sample time interval of Δt. In this case, the unicycle model $\boldsymbol{g}(\cdot)$ is utilized as the state propagation function:

$$\boldsymbol{x}_{i,t+1} = \boldsymbol{g}(\boldsymbol{x}_{i,t}, \boldsymbol{u}_{i,t}) = \boldsymbol{x}_{i,t} + \begin{bmatrix} \cos(\theta_{i,t}) & 0 \\ \sin(\theta_{i,t}) & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{u}_{i,t} \Delta t \tag{2}$$

The robot i is only aware of the nominal input velocity vector $\bar{\boldsymbol{u}}_{i,t} = [\bar{v}_{i,t}, \bar{\omega}_{i,t}]^\top$, while the actual input velocity vector $\boldsymbol{u}_{i,t} = \bar{\boldsymbol{u}}_{i,t} + \delta \boldsymbol{u}_{i,t}$ remains unknown. Here, the velocity noise $\delta \boldsymbol{u}_{i,t} = [\delta v_{i,t}, \delta \omega_{i,t}]^\top$ is assumed to be a zero-mean Gaussian noise:

$$\begin{cases} \delta v_{i,t} \sim \mathcal{N}(0, (\sigma_v \bar{v}_{i,t})^2) \\ \delta \omega_{i,t} \sim \mathcal{N}(0, (\sigma_\omega \bar{\omega}_{i,t})^2) \end{cases} \tag{3}$$

Thus, $\boldsymbol{u}_{i,t} = [v_{i,t}, \omega_{i,t}]^\top$ is treated as a random variable following a normal distribution when $\bar{\boldsymbol{u}}_{i,t}$ is known by the robot i:

$$\begin{cases} v_{i,t} \sim \mathcal{N}\left(\bar{v}_{i,t}, (\sigma_v \bar{v}_{i,t})^2\right) \\ \omega_{i,t} \sim \mathcal{N}\left(\bar{\omega}_{i,t}, (\sigma_\omega \bar{\omega}_{i,t})^2\right) \end{cases} \tag{4}$$

where σ_v, σ_ω are fixed variance coefficients.

In the multi-centralized framework, the state propagation function of all robots is denoted as:

$$\begin{aligned} \boldsymbol{X}_{t+1} &= \boldsymbol{g}_N(\boldsymbol{X}_t, \boldsymbol{u}_{1,t}, \ldots, \boldsymbol{u}_{N,t}) \\ &= \left[\boldsymbol{g}(\boldsymbol{x}_{1,t}, \boldsymbol{u}_{1,t})^\top, \ldots, \boldsymbol{g}(\boldsymbol{x}_{N,t}, \boldsymbol{u}_{N,t})^\top\right]^\top \end{aligned} \tag{5}$$

where $\boldsymbol{g}_N$ means motion model of all N robots.

B. Measurement Model

When the robot i obtains the relative measurement about the robot $j (j \in \{1, \ldots, N\} \setminus \{i\})$ at time $t+1$, a measurement model $\boldsymbol{h}(\cdot)$ from the states of the robot i and the robot j to the measurement $\boldsymbol{z}_{ij,t+1}$ is denoted as follows:

$$\boldsymbol{z}_{ij,t+1} = \boldsymbol{h}(\boldsymbol{x}_{i,t+1}, \boldsymbol{x}_{j,t+1}) + \delta \boldsymbol{z}_{ij,t+1} \tag{6}$$

where the measurement noise $\delta \boldsymbol{z}_{ij,t+1}$ is also a zero-mean Gaussian noise. Here, the relative pose $\boldsymbol{h} = [\Delta x, \Delta y, \Delta \theta]^\top$ is used as the measurement and the measurement model is:

$$\boldsymbol{h}(\boldsymbol{x}_{i,t+1}, \boldsymbol{x}_{j,t+1}) = \boldsymbol{\Gamma}(\theta_{i,t+1}) (\boldsymbol{x}_{j,t+1} - \boldsymbol{x}_{i,t+1}) \tag{7}$$

where

$$\boldsymbol{\Gamma}(\theta_{i,t+1}) = \begin{bmatrix} \cos\theta_{i,t+1} & \sin\theta_{i,t+1} & 0 \\ -\sin\theta_{i,t+1} & \cos\theta_{i,t+1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

The measurement model can be rewritten in the multi-centralized framework as:

$$\boldsymbol{z}_{ij,t+1} = \boldsymbol{h}(\boldsymbol{X}_{t+1}) + \delta \boldsymbol{z}_{ij,t+1} \tag{9}$$

IV. Multi-Robot Cooperative Localization Algorithm

In our multi-centralized framework, we incorporate three updates: the time propagation update, the measurement update, and the communication update. These updates are decoupled to handle failed communication environment effectively. In the time propagation update for the robot i, we approximate the distribution of $\boldsymbol{u}_t$ of other robots as a normal distribution and utilize the motion model to reasonably estimate unknown states of other robots. To validate this approximation, a numerical KL-divergence method is employed. For the measurement update, we use a M-estimation method to robustly fuse the measurements, particularly the biased measurements. For the communication part, we utilize a CI-based update method. The whole proposed DCL algorithm is illustrated in Algorithm 1.

Algorithm 1: The Whole Update of The Robot i.

Input:

$\hat{\boldsymbol{X}}_t^i, \hat{\boldsymbol{\Sigma}}_t^i, \boldsymbol{u}_t^i$;

$\{\boldsymbol{z}_{ij,t+1}; \ldots\}$ $(j \in \{1, \ldots, N\} \setminus \{i\})$;

$\left\{\left(\hat{\boldsymbol{X}}_{t+1|t+1}^j, \hat{\boldsymbol{\Sigma}}_{t+1|t+1}^j\right); \ldots\right\}$ $(j \in \{1, \ldots, N\} \setminus \{i\})$

Output: $\hat{\boldsymbol{X}}_{t+1}^i, \hat{\boldsymbol{\Sigma}}_{t+1}^i$

Time Propagation Update

1: $\left(\hat{\boldsymbol{X}}_{t+1|t}^i, \hat{\boldsymbol{\Sigma}}_{t+1|t}^i\right) \leftarrow (\hat{\boldsymbol{X}}_t^i, \hat{\boldsymbol{\Sigma}}_t^i)$ using Eq. (14)

