Cooperative Localization for Multi-Agents Based on Reinforcement Learning Compensated Filter

Ran Wang, *Graduate Student Member, IEEE*, Cheng Xu, *Member, IEEE*, Jing Sun, Shihong Duan, and Xiaotong Zhang, *Senior Member, IEEE*

***Abstract*— In modern navigation and positioning systems, accurate location information is crucial for ensuring system performance and user experience. Particularly, in scenarios involving the use of multiple agents such as robots and drones for rescue operations in unknown complex environments, accurate localization is fundamental for subsequent actions. However, traditional filtering-based localization algorithms may exhibit suboptimal performance and are sensitive to initial estimates and system noise. To address these issues, this paper proposes a multi-agent collaborative localization algorithm based on reinforcement learning compensation filtering to tackle localization problems in complex environments and improve the robustness and accuracy. Specifically, this paper introduces a value decomposition-based reinforcement learning network for filtering compensation to reduce overall localization error and address the credit allocation problem in multi-agent reinforcement learning. The main contributions of this paper are as follows: Firstly, a local localization estimation method based on reinforcement learning compensation Extended Kalman Filter (EKF) is proposed, which further corrects the results of the EKF algorithm and eliminates initial estimation errors. Secondly, a global collaborative localization estimation algorithm (MARL_CF) based on credit allocation in multi-agent reinforcement learning is proposed, which maximizes the reduction of overall localization error through information sharing and global optimization. Finally, the effectiveness of the proposed algorithms is validated through both numerical simulation and physical experiments. The results demonstrate that the proposed MARL_CF significantly improve the accuracy and robustness of localization in complex environments.**

***Index Terms*— Multi-agent reinforcement learning, nonlinear filtering, value decomposition, cooperative localization.**

I. Introduction

POSITION information plays a pivotal role in location-based services (LBS), such as smartphone navigation and autonomous vehicles [1]. The accuracy of positioning is crucial for both system performance and user experience, particularly in scenarios involving multiple agents, such as robots and drones, operating in complex and unfamiliar environments during rescue operations [2]. The primary objective in such scenarios is to estimate the pose and state information of each agent, as it directly influences the efficiency of path planning.

To achieve high precision and real-time performance, researchers have explored cooperative localization methods, with a focus on Ultra-Wideband (UWB) and Inertial Measurement Unit (IMU) technologies, both of which have garnered significant attention due to their unique advantages [3]. UWB technology excels in indoor and urban environments, providing high-precision distance measurements and robust anti-interference capabilities. In contrast, IMU technology offers attitude and motion information through the measurement and integration of acceleration and angular velocity. By combining UWB and IMU, we can leverage their respective strengths, resulting in improved localization accuracy and robustness.

In a cooperative network, individual agents utilize inertial devices to obtain position information and autonomously establish ranging relationships with others within the wireless network. Through the exchange of information, they collaboratively work towards optimizing the estimation of target localization [4], [5]. This cooperative approach integrates perceptual information collected by individual agents, fostering information gain and effective communication among agents in highly dynamic environments. The fusion of UWB and IMU technologies empowers these agents to effectively collaborate, enabling them to make informed decisions and take timely actions in real-time rescue scenarios.

However, UWB/IMU cooperative localization still faces several challenges and shortcomings that warrant further attention. Firstly, the issue of initial localization proves to be critical, particularly in scenarios lacking prior information

Manuscript received 16 September 2023; revised 19 March 2024; accepted 8 May 2024. Date of publication 14 June 2024; date of current version 18 September 2024. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62101029, in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2023A1515140071, and in part by China Scholarship Council Award under Grant 202006465043 and Grant 202306460078. *(Corresponding author: Cheng Xu.)*

Ran Wang is with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China, also with Shunde Innovation School, University of Science and Technology Beijing, Foshan 528399, China, and also with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: wangran423@foxmail.com).

Cheng Xu is with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China, also with Shunde Innovation School, University of Science and Technology Beijing, Foshan 528399, China, and also with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: xucheng@ustb.edu.cn).

Jing Sun, Shihong Duan, and Xiaotong Zhang are with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China, and also with Shunde Innovation School, University of Science and Technology Beijing, Foshan 528399, China (e-mail: sjxzweb@163.com; duansh@ustb.edu.cn; zxt@ies.ustb.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSAC.2024.3414599.

Digital Object Identifier 10.1109/JSAC.2024.3414599

or reference base stations [6]. Accurate initial localization is imperative for subsequent cooperative localization algorithms and overall system performance. Secondly, challenges related to error accumulation between UWB and IMU, as well as inconsistencies in error distribution among agents, need to be addressed [7]. These issues necessitate optimization methods to reduce cooperative errors and enhance positioning accuracy.

Numerous methods have been proposed to address the challenges encountered in UWB/IMU cooperative localization. Traditional approaches often rely on the classical Kalman filter, which provides an optimal solution for linear Gaussian problems and is commonly employed in practical applications [8]. However, real-world problems often exhibit nonlinear characteristics, necessitating the use of nonlinear filtering methods, which are prevalent in most real-world applications. Classic methods for tackling nonlinear filtering problems include the Extended Kalman Filter (EKF) [9], the Unscented Kalman Filter (UKF) [10], the Complementary Filter (CF) [11], Particle Filters, and others. Nevertheless, these methods have limitations in handling complex and nonlinear error distributions, often leading to high computational complexity. Therefore, there is a pressing need for more efficient and accurate methods to address the challenges in UWB/IMU cooperative localization.

However, the manually designed models and algorithms mentioned above perform well under specific conditions but lack reliability in complex and dynamic environments [12]. They are highly sensitive to initial state estimation and rely on empirical selection, making it challenging to ensure accuracy. Additionally, in complex unknown environments, the changing noise distribution results in continuous alterations in the environmental model structure, necessitating frequent adjustments of the filter gain [13]. Without considering gain adjustment, the estimation performance may converge slowly or even diverge, leading to unsatisfactory localization outcomes when using the EKF. To address these issues, machine learning techniques have been employed, utilizing data-driven approaches to solve attitude estimation problems [14], [15]. Supervised learning methods such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) have shown promise in pose estimation by modeling motion dynamics or extracting general features. However, these methods are often limited to specific environments and may struggle to converge without prior experience [16].

An alternative approach, proposed by Morimoto and Doya [17], involves using Reinforcement Learning (RL) to estimate hidden variables and parameters in nonlinear dynamic systems. However, there exists a substantial research gap between RL and localization estimation, and training models from scratch using RL methods can be inefficient and yield suboptimal performance. To address this challenge, this study introduces a novel approach that combines RL with EKF to solve control application problems [18], [19]. The hybrid method aims to leverage the strengths of both RL and EKF to achieve more robust and efficient localization in complex dynamic environments. This study considers the cooperative localization problem from the perspective of Multi-Agents based on Reinforcement Learning (MARL), with each agent corresponding to a network edge. The study proposes a new cooperative localization algorithm by combining centralized learning and decentralized training (CTDE) of multi-agent reinforcement learning with filtering methods. This optimization aims to distribute overall positioning errors more effectively among multiple agents. By employing this approach, we seek to enhance cooperation and information sharing among agents, ultimately leading to improved system performance and positioning accuracy.

In summary, the main contributions are as follows:

1) We propose a local localization method that leverages reinforcement learning compensation for the Extended Kalman Filter (EKF). This study addresses the challenge of unknown initial positions by employing reinforcement learning to refine the EKF's estimations. Through reinforcement learning, the model is trained to acquire compensatory filtering gain values, effectively rectifying errors stemming from initial estimation.
2) We introduce a global cooperative localization algorithm, MARL_CF, founded on the principles of credit assignment. This algorithm utilizes distance information between agents as input for the reinforcement learning network. Through training an optimization strategy with an Actor-Critic (AC) network, the results of local localization are further enhanced. The credit assignment network is responsible for allocating the overall error within the multi-agent system, effectively preventing the occurrence of "laziness" among agents [20].
3) We meticulously train and evaluate the MARL_CF algorithm, conducting experiments in both numerical simulation scenarios and physical environments. These experiments encompass diverse initial position ranges and noise covariance settings, facilitating comparisons and validations against state-of-the-art algorithms. The results unequivocally demonstrate the algorithm's effectiveness.

