FPECMV: Learning-based Fault-Tolerant Collaborative Localization under Limited Connectivity

Rong Ou, Guanqi Liang, and Tin Lun Lam[†]

***Abstract*— Collaborative localization (CL) has garnered substantial attention in the field of robotics in recent years. Nonetheless, conventional CL algorithms have faced challenges when dealing with practical issues such as spurious sensor data and limited or discontinued observation and communication in real-world settings. This paper proposes a fault-tolerant practical estimated cross-covariance minimum variance update method (FPECMV) designed to tackle these challenges under limited connectivity. The proposed algorithm uses a CNN-based method to evaluate confidence, along with a fault isolation module to identify faults and manage spurious data in real time. The proposed fault isolation module utilizes relative measurement information that randomly occurs, without requiring high observation and communication prerequisites. Notably, the algorithm takes into account correlations among agents to maintain consistency in localization filters and attain accurate localization despite constraints posed by limited connectivity. To evaluate the performance of the proposed algorithm, experiments were conducted in a collaborative multi-robot environment with spurious sensor data and limited connectivity, using both the BULLET simulation and physical mobile robots. The experimental results indicate that the overall localization performance of the proposed algorithm is improved by 21.0% compared to the state of the art. The experiment results demonstrate the effectiveness of our algorithm in localizing group agents in challenging and intricate scenarios with limited connectivity and spurious sensor data.**

I. Introduction

Multi-agent systems have gained significant attention owing to their notable advantages in terms of fault tolerance, extensive coverage, and high estimation accuracy through data fusion. These systems exhibit tremendous potential in diverse robotic applications, including swarm [1], SLAM [2], and modular robotics [3]. To effectively utilize these applications, precise localization plays a crucial role. Unlike traditional centralized localization solutions, distributed collaborative localization (DCL) achieves precise location by combining relative measurement information among agents and their location data [4], and has become a popular area of research in recent years due to its inherent advantages of flexibility and accuracy [5], [6].

In the context of collaborative localization, it is possible to improve the accuracy of the localization of mobile agents by establishing connectivity [7] with a measurement network composed of multiple agents. This enables each agent to receive real-time relative measurement feedback from its neighboring agents, thereby enhancing its localization. However, achieving such tight connectivity requirements imposes significant demands on various resources, including measurement, communication, computational, and storage resources [7], [8]. In practical situations, connectivity is often limited due to various factors, such as communication failures among agents [9], [10], or agents being in observation blind zones caused by external events, such as limited communication ranges [11], [12], obstacle blocking [13], [14], and limited measuring ranges [15]–[18]. In the past, various algorithms have been developed, such as those proposed in [19]–[23], which aim to relax the connectivity requirements by enabling agents to communicate only when measuring each other. However, these algorithms tend to adopt conservative update strategies by ignoring correlations among agents and failing to leverage measurement updates for the benefit of other agents within the system network. For instance, the CIF-based methods discussed in [22] adopt conservative bounds to account for the absence of cross-covariance information, thereby resulting in highly conservative estimates. On the other hand, in the SA-split-EKF method proposed by [8], each robot determines its position in the global coordinate system through local dead reckoning (DR) and subsequently adjusts its pose estimate upon receiving a relative measurement update message from the server. Notably, the computational and storage expenses per robot are of order $O(1)$ for any team size. However, it is worth noting that the algorithm of [8] requires the existence of a server within the agent group, and the centralized structure of the algorithm renders it unusable in the event of a server malfunction. Zhu [7] has proposed the use of the PECMV method, which is capable of processing relative measurements to correct both the local state and the corresponding error covariance without explicit knowledge of the cross-covariance. In addition, this algorithm has been optimized to balance localization accuracy and computational speed, resulting in reduced spatial complexity. To the best of our knowledge, PECMV is the state of the art for handling distributed collaborative localization in limited connected scenarios. However, these works, including PECMV, are designed for ideal limited connected scenarios. Once they are employed in real complex environments, common failures such as spurious sensor data [24] can lead to significant localization errors or even failures.

This work was supported in part by the National Natural Science Foundation of China under Grant 62073274; in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2023B1515020089; and in part by the Shenzhen Institute of Artificial Intelligence and Robotics for Society under Grant AC01202101103.

Rong Ou, Guanqi Liang, and Tin Lun Lam are with the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, and Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS), Shenzhen, Guangdong, 518172, P.R. China.

Rong Ou and Guanqi Liang contributed equally to this work.

[†]Corresponding author is Tin Lun Lam `tllam@cuhk.edu.cn`.

In intricate real-world scenarios, robots may experience their own instability and uncertain external conditions, resulting in unexpected situations, including spurious sensor data. Such inaccurate data can significantly degrade localization accuracy and cause significant deviations from the true value. Hence, fault tolerance is critical for collaborative localization. Previous research has focused on troubleshooting through multi-sensor information fusion and fault detection for fault-tolerant and accurate collaborative localization [24]–[26]. Although these works have yielded positive results in purely fault-tolerant localization, their fault-tolerance is based on the premise that each agent maintains continuous observation and communication in the agent network. In the case of limited connectivity, where data communication and observations are limited, resulting in a small sample space, these existing efforts on fault tolerance do not work properly. In summary, fault-tolerant collaborative localization under limited connectivity remains unresolved and challenging.