Measurement Update

2: **if** $\{\boldsymbol{z}_{ij,t+1}; \ldots\} \neq \phi$ **then**
3: **for** Measurement in $\{\boldsymbol{z}_{ij,t+1}; \ldots\}$ **do**
4: Update states and covariances using Eq. (27) and 30
5: **end for**
6: **end if**
7: $(\hat{\boldsymbol{X}}_{t+1|t+1}^i, \hat{\boldsymbol{\Sigma}}_{t+1|t+1}^i) \leftarrow (\hat{\boldsymbol{X}}_{t+1|t}^i, \hat{\boldsymbol{\Sigma}}_{t+1|t}^i)$

Communication Update

8: **if** $\left\{\left(\hat{\boldsymbol{X}}_{t+1|t+1}^j, \hat{\boldsymbol{\Sigma}}_{t+1|t+1}^j\right); \ldots\right\} \neq \phi$ **then**
9: **for** Information in $\left\{\left(\hat{\boldsymbol{X}}_{t+1|t+1}^j, \hat{\boldsymbol{\Sigma}}_{t+1|t+1}^j\right); \ldots\right\}$ **do**
10: Update states and covariances using CI
11: **end for**
12: **end if**
13: $\left(\hat{\boldsymbol{X}}_{t+1}^i, \hat{\boldsymbol{\Sigma}}_{t+1}^i\right) \leftarrow \left(\hat{\boldsymbol{X}}_{t+1|t+1}^i, \hat{\boldsymbol{\Sigma}}_{t+1|t+1}^i\right)$

A. Time Propagation Update

For the robot i, the time propagation update (line 1 in Algorithm. 1) needs to both update its own state and states of other robots. The robot i estimates its own state, denoted as $\hat{\boldsymbol{x}}_{i,t}^i$, using (2). In (2), the actual input velocity $\boldsymbol{u}_{i,t}$ is replaced with $\bar{\boldsymbol{u}}_{i,t}$, which represents the expected value of $\boldsymbol{u}_{i,t}$ ((4)). This substitution is possible because the distribution of $\boldsymbol{u}_{i,t}$ is known [20], owning to the known nominal input velocity $(\bar{v}_{i,t}, \bar{\omega}_{i,t})$. In order to update the covariance of its own state, $\hat{\boldsymbol{\Sigma}}_{i,t}^i$, a first-order linear approximation of the Taylor expansion is employed due to the nonlinearity of the motion model ((2)). This approximation requires obtaining the Jacobian matrix of the robot i, denoted as $\boldsymbol{F}_{i,t}^i(\hat{\boldsymbol{x}}_{i,t}^i), \boldsymbol{G}_{i,t}^i(\bar{\boldsymbol{u}}_{i,t})$.

Estimating the states of all other robots in our framework is crucial in addition to estimating its own state. However, this task becomes challenging for the robot i because it lacks knowledge about the nominal input velocity of other robots, like the nominal input velocity of robot j $\bar{\boldsymbol{u}}_{j,t}$, resulting in a non-normal distribution followed by $\boldsymbol{u}_{j,t}$. To address this issue, the resilient method [3] employs an upper bound-based approach to update the state of other robots, eliminating the need to estimate $\boldsymbol{u}_t$ and the orientation of other robots required in the motion model. Nevertheless, this approach may lead to conservation. To address this issue, we utilize a motion model ((2)) to estimate the states and state covariance of other robots. To do this, we need to model the actual input velocity $\boldsymbol{u}_t$ decided by the nominal input velocity $\bar{\boldsymbol{u}}_t$ and the velocity noise. In the case of the robot i, $\bar{\boldsymbol{u}}_{i,t}$ is a constant value. However, when the robot i estimates the state of all other robots, $\bar{\boldsymbol{u}}_t$ of all other robots like $\bar{\boldsymbol{u}}_{j,t}$, are unknown. Therefore, we assume that $\bar{\boldsymbol{u}}_{j,t}$ samples from a normal distribution $\bar{v}_{j,t} \sim \mathcal{N}(\mu_{\bar{v}}, \sigma_{\bar{v}})$ and $\bar{\omega}_{j,t} \sim \mathcal{N}(\mu_{\bar{\omega}}, \sigma_{\bar{\omega}})$ with predetermined and identical parameters. As a result of this assumption, $\bar{\boldsymbol{u}}_{j,t} = [\bar{v}_{j,t}, \bar{\omega}_{j,t}]^\top$ can be considered as a random variable for the robot i. By replacing $\bar{v}_{i,t}, \bar{\omega}_{i,t}$ with $\bar{v}_{j,t}, \bar{\omega}_{j,t}$, (4) can be seen as a conditional distribution denoted as $f(v_{j,t} \mid \bar{v}_{j,t})$ and $f(\omega_{j,t} \mid \bar{\omega}_{j,t})$, respectively. With the definition of conditional distribution, the probability density function (PDF) of the actual input velocity $\boldsymbol{u}_{j,t}$ can be seen as a marginal PDF obtained from the joint PDF of $f(v_{j,t}, \bar{v}_{j,t}), f(\omega_{j,t}, \bar{\omega}_{j,t})$:

$$\begin{cases} f(v_{j,t}) = \int_{-\infty}^{+\infty} f(v_{j,t}, \bar{v}_{j,t}) \mathrm{d}\bar{v}_{j,t} \\ f(\omega_{j,t}) = \int_{-\infty}^{+\infty} f(\omega_{j,t}, \bar{\omega}_{j,t}) \mathrm{d}\bar{\omega}_{j,t} \end{cases} \tag{10}$$

where

$$\begin{cases} f(v_{j,t}, \bar{v}_{j,t}) = f(v_{j,t} \mid \bar{v}_{j,t}) f(\bar{v}_{j,t}) \\ f(\omega_{j,t}, \bar{\omega}_{j,t}) = f(\omega_{j,t} \mid \bar{\omega}_{j,t}) f(\bar{\omega}_{j,t}) \end{cases} \tag{11}$$