The rest of this paper is organized as follows: Section II presents the overall framework of proposed method, and Section III and Section IV describe the local localization module and global cooperative localization module, respectively. Section V demonstrates the experiments and analysis. Finally, Section VI concludes the paper.

II. System Framework

This section introduces the framework for Multi-Agent Reinforcement Learning Compensated Filter (MARL_CF) in the context of cooperative localization.

A. Problem Definition

The central issue tackled in this paper pertains to multi-agent cooperative localization in intricate and unknown environments. Considering a network of N agents, the position of the agent i at time t is denoted as $X_{t,i}$. The overall goal of this paper is to estimate the positions of all agents within the network based on observational information, i.e., $\mathbf{X}_t = [X_{t,1}, X_{t,2}, .., X_{t,N}]$. For simplicity, we will ignore the

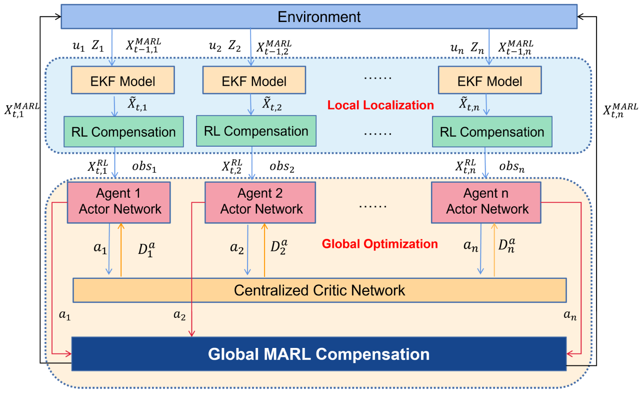


Fig. 1. The framework diagram of our proposed cooperative localization algorithm, MARL_CF.

subscript i in the case of not specifically referring to an agent, and use X_t to represent the position of any agent, that is:

$$X_t = (x_t, y_t, \phi_t) \tag{1}$$

where x_t and y_t represent the x and y coordinate at time t, and ϕ_t represents the heading angle at time t. The measurements obtained using the IMU sensor are defined as follows:

$$\mu_t = (\vartheta_t, \phi_t) \tag{2}$$

where ϑ_t and ϕ_t represent the linear acceleration and angular velocity of the IMU, respectively. The measurements of the agent is represented as follows:

$$Z_t = (d_t, \theta_t, C) \tag{3}$$

where d_t represents the straight-line distance between adjacent agents, θ_t represents the relative heading angle, and C represents the correspondence between adjacent agents.

B. Algorithm Framework

Noise and the ever-changing environmental conditions introduce errors when individual agents utilize the EKF filtering algorithm, which, in turn, leads to subpar localization performance. Furthermore, credit assignment problems can arise among multiple agents. In response to these challenges, we integrate MARL with filtering techniques to implement MARL corrections within the context of EKF localization. The framework comprises both a local localization and a global optimization module, illustrated in Fig. 1.

1) Local Localization Module: This module comprises Extended Kalman Filtering and single-agent reinforcement learning algorithms. It utilizes data from IMU and UWB sensors for Extended Kalman Filtering localization. Subsequently, reinforcement learning comes into play to train the compensatory Kalman gain matrix, yielding local localization results for each agent. The reinforcement learning network takes the residual observation of each agent as input, trains the compensatory Kalman gain matrix to rectify initial estimation errors, and thereby enhances localization accuracy.

2) Global Optimization Module: This module involves collaboration and information sharing among multiple agents to further improve their localization accuracy and obtain the global estimates. It includes the actor networks of each agent, a shared critic network, and the MARL correction module. During the training process, all agents share a joint critic network (central controller), but each agent has its own actor network (policy network). During the testing and execution process, each agent only needs to input its observation information o_t to the policy network, which outputs an action value a_t based on the policy $\pi(o_t)$. This action value is used to optimize the local localization result, obtaining the final global estimation.

III. Local Localization Optimization

This section demonstrates how we harness reinforcement learning to compensate for the Extended Kalman Filter (EKF). The fundamental structure is depicted in Fig. 2.

A. EKF for Local Localization

Throughout the motion process, each agent initiates by updating and propagating its individual state according to its motion model. Subsequently, it acquires observation values from neighboring agents through communication. The disparity between the observed and actual values is then utilized to refine the agent's localization estimate, resulting in a continuous reduction of errors and a gradual convergence towards the true position.

1) State Update: The IMU sensor furnishes the agent with motion-related information based on the initial position. Employing linear acceleration and angular velocity, in conjunction with the position information from the previous time step denoted as X_{t-1}, predictions are made to obtain the prior estimate of the current position $\bar{X}_t$ and covariance matrix $\bar{P}_t$. The motion model for updating the agent using IMU sensor data can be summarized as follows:

$$\bar{X}_t = f(u_t, x_{t-1}) = X_{t-1} + \begin{bmatrix} \vartheta_t \Delta t \cos(\theta + \frac{\omega_t \Delta t}{2}) \\ \vartheta_t \Delta t \sin(\theta + \frac{\omega_t \Delta t}{2}) \\ \omega_t \Delta t \end{bmatrix} \tag{4}$$

$$\bar{P}_t = F_{t-1} P_{t-1} F_{t-1}^T + Q_{t-1} \tag{5}$$

Here, $u_t = (\vartheta_t, \omega_t)$ represents the inputs, encompassing linear acceleration and angular velocity data from the IMU. These inputs inherently include noise values, assumed to follow an uncorrelated Gaussian white noise distribution. Here, Q represents the process noise covariance matrix. Furthermore, F_{t-1} denotes the linearized function of $f(X_{t-1})$, implemented utilizing the Jacobian matrix:

$$F_{t-1} = \frac{\partial f(u_t, x_{t-1})}{\partial x_{t-1}} = \begin{bmatrix} 1 & 0 & -\vartheta_t \Delta t \sin\left(\theta + \frac{\omega_t \Delta t}{2}\right) \\ 0 & 1 & \vartheta_t \Delta t \cos\left(\theta + \frac{\omega_t \Delta t}{2}\right) \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

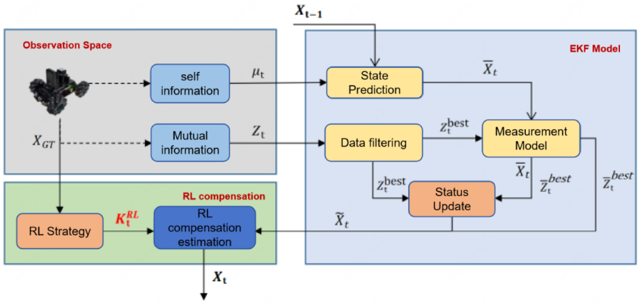


Fig. 2. The framework diagram of reinforment learning compesated EKF for local localization.