In this paper, we introduce a fault-tolerant distributed collaborative localization algorithm. Building on the previously developed PECMV [7] algorithm, we propose a learning-based confidence evaluation method and fault isolation module to effectively handle measurements with spurious sensor data. It is worth emphasizing that the proposed fault isolation module utilizes relative measurement information that randomly occurs, without requiring high observation and communication prerequisites. Our algorithm accounts for past correlations among agents, ensuring localization filter consistency and accurate localization despite limited connectivity and spurious sensor data faults. Furthermore, our algorithm does not restrict the type of sensors and supports a general measurement model. We evaluated the performance of our proposed approach using both simulations and physical experiments with vision-based measurement mobile robots. The results demonstrate that our proposed algorithm outperforms the PECMV algorithm in challenging and complex scenarios that involve limited connectivity and spurious sensor data faults.

The rest of the paper is organized as follows. Section II outlines the system models definition. Section III presents the proposed algorithm. Section IV validates the proposed algorithm through a series of simulations and experiments. Finally, the last section concludes the paper and discusses future research directions.

II. System Models Definition

A. Robot Motion Model

Let's consider a group of N agents equipped with communication and computation capabilities. The mathematical expression for the motion model of each agent $i \in V = \{1, ..., N\}$ at discrete time $t \in Z^+$ is given by

$$x^i(t) = f(x^i(t-1), u^i_m), \quad x^i \in R^{n^i}, \tag{1}$$

which describes the state of agent i at time step t, denoted by $x^i(t) \in R^{n^i}$, encompasses the agent's position, velocity, and orientation. The self-motion measurement command, $u^i_m = u^i(t) + \nu^i_u$, is obtained from sources such as odometry or an inertial measurement unit (IMU), where ν^i_u represents the measurement noise.

B. The Measurement Model

During the mission horizon, the agents' localization accuracy is reduced by noise in the self-motion measurements. If the availability of absolute measurements for correcting dead-reckoning-based localization is limited, joint processing of relative measurements between two agents is utilized to limit the error and enhance the accuracy. The relative measurement, which may include relative range, bearing, pose, or a combination of these, is denoted by $z^i_j(t)$ and is obtained by agent i from agent j at time t,

$$z^i_j(t) = h(x^i(t), x^j(t)) + \nu^i(t), \quad z^i_j(t) \in R^{n^i_z}, \tag{2}$$

where $h(x^i(t), x^j(t))$ is the measurement model, and $\nu^i(t) \in N(0, R^i(t))$ is the zero mean Gaussian measurement noise with covariance matrix $R^i(t) \in S^{++}_{n^i_z}$. At each time step $t \in Z^+$, each agent applies a local filter to estimate its own state $\hat{x}^{i-}(t) \in R^{n^i}$ and the corresponding error covariance matrix $P^{i-}(t) \in S^{++}_{n^i}$ using its motion model and sporadic access to absolute measurements from sources such as known landmarks. We denote the prior belief of agent i at time t as $bel^{i-}(t) = (\hat{x}^{i-}(t), P^{i-}(t))$. When there is no inter-agent measurement available to update the local belief, the propagated belief is considered the updated belief, i.e., $bel^{i+}(t) = bel^{i-}(t) = (\hat{x}^{i-}(t), P^{i-}(t))$. However, when a relative measurement is accessible, the local belief is updated using fusion approaches. These approaches are briefly described below.

III. Algorithm

A. Application Scenarios

The collaborative localization algorithm is improved based on the practical estimated cross-covariance minimum variance update method (PECMV) [7], and we propose an FPECMV method for application scenarios with limited connectivity, and spurious sensor data, which can fuse relative measurement data from multiple robots. The conditions of the scenario are described as follows.

- Limited connectivity: In the scenario, the observation and communication of robots are limited and discontinuous. The robot's pose is not observed at every moment due to obstacles, the robot's observation range, or robot malfunction. And robots may have communication failures due to environmental disturbances.
- Spurious sensor data: There are measurement fault in certain robots that are measuring relative poses.

B. Algorithm update procedure:

Fig. 1 shows the flow chart of our fault-tolerant distributed collaborative localization algorithm, which comprises three principal modules: CNN-based confidence evaluation, Fault Isolation, and Collaborative Localization. Different from PECMV, we developed a fault isolation module capable of utilizing relative measurement that is occasionally obtained.

