Reinforcement Learning-Based Dynamic Coverage Control of Multi-Rotor UAVs With Safety Priority

Zhuangzhuang Ma, Junjie You, Yunlin Zhang, Yuhua Cheng, and Jinliang Shao, *Member, IEEE*

***Abstract*— This paper considers the dynamic coverage control of multi-rotor Unmanned Aerial Vehicles (UAVs) with the limited sensory range, which aims to collect sensor information from all points of interest in the given task area until the desired prescribed level is reached. However, the unknown environments are usually unavoidable for coverage task, where the presence of various obstacles and communication interferences affect the flight safety and communication stability of UAVs. Therefore, collision avoidance and connectivity maintenance are considered as the two safety issues in this paper, in which connectivity maintenance ensures the communication environment for UAVs to collaboratively accomplish task, and collision avoidance is used for UAVs to avoid obstacles and neighbors. In order to realize dynamic coverage control with safety constraints based on local environment information, this paper proposes the reinforcement learning-based algorithm with shield, where the shield designed by discrete-time Control Barrier Function (CBF) not only ensures the safety of the UAVs in the learning and control phases, but also maximizes the coverage performance of UAVs. In addition, each UAV only relies on local information to generate safe actions for advancing the coverage process during the execution phase. Finally, the effectiveness of the algorithm is verified by numerical simulations and physical experiment.**

***Note to Practitioners*—A typical application scenario of dynamic coverage control is search and rescue (SAR), in which UAVs equipped with multiple sensors focus on monitoring areas where trapped people may be present, e.g., anomalous areas detected by infrared sensors due to human body temperature. Since SAR always occurs in unknown environments, it is crucial to ensure the safety of UAVs during missions, of which the safety issues considered in this paper include collision avoidance and connectivity maintenance. To perform the SAR mission safely, we construct the CBF-based shield, which minimizes corrections the exploration actions of the UAVs and ensures the safety of the cluster during the mission. In addition, UAVs are difficult to obtain global environmental information in unknown environments and only rely on their sensors to collect local information. Therefore, the reinforcement learning algorithm with shield proposed in this paper adopts the centralized training and decentralized execution strategy, where the UAVs only need local observation information to plan their next actions. Physical experiments were also conducted to validate the feasibility of implementing the proposed algorithm using real UAVs.**

***Index Terms*— Safety critical systems, reinforcement learning, dynamic coverage control, multi-rotor UAVs.**

I. Introduction

COVERAGE is a cooperative behavior of biological groups, and it is widely present in a variety of tasks such as ant foraging and wolf hunting. Inspired by the coverage behavior of clusters in nature, which can generate efficient and complex swarm intelligence, researchers have devoted a lot of attention to coverage problem and applied it to different scenarios, including rescue [1], tracking [2] and emergency communications [3].

A. Related Work & Motivation

Coverage control originated from the optimal mobile sensor placement problem, which causes agents to spread out over the task domain while aggregating in areas of high sensory interest. In a groundbreaking work [4], the coverage problem was formally established as minimizing a loss function with respect to location. To achieve optimal coverage, an iterative method based on Voronoi cells (known as the Lloyd algorithm) was proposed. The existing coverage control methods can be generally classified into Voronoi-based [4], probabilistic-based [5] and potential-based [6]. While the agents described earlier ultimately occupied optimal positions, it is possible that certain points of interest (PoIs) remain unexplored within their sensing domain, particularly when the sensing range falls short of covering the entire domain. Based on this fact, [7] initially introduced the concept of dynamic coverage, aiming to ensure that all involved PoIs are sensed through dynamic movement. Along this line, the literatures [8], [9], [10], [11] designed velocity control law for issues of safety and local optimum, respectively.

Multi-rotor UAVs are often chosen for dynamic coverage tasks due to their versatility in deployment and payload capabilities. Ensuring the safety of these mobile platforms throughout the dynamic coverage is a crucial concern, particularly as safety-critical systems where violations of constraints during learning or control may lead to unacceptable failures or damage. The first constraint is related to the agents' dynamics. The results [8], [9], [10], [11] mainly focus on first-order dynamics, and the proposed velocity controller cannot fully capture the high-order nonlinearity of the dynamics, which

Manuscript received 15 March 2024; revised 22 May 2024; accepted 21 June 2024. Date of publication 1 July 2024; date of current version 8 July 2025. This article was recommended for publication by Editor C. Huang upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant 62273077 and in part by the Natural Science Foundation of Sichuan Province under Grant 2024NSFJQ0013. *(Corresponding author: Jinliang Shao.)*

Zhuangzhuang Ma, Junjie You, Yunlin Zhang, and Yuhua Cheng are with the School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China.

Jinliang Shao is with the School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China, and also with the Laboratory of Electromagnetic Space Cognition and Intelligent Control, Beijing 100089, China (e-mail: jinliangshao@126.com).

Digital Object Identifier 10.1109/TASE.2024.3420094

restricts the applications in this scenario. The dynamics of multi-rotor UAVs consists of coupled trajectory dynamics and attitude dynamics. Thanks to the fact that the time constant of trajectory dynamics is much larger than that of attitude dynamics, literatures [12], [13], [14] used a two-layer control structure to achieve decoupling of dynamics, where the inner-loop controller determines the thrust of the UAV to track the attitude, while the outer-loop controller drives the UAV to reach the desired position. Hence, if the inner-loop controller is effective, it is sufficient to solely design the outer-loop controller for both coverage and safety purposes. This outer-loop controller is typically represented as a double-integrator in [12], [13], and [14].

In most collaborative control applications involving UAVs, effective behavior coordination largely relies on the successful exchange of information, including perception data and control commands. Consequently, maintaining a dependable wireless communication network is paramount for team performance. Nevertheless, in practical communication settings, wireless channels frequently encounter issues such as path loss, shadowing, and multipath fading. They may also be subject to unintentional or intentional electromagnetic interference, all of which can compromise channel reliability and limit the communication range [15], [16]. The condition for connectivity maintenance is usually that the second smallest eigenvalue of the graph Laplacian matrix, also known as algebraic connectivity [17], is greater than 0. Since algebraic connectivity is a concave function of the Laplacian matrix, an optimization-based connectivity controller is generated [18], which relies on maximizing the algebraic connectivity. Besides, in formation control, the CBF method based on the minimum cost spanning tree [19] and the potential function method [20] were also used to maintain cluster connectivity. However, [4], [5], [6], [7], [8], [9], [10], [11] ignored connectivity maintenance constraint in dynamic coverage. Recently, only the literatures [21] and [22] considered dynamic coverage control with connectivity constraint by maintaining the initial topology of the cluster, where [21] designed the connectivity control law by solving the gradient of the potential field function, and [22] obtained the safety policy by constructing the penalty function through reinforcement learning. Nevertheless, both of [21] and [22] fixed the communication topology with single-integrator model, which limits the diffusion of cluster, and further inhibits the coverage effect.

Another important constraint during the coverage process is collision avoidance. In the literatures [8], [9], [10], and [11], the collision avoidance control law based on potential field function was designed for the single-integral model, and the gain parameters were used to linearly combine it with the gradient-based coverage control to achieve coverage control. Reference [23] studied the finite-time formation problem and introduced the potential field method with the double-integrator model to achieve collision avoidance. Due to the conservative behavior caused by improper gain parameter selection in the potential field method, [24] used a quadratic programming controller based on Control Barrier Function (CBF) to achieve collision avoidance in autonomous driving, seeking a compromise between performance and safety.

In addition, coverage control methods [4], [5], [6], [7], [8], [9], [10], [11], [21] require global information to measure the coverage effect and formulate control laws. However, UAVs often perform search and rescue (SAR) in unknown environments, i.e., the UAVs only obtain local information rely on their sensors. Therefore, formulating the suitable strategy based on the known local information to accomplish the dynamic coverage task with safety constraints is a problem more in line with practical applications. In recent years, the Multi-Agent Reinforcement Learning (MARL) approach has shown its powerful learning ability when handling the difficulties brought by state constraints and multi-target [25], [26], [27].

However, direct usage of MARL in coverage may face some tough challenges as there is the need to constantly learn through trial-and-error. This way not only limits MARL's application in the real world, but also leads to a sparse reward space, which makes the actions slow to converge [28]. Safety assurance in traditional MARL [29], [30], [31] is accomplished by formulating violations in a cost function and limiting the expectation of cumulative costs below a threshold. The safety constraints actually act on the cost of safety violations, so it is difficult to effectively enforce hard reachability-based constraints, which means that [22], [32] did not guarantee the safety of the agents throughout the training phase. To avoid violations during the learning process, [33], [34] introduced the shields in the MARL framework to filter non-safe actions, i.e., modifying the action to stop when agent will violate the safety constraint. Although the method can effectively ensure safety, the shields ignore the limitations of the dynamic model and the corrections of the original actions are large, which is not conducive to agents accomplishing their tasks.

In summary, [4], [5], [6], [7], [8], [9], [10], [11] utilized the gradient-based coverage control law and potential field to achieve the dynamic coverage without connectivity maintenance constraint, where the agents are modeled as the single-integrator. Reference [21] further maintained the initial topology of the cluster with potential field method. Compared with [4], [5], [6], [7], [8], [9], [10], [11], and [21], which require the global environment information, MARL [22] can achieve dynamic coverage with collision avoidance and connectivity maintenance of initial topology based on local information. However, [22] cannot guarantee the safety of the agents during the training phase. Therefore, it is necessary to design algorithm based on local information to solve the dynamic coverage control problem with safety constraints.

B. Contributions & Outline

We utilize the idea of shields into MARL algorithm to solve the dynamic coverage control problem with safety constraints, which can avoid connectivity disruption and collision violations of the UAVs due to random exploration during the learning process. Different from the stopping strategy in literatures [33], [34], we use the concept of discrete time CBF to encode the safety constraints and design CBF-based shield to ensure UAVs remain within the safety set, which allows for minimal correction of hazardous actions by introducing CBF-based shield into the Quadratic Programming (QP) controllers and weakens the effect of the connectivity constraints on the coverage performance.

According to the above discussion, the primary contributions of this paper can be summarized in three aspects as follows.