2) Measurement Update: Following the state update, the accumulation of IMU errors invariably results in an increase in the agent's state uncertainty. Within this framework, each agent's measurements are presented in the form of distance and direction. The measurement can be expressed as $Z_t = (d_t, \theta_t, C)$, where d_t signifies the Euclidean distance between adjacent agents, θ_t represents the relative bearing angle, and C indicates the correspondence between this measurement value and a specific neighbour. In this study, we ultimately select the most nearest three neighbours' (if any) observations as inputs. The measurement update process proceeds iterately using measurement collected from each adjacent as follows:

$$\tilde{X}_t = \bar{X}_t + K_t(Z_t - \bar{Z}_t) \tag{7}$$

$$\bar{Z}_t = h(\bar{X}_t) \tag{8}$$

$$K_t = \bar{P}_t H_t^T (H_t \bar{P}_t H_t^T + R_t)^{-1} \tag{9}$$

$$\tilde{P} = (I - K_t H_t)\bar{P}_t \tag{10}$$

In this context, R_t signifies the covariance matrix, Z_t denotes the practical measurement data at time t, and $\bar{Z}_t$ represents the predicted measurement value. Employing the measurement model, the current position estimate $\bar{X}_t$ is translated into the corresponding distance and direction measurements, essentially generating measurement predictions, as described by the measurement function:

$$\bar{Z}_t = h(x_t, m) = \begin{bmatrix} d_j \\ \text{atan2}(m_{j,y} - \bar{X}_{t,y}, m_{j,x} - \bar{X}_{t,x}) - \bar{X}_{t,\phi} \\ m_{j,s} \end{bmatrix} \tag{11}$$

$$d_j^2 = (m_{j,x} - \bar{X}_{t,x})^2 + (m_{j,y} - \bar{X}_{t,y})^2 \tag{12}$$

where $m_{j,x}$, $m_{j,y}$ and $m_{j,s}$ correspond to the current x, y estimates, and the identifier of neighbor agent j, respectively. It is worth mentioning that the absolute coordinates of the neighbor agent are unknown, and only its coordinate estimates are used as inputs to the current measurement equation to obtain estimates of the pratical measurements. H_t takes on the role of the observation model matrix, being the Jacobian matrix of the nonlinear measurement function $h(\cdot)$, as follows:

$$H_t^i = \frac{\partial h(\bar{X}_t, m)}{\partial x_t} = \frac{1}{q}\begin{bmatrix} d\delta_x & -d\delta_y & 0 \\ \delta_y & \delta_x & -1 \\ 0 & 0 & 0 \end{bmatrix} \tag{13}$$

$$\delta = \begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix} = \begin{bmatrix} m_{j,x} - \bar{X}_{t,x} \\ m_{j,y} - \bar{X}_{t,y} \end{bmatrix} \tag{14}$$

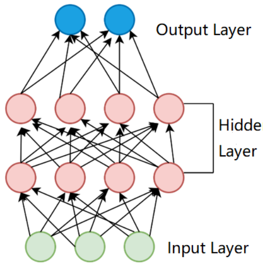


Fig. 3. MLP network structure diagram.

During the measurement process, the agent measures and computes the Euclidean distance between neighbors and the difference in yaw angles to construct the model predicted value $\bar{Z}_t$. Following this, the measurement prediction value is juxtaposed with the actual sensor observation Z_t.

B. RL Compensation

In this section, we delve into the relationship between reinforcement learning (RL) and the Extended Kalman Filter (EKF), and establish a Bayesian filter using Deep Reinforcement Learning (DRL). Initially, we represent EKF as a dynamic Markov Decision Process (MDP), and align the variables in the reinforcement learning environment with the components of EKF. This approach allows us to utilize DRL techniques to enhance the performance of EKF, enabling it to better adapt to uncertainty and complex environments.

1) MLP: Within the realm of DRL, deep neural networks (DNNs) are frequently employed as approximators for policy and value functions. In this paper, the Proximal Policy Optimization (PPO) algorithm [21] is utilized, where both the actor and critic components are constructed using fully connected Multi-Layer Perceptrons (MLPs) with Rectified Linear Unit (ReLU) activation functions. Unlike structures such as Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN), our problem involves continuous state and action spaces, making MLP the preferred network architecture. A typical MLP comprises three layers: input, hidden, and output. Each layer's neurons are interconnected with the previous and subsequent layers, receiving input from the previous layer, and transmitting it to the next layer. The impact of inputs is adjusted through weight tuning, and the sum is transformed into outputs via activation functions. The overall structure is illustrated in Fig. 3.

Regarding the policy network, the architecture of MLP relies on the dimensions of the input and output vectors, which correspond to the state and action sizes, respectively. The selection of the hidden layer size should align with the complexity of the problem at hand. In this paper, we employ an MLP with two hidden layers, which can be expressed mathematically as follows:

$$z = MLP_w^2(x) = w_2\left[f\left(w_1\left[f\left(w_0(x + b_0)\right) + b_1\right]\right) + b_2\right] \tag{15}$$

Here, x denotes the input state, $[w_0, w_1, w_2]$ symbolizes the weights assigned to each layer, and $f(\cdot)$ signifies the ReLU activation function, which is applied following each layer. ReLU, an activation function expressed as $f(x) = x^+ = \max(0, x)$, offers benefits during the backpropagation process.

2) Reinforcement Learning Filter: The ultimate EKF update can be formulated in the framework of MDP as follows:

$$\hat{x}_{t+1} = f(\bar{x}_t) + K_t(z_{t+1} - h(f(\hat{x}_t))) \tag{16}$$

$$\tilde{x}_{t+1} \sim \mathcal{P}(\tilde{x}_{t+1}|\tilde{x}_t, z_{t+1}, K_t), \quad \forall t \in \mathbb{Z}_+ \tag{17}$$

where $h(\cdot)$ indicates the estimated measurements. In the subsequent time step, the state solely depends on the current state $\tilde{x}_t$, the measurement z_{t+1}, and the Kalman gain K_t. Within the context of EKF, the Kalman gain is calculated based on the measurement innovation, which signifies the distinction between the true state x and the estimated state $\tilde{x}_t$. During the offline training process of DRL, assuming that the true state x is known, the mapping from $\tilde{x}_t$ to K_t can be represented as a nonlinear function. This nonlinear mapping function, considered as the reinforcement learning policy π, and the Kalman gain K as the reinforcement learning action, are utilized to train a compensatory gain value through reinforcement learning. The structure of RL state estimator in MDP tuple $< S, A, P, R, \gamma >$ is presented as follows:

- State: $\tilde{x}_t \in S$ signifies the system transition probability $\mathcal{P}(\tilde{x}_{t+1}|\tilde{x}_t, a(\hat{x}_t)) \in \mathcal{P}$, with $a(\cdot)$ representing the action value that corresponds to compensation gain of the filter.
- Action: A is a matrix representing the network's output, with values within the range of $[-0.1, 0.1]$. This action matrix combines the residuals of observed and model-predicted data to correct the localization estimate.
- Reward R: It quantifies the value of the action-state pair. This paper employs the mean squared error (MSE) between the true and estimated positions for this purpose. To clarify, in the context of our work, "true position" refers to the position used as a reference in the simulation environment for the purpose of training and evaluating our algorithm. It is not known to the localization algorithm itself but is used to compute the error metrics such as the mean squared error (MSE) for the training process.
- Policy $\pi(a_t, x_t)$: It employs the MLP approximator to map the estimated state to the compensation gain (action).

3) Reinforcement Learning Compensation: The number of measurement values obtained at each time step dynamically changes. Since the action space and observation input of the reinforcement learning network must have fixed sizes for computational convenience and consistency, this study opts to use the three closest pieces of observed information within the observation range as input to the reinforcement learning network. Additionally, it considers the Kalman gain value K_t^{RL} as the output of the reinforcement learning action.

For clarity, the EKF estimation result $\tilde{X}_t$ from the previous section is denoted as X_t^{EKF}, and the reinforcement learning compensation result is denoted as X_t^{RL}. These representations signify the final outcomes of the local localization process for each agent. The process of reinforcement learning compensation can be described as follows:

$$X_t^{\text{RL}} = X_t^{\text{EKF}} + \varepsilon_t^{\text{RL}} \tag{18}$$

$$\varepsilon_t^{\text{RL}} = K_t^{\text{RL}}(Z_t - \bar{Z}_t^{\text{RL}}) \tag{19}$$

$$\varepsilon_t^{\text{RL}} \sim \rho(\varepsilon_t^{\text{RL}}|\varepsilon_{t-1}^{\text{RL}}, K_t^{\text{RL}}) \tag{20}$$

where $K_t^{RL} \in A$ represents the compensation gain of the filter and serves as the output of the reinforcement learning network. The reward function for the network is defined as follows:

$$R = \sqrt{(x^{GT} - X_{t,x}^{\text{RL}})^2 + (y^{GT} - X_{t,y}^{\text{RL}})^2} \tag{21}$$

With this, a local localization framework based on the combination of reinforcement learning and filtering has been established, as depicted in Algorithm 1.