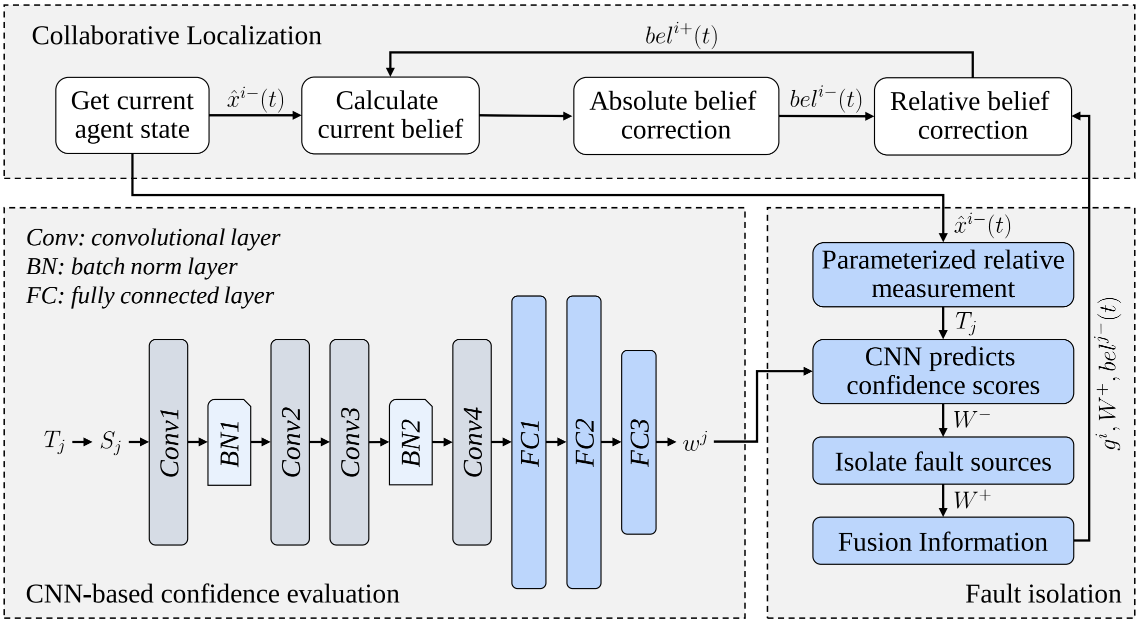

Fig. 1. The flowchart of the proposed FPECMV. The underlying collaborative localization module is built upon [7]. The proposed CNN-based confidence evaluation module computes the confidence score by analyzing the measurements. The proposed fault isolation module employs randomly occurring relative measurement data, without requiring high observation and communication prerequisites.

We take 5 frames of relative measurements from other robots at a time and combine them with the latest 5 frames of local states data to form the state matrix $T_j = [z_j^i(t_1), ..., z_j^i(t_5), x^{i-}(t_1), ..., x^{i-}(t_5)]$. Assuming a sensor frequency of 30HZ, each observation time is roughly 167ms, and a higher measurement frequency leads to a shorter observation time required by the algorithm. We parameterize T_j as the input vector S_k of the neural network according to Eq. (3) and Eq. (4), $T_j \in \mathbf{T}, S_j \in \mathbf{S}$.

Our network architecture consists of four convolutional layers followed by three fully connected layers. A batch normalization layer is placed between convolutional layer 1 and convolutional layer 2, and another batch normalization layer is placed between convolutional layer 3 and convolutional layer 4. These batch normalization layers are added to accelerate the convergence of the network. The activation functions of convolutional layers 1 and 2, as well as the two batch normalization layers, are Rectified Linear Unit (ReLU) functions. To prevent overfitting, we apply a dropout probability of 0.1 to the fully connected layer 1 and fully connected layer 2. The output layer uses the Sigmoid activation function $S(X) = \frac{1}{1+e^{-X}}$, which constrains the value of w_j to be within the range of [0,1]. The structure of our network is shown in Fig. 1.

We generated an augmented dataset in a simulated environment, which includes spurious sensor data and their corresponding confidence scores. The augmented dataset is generated based on the features extracted from the real environment, which also contains spurious sensor data. Specifically, 70% of the artificially generated spurious sensor data was allocated to the training dataset, while the remaining 30% was designated for the validation dataset.

During the training phase, the Adam optimizer is employed with a learning rate of 0.01 and an L2Decay value of 0.00005. A batch size of 32 is utilized for the training process, which continues for 80 epochs. After each epoch, the test accuracy is calculated, and the model parameters are saved persistently whenever the accuracy surpasses the accuracy achieved in the previous epoch. This approach facilitates the retention of the model with the highest accuracy, as opposed to the one obtained at the end of the final epoch.

The neural network functions as a rater and produces a self-adaptive confidence score $w^{j-} \in W^-$ for the observer. The observers are regarded as sources. We set a confidence score threshold β below which sources are deemed unreliable. Such sources are considered faulty, and they are excluded from subsequent information fusion steps to achieve fault isolation. Following fault isolation, we obtain $W^+ = [w^{1+}, ..., w^{j+}]$. The confidence score w^{j+} provided by the CNN network can reflect the potential presence of such data in relative measurements. By synthesizing the relative measurement state according to w^{j+}, the impact of spurious data can be greatly reduced. After normalizing the scores of each source and ensuring that their total sum equals 1, we merge the final data according to the confidence scores of all sources. We merge the information as follows: $g^i(t) = w^{j+} \times z_j^i(t_N), w^{j+} \in W^+$. Here, $g^i(t)$ denotes the relative measurement state of agent i.

$$\mathbf{S} = parameterization(\mathbf{T}) \tag{3}$$

$$Input = vectorization(\mathbf{S}) \tag{4}$$

$$labelW = vectorization(W) \tag{5}$$

After completing the fault isolation step, our CL module will commence operation. The local belief update procedure will commence in the following manner. We set the joint

```
Algorithm 1: FPECMV method
  Input:
  Relative measurements from robots set $K$: $\mathbf{T}$
  Isolation threshold: $\beta$
  The belief of last time: $bel^{i+}$
  The belief robots set $K$: $B$
  Output: Updated state of agent $i$
$x^{i-}(t) \leftarrow getCurrentState()$;
$A \leftarrow getCurrentMotion()$;
if $T$ is not None then
    foreach $j \in K,\ T_j \in \mathbf{T}, S_j \in \mathbf{S}$ do
        $S_j \leftarrow NormalizedAndMaped(T_j)$;
        $w^{j-} \leftarrow cnnPredict(S_j)$;
        $w^{j+} \leftarrow isolation(w^{j-}, \beta)$;
        $W^+[j] \leftarrow w^{j+}$;
    end
    foreach $w^{j+} \in W^+,\ z_j^i \in T,\ bel^{j-} \in B$ do
        $g^i \leftarrow g^i + w^{i+} * z_j^i(t_N)$;
        $bel^{j-} \leftarrow recalculateBelief(w^{i+}, bel^{j-})$
    end
$bel^{i+} \leftarrow moduleCL(bel^{i-}, bel^{j-}, z^i, A, g^i)$;
```

state of $\{i,j\}$ as $x_J(t) = (x^i(t)^\top, x^j(t)^\top)$, the joint belief of $\{i,j\}$ as $bel_J^-(t) = (\hat{x}_J^-(t), P_J^-(t))$, where

$$\hat{x}_J^-(t) = \begin{bmatrix}\hat{x}^{i-}(t)\\ \hat{x}^{j-}(t)\end{bmatrix}, \quad P_J^-(\mathbf{X}) = \begin{bmatrix}P^{i-} & P_{i,j}^-(t)\\ P_{i,j}^-(t)^\top & P^{j-}\end{bmatrix}. \quad (6)$$