1) This paper considers the dynamic coverage control problem subject to safety constraints, including collision avoidance and connectivity maintenance. Different from the existing coverage works [4], [5], [6], [7], [8], [9], [10], [11], that focus on theoretical results, we study dynamic coverage control in SAR mission of multi-rotor UAVs, and further consider the safety constraints in practical applications. Furthermore, to apply the algorithm to safety-critical systems, we designed a MARL-based controller with safety as the primary concern.
2) For dynamic coverage control, we select Multi-agent Deep Deterministic Policy Gradient (MADDPG) [35] as the underlying learning framework, and the shield is designed with the help of discrete-time CBF. Compared with the learning algorithms [22], [29], [30], [31], [32], [33], [34], we encode safety constraints using CBF, which not only ensures the safe at any stage in learning, but also minimizes the correction of dangerous actions and maintains the coverage performance of agents.
3) An efficient discrete-time CBF is designed based on position with velocity compensation. Meanwhile, with the help of the critical robot, CBF utilizing local information is proposed to maintain connectivity through distributed execution. Theory and simulations verify the effectiveness of the proposed discrete-time CBF and algorithm.

The rest of the paper is divided into five parts. Section II provides a brief rundown to connectivity and CBF. The mathematical form of dynamic coverage control is given in Section III. Section IV utilizes the framework of MADDPG with CBF shields to achieve safe policy learning. The simulations and physical experiment used to verify our algorithm are given in Section V. Finally, Section VI presents the conclusion.

II. Preliminaries

A. Notations

The set of real numbers and the set of positive real numbers are denoted by $\mathbb{R}$ and $\mathbb{R}_+$, respectively. $\mathbb{R}^{n\times m}$ refers to the $n \times m$-dimensional real matrix space. For the vector $\boldsymbol{x} \in \mathbb{R}^{n\times 1}$, $\|\boldsymbol{x}\|_2^2 = \sum_{i=1}^{n} (\boldsymbol{x}_i)^2$.

B. Connectivity

The undirected graph $\mathcal{G}(t_k) = (\mathcal{V}, \mathcal{E}(t_k))$ is used to model the interactions of n agents at time t_k, where $\mathcal{V} = \{1, 2, \ldots, n\}$ and $\mathcal{E}(t_k)$ are the set of agents and the set of communication links, respectively. Let $\boldsymbol{x}_i \in \mathbb{R}^{2\times 1}$ denote the position of agent i. Assumed that the communication capability of agents are limited and they can only exchange information with neighbors in a circular area of radius R_c. Once agent i establishes the communication connection with j, i.e., $\|\boldsymbol{x}_i(t_k) - \boldsymbol{x}_j(t_k)\|_2 \leq R_c$, then edge $(i, j) \in \mathcal{E}(t_k)$. Since the communication of agents is two-sided, $(j, i) \in \mathcal{E}(t_k)$, if $(i, j) \in \mathcal{E}(t_k)$. Further, the neighbor set of the i-th agent is defined as $\mathcal{N}_i(t_k) = \{j \in \mathcal{V} | (i, j) \in \mathcal{E}(t_k)\}$. Let $\mathcal{A} = [a_{ij}]_{n\times n}$ is the adjacency matrix associated with graph $\mathcal{G}(t_k)$, where $a_{ij} > 0$ if and only if $(i, j) \in \mathcal{E}(t_k)$; otherwise, $a_{ij} = 0$. If there is a series of edges $(i_0, i_1), (i_1, i_2), \ldots, (i_{r-1}, i_r) \in \mathcal{E}$, where $i_0, i_1, \ldots, i_r \in \mathcal{V}$ are distinct, then the path $\mathcal{P}_{i_0\to i_r}$ exists, which means that i_0 and i_r are connected. The graph $\mathcal{G}(t_k)$ is connected if there exists a path connecting any two agents. Algebraic connectivity (also known as the Fiedler value) λ_2 is often used mathematically to measure the connectivity of networks, and it is the second smallest eigenvalue of the Laplacian matrix of graph $\mathcal{G}(t_k)$. It can be seen from literature [36] that $\lambda_2 > 0$ if and only if the graph $\mathcal{G}$ is connected.

C. Control Barrier Function

This section introduces CBF controller to guarantee safety of agents. Consider a discrete-time control system as follows:

$$\boldsymbol{x}(t_{k+1}) = f(\boldsymbol{x}(t_k)) + g(\boldsymbol{x}(t_k))\boldsymbol{u}(t_k), \tag{1}$$

where $\boldsymbol{x}(t_k) \in \mathbb{R}^n$, $\boldsymbol{u}(t_k) \in \mathbb{R}^m$ denote the system state and control input at time t_k, respectively. $f(\cdot)$ and $g(\cdot)$ compose a known nominal model of the dynamics.

The purpose of CBFs is to keep the system (1) in a certain closed set. Assume that the set can be represented by a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$, then the safe set $\mathcal{C}_s$ and the non-safe set $\mathcal{C}_n$ are defined as:

$$\begin{aligned} \mathcal{C}_s &= \{\boldsymbol{x} \in \mathbb{R}^n | h(\boldsymbol{x}) \geq 0)\}, \\ \mathcal{C}_n &= \{\boldsymbol{x} \in \mathbb{R}^n | h(\boldsymbol{x}) < 0)\}. \end{aligned} \tag{2}$$

The function $h(\cdot)$ is called as the CBF in the discrete-time [37], if there exists $\eta \in (0, 1]$ such that:

$$h(\boldsymbol{x}(t_{k+1})) - h(\boldsymbol{x}(t_k)) \geq -\eta h(\boldsymbol{x}(t_k)), \tag{3}$$

where the lower bound of $h(\cdot)$ decreases exponentially with the rate $1 - \eta$.

Lemma 1 ([37]): Given the safe set $\mathcal{C}_s$ defined by (2) and the continuous function h. If h is a discrete-time CBF, any discrete-time controller $\boldsymbol{u}(t_k)$ satisfying (3) will render the set $\mathcal{C}_s$ forward invariant.

III. Problem Formation

This paper study the dynamic coverage control problem of a multi-rotor UAV system with connectivity maintenance as well as collision avoidance of neighbors and obstacles constraints in the unknown environment. Consider N multi-rotor UAVs are working in fixed height, whose positions are represented by $\boldsymbol{x}_i \in \mathbb{R}^2$, $i \in \{1, 2, \ldots, N\}$, to monitor M PoIs, whose positions are represented by $\boldsymbol{p}_j \in \mathbb{R}^2$, $j \in \{1, 2, \ldots, M\}$. PoIs are randomly distributed in the two-dimensional region $\mathcal{W}$, and it is assumed that $N < M$. In addition, there are L obstacles in the task area, whose center position is denoted as $\boldsymbol{p}_l^{ob}$, $l \in \{1, 2, \ldots, L\}$. Due to UAVs have limited communication range and coverage capabilities, they have to collaborate with others to complete the monitoring task while maintaining connectivity for information exchange.

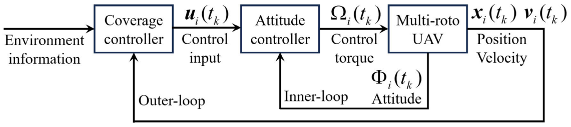


Fig. 1. The schematic diagram of the two-layer structure.

A. Multi-Rotor UAVs Control Model

Similar to literatures [12], [13], [14], a two-layer control structure is used for the multi-rotor UAV control in this paper, as shown in Fig. 1, where $\Omega(t_k)$ and $\Phi(t_k)$ represent the control torque and attitude, respectively. We assume that the inner-loop controller is efficient enough to allow us to focus on the design of the outer-loop controller, which can be represented as a double-integrator dynamics. Since the coverage effect of a multi-rotor UAV system at the same height is only related to position and velocity, a double-integrator is sufficient to model the problem. Specifically, the dynamics of the outer-loop controller can be written as

$$\begin{aligned}\boldsymbol{x}_i(t_{k+1}) &= \boldsymbol{x}_i(t_k) + \Delta t \boldsymbol{v}_i(t_k),\\ \boldsymbol{v}_i(t_{k+1}) &= \boldsymbol{v}_i(t_k) + \Delta t \boldsymbol{u}_i(t_k),\end{aligned} \tag{4}$$

where Δt is the time step, $\boldsymbol{x}_i(t_k) \in \mathbb{R}^{2\times 1}$, $\boldsymbol{v}_i(t_k) \in \mathbb{R}^{2\times 1}$, and $\boldsymbol{u}_i(t_k) \in \mathbb{R}^{2\times 1}$ capture the center of mass, velocity and control input of i-th UAV at time t_k, respectively. Meanwhile, we limit the maximum value of the UAVs velocity in order to fit the practice applications: $\|\boldsymbol{v}_i(t_k)\|_2 \leq v_{max}$.

B. Dynamic Coverage Control Model

Assumed that the sensor sensing area of a UAV is a circular area with the center $\boldsymbol{x}$ and radius R_s, and the peak sensing capability is $S_c > 0$. Due to the recognition of obstacles is not the focus of this paper, it is assumed that the UAVs can recognize the obstacles and model them as circles if they are in UAVs' sensing range. Different from the linear function used in literatures [7], [8], [9], [10], [11] to represent the relationship between sensing distance and sensing quality, a bell-shaped function is used here, whose reason will be explained later. The sensing model of i-th UAV at time t_k for the PoI $\boldsymbol{p}_j$ can be expressed as the following function $P(\cdot)$:

$$P(\boldsymbol{x}_i(t_k), \boldsymbol{p}_j) = \begin{cases} S_c \cdot e^{-\frac{d^2}{{R_s}^2}}, & d \leq R_s,\\ 0, & d > R_s,\end{cases} \tag{5}$$

where $d = \|\boldsymbol{x}_i(t_k) - \boldsymbol{p}_j\|_2$. According to the sensing model (5), it is clear that the sensing quality of PoI j reaches the peak when i-th UAV overlaps $\boldsymbol{p}_j$, and decreases when UAV moves away from j-th PoI.

Remark 1: Different from the traditional coverage control literature that assumes infinite-range sensors, the sensing model (5) captures a variety of sensors with finite sensing ranges such as vision-based cameras, infrared cameras, and radar. The bell-shaped function is used to measure the effect of distance on the sensor's capability, e.g., the closer a UAV carrying a vision-based camera gets to the target, the more pixel points of the target the camera captures, which means that the sensor perceives the target more clearly. The camera captures the most target pixel points when the UAV is directly above the target, i.e., the sensing quality reaches the peak. It is important to emphasize here that any model with a finite sensing range characteristic is allowed in this paper.