Algorithm 1 Local Localization Algorithm

```
Input: IMU measurement data $u_t$, UWB observation data $Z_t$, true position $X^{GT} = \{\mathcal{X}^{GT}, \mathcal{Y}^{GT}\}$, initial estimated position $X_0 \sim \mathcal{N}$.
Output: Local localization estimate $X_t^{RL}$
Initialize the system state values and covariance matrix $P_0$
for $t = 1, \ldots, T$ do
    EKF Correction:
        Prediction Update: Use $u_t$ to predict the location $\bar{X}_t$ and covariance matrix $\bar{P}_t$ based on formulas (4) and (5)
        Measurement Update: Utilize $Z_t$ for measurement update. Calculate the Kalman gain $K_t$ using formula (9)
        Update the localization result $\tilde{X}_t$ and covariance matrix $\tilde{P}_t$ based on formulas (7) and (10)
    RL Compensation Correction:
        Based on the previous error $\varepsilon_{t-1}^{RL}$, obtain the compensation gain matrix $K_t^{RL}$ using the RL policy network
        Calculate the current error state $\varepsilon_t^{RL}$ using formula (19) and compensate the current error value to the localization result $\tilde{X}_t$
        Update the local localization result $X_t^{RL}$ using formula (18)
end for
```

IV. Multi-Agent Global Localization

The primary objective of this section is to rectify local estimates by incorporating global data information. In multi-agent reinforcement learning methods, there exists a challenge known as the credit assignment problem, stemming from partial observability. In the context of multi-agent global localization, we tackle this credit assignment problem by attributing errors to specific local errors from a global localization perspective.

A. Credit Assignment

We initially introduce a global cooperative localization algorithm based on COMA [22], employing a centralized training and distributed execution (CTDE) structure. This algorithm leverages observation data and policy training among agents to learn from communication observations and the residuals of actual distances and orientations between agents, ultimately optimizing local estimates. The aim of this algorithm is to acquire an action value that fine-tunes the local localization result when executed, bringing it closer to the

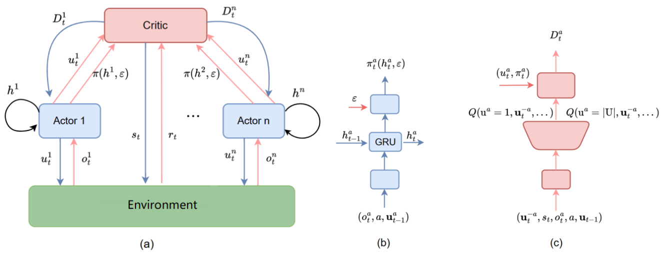


Fig. 4. Credit assignment network: (a) the centralized network structure of credit assignment algorithms, and the network structure of (b) Actors and (c) Critics.

true value. The components of the multi-agent reinforcement learning in this paper are defined as follows:

- **Observations** (O), encompasses the localization residuals of the current agent with respect to others. Local observations serve as input to each agent's actor network, while global observations serve as input to the centralized critic.
- **Action** (A), is defined as a discrete space consisting of eight action values to adjust current agent's orientations.
- **Reward** (R), represents the global reward function, which is inversely proportional to the localization error. A higher reward corresponds to better localization performance. Here, γ denotes the discount factor, given by:

$$R_t = \sum_{l=0}^{\infty} \gamma^l r_{t+l} \tag{22}$$

To address the credit assignment challenge in multi-agent cooperation, the "counterfactual baseline" method takes center stage in calculating advantage values for each agent within the context of the Actor-Critic (AC) framework's CTDE mode. The comprehensive network structure is visualized in Fig. 4.

Each agent possesses its own policy network, essentially implemented as a Recurrent Neural Network (RNN). At each time step, an agent utilizes its local observations as input and employs its policy network to select an action denoted as u_i. By aggregating the action values from all agents, a collective action is derived. Additionally, all agents share a common critic network, referred to as the Q-network. This network is responsible for computing the action-value function for each agent, thereby approximating the global reward. The individual reward D^a is subsequently employed to iteratively refine the agent's policy network, enhancing its action selection.

$$D^a = r(s,u) - r(s,(u^{-a}, c_a)) \tag{23}$$

In this context, D^a can be interpreted as a measure of how much better or worse if an action u is taken by agent a when compared to the default action c_a. During the subsequent training phase, agents will strive to maximize the value of D^a, indicating their aim to optimize the global reward. Here, u^{-a} denotes the collective action without the contribution of agent a, and (u^{-a}, c_a) signifies the collective action with agent a taking the default action.

Algorithm 2 Multi-Agent Reinforcement Learning Credit Assignment Algorithm

```
Input: Network parameters $\theta^c$, $\theta^{c'}$, $\theta^\pi$, discount factor $\gamma$,
    maximum number of iterations $T$, learning rate $\alpha$, number
    of training episodes $M$
Output: Optimal evaluation network parameters $\theta^c$, policy
    network parameters $\theta^\pi$
Initialize parameters and experience replay memory $\mathcal{M}$
for episode $\leftarrow 1$ to $M$ do
    for $t = 1, \cdots, T$ do
        for $i = 1, \cdots, N$ do
            Use Actor network to get the probability of each action, and select $a_i = \pi_i(o_i)$ according to the policy
        end for
        Execute joint action $u$ to obtain global reward $R_t$ and next observation $o'_t$
        Store $(\vec{o}_t, \vec{a}_t, r, \vec{o}'_t)$ in $\mathcal{M}$
    end for
    for $t = 1, \cdots, T$ do
        Sample data from $\mathcal{M}$ for parameter updates
        Compute the loss function for the Critic network using formula (26) and (27), and update parameters $\theta_t^c = \theta_t^c + \alpha\nabla\theta^c$
        Compute the policy gradient for the Actor network using formula (29) and update parameters $\theta_{t+1}^\pi = \theta_t^\pi + \alpha\nabla\theta^\pi$
        Update Critic parameters $\theta^{c'}$ using formula (28)
    end for
end for
```

To calculate D^a for all actions, one would traditionally replace each action with the default one and interact with the environment to determine the corresponding utility values. However, this approach poses two challenges: it demands extensive computations, significantly increasing computational complexity, and it doesn't provide a straightforward way to designate which action should serve as the default. Consequently, this paper introduces an approximate method that estimates the average utility value of all actions an agent can take using the utility value obtained when taking the default action. This can be expressed as:

$$r(s,(u^{-a}, c_a)) = Q(s, c_a) = \sum_{u'^a} \pi^a(u'^a|\tau^a) Q(s,(u^{-a}, u'^a)) \tag{24}$$

Subsequently, the computation of D^a can be equivalently expressed as $R^a(s,u)$, which is given by:

$$R^a(s,u) = Q(s,u) - \sum_{u'^a} \pi^a(u'^a|\tau^a) Q(s,(u^{-a}, u'^a)) \tag{25}$$

B. Parameter Updates

In the MARL_CF network, updates to the centralized critic network (Q-network) are carried out using the TD(λ) [23]. In particular, the critic parameters θ^c are updated by minibatch gradient descent to minimise the following equation:

$$\nabla\theta^c = (\kappa^{(\lambda)} - f^{\theta^c}(\cdot))^2 \tag{26}$$

$$\kappa^{(\lambda)} = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}G_t^{(n)} = \sum_{l=1}^{n}\gamma^{l-1}r_{t+l}+\gamma^{n}f^{\theta^{c}}(\cdot) \tag{27}$$

where $G_t^{(n)}$ is a mixture of n-step returns. The parameter update for the target critic network can be represented as:

$$\theta^{c'} = \iota\theta^{c} + (1-\iota)\theta^{c'} \tag{28}$$

where ι is learning rate. The policy network of each agent is updated based on the policy gradient, namely

$$\nabla\theta^{\pi} = \nabla_{\theta_{\pi}}\log\pi(u|\tau_t^{a})\cdot R^{a}(s,u) \tag{29}$$

where τ_t^a indicates the agent's own action-observation history. The entire workflow of the credit assignment network in the MARL_CF is depicted in Algorithm 2.

C. Error Correction

We employ a global cooperative localization method based on multi-agent reinforcement learning to refine the local localization outcomes X_t^{RL} and derive enhanced estimation results X_t^{MARL}. These results represent the agent's final position estimate at time t and are described as follows:

$$X_t^{MARL} = X_t^{RL} + a_t * \mathcal{D} \tag{30}$$

where $a_t \in A$ denotes the action value provided by the MARL_CF algorithm, signifying the adjustment direction, while $\mathcal{D}$ represents the adjustment displacement value.