The joint covariance matrix is subsequently set to

$$P_J^-(X) = \begin{bmatrix}P^{i-} & \mathbf{X}\\ \mathbf{X}^\top & P^{j-}\end{bmatrix}, \quad (7)$$

with the estimation of $\mathbf{X} = P_{i,j}^-(t)$. We obtain $\mathbf{X}^*$ from

$$\mathbf{X}^* = \arg\max_{\mathbf{X}} \det \begin{bmatrix}I_{n^i}\\ 0\end{bmatrix}^\top (P_J^-(X)^{-1} + H^{i^\top} R^{i^{-1}} H^i)^{-1} \begin{bmatrix}I_{n^i}\\ 0\end{bmatrix}, \quad (8)$$

subject to

$$\begin{bmatrix}P^{i-} & \mathbf{X}\\ \mathbf{X}^\top & P^{j-}\end{bmatrix} > 0. \quad (9)$$

It is noteworthy that $H^i = \begin{bmatrix}H_i^i & H_j^i\end{bmatrix}$ corresponds to the identity matrix I_{n^i}.

$P_{ij}^-(t)$ is computed by finding the value of $\mathbf{X}$ that results in the most conservative updated covariance. This optimization challenge is expressed as a convex matrix optimization with linear inequality constraints and we solve it using the method in PECMV [7]. Once $\mathbf{X}$ is obtained from Eq. (8), the belief is updated as

$$\text{bel}^{i+}(t) = (\hat{x}^{i+}(t), P^{i+}(t)), \quad (10)$$

for agent i is

$$\hat{x}^{i+} = \hat{x}^{i-} + K^i(z_j^i - \hat{z}_j^i), \quad (11)$$

$$P^{i+} = \begin{bmatrix}I_{n^i}\\ 0\end{bmatrix}^\top (P_J^-(\mathbf{X}^*)^{-1} + H^i R^{i^{-1}} H^i)^{-1} \begin{bmatrix}I_{n^i}\\ 0\end{bmatrix}, \quad (12)$$

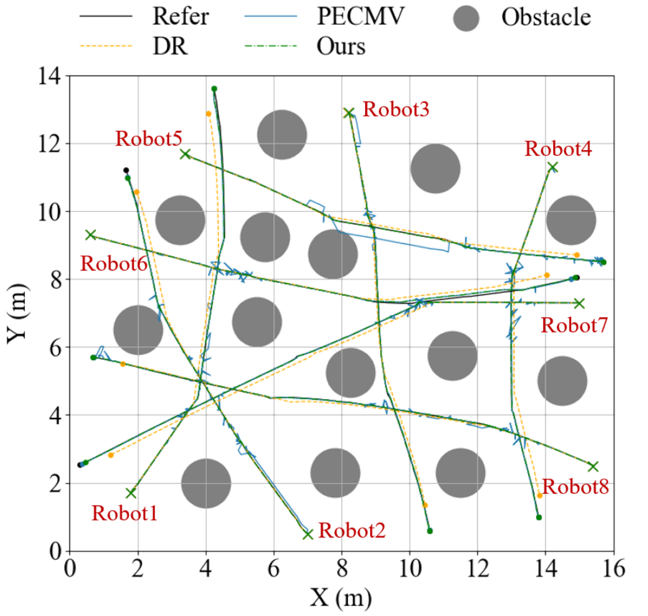

Fig. 2. Trajectory of Robots in simulation. The cross symbol represents the starting point and the solid circle symbol represents the ending point.

where

$$K^i = \begin{bmatrix}I_{n^i}\\ 0\end{bmatrix}^\top P_J^-(\mathbf{X}^*) H^{i^\top} \left(H^i P_J^-(\mathbf{X}^*) H^{i^\top} + R^i\right)^{-1}. \quad (13)$$

It satisfies $P^{i+}(t) \leq P^{i-}(t)$.

The decentralized CL algorithm based on FPECMV method is given by Algorithm 1.

IV. Simulations and Experiments

A. Simulation

The BULLET physics simulation engine is utilized to execute the simulation. The system under investigation consists of 8 mobile robots, and their equations of motion, which include linear acceleration $a^i(t)$, linear velocity $v^i(t)$, and angular velocity $w^i(t)$, are described as follows:

$$\begin{cases} a^i(t) = a_m^i(t) + \phi(t)\\ v^i(t) = v^i(t-1) + \delta t \times a^i(t-1)\\ w^i(t) = w^i(t-1) + \delta t \times a_w^i(t-1)\end{cases}, \quad (14)$$

where $i \in \{1,2,3,4,5,6,7,8\}$, $a_m^i(t)$ and $a_w^i(t)$ are measured linear and angular acceleration, while $\phi(t)$ is the corresponding measurement noises. We assume that the self-measurement noise of agents is 10% of the linear velocity and 5% of the angular velocity. In addition, the simulation employs relative pose measurements that are corrupted by a relative measurement noise with a standard deviation of [0.01m, 3 degrees]. Absolute range measurements with respect to landmarks that have known positions can also be obtained occasionally. During the simulation, a random spurious sensor data error was introduced when the robots measured each other's pose. The probability of generating spurious sensor data was set to 10%, which was consistent with the probability of the spurious sensor data occurring in our physical experiment. Additionally, the standard deviation of the error was set to 0.2m. Fourteen cylindrical obstacles