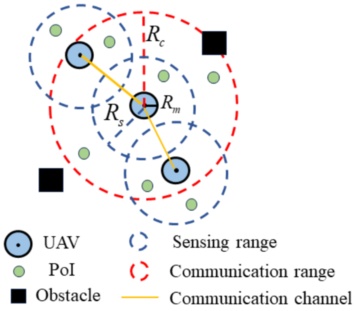


Fig. 2. UAVs performing the coverage task with connectivity maintenance.

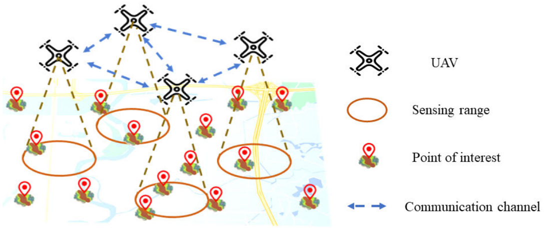


Fig. 3. An example of dynamic coverage control.

The goal of dynamic coverage control is to maximize the coverage capability of the UAVs team while ensuring information exchange and collision avoidance based on the communication radius R_c, physical radius R_m and sensing radius R_s of the UAV model defined above. Fig. 2 provides an example of the constrained coverage task, where the blue circles, green circles and black squares denote the UAVs, PoIs and obstacles, respectively. The blue dotted circles and red dotted circles are the maximum coverage and communication range of the UAVs, respectively. The yellow lines indicate the communication channel.

The effective coverage of all UAVs for PoI j is defined as:

$$\Gamma_N(\boldsymbol{p}_j, t_k) = \sum_{t=t_0}^{t_k} \sum_{i \in N} P(\boldsymbol{x}_i(t), \boldsymbol{p}_j). \tag{6}$$

The decision variables of formula (6) are the UAVs' trajectories. Let C^* be the desired attained effective coverage at all PoIs. The goal is to attain the effective coverage with $\Gamma_N(\boldsymbol{p}_j, t_k) = C^*$ for each PoI. Define the error function:

$$e(t_k) = \sum_{j=1}^{M} f_p(C^* - \Gamma_N(\boldsymbol{p}_j, t_k)), \tag{7}$$

where $f_p(x) = \max\{0, x\}$ is a penalty function if the effective coverage of j-th PoI does not reach C^*, i.e., $\Gamma_N(\boldsymbol{p}_j, t_k) < C^*$. Once $\Gamma_N(\boldsymbol{p}_j, t_k) \geq C^*$, the error function is zero at this PoI, that is, the extra coverage time spent at this PoI will not increase the coverage performance. The dynamic coverage control task is completed if $e(t_k) = 0$. We show an example of dynamic coverage control in Fig. 3.

At this point, the dynamic coverage control problem is formulated as a optimization problem subject to a group of complex constraints:

$$\arg\min_{\boldsymbol{u}} e(t_k) \tag{8}$$

$$s.t. \quad \lambda_2(t_k) > 0, \tag{9}$$

$$\|\boldsymbol{x}_i(t_k) - \boldsymbol{x}_j(t_k)\|_2 \geq 2R_m, \forall i \neq j, \tag{10}$$

$$\|\boldsymbol{x}_i(t_k) - \boldsymbol{p}_l^{ob}\|_2 \geq R_o, l \in \{1, \dots, L\}, \tag{11}$$

$$\begin{aligned} \boldsymbol{x}_i(t_k) &= \boldsymbol{x}_i(t_{k-1}) + \Delta t \boldsymbol{v}_i(t_{k-1}), \\ \boldsymbol{v}_i(t_k) &= \boldsymbol{v}_i(t_{k-1}) + \Delta t \boldsymbol{u}_i(t_{k-1}), \end{aligned} \tag{12}$$

where $\min_{t_k} e(t_k)$ indicates the minimum time to complete the dynamic coverage, R_m is the distance from the boundary to the center of UAV, and R_o is the safe distance between the UAVs and the obstacle. The condition (9) indicates that the cluster needs to maintain communication connectivity at all times. Conditions (10) and (11) ensure that the UAVs need to avoid collisions with neighbors and obstacles, respectively. Finally, condition (12) is the dynamic model constraint of all UAVs, which contributes to the implementation of the control law on the actual platform. If the connectivity maintenance constraint (9) is not considered, and the dynamic model is simplified to single-integrator, the problem (8) can be solved by the gradient-based algorithm with the help of global environment information. In order to balance the coverage effect and the connectivity maintenance constraint of the cluster, this paper applies MARL algorithm to learn control strategies with safety priority.

IV. Main Results

This section details the safe MARL algorithm used to solve the constrained dynamic coverage problem.

Traditional reinforcement learning methods such as Q-learning cannot be applied to the multi-agent systems, the reason is that for each agent, the others are part of the environment. This leads to an unstable environment and violates the Markov assumptions required for Q-learning convergence. In this paper, we adopts the centralized training and decentralized execution (CTDE) strategy to ensure that the agents are in a stable environment, and Multi-Agent Markov Decision Process (MAMDP) to model the dynamic coverage control scenario.

A. MAMDP

A MAMDP is a tuple $(S, A, r, N, \mathcal{P})$ consisting of the joint state $\boldsymbol{s} \in S^N$, joint action $\boldsymbol{a} \in A^N$, rewards $\boldsymbol{r} \in \mathbb{R}^N$ observed by the N agents in the environment, and state transition probability $\mathcal{P}(\boldsymbol{s}(t_{k+1})|\boldsymbol{s}(t_k), \boldsymbol{a}(t_k))$. From the literature [36], the continuous-time double-integrator could be transformed to MDP by discretization. Here we consider a homogeneous setting, where all agents have the same state space, action space, and reward function. In the dynamic coverage task, the state $\boldsymbol{s}_i(t_k)$ of i-th UAV contains PoIs, obstacles, the position and velocity of itself and its neighbors, i.e.,

$$\begin{aligned} \boldsymbol{s}_i(t_k) = \big\{&\boldsymbol{x}_i(t_k), \boldsymbol{v}_i(t_k), \{\boldsymbol{p}_l^{ob}\}_{l\in L}, \{\boldsymbol{x}_j(t_k)\}_{j\in N, j\neq i}, \\ &\{\boldsymbol{v}_j(t_k)\}_{j\in N, j\neq i}, \{\boldsymbol{p}_m, C^* - \Gamma_N(\boldsymbol{p}_m, t_k)\}_{m\in M}\big\}. \end{aligned}$$

In the execution phase, the fact that the UAVs only access the local environmental information based on the limited communication radius allows them to rely only on interactions with their neighbors to obtain information of other PoIs.

In our case, the action of i-th UAV is

$$\boldsymbol{a}_i(t_k) = \boldsymbol{u}_i(t_k) \in \mathbb{R}^{2\times 1}.$$

For each time step t_k, each agent receives the state $\boldsymbol{s}_i(t_k)$ and takes the action $\boldsymbol{a}_i(t_k)$ based on the deterministic policy $\boldsymbol{\pi}_i^{\boldsymbol{\theta}}$, which is parameterized by $\boldsymbol{\theta}$. Then the state is transferred to $\boldsymbol{s}_i(t_{k+1})$ according to the state transition probability $\mathcal{P}(\boldsymbol{s}_i(t_{k+1})|\boldsymbol{s}_i(t_k), \boldsymbol{a}_i(t_k))$, which remains unknown to the agent. The i-th UAV receives reward $r_i(t_k)$ based on state $\boldsymbol{s}_i(t_k)$ and action $\boldsymbol{a}_i(t_k)$, and its aim is to maximize its total reward $R = \sum_{k=0}^{T} \gamma^k r_i(t_k)$, where γ is a discount factor and T is the time horizon.

B. Safe Multi-Agent Reinforcement Learning Framework

This paper selects MADDPG as the underlying algorithm to solve the dynamic coverage task, and MADDPG uses the Actor-Critic framework, including an Actor network and a Critic network of each agent. The Actor network plays the role of decision making and the Critic network evaluates the expected reward under the current joint state-action pair and guiding the policy update. Actor network makes decisions based on the observation $\boldsymbol{o}$ about the environment. Since this paper considers connectivity maintenance constraint, the agents can obtain PoIs' information by communicating with their neighbors, i.e., the observations of the agents are equivalent to the states in our dynamic coverage problem. In the following, we still use the term observation value in our algorithm.

The main reason for selecting MADDPG as the underlying algorithm for dynamic coverage task is that

1) the dynamic coverage control problem is considered with UAVs as the mobile carriers, given that the action space during the learning process is continuous, which is in line with the mission scenario of MADDPG.
2) MADDPG adopts the CTDE strategy, where the UAVs only utilize local observation information for decision making during execution, which facilitates the distributed execution of dynamic coverage tasks.
3) MADDPG does not require the assumption of the environment model and the specific communication structure between all UAVs.

It is important to note that the traditional MADDPG to addressing safety during learning process is to encode safe constraints using the cost function, which limits the policy search space through penalties. However, it is often difficult to enforce hard safe constraints, i.e., the system state may reach unsafe regions. Therefore, this approach to learning policies is unacceptable for safety-critical systems because violating safety constraints can have devastating consequences. Due to the coverage task and safe constraints are tightly coupled, we introduce the safety MARL framework into dynamic coverage control, inspired by literature [34].

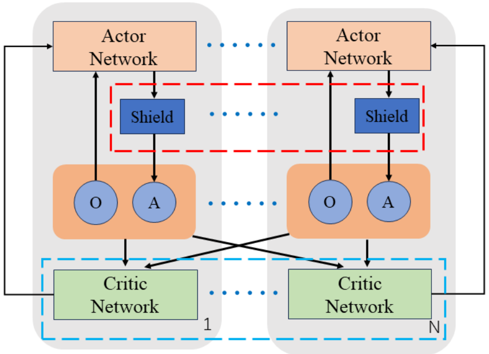


Fig. 4. The safe MARL framework with shield.

The main improvement of safe MARL in our work is the introduction of shield (the parts within the red dashed box in Fig. 4) as a policy filter in the direct interaction of UAVs with the environment, which corrects the risky actions during the training process. If the shield detects that the action violates safety constraints in the current state, it replaces the original action with a safe action and gives a penalty to help UAVs understand the safety boundary. To ensure that all UAVs remain within the safety region during the learning process, we construct CBF-based shields to directly encode the hard safety constraints at each point on the UAVs' trajectory. Next, we present the construction of the CBFs for maintaining cluster connectivity and collision avoidance, and provide the theoretical analysis of effectiveness.