The entire training process of the proposed MARL_CF algorithm is delineated in Algorithm 3. It commences with the initialization of the environment and network parameters for reinforcement learning. Input data encompasses IMU measurements, UWB observations, and the true positions of each agent. Subsequently, sensor data undergoes processing via the local localization method to yield local estimates, as illustrated in Algorithm 1. Following this, the current observation information o_t for each agent is acquired and fed into their policy networks to determine actions a_t. These actions contribute to the calculation of the global localization estimate, representing the final localization value.

Following the estimation phase, new observations o_t' and the global reward r are obtained, and this data $(\vec{o_t}, \vec{a_t}, r, \vec{o_t}')$ is stored in the experience replay memory. Subsequently, the global localization network parameters in the MARL_CF algorithm receive updates using the reward value r. This iterative process continues until the most effective strategies for this scenario are learned.

Algorithm 3 Cooperative Localization Algorithm Based on Reinforcement Learning Compensated Filtering (MARL_CF)

```
Input: MARL_CF network parameters, maximum iteration rounds $T$, number of training rounds $M$, number of agents $N$, IMU measurement data $u_t$, UWB observation data $Z_t$, true positions $X^{GT} = \mathcal{X}^{GT}, \mathcal{Y}^{GT}$
Output: Corrected final localization estimate $X_t^{MARL}$
for round_number $\leftarrow 1, \cdots, M$ do
    Initialize environment and network parameters
    Obtain initial observations for each agent
    for $t = 1, \cdots, T$ do
        for $i = 1, \cdots, N$ do
            Utilize Algorithm 2 with $u_{t,i}$ and $Z_{t,i}$ to perform local localization estimation with EKF followed by RL compensation, obtaining localization result $X_{t,i}^{RL}$
        end for
        Calculate the localization residuals of each agent, distances, and angles with respect to other agents as observation information $o$
        for $i = 1, \cdots, N$ do
            Obtain action value $a_i = \pi_i(o_i)$ for agent $i$ using Algorithm 2, and execute the action
            Perform global localization correction: $X_{t,i}^{MARL} = X_{t,i}^{RL} + a_i * \mathcal{D}$
        end for
    end for
end for
```

V. Experiment and Analysis

This section is dedicated to validating the effectiveness of the MARL_CF algorithm through concrete experiments. To achieve this, we will begin by introducing the experimental environment and parameter settings utilized for evaluation purposes. Following that, we will perform a statistical analysis of the algorithm's performance and validate its effectiveness under varying experimental conditions. Furthermore, we will conduct a comparative assessment between MARL_CF and other state-of-the-art methods to underscore its advantages. Additionally, we will conduct ablation experiments to showcase the effectiveness of different components within the MARL_CF framework. Lastly, field experiments will be carried out to provide further validation of the proposed method.

A. Environment and Parameter Settings

In the simulation phase, the hardware configuration of the personal computer comprises a 6-core Intel i7 CPU and 16 GB of RAM, running on a 64-bit Windows 10 operating system. The experimental arena is configured as a square area measuring 30m × 30m, housing 20 stationary agents. These stationary agents are designed to accelerate the training model's convergence. It's worth mentioning that these nodes do not share their actual coordinates with other agents; their only difference from others is that they do not engage in random/regulated motion. This environment involves four moving agents, each operating independently while observing pertinent information from others. To introduce diversity into the training process, the positions of stationary agents are randomly generated, wherein agents traverse distinct paths. The input data for each agent encompasses forward linear velocity and angular velocity at each time step:

$$u_t = \bar{u}_t + e_t = \begin{pmatrix} \bar{v}_t + v_t \\ \bar{\omega}_t + \omega_t \end{pmatrix} \tag{31}$$

where $\bar{v}_t = 0.5$ m/s represents the true forward linear velocity, and $\bar{\omega}_t \in (-1, 1)$ rad/s the true angular velocity. The

TABLE I
EXPERIMENTAL PARAMETER SETTINGS FOR MARL_CF

Parameter	Value (Unit)
Sampling time interval (Δt)	0.1 (s)
Random change in direction angle (φ)	(-1, 1) (rad/s)
Velocity (v)	0.5 (m/s)
Number of agents	4
Number of action spaces	8
State space range	[-15, 15]
Discount factor (γ)	0.95
Number of training episodes	500
Number of iterations per episode	300
Number of neurons	64

TABLE II
PPO2 PARAMETER SETTINGS

Parameter	Value (Unit)
MLP value function dimensions	(256, 128, 64)
MLP policy function dimensions	(256, 128, 64)
Discount factor (γ)	0.97
Action space range	[-0.02, 0.02]
State space range	[-15, 15]
Learning rate	0.002
Value function coefficient	0.0003

process noise $e_u \sim \mathcal{N}(0, R_m)$, where $R_m = \begin{bmatrix} 1 & 0 \\ 0 & 0.1745 \end{bmatrix}$, represents the Gaussian noise. The measurement for each agent includes the relative distance and angle between the sensors at this agent and others at each time step:

$$\bar{Z}_t = \begin{bmatrix} \sqrt{(m_{j,x} - \bar{X}_{t,x})^2 + (m_{j,y} - \bar{X}_{t,y})^2} \\ \text{atan2}(m_{j,y} - \bar{X}_{t,y}, m_{j,x} - \bar{X}_{t,x}) - \bar{X}_{t,\theta} \end{bmatrix} + e_{t,z} \tag{32}$$

where $e_u \sim \mathcal{N}(0, Q_m)$ and $Q_m = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.01745 \end{bmatrix}$ represents Gaussian noise added to the measurements.

The simulation experiment encompasses two motion scenarios: regular motion and random walk. In the regular motion model, four agents move in a diagonal formation, converging toward the center at a 45-degree angle. Conversely, in the random walk scenario, the initial positions $x_0 = (x_0, y_0)$ and heading angles θ_0 for each agent are configured before each experiment. At each step, the direction of movement is randomly chosen, while the velocity remains constant. This random movement persists for 300 steps within a two-dimensional environment. Throughout this process, the agents execute actions and collect environmental observations, facilitating continuous updates to their positions.

For the MARL_CF algorithm, we set the number of iterations to 500 episodes, and the network and training parameters are specified as presented in Table I. The local localization method utilizes Proximal Policy Optimization (PPO2) reinforcement learning [24], with the network parameters detailed in Table II.

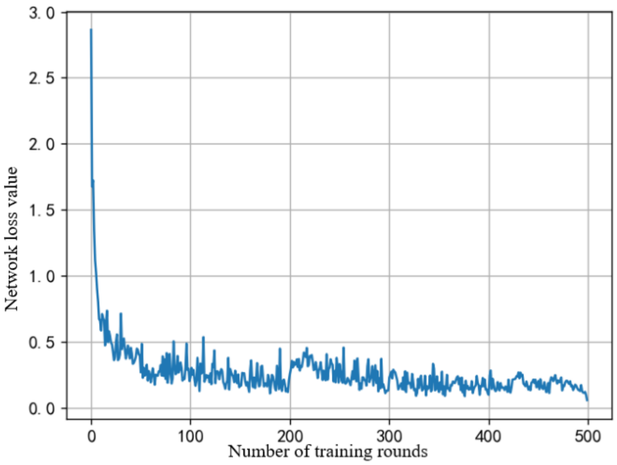


Fig. 5. The training loss of MARL_CF.

B. Model Training Performance

In the model training phase, we conducted 500 episodes of iterations, with each episode spanning 300 time steps. Fig. 5 depicts the training loss curve for MARL_CF, where the goal is to achieve larger rewards and smaller loss values. The loss curve reveals a gradual reduction in the overall loss value over time, with a notably significant drop in the first 50 episodes. Around the 100*th* episode, the loss value stabilizes. In the context of reinforcement learning, smaller loss values indicate that the current policy selection becomes more rational and accurate. Additionally, it suggests that the network's structure and parameter settings are apt, enabling swift convergence and enhancing the accuracy of the agent's localization estimates.