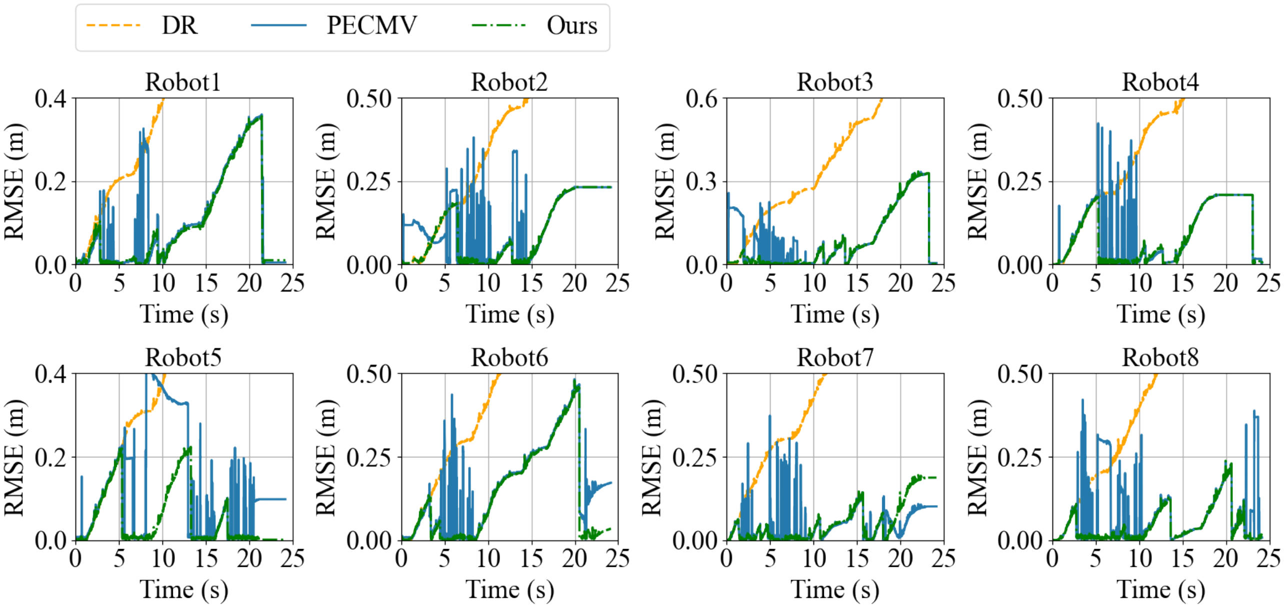

Fig. 3. RMSE over time of each robot position estimation for simulation scenario.

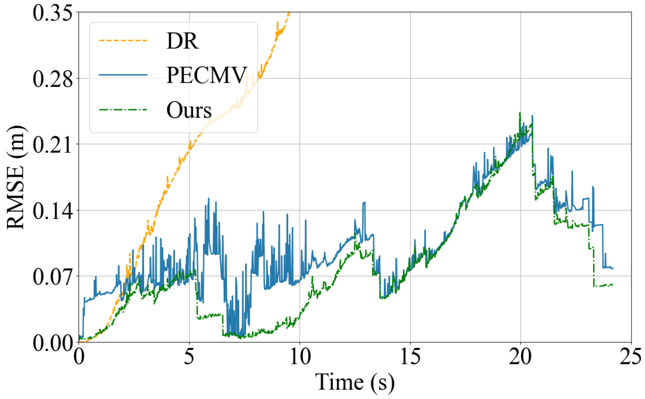

Fig. 4. RMSE of robots position estimation for simulation scenario.

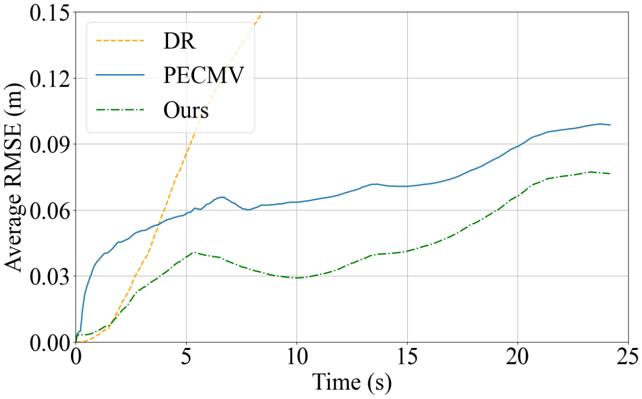

Fig. 5. Average RMSE of robots position estimation for simulation scenario.

with a diameter of 1 m and a height of 0.8 m were placed in the simulation environment, and they were used to limit the observation between robots. Its placement is shown in the Fig. 2. Finally, each robot obtained only one absolute measurement at the start point.

The trajectory diagram in Fig. 2 illustrates that the orange curve (DR) gradually deviates from the true value, as a result of the accumulated error of IMU. In contrast, the blue and green curves representing the PECMV and our proposed method, respectively, exhibit a correction that leads them to return to a trajectory closer to the true value (refer line) after observing each other. However, the presence of spurious sensor data causes the blue curve to exhibit more significant deviations compared to the green curve. Notably, the green curve (Ours) appears to be less affected by the spurious sensor data, especially during the mutual measurement phase, thus closely approximating the true value trajectory.