C. CBF-Based Shields

In the discrete decision space, a common shielding method is cancel UAVs' action and return when moving in danger region. However, this method greatly reduce the exploration efficiency in the continuous decision space, so we design the CBF to fine-tune the dangerous action to safe.

Since the discrete-time dynamic model is considered, the action $\boldsymbol{u}_i(t_k)$ of each agent $i \in \{1, 2, \ldots, N\}$ at time t_k will continue until $t_k + \Delta t$, which indicates that the velocity tracking of agent is hysteretic. The position $\boldsymbol{\xi}_i(t_k)$ with the velocity compensation at time step t_k is defined as:

$$\boldsymbol{\xi}_i(t_k) = \boldsymbol{x}_i(t_k) + \Delta t \boldsymbol{v}_i(t_k). \tag{13}$$

The following analysis is performed based on the agent's compensation position $\boldsymbol{\xi}_i(t_k)$. In order to distinguish the neighbors set of the agent's real position, $\tilde{\mathcal{N}}_i(t_k) = \{j \mid \|\boldsymbol{\xi}_{ij}(t_k)\|_2 \le R_c\}$ is defined as the neighbors set of the agent's compensation position, where $\boldsymbol{\xi}_{ij}(t_k) = \boldsymbol{\xi}_i(t_k) - \boldsymbol{\xi}_j(t_k)$.

To enable distributed execution of the proposed algorithm, design the distributed discrete-time CBF to maintain cluster connectivity and collision avoidance is necessary. By introducing the concept of critical robots, each robot can understand its local connection status.

Definition 1 (Critical Robot [38]): The robot j is called *Critical Robot* of robot i, if $j \in \tilde{\mathcal{N}}_i^f$ and $\nexists q : q \in \tilde{\mathcal{N}}_i^n \cap \tilde{\mathcal{N}}_j^n$, where $\tilde{\mathcal{N}}_i^f = \{j \in \tilde{\mathcal{N}}_i \mid \|\boldsymbol{\xi}_{ij}\|_2 > R_\epsilon\}$, $0 < R_\epsilon < R_c$ and $\tilde{\mathcal{N}}_i^n = \tilde{\mathcal{N}}_i - \tilde{\mathcal{N}}_i^f$ denote the *far-neighbor set* and *near-neighbor set* of robot i, respectively.

Critical robot actually refers to the agent, which belongs to far neighbors without a common near-neighbor. Defining the critical robot neighbor set $\tilde{\mathcal{N}}_i^c$ of robot i, and let $\tilde{E}_i^c = \{\tilde{e}_{ij} : j \in \tilde{\mathcal{N}}_i^c\}$ represent its relative critical edges set. At this time, the connectivity maintenance constraint of the cluster is transformed into the connectivity of the critical edges, that is, the edges $\tilde{e}_{ij}$ of robot $j \in \tilde{\mathcal{N}}_i^c$. The next lemma illustrates the importance of maintaining critical edges connectivity.

Lemma 2: Assume the network $\tilde{\mathcal{G}}$ based on velocity compensation position $\boldsymbol{\xi}$ is initially connected and the maximum speed of each robot satisfies $v_{max} \le \frac{R_c - R_\epsilon}{6\Delta t}$. For any existed edge $\tilde{e}_{ij}$ at time t_k, if the interruption of $\tilde{e}_{ij}$ at time $t_k + \Delta t$ would cause $\tilde{\mathcal{G}}$ unconnected, the $\tilde{e}_{ij} \in \tilde{E}_i^c$ at time t_k.

Proof: Based on the constraints of v_{max}, we can get that

$$\begin{aligned}
&\|\boldsymbol{\xi}_{ij}(t_k + \Delta t) - \boldsymbol{\xi}_{ij}(t_k)\|_2 \\
&\quad = \|\Delta t(\boldsymbol{v}_i(t_k) - \boldsymbol{v}_j(t_k)) + \Delta t^2(\boldsymbol{u}_i(t_k) - \boldsymbol{u}_j(t_k))\|_2 \\
&\quad \le 6\Delta t \cdot v_{max} \le R_c - R_\epsilon,
\end{aligned}$$

which also means that

$$\|\boldsymbol{\xi}_{ij}(t_k)\| \ge \|\boldsymbol{\xi}_{ij}(t_k + \Delta t)\| - R_c + R_\epsilon > R_\epsilon,$$

if $\|\boldsymbol{\xi}_{ij}(t_k + \Delta t)\| > R_c$. That is to say, for agents i, the agent j can be regarded as the far neighbor of the agent i i.e., $j \in \tilde{\mathcal{N}}_i^f$ before this edge $\tilde{e}_{ij}$ is interruption. Based on the above conclusion, proof by contradiction is used to analyze the conclusion of Lemma 2. According to Definition 1, if the edge $\tilde{e}_{ij}$ is not critical, there should be at least one common agent l such that $l \in \tilde{\mathcal{N}}_i^n \cap \tilde{\mathcal{N}}_j^n$. Therefore, the edge $\tilde{e}_{ij}$ disruption would not make $\tilde{\mathcal{G}}$ disconnected, which contradicts the proposition. This completes the proof. ■

Lemma 2 provides an idea for designing distributed CBF with connectivity maintenance. Due to any interruption of edge that may cause network disconnected is always detected as a critical edge in advance, the connectivity problem of cluster, which requires global information to measurement, is restated the maintenance of critical edges with local information about the agent and its neighbors.

Remark 2: Note that Lemma 2 is only a sufficient but not necessary condition, that is, the interruption of some critical edge will not destroy the cluster connectivity. Therefore, in order to improve the coverage effect of the cluster, critical edges that do not affect connectivity are allowed to be disconnected. Since the agents can only receive real-time information from their neighbors, an edge selection strategy is designed for the three agents whose communication topology forms a triangular configuration with their edges being critical edges, i.e., the edge $\tilde{E}_i^a = \{\tilde{e}_{ij} : \exists k \in \tilde{\mathcal{N}}_i \cap \tilde{\mathcal{N}}_j, \|\boldsymbol{\xi}_{ij}\| < \|\boldsymbol{\xi}_{jk}\|, \|\boldsymbol{\xi}_{ij}\| < \|\boldsymbol{\xi}_{ki}\|\}$, which has the closest distance from critical edge set $\tilde{E}_i^c$, is eliminated.

Using the concept of critical robots and edge selection strategies, we design the CBFs at time t_k for communication maintenance as shown below:

$$h_{ij}^c(\boldsymbol{\xi}(t_k)) = R_c^2 - \|\boldsymbol{\xi}_{ij}(t_k)\|_2^2, \ \tilde{e}_{ij} \in \tilde{E}_i^c - \tilde{E}_i^a, \tag{14}$$

where $\boldsymbol{\xi}(t_k) = [\boldsymbol{\xi}_1(t_k)^T, \ldots, \boldsymbol{\xi}_N(t_k)^T]^T$ and $h_{ij}^c = h_{ji}^c$.

Similarly, the CBF is designed to avoid collisions between UAVs i and $j \in \widetilde{\mathcal{N}}_i(t_k)$ based on condition (10):

$$h_{ij}^{a}(\boldsymbol{\xi}(t_k)) = \|\boldsymbol{\xi}_{ij}(t_k)\|_2^2 - 4R_m^2, \tag{15}$$

where $h_{ij}^a = h_{ji}^a$. This paper consider static obstacles, so they can be treated as UAVs with velocity 0, i.e., the CBF for obstacle avoidance between i-th UAV and l-th obstacle is defined as:

$$h_{il}^{o}(\boldsymbol{\xi}(t_k)) = \|\boldsymbol{\xi}_{il}^{o}(t_k)\|_2^2 - R_o^2, \tag{16}$$

where $\boldsymbol{\xi}_{il}^{o}(t_k) = \boldsymbol{\xi}_i(t_k) - \boldsymbol{p}_l^{ob}$.

Denote the output of the Actor network of UAV i by $\tilde{\boldsymbol{u}}_i(t_k)$ at time t_k. The goal of constructing CBFs is to minimize the modifications of the output of Actor network to satisfy collision avoidance and connectivity maintenance constraints. Therefore, the following optimization problem, which is constructed based on designed CBFs, is the shield of agent i at time t_k:

$$\boldsymbol{u}_i(t_k) = \arg\min_{\boldsymbol{u}_i(t_k)} \frac{1}{2}\|\boldsymbol{u}_i(t_k) - \tilde{\boldsymbol{u}}_i(t_k)\|_2 \tag{17}$$

$$s.t.\ h_{ij}^{c}(\boldsymbol{\xi}(t_{k+1})) \geq (1-\eta)h_{ij}^{c}(\boldsymbol{\xi}(t_k)), \tilde{e}_{ij} \in \tilde{E}_i^c - \tilde{E}_i^a \tag{18}$$

$$h_{ij}^{a}(\boldsymbol{\xi}(t_{k+1})) \geq (1-\eta)h_{ij}^{a}(\boldsymbol{\xi}(t_k)),\ \ j \in \widetilde{\mathcal{N}}_i(t_k), \tag{19}$$

$$h_{il}^{o}(\boldsymbol{\xi}(t_{k+1})) \geq (1-\eta)h_{il}^{o}(\boldsymbol{\xi}(t_k)),\ \ l \in \check{\mathcal{N}}_i(t_k), \tag{20}$$

where $\check{\mathcal{N}}_i(t_k) = \{l \mid \|\boldsymbol{\xi}_i(t_k) - \boldsymbol{p}_l^{ob}\|_2 \leq R_s\}$.

The following theorem demonstrates the effectiveness of designed shield.

Theorem 1: Assume the initial actually positions of the UAVs satisfy the connectivity maintenance constraint (9), collision avoidance constraints (10), (11), and $v_{max} \leq \frac{R_c - R_\epsilon}{6\Delta t}$, the safety constraints of position with velocity compensation are satisfied at time t_k by controller (17) with the output $\boldsymbol{u}(t_k)$.