Obviously, The PDF of $\boldsymbol{u}_{j,t}$ does not have an analytical formula, and we cannot make definitive inferences about it following a specific distribution. It's difficult to propagate the state covariances of other robots with a non-normal distribution followed by $\boldsymbol{u}_{j,t}$ [20]. To address this, we first approximate $\boldsymbol{u}_{j,t}$ as normal distributions. In order to ensure a reasonable approximation, we assume that the expectation and variance of the approximated normal distributions are the same as those of $\boldsymbol{u}_{j,t} = [v_{j,t}, \omega_{j,t}]^\top$. In order to obtain the expectation and variance, we used the laws of conditional probability [21]:

$$\begin{cases} \mathbb{E}[v] = \mathbb{E}\{\mathbb{E}[v \mid \bar{v}]\} & \mathrm{Var}[v] = \mathrm{Var}\{\mathbb{E}[v|\bar{v}]\} + \mathbb{E}\{\mathrm{Var}[v|\bar{v}]\} \\ \mathbb{E}[\omega] = \mathbb{E}\{\mathbb{E}[\omega \mid \bar{\omega}]\} & \mathrm{Var}[\omega] = \mathrm{Var}\{\mathbb{E}[\omega|\bar{\omega}]\} + \mathbb{E}\{\mathrm{Var}[\omega|\bar{\omega}]\} \end{cases} \tag{12}$$

where Var means variance. By (12), the expectation and the variance of $\boldsymbol{u}_{j,t}$ can be calculated as:

$$\begin{cases} \mathbb{E}[v_{j,t}] = \mu_{\bar{v}} & \mathrm{Var}[v_{j,t}] = \sigma_{\bar{v}}^2 + \sigma_v^2(\sigma_{\bar{v}}^2 + \mu_{\bar{v}}^2) \\ \mathbb{E}[\omega_{j,t}] = \mu_{\bar{\omega}} & \mathrm{Var}[\omega_{j,t}] = \sigma_{\bar{\omega}}^2 + \sigma_\omega^2(\sigma_{\bar{\omega}}^2 + \mu_{\bar{\omega}}^2) \end{cases} \tag{13}$$

To ensure the reasonableness of the approximation, the numerical KL-divergence is utilized to measure the similarity between the approximated normal distribution and the actual distribution of v_t and ω_t. By setting a grid density of 0.001, the numerical KL-divergence for $v_{j,t}$ is 0.046, and for $\omega_{j,t}$, it is 0.053. These results indicate that the approximation is reasonable and provides a good fit to the actual distributions.

With the reasonable approximation, we can use the expectation of $\boldsymbol{u}_{j,t}$ to propagate the states. Therefore, the time propagation update of both state and covariance can be obtained as:

$$\begin{cases} \hat{\boldsymbol{X}}^i_{t+1|t} = \boldsymbol{g}_N\left(\hat{\boldsymbol{X}}^i_t; \mathbb{E}[\boldsymbol{u}_{1,t}], \ldots, \bar{\boldsymbol{u}}_{i,t}, \ldots, \mathbb{E}[\boldsymbol{u}_{N,t}]\right) \\ \hat{\boldsymbol{\Sigma}}^i_{t+1|t} = \boldsymbol{F}^i_t \hat{\boldsymbol{\Sigma}}^i_t \boldsymbol{F}^{i\top}_t + \boldsymbol{G}^i_t \boldsymbol{Q}^i_t \boldsymbol{G}^{i\top}_t \end{cases} \tag{14}$$

where $(\hat{\boldsymbol{X}}^i_t, \hat{\boldsymbol{\Sigma}}^i_t)$ and $(\hat{\boldsymbol{X}}^i_{t+1|t}, \hat{\boldsymbol{\Sigma}}^i_{t+1|t})$ represent the state and the state covariance of all robots estimated by the robot i after the whole update at time t and after the time propagation update at time $t+1$. $\boldsymbol{Q}^i_t = \mathrm{diag}(\mathrm{Var}[v_{1,t}], \mathrm{Var}[\omega_{1,t}], \ldots, \mathrm{Var}[v_{i,t}], \mathrm{Var}[\omega_{i,t}], \ldots, \mathrm{Var}[v_{N,t}], \mathrm{Var}[\omega_{N,t}])$ and:

$$\begin{cases} \boldsymbol{F}^i_t = \mathrm{diag}\left(\boldsymbol{F}^i_{1,t}(\hat{\boldsymbol{x}}^i_{1,t}, \mathbb{E}[\boldsymbol{u}_{1,t}]), \ldots, \boldsymbol{F}^i_{i,t}(\hat{\boldsymbol{x}}^i_{i,t}, \bar{\boldsymbol{u}}_{i,t}), \right. \\ \qquad \left. \ldots, \boldsymbol{F}^i_{N,t}(\hat{\boldsymbol{x}}^i_{N,t}, \mathbb{E}[\boldsymbol{u}_{N,t}])\right) \\ \boldsymbol{G}^i_t = \mathrm{diag}\left(\boldsymbol{G}^i_{1,t}(\mathbb{E}[\boldsymbol{u}_{1,t}]), \ldots, \boldsymbol{G}^i_{i,t}(\bar{\boldsymbol{u}}_{i,t}), \ldots, \right. \\ \qquad \left. \boldsymbol{G}^i_{N,t}(\mathbb{E}[\boldsymbol{u}_{N,t}])\right) \end{cases} \tag{15}$$

B. M-Estimation-Based Measurement Update

The robot i has estimated its own state and the state of other robots. Once a relative measurement is obtained, the measurement update will be carried out to enhance the accuracy of the state estimation (line 2 to 7 in Algorithm. 1). However, when the measurement is biased, the state estimation error may be increased rapidly. In order to cope with the biased measurement, Most methods like [17], [18] and [19] adopt FDE to reject measurements that exceed a preset threshold. However, the performance of FDE-based methods is highly dependent on the threshold, which is challenging to pick. To overcome this limitation, we have chosen a more robust method, M-estimation, for the measurement update. Unlike FDE, which determines measurement biases based on the similarity judgement before and after the measurement update such as the Mahalanobis distance, the M-estimation assigns different weights according to the measurement residuals between actual measurement values and calculated values based on the estimated states from the time propagation update, iteratively obtaining the most suitable weights for different measurements.

To apply the M-estimation in the measurement update, we first employ a first-order linear approximation of the Taylor expansion to (9). This linearized measurement model can be expressed as:

$$\boldsymbol{z}_{ij,t+1} = \boldsymbol{h}(\hat{\boldsymbol{X}}^i_{t+1|t}) + \boldsymbol{H}^i_{t+1}\left(\boldsymbol{X}_{t+1} - \hat{\boldsymbol{X}}^i_{t+1|t}\right) + \delta\boldsymbol{z}_{ij,t+1} \tag{16}$$

where $\boldsymbol{H}^i_{t+1} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{X}_{t+1}}\Big|_{\boldsymbol{X}_{t+1} = \hat{\boldsymbol{X}}^i_{t+1|t}}$.