C. Performance Under Different Parameter Settings

To assess the performance of the MARL_CF algorithm under various experimental conditions, this subsection employs the controlled variable method to analyze the influence of initial position errors and model covariance settings on localization outcomes. Furthermore, we conducted a comparative evaluation of the MARL_CF algorithm against other state-of-the-art methods, including the Extended Kalman Filter (EKF) [9], Dead Reckoning (DR) [25], and Particle Filter (PF) [26].

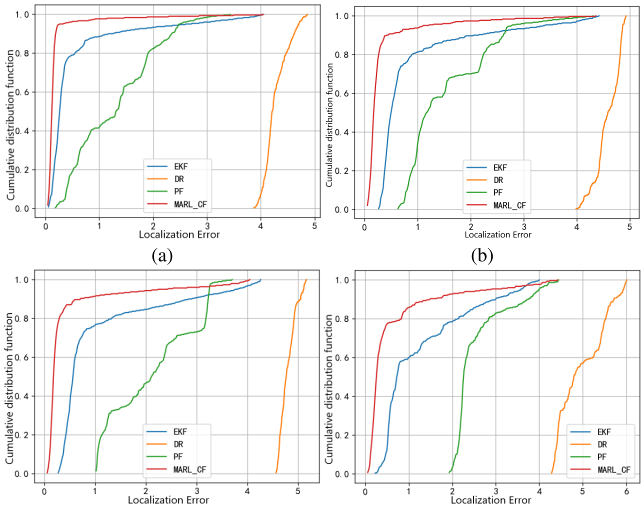
1) Impact of Initial Position Errors: To further examine the influence of initial estimation errors on the MARL_CF algorithm, this section conducted tests in four distinct initial estimation error ranges: $[-3, 3]$, $[-5, 5]$, $[-8, 8]$, and $[-10, 10]$. The experimental results are presented in Table III. The mean squared error (MSE) for multi-agent collaborative localization in the experimental results can be defined as follows:

$$RMSE = \frac{1}{M}\sum_{i=1}^{M}\left(\frac{1}{N}\sum_{j=1}^{N}\sqrt{e^2}\right) \tag{33}$$

The findings reveal that the MARL_CF algorithm consistently outperforms the EKF, DR, and PF algorithms by producing smaller localization errors, regardless of the initial estimation range. As the initial estimation error range increases, all algorithms exhibit a gradual rise in localization

TABLE III
MEAN SQUARED ERRORS OF VARIOUS ALGORITHMS UNDER DIFFERENT INITIAL ESTIMATE RANGES

Algorithm	[-3,3]	[-5,5]	[-8,8]	[-10,10]
DR [25]	3.392	4.586	7.497	8.161
EKF [9]	0.739	0.879	1.715	1.990
PF [26]	1.440	1.534	2.191	3.661
MARL_CF	**0.260**	**0.336**	**0.627**	**0.743**



(a) (b)

(c) (d)

Fig. 6. Mean squared error (MSE) curves of algorithms under different noise covariances. In (a)-(d), the noise covariance matrices are set to 0.5, 1, 2, and 4 times their original values.

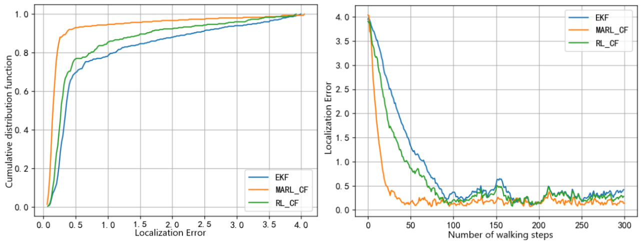
TABLE IV
MEAN SQUARED ERRORS OF VARIOUS ALGORITHMS UNDER DIFFERENT NOISE COVARIANCES

Algorithm	0.5	1	2	4
DR [25]	4.278	4.586	4.797	4.979
EKF [9]	0.545	0.879	1.032	1.271
PF [26]	1.301	1.534	2.115	2.559
MARL_CF	**0.205**	**0.336**	**0.438**	**0.610**

errors. However, the MARL_CF demonstrates the most modest increase in errors, highlighting its effectiveness in mitigating initial estimation errors. It is crucial to avoid an excessively large initial estimation range, as it could exceed the observable range of sensors, rendering it irrelevant as a reference.

In conclusion, the experimental results consistently demonstrate that the MARL_CF algorithm achieves smaller localization errors under different initial estimation ranges and outperforms other algorithms, including EKF, DR, and PF, as shown in Table III.

2) Impact of Noise Covariance: The localization performance of the algorithm is influenced by the noise levels in both the input data and measurement data, as reflected in the noise covariance matrices Q and R. This section aims to investigate how the algorithm's performance is impacted by these covariance matrices. To maintain variable uniqueness, we fix the initial estimation range at $[-5, 5]$. To analyze the impact of noise covariance matrices, we adjust their values based on the findings from Section V-C, setting them to 0.5, 1, 2, and 4 times their original values. The experimental results are presented in Fig. 6 and Table IV.



(a) (b)

Fig. 7. Localized RL ablation experiment: (a)cumulative distribution error, and (b) error variation in time sequence.

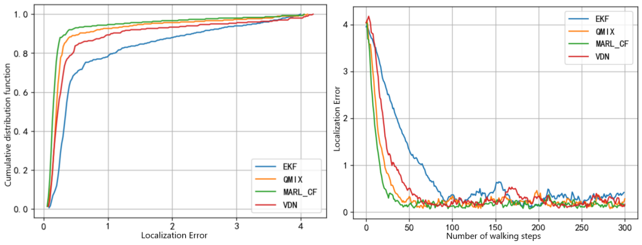
The results demonstrate that as the noise covariance values increase, the localization errors also increase. However, the MARL_CF algorithm exhibits better performance in handling increased noise uncertainty, as it allows more room for compensation. In comparison to other algorithms, the MARL_CF algorithm is the least sensitive to changes in noise covariance. This advantage is evident in Fig. 6, which displays the cumulative error distribution curves of different algorithms under varying covariance matrices. The MARL_CF algorithm shows a higher probability of achieving localization within a smaller error range. In summary, the experimental results highlight that increasing noise covariance leads to higher localization errors. Nonetheless, the MARL_CF algorithm remains robust in the face of noise, achieving higher localization accuracy within a smaller error range.

D. Ablation Experiments

To verify the effectiveness of the local localization module and the global cooperative localization module proposed in the MARL_CF algorithm, this section focuses on discussing ablation experiments for both RL local localization and MARL global cooperative localization.

1) RL Local Localization Ablation: This section analyzes the localization results of three algorithms: the Extended Kalman Filter (EKF), the local compensation localization (RL_CF), and the multi-agent cooperative localization (MARL_CF). The experimental results are presented in Fig. 7-(a). Within a localization error of 0.5m, the probability distribution of EKF localization is 69.7%, while RL_CF localization has a probability of 77.6%, and MARL_CF localization has a probability of 93.3%. The local localization module improves accuracy, increasing the probability by 11.3% compared to EKF, and the MARL_CF further increases it by 20.2%, demonstrating the significant advantage of the global cooperative localization algorithm in reducing errors. Additionally, Fig. 7-(b) illustrates that the RL_CF method achieves smaller localization errors at each time step compared to EKF, converging at around 90 steps. On the other hand, the MARL_CF reaches convergence at around 50 steps.

In conclusion, the experimental results highlight the effectiveness of both the local compensation localization module and the global cooperative localization algorithm in



(a) (b)

Fig. 8. Global positioning MARL ablation experiment: (a) cumulative distribution error, and (b) error variation in time sequence.

reducing localization errors. Notably, the multi-agent global cooperative localization module has a more substantial impact on error reduction compared to the local localization module.