Based on the results presented in Fig. 3 and Fig. 4, it is evident that the RMSE curves of the PECMV algorithms exhibit notable fluctuations (ranging from approximately 0.1 to 0.4 meters) during the mutual observation period of the robots (e.g., 4-8 seconds, 13-15 seconds, and 21-24 seconds). These fluctuations are attributed to the occurrence of spurious data failures among the robots. Conversely, the RMSE curve of our proposed algorithm (shown in green) is relatively stable during mutual measurement, indicating that it is more resilient to the perturbations caused by measurement faults when faced with a considerable amount of spurious sensor data.

Moreover, the use of relative measurement information by the DCL algorithm significantly reduces the position errors of each robot during mutual observation. This is evident in Fig. 5, which shows the variation of the average RSME over time for all eight robots. The results clearly demonstrate the better positioning accuracy of our algorithm in comparison to the PECMV algorithm in this scenario.

B. Experiment

The experiment was conducted in an indoor area measuring 6 by 4 meters, using the wheeled robot named Spark

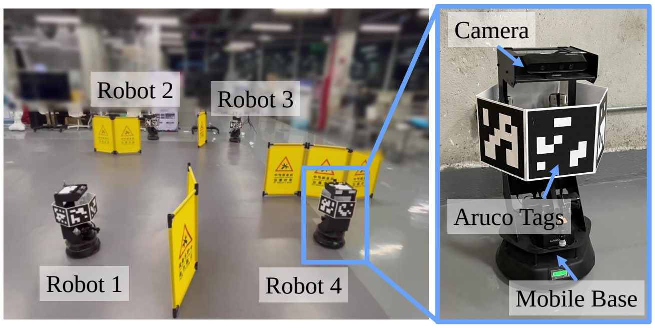

Fig. 6. Experimental environment and robot configuration.

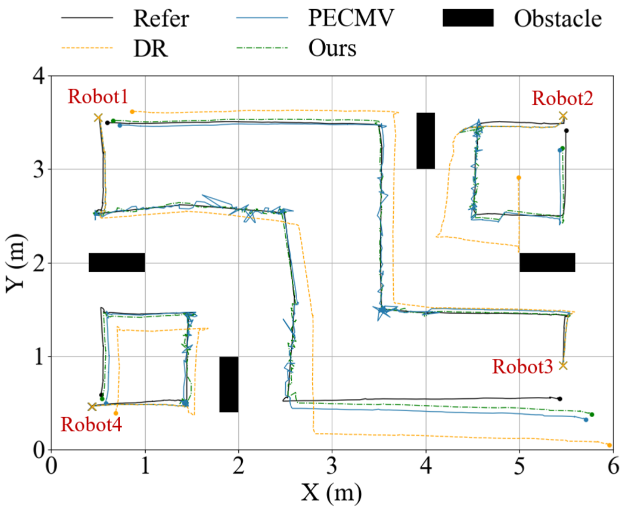

Fig. 7. Trajectory of robots in experiment. The cross symbol represents the starting point and the solid circle symbol represents the ending point.

equipped with an integrated IMU and camera and operated through the Robot Operating System (ROS), as illustrated in Fig. 6. The experiment entailed deploying four Spark robots, each commencing from a corner in Fig. 7 and subsequently following a predetermined trajectory to a specified end point at a forward speed of 0.2 m/s. A vision measurement system was employed to determine the relative position of the robots. This system comprised six stationary Aruco tags mounted around the robot, with the camera on top of the robot capturing the relative position and pose of the Aruco, which was then converted into the relative orientation of the robot. The effective measuring distance of Our Aruco-based vision measurement system is 6m. The Optitrack motion capture device was utilized to obtain a reference value for the experiment, recording the trajectory and pose of each robot. Additionally, four obstacles were strategically placed in the experimental environment to limit the observation between robots, as illustrated in Fig. 7. It is worth noting that each robot only receives an absolute measurement at the start point.

The robot's elevated center of gravity and limited chassis performance resulted in significant motion-induced jitter, leading to distorted or blurry images, especially when both the observer and the observed moved concurrently. Moreover, deficiencies in lighting conditions and erratic measurements of Aruco tags at certain angles posed realistic interferences or failure factors, leading to the generation of spurious data in vision measurements. As a result, we analyzed the statistics of spurious data occurrences in visual measurements, which showed an error probability of about 10% and a range of 10-30 cm. These findings corroborated the expected incidence of spurious sensor data conditions.

The experimental results, shown in Fig. 7, illustrate that the orange curve (DR) progressively diverges from the reference value due to cumulative errors in the IMU. In contrast, the blue and green curves (PECMV, Ours) effectively correct the trajectory when the robots measure each other. However, due to the impact of random spurious sensor data faults, the blue curve (PECMV) exhibit some degree of deviation, whereas the green curve (Ours) remains unaffected by the spurious data and more closely approximates the true value (represented by the black reference line generated by the motion capture system). The experimental results align with the simulation results.

Fig. 8 and 9 provides further insights into the experiment by presenting the RMSE results for the positions of the four robots under different algorithms. During the phase of mutual measurement among the robots, the errors in their positions were significantly reduced as anticipated. However, when subjected to the interference of the spurious data caused by measurement fault, the robots using the PECMV algorithms exhibited significant fluctuations in their position errors, indicating the influence of the faults. In contrast, the robots implementing our proposed algorithm demonstrated minimal fluctuations, which suggests that it is less affected by spurious sensor data faults and possesses good fault tolerance.