Then the solution of the state can be rewritten as a linear regression model:

$$\begin{Bmatrix} \boldsymbol{z}_{ij,t+1} - \boldsymbol{h}(\hat{\boldsymbol{X}}^i_{t+1|t}) + \boldsymbol{H}^i_{t+1}\hat{\boldsymbol{X}}^i_{t+1|t} \\ \hat{\boldsymbol{X}}^i_{t+1|t} \end{Bmatrix} = \begin{bmatrix} \boldsymbol{H}^i_{t+1} \\ \boldsymbol{I}_{N\times N} \end{bmatrix} \boldsymbol{X}_{t+1} + \begin{Bmatrix} \delta\boldsymbol{z}_{ij,t+1} \\ \boldsymbol{\delta}_k \end{Bmatrix} \tag{17}$$

where $\boldsymbol{I}$ is an identity matrix and $\boldsymbol{\delta}_k$ means the state estimation error in the time propagation update.

To ensure that each component in the residual term in the regression equation is independent and identically distributed, which is required by the application of the M-estimation, we need to transform (17) as follows [6]:

$$\boldsymbol{p}_{t+1} = \boldsymbol{M}_{t+1}\boldsymbol{X}_{t+1} + \boldsymbol{\xi}_{t+1} \tag{18}$$

where:

$$\boldsymbol{p}_{t+1} = \boldsymbol{S}^{-1/2}_{t+1} \begin{Bmatrix} \boldsymbol{z}_{ij,t+1} - \boldsymbol{h}(\hat{\boldsymbol{X}}^i_{t+1|t}) + \boldsymbol{H}^i_{t+1}\hat{\boldsymbol{X}}^i_{t+1|t} \\ \hat{\boldsymbol{X}}^i_{t+1|t} \end{Bmatrix} \tag{19}$$

$$\boldsymbol{M}_{t+1} = \boldsymbol{S}^{-1/2}_{t+1} \begin{bmatrix} \boldsymbol{H}^i_{t+1} \\ \boldsymbol{I}_{N\times N} \end{bmatrix} \tag{20}$$

$$\boldsymbol{\xi}_{t+1} = \boldsymbol{S}^{-1/2}_{t+1} \begin{Bmatrix} \delta\boldsymbol{z}_{ij,t+1} \\ \boldsymbol{\delta}_k \end{Bmatrix} \tag{21}$$

$$\boldsymbol{S}_{t+1} = \begin{bmatrix} \boldsymbol{R}_{ij,t+1} & \boldsymbol{0} \\ \boldsymbol{0} & \hat{\boldsymbol{\Sigma}}^i_{t+1|t} \end{bmatrix} \tag{22}$$

where $\boldsymbol{R}_{ij,t+1}$ is the fixed covariance of $\delta \boldsymbol{z}_{ij,t+1}$. The goal of the measurement update is to reduce the estimation error, known as the residual $\boldsymbol{\xi}_{t+1}$. Therefore, the measurement update process can be reformulated as a regression minimization problem stated in (18) and this problem can be solved using Huber's generalized maximum likelihood technique, which minimizes a cost function:

$$J\left(\hat{\boldsymbol{X}}^{i}_{t+1|t}\right) = \sum_{k=1}^{3N+m} \rho(\zeta_k) \tag{23}$$

where m represents for the dimension of $\boldsymbol{z}_{ij,t+1}$ which is 3 here, $\rho(\cdot)$ is a kernel function and ζ_k is the k-th component of the residual $\boldsymbol{\zeta} = \boldsymbol{M}_{t+1}\hat{\boldsymbol{X}}^{i}_{t+1|t} - \boldsymbol{p}_{t+1}$.

To solve (23), a robust kernel function called the Huber kernel function is used. This function combines the ℓ_1 and the ℓ_2 norm functions, making it more robust than using the pure ℓ_2 norm (least squares method) function alone. The Huber kernel function is defined as follows:

$$\rho(\zeta_k) = \begin{cases} \frac{1}{2}\zeta_k^2 & \text{for } |\zeta_k| < \gamma \\ \gamma|\zeta_k| - \frac{1}{2}\gamma^2 & \text{for } |\zeta_k| \geq \gamma \end{cases} \tag{24}$$

where γ is a threshold and usually chosen as 1.345 [6].

Since $\rho(\zeta_k)$ is differentiable, the minimization of (23) can be solved in this implicit equation:

$$\sum_{k=1}^{3N+m} \phi(\zeta_k) \frac{\partial \zeta_k}{\partial \hat{\boldsymbol{X}}^{i}_{t+1|t}} = 0 \tag{25}$$

where $\phi(\zeta_k) = \rho'(\zeta_k)$. Define $\psi(\zeta_k) = \phi(\zeta_k)/\zeta_k$ and $\boldsymbol{\Psi} = \text{diag}[\psi(\zeta_k)], k = 1, \ldots, 3N+m$. Then (25) can be rewritten as:

$$\boldsymbol{S}_{t+1}^{\top} \boldsymbol{\Psi} \left(\boldsymbol{S}_{t+1}\hat{\boldsymbol{X}}_{t+1|t} - \boldsymbol{p}_{t+1}\right) = \boldsymbol{0} \tag{26}$$