2) Global Localization Ablation: To validate the effectiveness of the credit assignment-based global cooperative localization algorithm, this section conducted a comparative experiment with the Extended Kalman Filter (EKF) algorithm and other multi-agent reinforcement learning algorithms based on credit assignment, including the Value Decomposition Networks (VDN) [27] and the QMIX [28]. The experimental results are shown in Fig. 8-(a) and Fig. 8-(b). In the comparison, the EKF algorithm achieved a localization error of 0.87m, while the MARL_CF, VDN, and QMIX algorithms achieved localization errors of 0.33m, 0.52m, and 0.42m, respectively. It is evident that the MARL_CF algorithm outperforms the VDN and QMIX algorithms, achieving higher localization accuracy by 36.5% and 21.4%, respectively. Additionally, the MARL_CF exhibits slightly faster convergence speed.

In conclusion, the experimental results show that the credit assignment-based global cooperative localization algorithm (MARL_CF) has significant advantages in localization tasks. Compared to other algorithms (EKF, VDN, and QMIX), the MARL_CF algorithm not only provides higher localization accuracy but also exhibits faster convergence speed.

Based on the two sets of experimental analyses above, it is evident that the MARL_CF algorithm effectively reduces localization errors. Both the local localization module and the credit assignment-based global cooperative localization module contribute positively to reducing localization errors. By integrating both the local localization module and the global cooperative localization module, the MARL_CF algorithm demonstrates remarkable capability in reducing localization errors and enhancing precision. Especially, the multi-agent global cooperative localization module performs better in reducing localization errors compared to the local module.

E. Physical Experiments

To verify the effectiveness of the proposed MARL_CF, we deployed four auto-robots [29] in a physical field measuring 10m × 10m, as depicted in Fig. 9. Noted that the arena size of the physical filed is different from that used in the simulation due to the limitations of the motion capture system requirements and room space. Simulations offer the flexibility to explore larger and more complex environments without the physical and logistical constraints present in real-world settings. In contrast, physical experiments provide essential validation of the algorithm's efficacy in real environments, albeit on a smaller scale. This dual approach ensures a balanced assessment, capturing both the theoretical robustness and practical applicability of the proposed solution. These robots operated in two modes: random walk and regular motion, for a duration of 100 seconds each. Ground truth data was captured using the NOKOV optics motion tracking system [30]. Subsequently, the trajectory and localization errors of the agents were analyzed to evaluate the algorithm's performance.

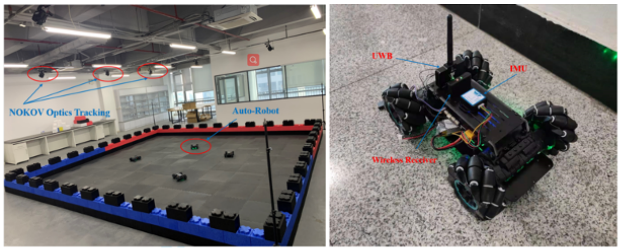
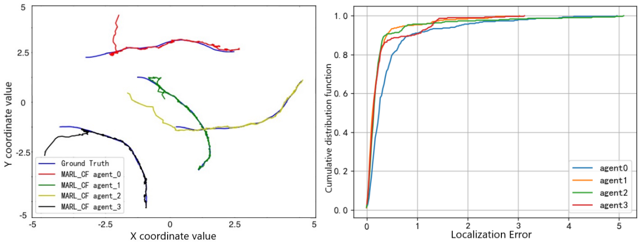


Fig. 9. **Left:** The scene for physical experiment. **Right:** The auto-robot and sensors used for the test.

1) Random Walk Scenario: In this scenario, four agents are deployed, and they start moving randomly in different directions from unknown random initial positions. Fig. 10-(a) shows the localization results of the four agents. It can be observed that with the MARL_CF, each agent can quickly converge to the true trajectory when the initial positions are unknown. Furthermore, Fig. 10-(b) shows the cumulative distribution of localization errors for each agent, which indicates that each agent has a probability of over 80% to reduce the localization error within 0.5 meters. From the above-mentioned two figures, it can be concluded that the MARL_CF demonstrates higher accuracy and robustness in multi-agent localization. Even in situations where the initial positions are unknown, the agents can quickly converge to the correct trajectory, and the error remains within a small range.

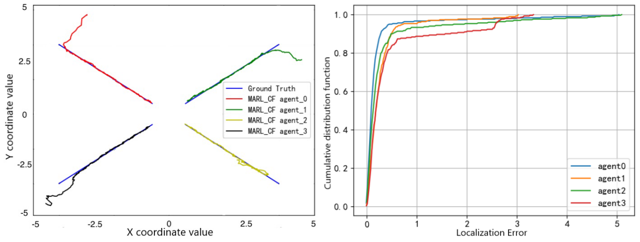
2) Regular Motion Scenario: In this experiment, four agents initiate from the corners of a rectangular structure and converge toward the center at a 45° angle. The localization results are illustrated in Fig. 11-(a) and Fig. 11-(b). Notably, in the scenario of regular motion, the localization errors for each agent are considerably smaller compared to those observed in the random walk scenario. To be specific, each agent exhibits a probability of over 85% to reduce localization errors within a 0.5-meter range. This underscores the significance of a regular topological structure among the agents' movements, which notably contributes to enhanced localization accuracy. Predictable relative positions and relationships between agents facilitate better cooperation, resulting in improved precision.

In summary, these experimental outcomes demonstrate that in multi-agent environments characterized by a regular topological structure, cooperative localization achieves higher accuracy. These findings offer valuable insights for the further comprehension and optimization of multi-agent cooperative localization algorithms. Moreover, we conducted an exhaustive



(a) (b)

Fig. 10. (a) The path of various agents' trajectories and (b) the cumulative distribution function (CDF) of each agent's localization errors in *random walk* scenario.



(a) (b)

Fig. 11. (a) The path of various agents' trajectories and (b) the cumulative distribution function (CDF) of each agent's localization errors in *regular motion* scenario.

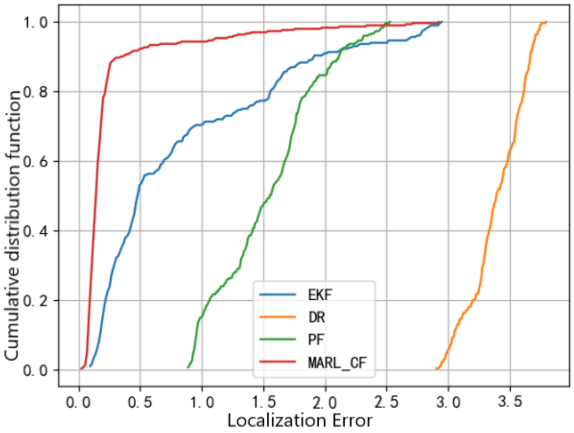


Fig. 12. Cumulative distribution function (CDF) curves of mean squared error (MSE) in physical environment test.

analysis of the agents' trajectory paths and cumulative distribution functions depicting localization errors for various algorithms, including the Extended Kalman Filter (EKF) [9], Dead Reckoning (DR) [25], and Particle Filter (PF) [26]. The trajectory errors of these algorithms are visually presented in Fig. 12, aligning with the simulation results and reaffirming the superior performance of our proposed MARL_CF algorithm when compared to state-of-the-art methods.

VI. Conclusion and Perspective

This paper presents a novel multi-agent collaborative localization algorithm that integrates reinforcement learning and filter correction to address challenges encountered by traditional filtering algorithms. These challenges include sensitivity to initial estimates and adaptability to dynamic environments. The algorithm introduces two methodologies: local localization and global cooperative localization, both aimed at enhancing localization accuracy.

The local localization method utilizes a reinforcement learning-based compensatory Extended Kalman Filter (EKF) algorithm to reduce localization errors. By leveraging reinforcement learning, the EKF algorithm undergoes correction to eliminate errors stemming from initial estimates. Conversely, the global cooperative localization method employs a multi-agent reinforcement learning algorithm based on credit assignment (MARL_CF). This algorithm facilitates information sharing among agents and optimizes overall localization errors through value function decomposition. Agents' policies are refined using an Actor-Critic (AC) network to enhance local localization results. Additionally, the credit assignment network mitigates imbalances among agents, effectively minimizing overall localization errors.

Through simulation and physical experiments conducted within a proposed scenario, the MARL_CF algorithm undergoes comprehensive evaluation under various conditions. The results unequivocally demonstrate its significant superiority over the EKF, DR, and PF algorithms, resulting in enhanced localization accuracy within the localization estimation task.