Proof: By the Lemma 1, if the discrete-time controller $\boldsymbol{u}_i(t_k)$ satisfying (18), (19) and (20), then the set $\mathcal{C}_s^c \bigcup \mathcal{C}_s^a \bigcup \mathcal{C}_s^o$ is the forward invariant, where

$$\begin{aligned} \mathcal{C}_s^c &= \{\boldsymbol{\xi}(t_k) | h_{ij}^{c}(\boldsymbol{\xi}(t_k)) \geq 0, \tilde{e}_{ij} \in \tilde{E}_i^c - \tilde{E}_i^a\}, \\ \mathcal{C}_s^a &= \{\boldsymbol{\xi}(t_k) | h_{ij}^{a}(\boldsymbol{\xi}(t_k)) \geq 0, j \in \widetilde{\mathcal{N}}_i(t_k)\}, \\ \mathcal{C}_s^o &= \{\boldsymbol{\xi}(t_k) | h_{ij}^{o}(\boldsymbol{\xi}(t_k)) \geq 0, l \in \check{\mathcal{N}}_i(t_k)\}, \end{aligned}$$

which means that the output $\boldsymbol{u}_i(t_k)$ of controller (17) is able to constrain the agent within its safe set. Therefore, the effectiveness of designed CBFs depends on whether the optimization problem (17) has a solution. Since we are concerned with the existence of the solution, let $\eta = 1$. For the condition (18), we need to get the $\boldsymbol{u}_i(t_k)$ to satisfy

$$\begin{aligned} &h_{ij}^{c}(\boldsymbol{\xi}(t_{k+1})) \\ &= R_c^2 - \|\boldsymbol{x}_{ij}(t_k) + 2\Delta t \boldsymbol{v}_{ij}(t_k) + \Delta t^2 \boldsymbol{u}_{ij}(t_k)\|_2^2 \\ &= R_c^2 - \|\boldsymbol{\xi}_{ij}(t_k) + \Delta t \boldsymbol{v}_{ij}(t_k) + \Delta t^2 \boldsymbol{u}_{ij}(t_k)\|_2^2 \geq 0, \end{aligned} \tag{21}$$

where $\boldsymbol{x}_{ij}(t_k) = \boldsymbol{x}_i(t_k) - \boldsymbol{x}_j(t_k)$, $\boldsymbol{v}_{ij}(t_k) = \boldsymbol{v}_i(t_k) - \boldsymbol{v}_j(t_k)$ and $\boldsymbol{u}_{ij}(t_k) = \boldsymbol{u}_i(t_k) - \boldsymbol{u}_j(t_k)$.

Consider the solution $\boldsymbol{u}_i(t_k) = -\frac{\boldsymbol{v}_i(t_k)}{\Delta t}$, then formula (21) satisfies

$$h_{ij}^{c}(\boldsymbol{\xi}(t_{k+1})) = R_c^2 - \|\boldsymbol{\xi}_{ij}(t_k)\|_2^2, \tag{22}$$

which means that $h_{ij}^{c}(\boldsymbol{\xi}(t_{k+1})) \geq 0$, if $\|\boldsymbol{\xi}_{ij}(t_k)\|_2^2 \leq R_c^2$. In others words, if $\|\boldsymbol{\xi}_{ij}(t_0)\|_2^2 \leq R_c^2$, then $\boldsymbol{u}_i(t_k) = -\frac{\boldsymbol{v}_i(t_k)}{\Delta t}$ makes inequality (21) hold. Based on the equation (13) and $\|\boldsymbol{x}_{ij}(t_0)\|_2^2 \leq R_c^2$, we can get $\|\boldsymbol{\xi}_{ij}(t_0)\|_2^2 \leq R_c^2$.

For the condition (19), we also consider $\boldsymbol{u}_i(t_k) = -\frac{\boldsymbol{v}_i(t_k)}{\Delta t}$, then

$$h_{ij}^{a}(\boldsymbol{\xi}(t_{k+1})) = \|\boldsymbol{\xi}_{ij}(t_k)\|_2^2 - 4R_m^2. \tag{23}$$

Due to the actually positions of the UAVs at time t_k satisfies $\|\boldsymbol{x}_{ij}(t_k)\|_2^2 \geq 4R_m^2$, $\boldsymbol{u}_i(t_k) = -\frac{\boldsymbol{v}_i(t_k)}{\Delta t}$ is a feasible solution such that $h_{ij}^{a}(\boldsymbol{\xi}(t_{k+1})) \geq 0$.

Obstacle can be regarded as stationary agent, so $\boldsymbol{u}_i(t_k) = -\frac{\boldsymbol{v}_i(t_k)}{\Delta t}$ is a feasible solution for condition (20).

To sum up, our shield is effective and ensure the safety of the agent with position $\boldsymbol{\xi}(t_k)$. This completes the proof. ■

Remark 3: The objective function (17) shows that the output $\tilde{\boldsymbol{u}}_i(t_k)$ of the Actor network is modified as little as possible to preserve the coverage performance. Condition (18) is the discrete-time CBF with connectivity maintenance. Condition (19) and condition (20) are CBFs for UAVs to collision avoidance with neighbors and obstacles, respectively. In the proposed safe MARL framework with shield, the output $\tilde{\boldsymbol{u}}_i(t_k)$ at time t_k from the Actor network does not directly act on the agent, but is corrected into a safe action $\boldsymbol{u}_i(t_k)$ through the CBF-based shield and then adopted by the UAV.

Remark 4: It is noteworthy that although the constraints (18), (19) and (20) involved in the optimization problem (17) are all hard constraints, the shield controller guarantees the existence of at least one solution $\boldsymbol{u}_i(t_k) = -\frac{\boldsymbol{v}_i(t_k)}{\Delta t}$, $i \in \{1, 2, \ldots, N\}$, which ensures the feasibility of the controller. The aforementioned feasible solution essentially serves as an emergency stop strategy for all agents, where agents reduce their velocities to 0 within one time step to avoid dangerous behaviors.

D. Reward Functions

Although CBF-based shields can ensure that all UAVs are always in the safe set to avoid the trap of sparse rewards, the corrective mechanism is also needed to help UAVs aware of safety boundaries. Therefore, we design the reward functions to help UAVs learn the corrective mechanism. The reward r_i of i-th UAV including coverage reward R_i^{cov}, connectivity reward R_i^{con} and collision avoidance reward R_i^{col}:

$$r_i(\boldsymbol{s}_i(t_k), \boldsymbol{a}_i(t_k)) = R_i^{\text{cov}} + R_i^{\text{con}} + R_i^{\text{col}}. \tag{24}$$

Since the coverage task has the long-term reward, i.e., the system can obtain the final reward $R^a > 0$ only when $e(t_k) = 0$, which is difficult at the early stage of training. Therefore, a staged reward $R^s > 0$ is defined, and it is obtained when the coverage of a PoI is completed. Let $|\mathcal{M}(t_k)|$ denotes the number of PoIs that have already been covered at time t_k, i.e., $\mathcal{M}(t_k) = \{j | \Gamma_N(\boldsymbol{p}_j, t_k) \geq C^*\}$. Due to the dynamic coverage task rewards are sparse, the distance between the UAVs and the uncovered PoIs is also treated as a penalty to speed up the task completion. At this point, we can obtain the coverage reward as:

$$R_i^{\text{cov}} = R_g^a + R^s(|\mathcal{M}(t_k)| - |\mathcal{M}(t_{k-1})|) - R^d \sum_{j \in M - \mathcal{M}(t_k)} \min_{i=1:N} \{\|\boldsymbol{x}_i(t_k) - \boldsymbol{p}_j\|_2\}, \tag{25}$$

where $R^d > 0$ is the penalty factor and

$$R_g^a = \begin{cases} R^a, & |\mathcal{M}(t_k)| = M, \\ 0, & |\mathcal{M}(t_k)| < M. \end{cases}$$

R_i^{con} is the connectivity interruption penalize of the UAVs' network:

$$R_i^{\text{con}} = \begin{cases} 0, & \lambda_2 \geq \epsilon + \eta, \\ -R^c, & 0 \leq \lambda_2 < \epsilon + \eta, \end{cases}$$

where η is the buffer value and $R^c > 0$ is the penalty for connectivity less than the threshold $\epsilon + \eta$. Due to the UAV under shield never goes cross the safety boundaries, its necessary to set the η to correct UAV's policy.

R_i^{col} is the penalty for the occurrence collision, which contains two parts: collision with other UAVs $j \in \widetilde{\mathcal{N}}_i(t_k)$ and collision with the obstacle $l \in \check{\mathcal{N}}_i(t_k)$:

$$R_i^{\text{col}} = R_i^{\text{uav}} + R_i^{\text{obs}},$$

where

$$R_i^{\text{uav}} = \begin{cases} 0, & \|\boldsymbol{\xi}_{ij}(t_k)\|_2 \geq 2R_m + \psi, \\ -R^o, & 2R_m \leq \|\boldsymbol{\xi}_{ij}(t_k)\|_2 < 2R_m + \psi, \end{cases}$$

$$R_i^{\text{obs}} = \begin{cases} 0, & \|\boldsymbol{\xi}_{il}^o(t_k)\|_2 \geq R_o + \psi, \\ -R^o, & R_o \leq \|\boldsymbol{\xi}_{il}^o(t_k)\|_2 < R_o + \psi, \end{cases}$$

with ψ is the buffer value and $R^o > 0$ is the penalty for collision.

E. Dynamic Coverage Algorithms Based on MADDPG With CBF-Based Shield

As the network framework shown in Fig. 4, the MADDPG assigns to each UAV an Actor network and a Critic network. The Actor network is responsible for learning the control strategy $\boldsymbol{\pi}$, where the network input is local observations $\boldsymbol{o}$ and the output is the action. The Critic network is responsible for evaluating $\boldsymbol{\pi}$, where the network input is the states $\boldsymbol{s}$ and action $\boldsymbol{a}$ of UAVs, and the output is the action-value function $Q_{\boldsymbol{\pi}}(\boldsymbol{s}, \boldsymbol{a})$, which represents the expected cumulative reward that can be obtained by taking optimal action in the current state and is parameterized by $\boldsymbol{\psi}$.

In order to improve the training stability, the target network is used, i.e., each UAV corresponds to four networks: the Actor network, Target Actor network, Critic network and Target Critic network, where Actor network and Critic network are the object of network parameters optimization. During the training process, at each N_s-step, the target network synchronizes its parameters with the corresponding network, which is to effectively avoid the moving target phenomenon during training due to a network that performs both prediction and supervision. Using $\boldsymbol{\theta}'$ and $\boldsymbol{\psi}'$ to denote the network parameters of the Target Actor network and Target Critic network, respectively. The network parameters $\boldsymbol{\theta}'$ is updated by $\boldsymbol{\theta}' = \tau\boldsymbol{\theta}' + (1 - \tau)\boldsymbol{\theta}$, where $0 < \tau < 1$. Accordingly, $\boldsymbol{\psi}'$ is updated by $\boldsymbol{\psi}' = \tau\boldsymbol{\psi}' + (1 - \tau)\boldsymbol{\psi}$.