The solution is obtained by iterating

$$\hat{\boldsymbol{X}}^{i}_{t+1|t^{(n+1)}} = \hat{\boldsymbol{X}}^{i}_{t+1|t} + \boldsymbol{K}^{(n)}_{t+1}\left[\boldsymbol{z}^{ij}_{t+1} - \boldsymbol{h}(\hat{\boldsymbol{X}}^{i}_{t+1|t})\right] \tag{27}$$

where

$$\begin{aligned} \boldsymbol{K}^{(n)}_{t+1} &= \hat{\boldsymbol{\Sigma}}^{i}_{t+1|t^{1/2}} \boldsymbol{\Psi}_{\boldsymbol{X}}^{-1} \hat{\boldsymbol{\Sigma}}^{i}_{t+1|t^{1/2}} {\boldsymbol{H}^{i}_{t+1}}^{\top} \\ &\quad \left(\boldsymbol{H}^{i}_{t+1}\hat{\boldsymbol{\Sigma}}^{i}_{t+1|t^{1/2}} \boldsymbol{\Psi}_{\boldsymbol{X}}^{-1} \hat{\boldsymbol{\Sigma}}^{i}_{t+1|t^{1/2}} {\boldsymbol{H}^{i}}_{t+1^{\top}}\right. \\ &\quad \left. + {\boldsymbol{R}_{ij,t+1}}^{1/2} \boldsymbol{\Psi}_{\boldsymbol{z}}^{-1} {\boldsymbol{R}_{ij,t+1}}^{1/2}\right)^{-1} \end{aligned} \tag{28}$$

n means the iteration times. The measurement residual part $\boldsymbol{\Psi}_{\boldsymbol{z}}$ and the state prediction residual part $\boldsymbol{\Psi}_{\boldsymbol{X}}$ come from the decomposition of $\boldsymbol{\Psi}$:

$$\boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{\Psi}_{\boldsymbol{z}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Psi}_{\boldsymbol{X}} \end{bmatrix} \tag{29}$$

$\boldsymbol{\Psi} = \boldsymbol{I}$ when $n = 0$. The convergence of the iteration appears when ψ function is non-increasing. Practically, the iteration will stop when $\| \hat{\boldsymbol{X}}^{i}_{t+1|t^{(n+1)}} - \hat{\boldsymbol{X}}^{i}_{t+1|t^{(n)}} \|$ is small enough. With $\boldsymbol{\Psi}_{\boldsymbol{X}}$, the covariance is updated by:

$$\hat{\boldsymbol{\Sigma}}_{t+1|t} = (\boldsymbol{I} - \boldsymbol{K}_{t+1}\boldsymbol{H}^{i}_{t+1})\hat{\boldsymbol{\Sigma}}^{i}_{t+1|t^{1/2}} \boldsymbol{\Psi}_{\boldsymbol{X}}^{-1} \hat{\boldsymbol{\Sigma}}^{i}_{t+1|t^{1/2}} \tag{30}$$

Equation (27), (28) and (30), which represents the M-estimation, is formally similar to the extension Kalman filter (EKF). As $\boldsymbol{\Psi} \to \boldsymbol{I}$, the M-estimation reduces to the EKF. The difference lies in the measurement update. When accepting a biased measurement, the EKF weights it as a normal measurement. Nevertheless, the M-estimation reweights it by $\boldsymbol{\Psi}_{\boldsymbol{z}}$ so that the impact of the biased measurement can be reduced by allocating lower weights, making the state estimation less affected by biased measurements.

C. Communication Update

With our proposed time propagation update and measurement update methods, our DCL method can robustly work even under the failed communication and biased measurement environment. Nevertheless, we can also use the communication information to further improve our performance if the communication is not totally failed. Specifically, the robot i uses CI to fuse states estimated by itself and other robots communicating with it. The fusion result is used as the estimated state of the robot i. CI is an information fusion method shown in [10] and omitted for brevity here.

V. Simulation Experiments

In order to validate the performance of the proposed algorithm, it is essential to create scenarios that simulate failed communication and biased measurements, ensuring their resemblance to real-world environments. Additionally, we conduct a comparative analysis of our algorithm with several existing methods to showcase its robustness in these challenging scenarios.

A. Setup

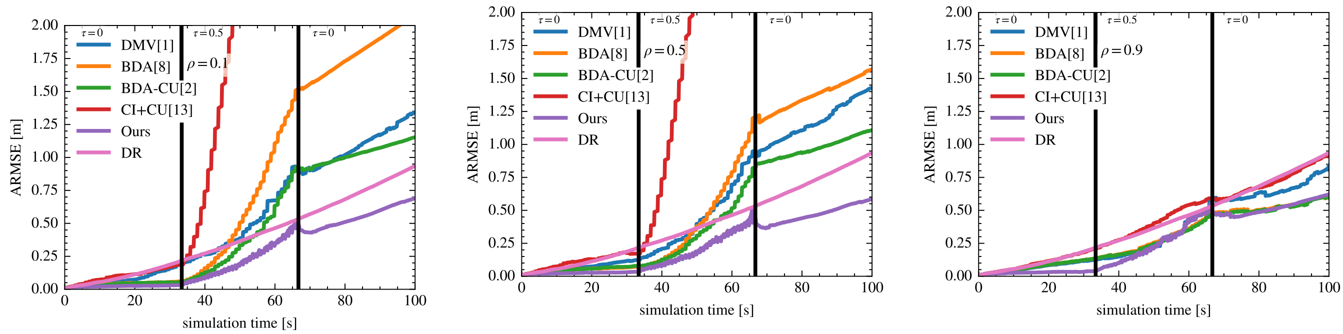
To accurately simulate real-world situations, it is essential to design scenarios that closely resemble them. One way to simulate the communication environment is by introducing a fixed probability of failed communication, denoted as $\rho \in \{0.1, 0.5, 0.9\}$, which indicates the level of failed communication. By incorporating low, medium, and high levels of failed communication, a wide range of scenarios can be covered. To simulate biased measurements, which can occur due to factors like electromagnetic interference, environmental conditions, or hardware malfunctions, two situations can be considered: sudden occurrence and gradual increase of biased measurements with a certain occurrence probability. The occurrence probability of biased measurements can be denoted as τ and can take values in the range $[0, 0.5]$.

Table I lists important parameters used in our simulations. Specifically, 6 robots are used in our simulation with fixed initial states and covariances. They are driven by velocities v and angular velocities ω, which is generated in (4) with nominal input velocities $\bar{v}$ and variance coefficient σ_v, as well as angular velocities $\bar{\omega}$ and variance coefficient σ_ω. In order to show the robustness of our method to varying robot movements, we assign different σ_ω (as shown in Table I) to generate greater randomness in the rotation of the robots in experiment 1, causing the robots to exhibit varied trajectories. The parameters of the normal distribution followed by $\bar{v}_{j,t}, \bar{\omega}_{j,t}$ come from:

$$\mu_{\bar{v}} = (v_{\max} + v_{\min})/2 \qquad \mu_{\bar{\omega}} = (\omega_{\max} + \omega_{\min})/2 \tag{31}$$

$$\sigma_{\bar{v}} = (v_{\max} - \mu_v)/3 \qquad \sigma_{\bar{\omega}} = (\omega_{\max} - \mu_\omega)/3 \tag{32}$$

where the subscript $\max, \min$ means the maximum velocity and the minimum velocity. Specific values are presented in Table I.