The reason for the curves partially overlap is that despite the existence of spurious data, the normal measurement results can still be obtained, and PECMV will correct the error when the normal relative measurement is obtained. However, it does not mean that PECMV can resist the interference of spurious sensor data. As shown by the RMSE curve of Robot4 in Fig. 8, around 60s, robot4 obtained a relative measurement state containing spurious data, which caused its error to increase sharply and maintained this error in the following time. On the contrary, our algorithm can effectively isolate the spurious data and correct the position error.

Moreover, Fig. 10 shows the average RMSE curve of the four robots, demonstrating a clear downward trend for our algorithm after acquiring relative measurement information. Notably, curve of our algorithm remained consistently lower than that of the PECMV over time, highlighting the effectiveness of our approach in practical scenarios. In the experiment, the average RMSE of our algorithm and PECMV are 0.0645m, 0.0815m, respectively. Our approach exhibits an average RMSE reduction of 21% compared to PECMV.

These experimental findings collectively indicate that our proposed algorithm outperforms the PECMV algorithms in situations characterized by both limited connectivity and spurious sensor data.

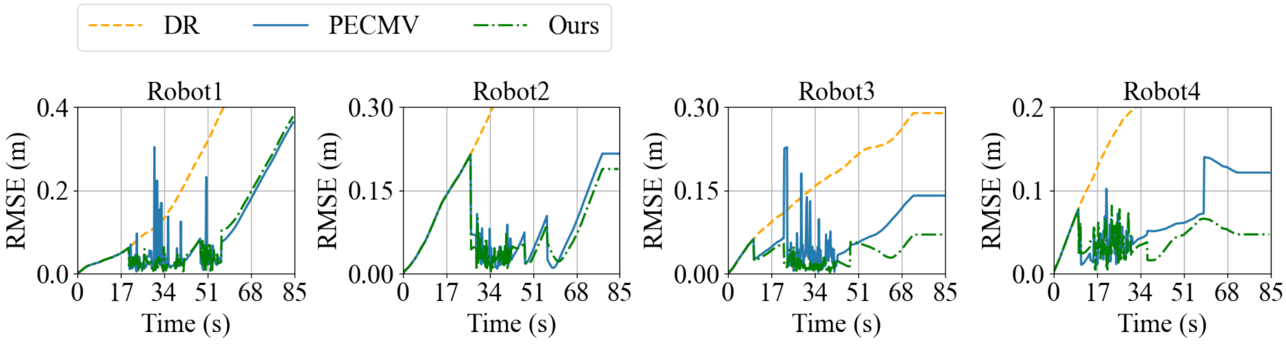

Fig. 8. RMSE over time of each robot position estimation in experiment.

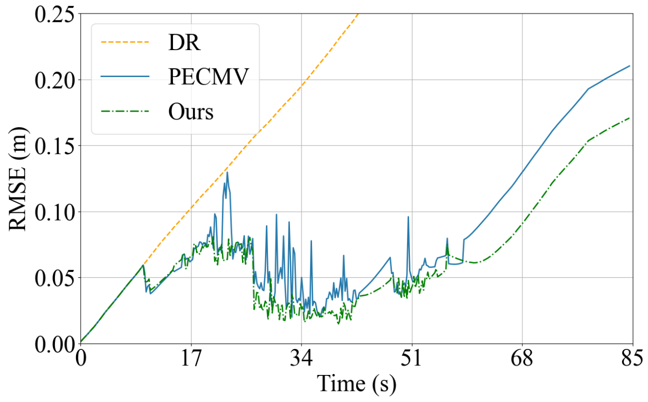

Fig. 9. RMSE of robots position estimation in experiment

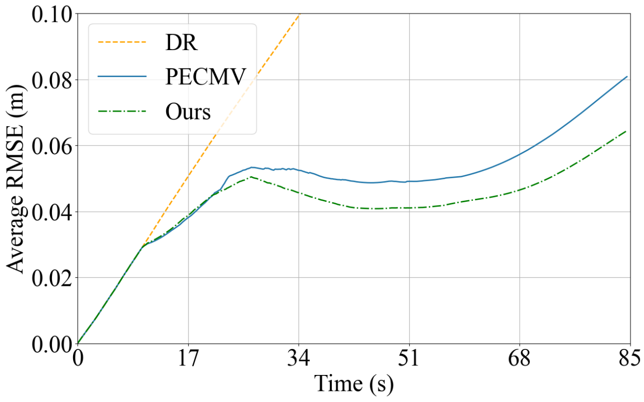

Fig. 10. Average RMSE of robots position estimation in experiment

V. Conclusion

This paper introduces a fault-tolerant distributed collaborative localization algorithm that incorporates a learning-based confidence evaluation method to handle measurements containing spurious sensor data. The proposed algorithm maintains localization filter consistency by considering past agent correlations implicitly. The simulation and experimental results demonstrate the effectiveness of our distributed algorithm in achieving accurate localization under spurious sensor data and limited connectivity constraints. Moreover, our algorithm supports generic measurement models since we do not prescribe the type of sensor.

Furthermore, we aim to enhance the algorithm's fault-tolerance performance by detecting various types of data faults, thereby increasing its overall robustness. Due to the proposed localization algorithm's lightweight and robust nature, we plan to deploy this solution to our developing modular self-reconfigurable robot [27], [28] and implement a modular self-reconfigurable robot system that includes visual self-localization.

References

[1] X. Zhou, X. Wen, Z. Wang, Y. Gao, H. Li, Q. Wang, T. Yang, H. Lu, Y. Cao, C. Xu *et al.*, "Swarm of micro flying robots in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm5954, 2022.

[2] X. Guo, J. Hu, J. Chen, F. Deng, and T. L. Lam, "Semantic histogram based graph matching for real-time multi-robot global localization in large scale environment," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8349–8356, 2021.