In addition, the Experience Replay is also used to solve the problem that the samples in the learning are not independent and identically distributed, i.e., the replay buffer is used to store the state transitions generated by the UAVs interacting with the environment. Specifically, batches of state transitions from different episodes are randomly read during training, and the correlation between these transitions is small.

Due to the dynamic coverage task is to plan the trajectory of the UAVs, the Actor and Critic networks use the Multi-Layer Perceptron (MLP) structure. The networks are trained by the Back Propagation (BP) algorithm. Here, assume that the four networks of each UAV have the same hidden layer structure, which are composed of three full connection layers, and the hidden layer contains 512 neurons.

Assume the bath size is $|\mathcal{B}|$, the loss function of Actor network is

$$L(\boldsymbol{\theta}_i) = -\frac{1}{|\mathcal{B}|}\sum_{j=1}^{|\mathcal{B}|} Q_i(\boldsymbol{o}_i, \boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_N, \boldsymbol{\psi}_i), \tag{26}$$

and the loss function of Critic network is

$$L(\boldsymbol{\psi}_i) = \frac{1}{|\mathcal{B}|}\sum_{j=1}^{|\mathcal{B}|} (Q_i(\boldsymbol{o}_i, \boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_N, \boldsymbol{\psi}_i) - y_j)^2, \tag{27}$$

where $y_j = r_i + \gamma Q_i'(\boldsymbol{o}_i, \boldsymbol{a}_1', \boldsymbol{a}_2', \ldots, \boldsymbol{a}_N', \boldsymbol{\psi}_i')$, $\boldsymbol{a}_i'$ and Q_i' are the outputs of the Target Actor network and Target Critic network, respectively.

During the training process of our algorithm, i-th UAV obtains its original action $\boldsymbol{a}_i^c$ from the Actor network based on local observation $\boldsymbol{o}_i$, and the action $\boldsymbol{a}_i^c$ may be dangerous. Therefore, $\boldsymbol{a}_i^c$ is placed into the CBF-based shield with new position $\boldsymbol{\xi}_i$ to obtain a corrected safe action $\boldsymbol{a}_i^s$. Then, the reward r_i and the UAV's state $\boldsymbol{s}_i$ are stored into replay buffer. Meanwhile, the Actor network and Critic network are updated using (26) and (27), respectively. The Target networks are periodically updated according to τ. The overall process of our method, which is called MARLSP, can be expressed as Algorithm 1.

V. Experiment

A. Experimental Configuration

This section provides several numerical simulations to demonstrate the effectiveness of the proposed dynamic coverage method. The task is performed by 4 multi-rotor UAVs with identical configurations, where the physical radius of the UAVs is $R_m = 0.2m$, the sensor sensing radius is $R_s = 2.5m$, and the maximum communication range is $R_c = 6m$. Considering the safety constraints, it is assumed that all UAVs maintain the safe distance $R_o = 0.5m$ from the obstacle and the desired connectivity of the cluster is set to $\epsilon = 0.1$. The UAV cluster dynamically coverages 20 PoIs, whose positions are randomly generated in the $20m * 20m$ square area. The peak sensing capability is $S_c = 2$ and the desired attained effective coverage is $C^* = 4$. The parameters of the reward are set to: $R^a = 1500$, $R^s = 75$, $R^d = 1$, $R^c = 50$ and $R^o = 30$.

Algorithm 1 Dynamic Coverage Control Algorithm MARLSP

```
Input: episode iterations $N_e$, period $T$, discount factor $\gamma$,
       synchronization parameter $\tau$.
Output: policy $\boldsymbol{\pi}$.
for $i$=1:N do
    Initialize replay buffer $\mathcal{D}_i^b$, initialize networks
    parameters $\boldsymbol{\theta}_i$, $\boldsymbol{\theta}_i'$, $\boldsymbol{\psi}_i$, $\boldsymbol{\psi}_i'$.
for episode=1:$N_e$ do
    Initialize environment, obstacle $\boldsymbol{p}_l^{ob}$, $l = \{1, \dots, L\}$,
    get the PoIs distribution $\boldsymbol{p}_j$, reset coverage process
    $\Gamma_N(\boldsymbol{p}_j, t_k)$, $j = \{1, \dots, M\}$, initialize UAV state $\boldsymbol{s}_i$,
    update observations $\boldsymbol{o}_i$, $i = \{1, \dots, N\}$, initialize
    random process $\mathcal{N}^k$.
    for $k = 1 : T$ for each UAV $i$ do
        Update $\boldsymbol{o}_i(t_k)$ based on communication topology.
        Select action $\boldsymbol{a}_i^c(t_k) = \boldsymbol{a}_i(\boldsymbol{o}_i(t_k), \boldsymbol{\theta}_i) + \mathcal{N}^k$.
        Pre-executed action $\boldsymbol{a}_i^c(t_k)$.
        Store the tuple $\{\boldsymbol{o}_i(t_k), \boldsymbol{a}_i^c(t_k), r_i, \boldsymbol{o}_i(t_{k+1})\}$ to $\mathcal{D}_i^b$.
        Obtain the filter position $\boldsymbol{\xi}_i(t_k)$.
        Construct the CBF-based shield (17), get the safe
        action $\boldsymbol{a}_i^s(t_k)$.
        Implement $\boldsymbol{a}_i^s(t_k)$ and receive the observation
        $\boldsymbol{o}_i(t_{k+1})$ and reward $r_i$.
        Store the tuple $\{\boldsymbol{o}_i(t_k), \boldsymbol{a}_i^s(t_k), r_i, \boldsymbol{o}_i(t_{k+1})\}$ to $\mathcal{D}_i^b$.
        Take samples from $\mathcal{D}_i^b$ to update $\boldsymbol{\theta}_i$ according
        to (26), update $\boldsymbol{\psi}_i$ according to (27).
        Update the target networks with
        $\boldsymbol{\theta}_i' = \tau\boldsymbol{\theta}_i' + (1-\tau)\boldsymbol{\theta}_i$ and $\boldsymbol{\psi}_i' = \tau\boldsymbol{\psi}_i' + (1-\tau)\boldsymbol{\psi}_i$.
```

In the training phase, the maximum number of training episode is $N_e = 1.5 \times 10^5$, and the maximum number of steps in a episode is $T = 80$. The batch size is 1024 and the target network is updated every 100 steps with the update rate $\tau = 0.75$. Besides, the discount factor is $\gamma = 0.95$ and the replay buffer contains 1024∗60 state transitions. The MARLSP algorithm was run with the TensorFlow 1.14 deep learning framework.

B. Performance Evaluation

The advantages of the proposed MARLSP algorithm are emphasized by comparing MADDPG, Independent Q-Learning (IQL) and gradient-based algorithm in the same environment, where MADDPG and IQL algorithms use only reward functions to prompt the agent to learn safe strategies. Since the gradient-based algorithm in reference [8] not only exhibits the characteristics of model-free algorithms, which require precise environmental models and dynamics models of agents to formulate control laws, but also serves as the foundation for numerous model-based methods, it is able to represent the performance of model-based algorithms. Specifically, the gradient-based control law without connectivity maintenance mentioned in reference [8] is expressed as follows:

$$\boldsymbol{u}_i(t) = -k_i^{\mathrm{cov}}\boldsymbol{u}_i^{\mathrm{cov}}(t) - k_i^{\mathrm{col}}\boldsymbol{u}_i^{\mathrm{col}}(t), \tag{28}$$

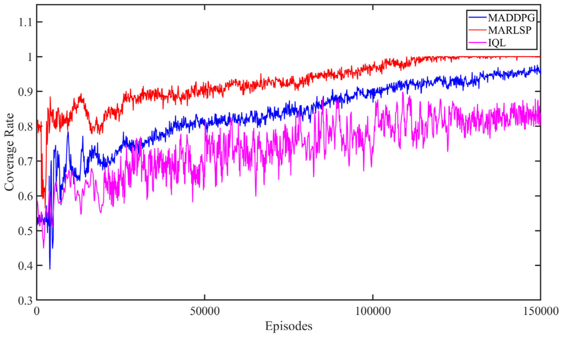


Fig. 5. Coverage rate of multi-rotor UAVs during training.

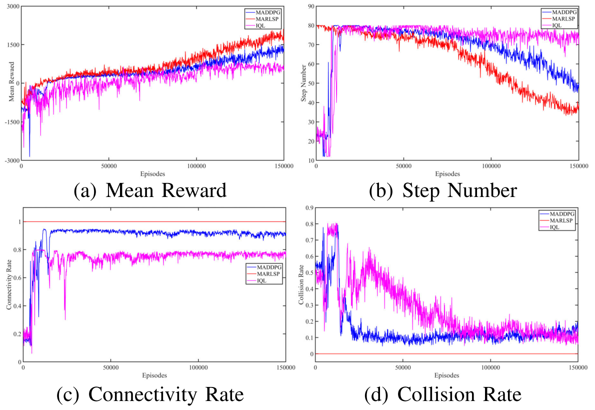
where $\boldsymbol{u}_i^{\mathrm{cov}}(t) = \frac{\partial e(t)}{\partial t}$ and $\boldsymbol{u}_i^{\mathrm{col}}(t) = \sum_{j=1, j\neq i}^{N} \frac{\partial P_{ij}}{\partial \boldsymbol{x}_i}$ with

$$P_{ij}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(\min\left\{0, \frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 - R_s^2}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 - (2 * R_m)^2}\right\}\right)^2.$$

The control law in equation (28) is extremely sensitive to the feedback gains $k_i^{\mathrm{cov}} > 0$ and $k_i^{\mathrm{col}} > 0$. For different PoIs settings, obstacle configurations and UAV fleet sizes, we have to choose the appropriate feedback gains to achieve dynamic coverage. In our simulation scenario, let $k_i^{\mathrm{cov}} = 1$, $k_i^{\mathrm{col}} = 0.1$, and the settings of R_s and R_m remain unchanged. The MADDPG without shield is chosen as the baseline algorithm for comparison in this work.