(a) $\rho = 0.1$ (b) $\rho = 0.5$ (c) $\rho = 0.9$

Fig. 1. ARMSE of robots' state estimation over time with sudden $\tau = 0.5, 0$ and $\rho \in \{0.1, 0.5, 0.9\}, \sigma_\omega = 0.3$

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Sample time Δt	0.1s
Range of $\bar{v}, \bar{\omega}$	$[0, 0.2](\mathrm{m/s}), [-1, 1](\mathrm{Rad/s})$
$\bar{v}_{j,t}, \bar{\omega}_{j,t}$	$\mathcal{N}(0.1, (1/30)^2), \mathcal{N}(0, (1/3)^2)$
σ_v, σ_ω for v, ω in Eq.4	0.7, 0.3 for experiment 1 and 2; 0.7, 0.8 for experiment 1 only
Measurement noise $\delta \boldsymbol{z}_{ij,t+1}$	$\mathcal{N}\left([0,0,0]^\top,\ \mathrm{diag}(0.05\mathrm{m}, 0.05\mathrm{m}, \pi/180\mathrm{Rad})^2\right)$
Sensing distance and bearing range	$(0, 5)(\mathrm{m}),\ [-\pi, \pi](\mathrm{Rad/s})$
Bias coefficient R_v	$0.27(t-29.7)$ in the experiment 1 and 0.5 in the experiment 2
Bias	$R_v \times (1.5\mathrm{m}, 1.5\mathrm{m}, \pi/6\mathrm{Rad})$
Communication Rate	$10\Delta t$
Initial states of robots	$[0, 12, 0]^\top, [1, 9, 0]^\top, [0, 6, 0]^\top$ $[1, 3, 0]^\top, [0, 0, 0]^\top, [1, -3, 0]^\top$
Initial covariance of each robot	$\mathrm{diag}(0.05^2, 0.05^2, (\pi/180)^2)$

The bias is added in the normal measurement. 20 Monte Carlo runs are executed. Within the scope of each run, all robots move simultaneously while concurrently performing measurements and engaging in inter-robot communication.

To demonstrate the superior performance of our algorithm, we have selected four state-of-the-art DCL methods for comparison, including the DMV [13], the BDA [12], the BDA-CU [4] and the CI+CU [17]. Moreover, to show the impacts of measurements, Dead Reckoning (DR) is also used for comparison.

B. Metric

To evaluate the accuracy of the estimation, we utilize the average root mean square error (ARMSE) as an index to quantify the disparity between the ground-truth and the estimation. The ARMSE is denoted as:

$$\mathrm{ARMSE}_t = \frac{1}{20N} \sum \sum_{i=1}^{N} \mathrm{RMSE}_i \tag{33}$$

$$= \frac{1}{20N} \sum \sum_{i=1}^{N} \sqrt{\frac{1}{2}\left[\left(\hat{x}_{i,t}^i - x_{i,t}\right)^2 + \left(\hat{y}_{i,t}^i - y_{i,t}\right)^2\right]} \tag{34}$$

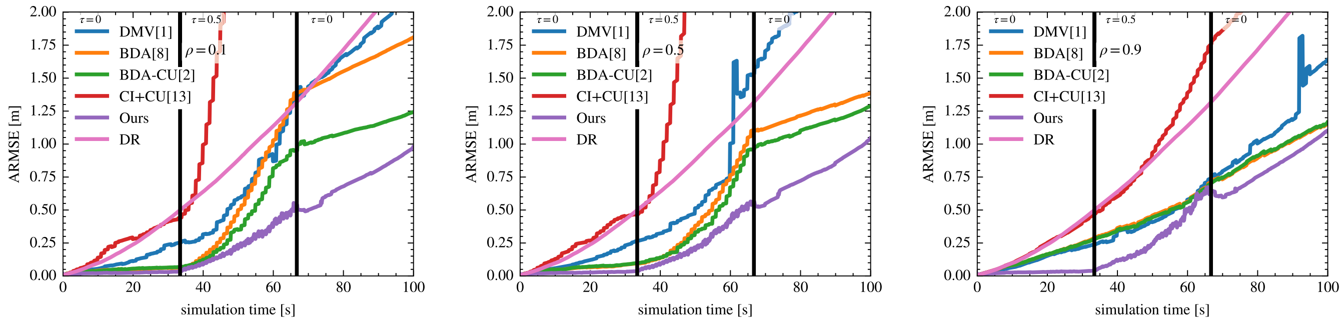
which consists of averaging the number of robots N and 20 Monte Carlo runs.

C. Experiment Results

1) Experiment 1: Experiment on Fixed Probability of Failed Communication and Sudden Occurrence and Absence Probability of Increased Biased Measurements: The first type of simulation is conducted in a specific environment where $\tau = 0$ for the initial 33.3 seconds and then changes to 0.5 for the next 33.3 seconds and finally changes to 0 for the last 33.3 seconds. Additionally, the level of bias in the measurements gradually increases over time, as listed in Table I. This scenario simulates a situation where the exteroceptive sensor's performance suddenly deteriorates, resulting in the measurement data with increased bias. In each simulation, the probability of failed communication remains constant, and ρ is held at a fixed value. Figs. 1 and 2 illustrate results of the simulation experiments conducted under various conditions and Table II displays the results obtained from Fig. 1 averaged over each 33.3 seconds.

The first observation from Figs. 1(a) and 2(a) is that the ARMSE of the proposed algorithm (indicated by the purple line) exhibits the slowest growth rate during the initial 33.3 seconds, approaching nearly zero, even when communication is severely disrupted ($\rho = 0.9$). In contrast, the ARMSE of other methods that couple communication with measurements increases with ρ. This demonstrates the remarkable robustness of our proposed multi-centralized framework which decouples measurement and communication. By decoupling these two components, the failed communication does not impact our measurement update if there is measurement obtained. Additionally, accurately approximating the distribution of the actual input velocity of other robots is also crucial in achieving robustness, as it enables a reasonable estimation of the state of other robots.