[3] G. Liang, Y. Tu, L. Zong, J. Chen, and T. L. Lam, "Energy sharing mechanism for a freeform robotic system-freebot," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 4232–4238.

[4] M. Li, G. Liang, H. Luo, H. Qian, and T. L. Lam, "Robot-to-robot relative pose estimation based on semidefinite relaxation optimization," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4491–4498.

[5] M. Li, T. L. Lam, and Z. Sun, "3-d inter-robot relative localization via semidefinite optimization," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 081–10 088, 2022.

[6] Y. Wang, M. Lin, X. Xie, Y. Gao, F. Deng, and T. L. Lam, "Asymptotically efficient estimator for range-based robot relative localization," *IEEE/ASME Transactions on Mechatronics*, 2023.

[7] J. Zhu and S. S. Kia, "Cooperative localization under limited connectivity," *IEEE Transactions on Robotics*, vol. 35, no. 6, pp. 1523–1530, 2019.

[8] S. S. Kia, J. Hechtbauer, D. Gogokhiya, and S. Martinez, "Server-assisted distributed cooperative localization over unreliable communication links," *IEEE Transactions on Robotics*, vol. 34, no. 5, pp. 1392–1399, 2018.

[9] N. Saeed, A. Celik, T. Y. Al-Naffouri, and M.-S. Alouini, "Underwater optical wireless communications, networking, and localization: A survey," *Ad Hoc Networks*, vol. 94, p. 101935, 2019.

[10] B.-C. Renner, J. Heitmann, and F. Steinmetz, "Ahoi: Inexpensive, low-power communication and localization for underwater sensor networks and μauvs," *ACM Transactions on Sensor Networks (TOSN)*, vol. 16, no. 2, pp. 1–46, 2020.

[11] J. González-García, A. Gómez-Espinosa, E. Cuan-Urquizo, L. G. García-Valdovinos, T. Salgado-Jiménez, and J. A. Escobedo Cabello, "Autonomous underwater vehicles: Localization, navigation, and communication for collaborative missions," *Applied sciences*, vol. 10, no. 4, p. 1256, 2020.

[12] L. Bai, Y. Yang, C. Feng, C. Guo, and B. Jia, "Novel visible light communication assisted perspective-four-line algorithm for indoor localization," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 2020, pp. 1–6.

[13] B. Hussain, Y. Wang, R. Chen, H. C. Cheng, and C. P. Yue, "Lidr:

Visible-light-communication-assisted dead reckoning for accurate indoor localization," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 15 742–15 755, 2022.

[14] L. Bertoni, S. Kreiss, and A. Alahi, "Monoloco: Monocular 3d pedestrian localization and uncertainty estimation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6861–6871.

[15] Y. Yu, R. Chen, W. Shi, and L. Chen, "Precise 3d indoor localization and trajectory optimization based on sparse wi-fi ftm anchors and built-in sensors," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 4, pp. 4042–4056, 2022.

[16] A. Kim, A. Ošep, and L. Leal-Taixé, "Eagermot: 3d multi-object tracking via sensor fusion," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11 315–11 321.

[17] F. Shamsfakhr, B. Sadeghi Bigham, and A. Mohammadi, "Indoor mobile robot localization in dynamic and cluttered environments using artificial landmarks," *Engineering Computations*, vol. 36, no. 2, pp. 400–419, 2019.

[18] J. Zheng, K. Li, and X. Zhang, "Wi-fi fingerprint-based indoor localization method via standard particle swarm optimization," *Sensors*, vol. 22, no. 13, p. 5051, 2022.

[19] S. E. Webster, J. M. Walls, L. L. Whitcomb, and R. M. Eustice, "Decentralized extended information filter for single-beacon cooperative acoustic navigation: Theory and experiments," *IEEE Transactions on Robotics*, vol. 29, no. 4, pp. 957–974, 2013.

[20] L. Luft, T. Schubert, S. I. Roumeliotis, and W. Burgard, "Recursive decentralized collaborative localization for sparsely communicating robots." in *Robotics: science and systems*. New York, 2016.

[21] J. Čurn, D. Marinescu, N. O'Hara, and V. Cahill, "Data incest in cooperative localisation with the common past-invariant ensemble kalman filter," in *Proceedings of the 16th International Conference on Information Fusion*. IEEE, 2013, pp. 68–76.

[22] H. Li and F. Nashashibi, "Cooperative multi-vehicle localization using split covariance intersection filter," *IEEE Intelligent transportation systems magazine*, vol. 5, no. 2, pp. 33–44, 2013.

[23] S. Fang, H. Li, and M. Yang, "Lidar slam based multivehicle cooperative localization using iterated split cif," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 21 137–21 147, 2022.

[24] 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," *Information Fusion*, vol. 37, pp. 61–76, 2017.

[25] J. Klingner, N. Ahmed, and N. Correll, "Fault-tolerant covariance intersection for localizing robot swarms," *Robotics and Autonomous Systems*, vol. 122, p. 103306, 2019.

[26] M. Mukherjee, A. Banerjee, A. Papadimitriou, S. S. Mansouri, and G. Nikolakopoulos, "A decentralized sensor fusion scheme for multi sensorial fault resilient pose estimation," *Sensors*, vol. 21, no. 24, p. 8259, 2021.

[27] G. Liang, H. Luo, M. Li, H. Qian, and T. L. Lam, "Freebot: A freeform modular self-reconfigurable robot with arbitrary connection point-design and implementation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 6506–6513.

[28] T. L. Lam and G. Liang, "Self-reconfigurable robot module and self-reconfigurable robot," Jul. 8 2021, uS Patent App. 17/134,066.