Coverage rate is the core metric, i.e., the ratio of the number of PoIs that are effectively coverage to the total PoIs, which is demonstrated in Fig. 5, where the red, blue and purple curves indicate the changes in coverage rate during training for the MARLSP and MADDPG and IQL algorithm, respectively. It can be seen that MARLSP algorithm's coverage rate improves much faster. This is because the dynamic coverage task reward is sparse, and most of the exploration experience in the early training stage does not provide effective guidance for the UAVs. Without the shield, MADDPG and IQL algorithms produce unsafe actions that cause UAVs to frequently cross safety region, which results in repeated penalties that limit learning of effective coverage strategies. Due to coverage and connectivity are conflicting in behavior, the baseline algorithm that only motivates UAVs to learn strategies through rewards and punishments cannot achieve coverage of all PoIs. Meanwhile, our proposed MARLSP algorithm adds the CBF-based shield to correct the UAVs' actions, which increases the number of valid samples. Simultaneously, CBF-based shield minimizes modifications to the original exploration actions of the UAVs, which also helps the UAVs to learn effective coverage strategies with safety constraints.

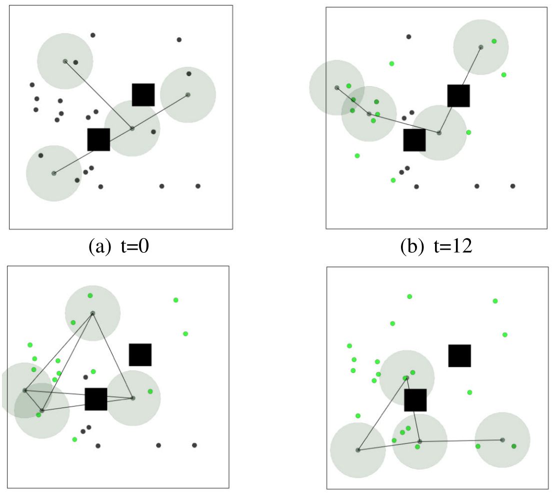
Fig. 6(a) depicts the rewards convergence process of the three algorithms. At the beginning of training, the UAVs cannot get the total reward for completing the task, and they learn more about the safety strategy, as can also be seen from the reward curve, where the reward is negative. In the middle of training, the UAVs gradually learn the coverage strategy. However, the MADDPG and IQL algorithms learn the strategy according to the rewards and penalties, they are unable to harmonize the safety constraints and tasks, which results in lower rewards than MARLSP later in training. There is a slight



(a) Mean Reward (b) Step Number

(c) Connectivity Rate (d) Collision Rate

Fig. 6. The comparison of training curves of MARLSP and MADDPG and IQL algorithms.



(a) t=0 (b) t=12

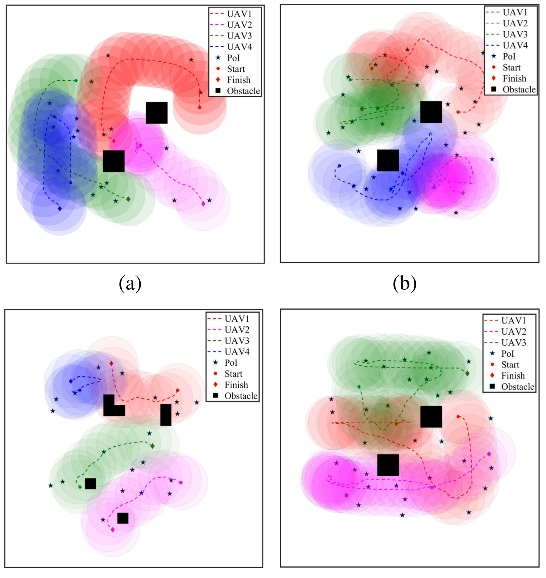
(c) t=27 (d) t=43

Fig. 7. The trajectory evolution of all UAVs based on MARLSP algorithm.

fluctuation in the reward due to the presence of variance in the action sampling, which only affects the trajectory of the UAVs, and has less effect on the coverage rate and the number of steps. The variation of the step number of UAVs is specifically provided by Fig. 6(b).

To verify the safety of the proposed algorithm, Fig. 6(c),(d) show the connectivity rate and collision rate, which are the ratio of the number of safe steps to the total step numbers during training, respectively. The MARLSP algorithm can maintain the communication of UAV cluster and the UAVs do not collide with neighbors or obstacles at any stage of learning, which verifies Theorem 1. The MADDPG and IQL algorithms always suffer from the risk of cluster connectivity disruptions and collisions, and the UAVs are unable to learn a strategy that balances coverage and connectivity. The MADDPG algorithm is better than the IQL algorithm of dynamic coverage task, as it uses global information during training to learn strategies. In IQL algorithm, other agents are treated as part of the environment, which results in a non-stationary environment with the worst coverage performance.

Fig. 7 shows the trajectory generated by the learning strategy of MARLSP algorithm, where the circles indicate



(a) (b)

(c) (d)

Fig. 8. The trajectories of MARLSP algorithm in different dynamic coverage scenarios.

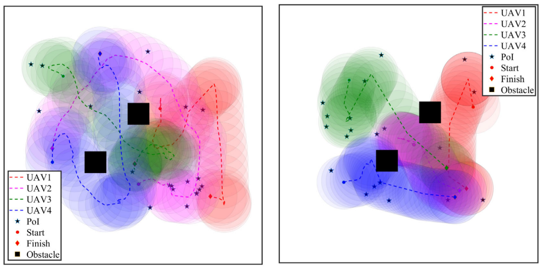
the coverage range of UAVs, the connecting lines indicate that the UAVs can communicate, the black and green dots indicate PoIs which are not covered and have reached effective coverage, respectively. These trajectories achieve the dynamic coverage task while satisfying safety constraints, and the overall trajectories of the UAVs are shown in Fig. 8(a).

To more rigorously test the proposed algorithm's performance, the number of PoIs, obstacle configurations and UAV fleet sizes are varied in dynamic coverage scenarios. Fig. 8 shows the trajectories generated by MARLSP algorithm for four different dynamic coverage scenarios, where scenario (b) has 32 PoIs, scenario (c) has multiple obstacles and scenario (d) has only 3 UAVs. The algorithm was tested 100 times for randomly positioned PoIs in each scenario, with results summarized in Table I.

As shown in Table I, MARLSP's dynamic coverage rate is not always 100%. The reason is that the cluster can sacrifice part of the coverage effectiveness to maintain communication connections in some special cases, such as a few PoIs are located alone at the boundary of the task area. In these cases, increasing the maximum training steps can help agents achieve full coverage. Fig. 9(a) provides a 62-step trajectory of agents, which is higher than the average number of steps 37. While always achieving full coverage, the gradient-based algorithm ignores connectivity maintenance constraint. Besides, the global information of the environment is necessary for agents to compute gradient-based control laws and the feedback gains have to be re-selected once the environment changes. As shown in Fig. 9(b), there is no cooperative relationship between the agents, and the goal of each agent is to move towards the uncovered PoI. All agents eventually move toward the PoIs located on the lower right side of the task area, which results in more time spent of cluster to complete task.

TABLE I
DYNAMIC COVERAGE PERFORMANCE COMPARISON AMONG FOUR ALGORITHMS

Algorithms	Scenario (a)				Scenario (b)				Scenario (c)				Scenario (d)			
	Coverage	Step	Connectivity	Collision	Coverage	Step	Connectivity	Collision	Coverage	Step	Connectivity	Collision	Coverage	Step	Connectivity	Collision
MARLSP	**0.98**	**37**	**1.0**	**0.0**	**0.98**	**61**	**1.0**	**0.0**	**0.98**	**40**	**1.0**	**0.0**	**0.94**	**73**	**1.0**	**0.0**
MADDPG	0.91	48	0.91	0.13	0.89	77	0.85	0.15	0.91	52	0.87	0.21	0.82	79	0.64	0.11
IQL	0.76	78	0.78	0.12	0.74	79	0.63	0.17	0.76	78	0.75	0.21	0.59	80	0.52	0.10
Gradient	1.0	62	0.0	0.06	1.0	96	0.0	0.07	1.0	67	0.0	0.11	1.0	121	0.0	0.03



(a) MARLSP algorithm. (b) Gradient-based algorithm.

Fig. 9. The trajectories with scattered PoIs distribution of MARLSP algorithm and Gradient-based algorithm.

C. Complexity Analysis

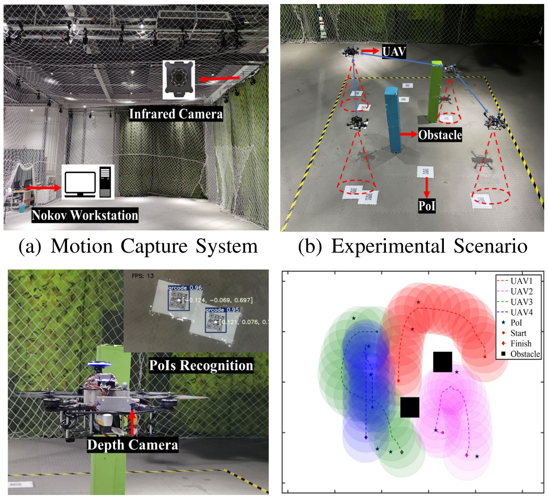
The complexity of the algorithm is further analyzed by calculating the number of floating-point operations (FLOPs) and parameters. The MARLSP algorithmic utilizes a three-layer fully connected network, where the dimension of the hidden layer is Λ. According to the local observation information $\boldsymbol{o}_i(t_k)$ of each UAV, the input layer contains PoIs, obstacles, the position and velocity of itself and its neighbors, i.e., the dimension of the input layer is $4N + 3M + 2K$. The output layer is the desired velocity of the UAV with a dimension of 2.

With the help of conclusions from literature [39] on computing FLOPs in fully connected layers:

$$\text{FLOPs} = (2I - 1)O,$$

where I is the input dimensionality and O is the output dimensionality, the FLOPs of proposed MARLSP algorithm is $\Lambda(8N+6M+4K+2)-2$. When the number of neurons in the hidden layer is fixed, the FLOPs of MARLSP algorithmic increases linearly with the number of UAVs, POIs, and obstacles, i.e., the complexity of the algorithm is $o(N + M + K)$.