The second observation from Figs. 1(b) and 2(b) is that the ARMSE of the proposed algorithm exhibits the slowest growth rate during the next 33.3 seconds when the probability of failed communication ρ is at 0.1 and 0.5. When the level of the failed communication is not too high ($\rho \leq 0.5$), nearly all the methods can use the measurement to update. However, the biased measurement leads to an increase in the ARMSE when updating with biased measurements. Our M-estimation based measurement update can mitigate this increase by assigning a small weight to the biased measurement, thus weakening the impact of bias on the state estimation. Comparatively, the ARMSE of the BDA-CU is lower than that of the BDA when $\rho = 0.1$ and 0.5. As concluded in [4], the effectiveness of the CU method in mitigating the impact of bias is validated. For the CI+CU algorithm, despite



(a) $\rho = 0.1$ (b) $\rho = 0.5$ (c) $\rho = 0.9$

Fig. 2. ARMSE of robots' state estimation over time with sudden $\tau = 0.5, 0$ and $\rho \in \{0.1, 0.5, 0.9\}, \sigma_\omega = 0.8$

TABLE II
TIME-AVERAGED ARMSE OF STATE ESTIMATION OF ROBOTS

	DMV[1]			BDA[8]			BDA-CU[3]			CI+CU[13]			Ours			DR		
$\rho \setminus \tau$	0	0.5	0	0	0.5	0	0	0.5	0	0	0.5	0	0	0.5	0	0	0.5	0
0.1	0.077	0.464	1.083	0.042	0.596	1.787	0.040	0.367	1.024	0.102	2.533	6.362	**0.024**	**0.219**	**0.545**	0.099	0.368	0.727
0.5	0.066	0.436	1.168	0.049	0.448	1.368	0.051	0.333	0.968	0.108	2.432	6.472	**0.025**	**0.211**	**0.467**	0.099	0.368	0.727
0.9	0.070	0.271	0.651	0.074	0.268	0.527	0.074	0.260	0.518	0.100	0.403	0.720	**0.028**	**0.248**	**0.517**	0.099	0.368	0.727

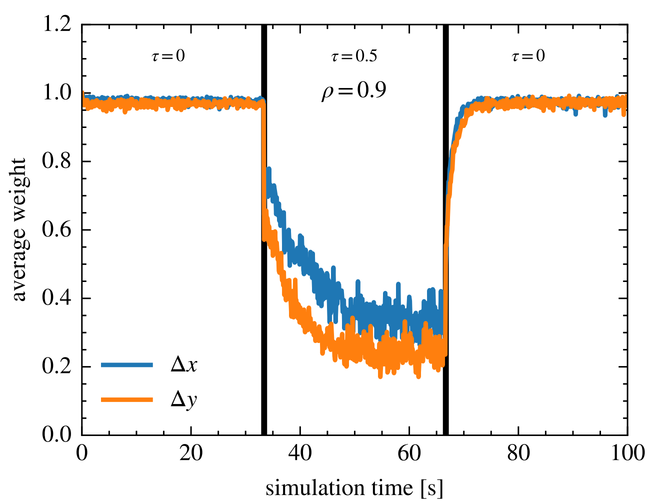


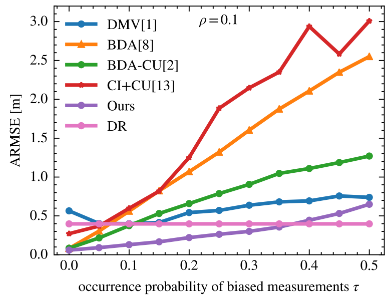
Fig. 3. Average weight of measurement in the M-estimation over time with sudden $\tau = 0.5, 0$ and $\rho = 0.9$

being a FDE-based method, it exhibits the highest ARMSE. One possible reason for this is the challenge of determining the threshold for FDE without prior information about the bias. We set thresholds utilized in the CI+CU after conducting multiple trials; specifically, $d_{\text{TOSS}} = 0.1, d_{\text{CU}} = 0.025$. Additionally, the bias in our scenario is not fixed but increases over time, further contributing to the higher ARMSE.

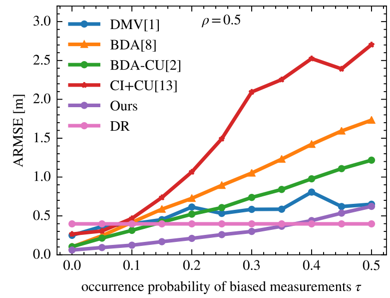
The third observation from Figs. 1(c) and 2(c) is that the performance of our algorithm becomes worse than other methods from 60 to 66.6 seconds when communication is severely disrupted ($\rho = 0.9$). This can be explained by the fact that algorithms that couple measurement and communication, such as the DMV, the BDA and the BDA-CU, do not conduct measurement updates when communication is barely blocked. Although they can obtain measurement about other robots, the estimated states of other robots cannot be obtained through communication, so the biased measurements will not be used to update the estimated state. While preventing the measurement update with failed communication may benefit these localization methods in certain specific scenarios, it ultimately undermines their robustness in dealing with complex environments. A primary challenge lies in accurately predicting whether the measurement is biased. Once the bias is removed, failed communication also prevents the measurement update, causing the robot to rely solely on the time propagation update. As a result, rapid error accumulation occurs, ultimately causing the localization to no longer converge. This phenomenon is apparent during the last 33.3 seconds in both Figs. 1(c) and 2(c). Notably, when the bias is removed, our method outperforms all others, demonstrating the least ARMSE. Moreover, it's also observed that our performance is better in Fig. 2(c). The primary reason is that, due to communication blockages, other methods rarely perform measurement updates. The augmented angular velocity noise consequently contributes to the rapid error accumulation in the time propagation update.

The fourth observation from Table II indicates that, overall, our method exhibits superior performance compared to other methods in average. When ρ is increased, our performance is only minimally affected compared to other methods. Additionally, in terms of measurements, our method demonstrates the least sensitivity to biased measurements.