In conclusion, this study introduces reinforcement learning and cooperative localization techniques, giving rise to the development of local localization and global cooperative localization methods. These methodologies effectively bolster multi-agent localization accuracy. These research findings robustly advocate for the application of localization technology in navigation and autonomous driving fields, while also indicating promising avenues for further research and development. Looking ahead, we are committed to advancing our research by refining our reward structure to incorporate more sophisticated mechanisms of reinforcement learning, such as improved credit assignment algorithms. Such enhancements are designed to optimize the performance of our localization system further, by more efficiently allocating rewards to actions that substantially enhance accuracy and robustness. These future efforts aim to push the boundaries of what our localization technologies can achieve, continuing to drive innovation in the field.

References

[1] A. Coluccia and A. Fascista, "A review of advanced localization techniques for crowdsensing wireless sensor networks," *Sensors*, vol. 19, no. 5, p. 988, Feb. 2019.

[2] R. Wang, C. Xu, H. Wu, Y. Shi, S. Duan, and X. Zhang, "Gaussian condensation filter based on cooperative constrained particle flow," *IEEE Internet Things J.*, vol. 10, no. 15, pp. 13533–13543, May 2023.

[3] Y. Han, C. Wei, R. Li, J. Wang, and H. Yu, "A novel cooperative localization method based on IMU and UWB," *Sensors*, vol. 20, no. 2, p. 467, Jan. 2020.

[4] J. Zhang, Z. Zhao, and S. Zhou, "Vector task map: A progressive distribution method for crowd-sensing tasks," *J. Comput. Sci.*, vol. 40, no. 8, pp. 1946–1960, 2017.

[5] H. Cao, P. Zhang, T. Lu, H. Gu, and N. Gu, "Real-time user activity recognition modeling method based on sensor distance," *Comput. Eng.*, vol. 45, no. 2, pp. 1–6, 2019.

[6] M. Ridolfi, A. Kaya, R. Berkvens, M. Weyn, W. Joseph, and E. D. Poorter, "Self-calibration and collaborative localization for UWB positioning systems: A survey and future research directions," *ACM Comput. Surv.*, vol. 54, no. 4, pp. 1–27, May 2021, doi: 10.1145/3448303.

[7] H. Ahmed and M. Tahir, "Improving the accuracy of human body orientation estimation with wearable IMU sensors," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 3, pp. 535–542, Mar. 2017.

[8] P. Kim and L. Huh, *Kalman Filter for Beginners: With MATLAB Examples*. Seattle, WA, USA: CreateSpace, 2011.

[9] Y. Huang, Y. Zhang, B. Xu, Z. Wu, and J. A. Chambers, "A new adaptive extended Kalman filter for cooperative localization," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 1, pp. 353–368, Feb. 2018.

[10] B. Allotta et al., "A comparison between EKF-based and UKF-based navigation algorithms for AUVs localization," in *Proc. OCEANS*, May 2015, pp. 1–5.

[11] S. P. Tseng, W.-L. Li, C.-Y. Sheng, J.-W. Hsu, and C.-S. Chen, "Motion and attitude estimation using inertial measurements with complementary filter," in *Proc. 8th Asian Control Conf. (ASCC)*, May 2011, pp. 863–868.

[12] C. Xu, Y. Shi, J. Wan, and S. Duan, "Uncertainty-constrained belief propagation for cooperative target tracking," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 19414–19425, Oct. 2022.

[13] R. Liu, C. Yuen, T.-N. Do, D. Jiao, X. Liu, and U.-X. Tan, "Cooperative relative positioning of mobile users by fusing IMU inertial and UWB ranging information," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 5623–5629.

[14] L. Hu, C. Wu, and W. Pan, "Lyapunov-based reinforcement learning state estimator," 2020, *arXiv:2010.13529*.

[15] L. Hu, Y. Tang, Z. Zhou, and W. Pan, "Reinforcement learning for orientation estimation using inertial sensors with performance guarantee," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 10243–10249.

[16] J. Michels, A. Saxena, and A. Y. Ng, "High speed obstacle avoidance using monocular vision and reinforcement learning," in *Proc. 22nd Int. Conf. Mach. Learn. (ICML)*, 2005, pp. 593–600.

[17] J. Morimoto and K. Doya, "ERRATUM: Reinforcement learning state estimator," *Neural Comput.*, vol. 19, no. 3, pp. 730–756, 2007.

[18] Q. Zhang, W. Pan, and V. Reppa, "Model-reference reinforcement learning for collision-free tracking control of autonomous surface vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 8770–8781, Jul. 2022.

[19] J. Hwangbo, I. Sa, R. Siegwart, and M. Hutter, "Control of a quadrotor with reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 2096–2103, Oct. 2017.

[20] M. Zhou et al., "Learning implicit credit assignment for cooperative multi-agent reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 11853–11864.

[21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.

[22] W. Li, W. Liu, S. Shao, S. Huang, and A. Song, "Attention-based intrinsic reward mixing network for credit assignment in multiagent reinforcement learning," *IEEE Trans. Games*, vol. 16, no. 2, pp. 270–281, Jun. 2024, doi: 10.1109/TG.2023.3263013.

[23] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 1–12.

[24] H. Gao, W. Huang, T. Liu, Y. Yin, and Y. Li, "PPO2: Location privacy-oriented task offloading to edge computing using reinforcement learning for intelligent autonomous transport systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 7, pp. 7599–7612, Jul. 2023, doi: 10.1109/TITS.2022.3169421.

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

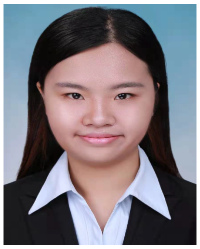
[26] L. Wielandner, E. Leitinger, F. Meyer, and K. Witrisal, "Message passing-based 9-D cooperative localization and navigation with embedded particle flow," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 9, pp. 95–109, 2023.

[27] P. Sunehag et al., "Value-decomposition networks for cooperative multi-agent learning," 2017, *arXiv:1706.05296*.

[28] T. Rashid, M. Samvelyan, C. S. De Witt, G. Farquhar, J. Foerster, and S. Whiteson, "Monotonic value function factorisation for deep multi-agent reinforcement learning," *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 7234–7284, 2020.

[29] Nooploop. *Autorobo*. Accessed: Jul. 21, 2023. [Online]. Available: https://www.nooploop.com/en/autorobo/

[30] *Nokov*. Accessed: Jul. 21, 2023. [Online]. Available: https://en.nokov.com/direct



Ran Wang (Graduate Student Member, IEEE) received the B.E. degree from Beijing Information Science and Technology University, China, in 2013, and the M.S. degree from the University of Science and Technology Beijing (USTB), China, in 2016, where she is currently pursuing the Ph.D. degree. Her research interests include distributed security, blockchain, and the Internet of Things.

Cheng Xu (Member, IEEE) received the B.E., M.S., and Ph.D. degrees from the University of Science and Technology Beijing (USTB), China, in 2012, 2015, and 2019, respectively. He is currently an Associate Professor with the School of Computer and Communication Engineering, USTB. He is supported by the Post-Doctoral Innovative Talent Support Program from Chinese Government in 2019. His current research interests include swarm intelligence, multi-agent reinforcement learning, distributed security, and the Internet of Things.

Jing Sun received the master's degree from the University of Science and Technology Beijing in 2023. Her research interests include distributed security, multi-modal navigation, and the Internet of Things.

Shihong Duan received the Ph.D. degree in computer science from the University of Science and Technology Beijing (USTB). She is currently an Associate Professor with the School of Computer and Communication Engineering, USTB. Her research interests include wireless indoor positioning, multi-robots networks, and the Internet of Things.

Xiaotong Zhang (Senior Member, IEEE) received the M.S. and Ph.D. degrees from the University of Science and Technology Beijing in 1997 and 2000, respectively. He was a Professor with the Department of Computer Science and Technology, University of Science and Technology Beijing. His research interests include material big-data, database systems, and the Internet of Things.