Further, the number of parameters with MARLSP algorithm is $\Lambda(4N + 3M + 2K + 3) + 2$ and it also increases linearly with the size of SAR task. During the actual training process, 8 hours were spent to get the required model. According to our calculations, the time of the agent to complete one step training is $0.006s$, of which $0.002s$ is used to solve the QP problem. Correspondingly, we calculate the FLOPs of the gradient-based algorithm is $o(MN+N+K)$. Compare with the traditional gradient-based algorithm without connectivity maintenance constraint, although reinforcement learning takes some time in the training phase, it outperforms the gradient-based algorithm in the execution phase only in terms of complexity.



(a) Motion Capture System (b) Experimental Scenario

(c) Effective Coverage (d) UAVs' Trajectories

Fig. 10. Physical experimentation with 4 multi-rotor UAVs and 10 PoIs.

D. Physical Experimentation

A physical experimental platform was constructed to validate the algorithm's feasibility with multi-rotor UAVs. It is consisted of three key components: a NOKOV motion capture system, flight workstation, and custom multi-rotor UAVs. The NOKOV, which is shown by Fig. 10(a), serves as optical 3D motion capture system consisting of 44 infrared cameras and a workstation, and it can provide high-precision position data for the UAVs. The UAVs use Pixhawk4 as the flight control system and are equipped with D435i depth camera as the sensor, in which the camera can provide depth images and RGB images in real time, and the maximum depth range is $10m$.

Fig. 10(b) indicates the experimental scenario: four UAVs equipped with depth cameras and training models are covering 10 PoIs, which are represented by 2-dimensional bar code (QR code), in the $4m * 4m$ area. The UAVs can get accurate information of positions, velocities and obstacles by the motion capture system. The process of UAV to coverage PoI is simulated by the depth camera to recognize the QR code in Fig. 10(c). Each UAV is equipped with a D435i depth camera underneath, whose field of view is related to the UAV's flight height, and the mission height is set to be $1m$ here. The recognition of QR code adopts the target detection algorithm based on YOLOv5, which not only can detect the moving targets in the complex environment, but also be fused with the subsequent attitude detection to obtain the center point of PoI as well as the 3D coordinate information. The distance between the UAV and the PoI is further determined. According to the trajectories in Fig. 10(d), it can be seen that the UAVs carrying the model learned by the MARLSP algorithm can realize the dynamic coverage task with safety constraints.

VI. Conclusion

This paper addressed the SAR mission for multi-rotor UAVs by modeling it as a constrained dynamic coverage control problem with safety as the priority for safety-critical systems. To solve the constrained dynamic coverage problem with unknown environments, we propose a safe MARL framework with a CBF-based shield, which is characterized by the ability to modify the actions generated by the UAVs in MADDPG algorithm to satisfy the constraints. In addition, the CBF-based shield was hard-coded to enforce safety constraints and was theoretically proven effective. Finally, the effectiveness of the method is verified by simulations and physical experiment.

References

[1] N. Geng, Q. Meng, D. Gong, and P. W. H. Chung, "How good are distributed allocation algorithms for solving urban search and rescue problems? A comparative study with centralized algorithms," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 1, pp. 478–485, Jan. 2019.

[2] L. Liu, Z. Wang, and H. Zhang, "Adaptive fault-tolerant tracking control for MIMO discrete-time systems via reinforcement learning algorithm with less learning parameters," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 299–313, Jan. 2017.

[3] C. H. Liu et al., "Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2059–2070, Sep. 2018.

[4] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robot. Autom.*, vol. 20, no. 2, pp. 243–255, Apr. 2004.

[5] C. G. Cassandras and W. Li, "Sensor networks and cooperative control," in *Proc. 44th IEEE Conf. Decis. Control*, Dec. 2005, pp. 4237–4238.

[6] A. Howard, M. J. Matari, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *Distributed Autonomous Robotic Systems*, vol. 5. Tokyo, Japan: Springer, Jun. 2002, pp. 299–308.

[7] I. I. Hussein and D. M. Stipanovic, "Effective coverage control using dynamic sensor networks," in *Proc. 45th IEEE Conf. Decis. Control*, Dec. 2006, pp. 2747–2752.

[8] I. I. Hussein and D. M. Stipanovic, "Effective coverage control for mobile sensor networks with guaranteed collision avoidance," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 4, pp. 642–657, Jul. 2007.

[9] I. I. Hussein and D. M. Stipanovic, "Effective coverage control using dynamic sensor networks with flocking and guaranteed collision avoidance," in *Proc. Amer. Control Conf.*, Jul. 2007, pp. 3420–3425.

[10] G. M. Atınç, D. M. Stipanović, and P. G. Voulgaris, "Supervised coverage control of multi-agent systems," *Automatica*, vol. 50, no. 11, pp. 2936–2942, Nov. 2014.

[11] G. M. Atınç, D. M. Stipanović, and P. G. Voulgaris, "A swarm-based approach to dynamic coverage control of multi-agent systems," *Automatica*, vol. 112, Feb. 2020, Art. no. 108637.

[12] A. Karimoddini, H. Lin, B. M. Chen, and T. Heng Lee, "Hybrid formation control of the unmanned aerial vehicles," *Mechatronics*, vol. 21, no. 5, pp. 886–898, Aug. 2011.

[13] X. Dong, Y. Hua, Y. Zhou, Z. Ren, and Y. Zhong, "Theory and experiment on formation-containment control of multiple multirotor unmanned aerial vehicle systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 1, pp. 229–240, Jan. 2019.

[14] Y. Kuriki and T. Namerikawa, "Consensus-based cooperative formation control with collision avoidance for a multi-UAV system," in *Proc. Amer. Control Conf.*, Jun. 2014, pp. 2077–2082.

[15] M. Malmirchegini and Y. Mostofi, "On the spatial predictability of communication channels," *IEEE Trans. Wireless Commun.*, vol. 11, no. 3, pp. 964–978, Mar. 2012.

[16] Q. Liu, S. Zhou, and G. B. Giannakis, "Cross-layer combining of adaptive modulation and coding with truncated ARQ over wireless links," *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 1746–1755, Sep. 2004.

[17] M. M. Zavlanos, M. B. Egerstedt, and G. J. Pappas, "Graph-theoretic connectivity control of mobile robot networks," *Proc. IEEE*, vol. 99, no. 9, pp. 1525–1540, Sep. 2011.

[18] M. C. De Gennaro and A. Jadbabaie, "Decentralized control of connectivity for multi-agent systems," in *Proc. 45th IEEE Conf. Decis. Control*, Dec. 2006, pp. 3628–3633.

[19] J. Fu, G. Wen, X. Yu, and Z.-G. Wu, "Distributed formation navigation of constrained second-order multiagent systems with collision avoidance and connectivity maintenance," *IEEE Trans. Cybern.*, vol. 52, no. 4, pp. 2149–2162, Apr. 2022.

[20] D. V. Dimarogonas and K. H. Johansson, "Decentralized connectivity maintenance in mobile networks with bounded inputs," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 1507–1512.

[21] S. Gao and Z. Kan, "Effective dynamic coverage control for heterogeneous driftless control affine systems," *IEEE Control Syst. Lett.*, vol. 5, no. 6, pp. 2018–2023, Dec. 2021.

[22] S. Meng and Z. Kan, "Deep reinforcement learning-based effective coverage control with connectivity constraints," *IEEE Control Syst. Lett.*, vol. 6, pp. 283–288, 2022.

[23] X. He, Q. Wang, and Y. Hao, "Finite-time adaptive formation control for multi-agent systems with uncertainties under collision avoidance and connectivity maintenance," *Sci. China Technolog. Sci.*, vol. 63, no. 11, pp. 2305–2314, Nov. 2020.

[24] H. Abdi, G. Raja, and R. Ghabcheloo, "Safe control using vision-based control barrier function (V-CBF)," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 782–788.

[25] C. Yu, Y. Dong, Y. Li, and Y. Chen, "Distributed multi-agent deep reinforcement learning for cooperative multi-robot pursuit," *J. Eng.*, vol. 2020, no. 13, pp. 499–504, Jul. 2020.

[26] Z. Yan, A. R. Kreidieh, E. Vinitsky, A. M. Bayen, and C. Wu, "Unified automatic control of vehicular systems with reinforcement learning," *IEEE Trans. Autom. Sci. Eng.*, vol. 20, no. 2, pp. 789–804, Apr. 2023.

[27] O. Vinyals et al., "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, Nov. 2019.

[28] V. G. Goecks, G. M. Gremillion, V. J. Lawhern, J. Valasek, and N. R. Waytowich, "Integrating behavior cloning and reinforcement learning for improved performance in dense and sparse reward environments," in *Proc. 19th Int. Conf. Auto. Agents MultiAgent Syst.*, May 2020, pp. 465–473.

[29] J. Hu, H. Niu, J. Carrasco, B. Lennox, and F. Arvin, "Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 14413–14423, Oct. 2020.

[30] M. Theile, H. Bayerlein, R. Nai, D. Gesbert, and M. Caccamo, "UAV coverage path planning under varying power constraints using deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 1444–1449.

[31] M. Hassan and D. Liu, "PPCPP: A predator–prey-based approach to adaptive coverage path planning," *IEEE Trans. Robot.*, vol. 36, no. 1, pp. 284–301, Feb. 2020.

[32] Z. Sun, N. Wang, H. Lin, and X. Zhou, "Persistent coverage of UAVs based on deep reinforcement learning with wonderful life utility," *Neurocomputing*, vol. 521, pp. 137–145, Feb. 2023.

[33] O. Bastani, S. Li, and A. Xu, "Safe reinforcement learning via statistical model predictive shielding," in *Robotics: Science and Systems*, Jul. 2021, pp. 1–13.

[34] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, "Safe reinforcement learning via shielding," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2018, vol. 32, no. 1, pp. 2669–2678.

[35] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. 31st Conf. Neural Inf. Process. Syst.*, 2017, pp. 6382–6393.

[36] C. Godsil, and G.F. Royle, *Algebraic Graph Theory*. Cham, Switzerland: Springer, 2001.

[37] J. Zeng, Z. Li, and K. Sreenath, "Enhancing feasibility and safety of nonlinear model predictive control with discrete-time control barrier functions," in *Proc. 60th IEEE Conf. Decis. Control (CDC)*, Dec. 2021, pp. 6137–6144.

[38] P. D. Hung, T. Q. Vinh, and T. D. Ngo, "Hierarchical distributed control for global network integrity preservation in multirobot systems," *IEEE Trans. Cybern.*, vol. 50, no. 3, pp. 1278–1291, Mar. 2020.

[39] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, Nov. 2016.