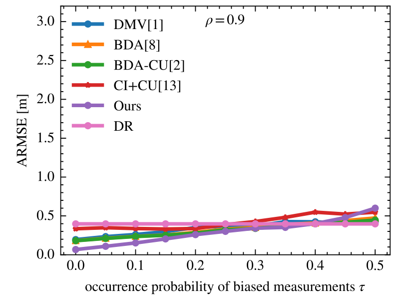
To show the working mechanism of our M-estimation process, Fig. 3 illustrates the effect of the M-estimation for Fig. 1(c). The average weight (averaged over N and 20 Monte Carlo running times) about the measurement over simulation times is shown. Specifically, the average weight comes from all the unbiased measurement during the first 33.3 and the last 33.3 seconds and all the biased measurement during the middle 33.3 seconds. In Fig. 3, the average weight is close to 1 during the first 33.3 and the last 33.3 seconds while they decrease during the middle 33.3 seconds. That's because when biased measurement arrives, $|\zeta_k|$ becomes larger. If $|\zeta_k|$ is larger than γ, the proposed



(a) $\rho = 0.1$



(b) $\rho = 0.5$



(c) $\rho = 0.9$

Fig. 4. ARMSE of robots' state estimation over each τ with $\rho \in \{0.1, 0.5, 0.9\}$

algorithm treats the measurement as a bias and gives it a lower weight.

2) Experiment 2:Experiment on Fixed Probability of Failed Communication and Increased Occurrence Probability of Biased Measurements: In this situation, we want to mimic an extreme scenario where the occurrence of bias measurement gradually increased. With an increased occurrence probability of bias measurement, we are able to simulate a process of randomly generating biased measurements. The whole simulation time is also 100 seconds and the bias is fixed as listed in Table I. Fig. 4 shows the results of averaged over 20 Monte Carlo running times.

We can observe from Fig. 4 that our proposed algorithm outperforms other methods. It exhibits remarkable performance when the occurrence probability of biased measurements τ is less than 0.4, significantly improving estimation accuracy regardless of failed communication. In contrast, most other methods deteriorate when τ exceeds 0.1, achieving better performance only when the measurement update is nearly disabled due to high level of the failed communication (Fig. 4(c)). Through the fusion of a multi-centralized framework and M-estimation, our proposed algorithm demonstrates strong robustness against scenarios with failed communication and biased measurements.

VI. Conclusion

In this study, we have presented a robust cooperative localization algorithm that utilizes a multi-centralized framework and the M-estimation to address challenges arising from failed communication and biased measurements. Through simulation experiments, we have demonstrated superior localization accuracy in challenging scenarios. The next phase of our research involves conducting real-world experiments to further validate the proposed algorithm.

References

[1] N. Trawny, S. I. Roumeliotis, and G. B. Giannakis, "Cooperative multi-robot localization under communication constraints," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2009, pp. 4394–4400.

[2] E. D. Nerurkar, K. X. Zhou, and S. I. Roumeliotis, "A hybrid estimation framework for cooperative localization under communication constraints," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2011, pp. 502–509.

[3] T.-K. Chang, K. Chen, and A. Mehta, "Resilient and consistent multirobot cooperative localization with covariance intersection," *IEEE Trans. Robot.*, vol. 38, no. 1, pp. 197–208, Feb. 2022.

[4] X. Wang, S. Sun, T. Li, and Y. Liu, "Fault tolerant multi-robot cooperative localization based on covariance union," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 7799–7806, Oct. 2021.

[5] P. J. Huber, *Robust Statistics*, vol. 523. Hoboken, NJ, USA: Wiley, 2004.

[6] C. Karlgaard and H. Schaub, "Comparison of several nonlinear filters for a benchmark tracking problem," in *Proc. AIAA Guid., Navigation, Control Conf. Exhibit*, 2006, pp. 1–9.

[7] S. S. Kia, J. Hechtbauer, D. Gogokhiya, and S. Martínez, "Server-assisted distributed cooperative localization over unreliable communication links," *IEEE Trans. Robot.*, vol. 34, no. 5, pp. 1392–1399, Oct. 2018.

[8] J. M. Walls, A. G. Cunningham, and R. M. Eustice, "Cooperative localization by factor composition over a faulty low-bandwidth communication channel," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 401–408.

[9] S. Roumeliotis and G. Bekey, "Distributed multirobot localization," *IEEE Trans. Robot. Automat.*, vol. 18, no. 5, pp. 781–795, Oct. 2002.

[10] L. C. Carrillo-Arce, E. D. Nerurkar, J. L. Gordillo, and S. I. Roumeliotis, "Decentralized multi-robot cooperative localization using covariance intersection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 1412–1417.

[11] H. Li, F. Nashashibi, and M. Yang, "Split covariance intersection filter: Theory and its application to vehicle localization," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1860–1871, Dec. 2013.

[12] L. Luft, T. Schubert, S. I. Roumeliotis, and W. Burgard, "Recursive decentralized localization for multi-robot systems with asynchronous pairwise communication," *Int. J. Robot. Res.*, vol. 37, no. 10, pp. 1152–1167, 2018.

[13] J. Zhu and S. S. Kia, "Cooperative localization under limited connectivity," *IEEE Trans. Robot.*, vol. 35, no. 6, pp. 1523–1530, Dec. 2019.

[14] C. Chen and S. S. Kia, "Cooperative localization using learning-based constrained optimization," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 7052–7058, Jul. 2022.

[15] A. Bahr, M. R. Walter, and J. J. Leonard, "Consistent cooperative localization," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2009, pp. 3415–3422.

[16] P. Zhu and W. Ren, "Multi-robot joint localization and target tracking with local sensing and communication," in *Proc. Amer. Control Conf.*, 2019, pp. 3261–3266.

[17] J. Klingner, N. Ahmed, and N. Correll, "Fault-tolerant covariance intersection for localizing robot swarms," *Robot. Auton. Syst.*, vol. 122, 2019, Art. no. 103306.

[18] J. Al Hage, M. E. El Najjar, and D. Pomorski, "Multi-sensor fusion approach with fault detection and exclusion based on the Kullback–Leibler divergence: Application on collaborative multi-robot system," *Inf. Fusion*, vol. 37, pp. 61–76, 2017.

[19] Z. El Mawas, C. Cappelle, and M. El Badaoui El Najjar, "Fault tolerant cooperative localization using diagnosis based on Jensen Shannon divergence," in *Proc. 25th Int. Conf. Inf. Fusion*, 2022, pp. 1–8.

[20] S. Thrun, "Probabilistic robotics," *Commun. ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[21] D. Bertsekas and J. N. Tsitsiklis, *Introduction to Probability*, vol. 1. Nashua, NH, USA: Athena Sci., 2008.