Safety-Guaranteed Learning-Based Flocking Control Design

Mingzhe Liu and Yan Chen, *Member, IEEE*

***Abstract*—This letter aims to develop a new learning-based flocking control framework that ensures inter-agent free collision. To achieve this goal, a leader-following flocking control based on a deep Q-network (DQN) is designed to comply with the three Reynolds' flocking rules. However, due to the inherent conflict between the navigation attraction and inter-agent repulsion in the leader-following flocking scenario, there exists a potential risk of inter-agent collisions, particularly with limited training episodes. Failure to prevent such collision not only caused penalties in training but could lead to damage when the proposed control framework is executed on hardware. To address this issue, a control barrier function (CBF) is incorporated into the learning strategy to ensure collision-free flocking behavior. Moreover, the proposed learning framework with CBF enhances training efficiency and reduces the complexity of reward function design and tuning. Simulation results demonstrate the effectiveness and benefits of the proposed learning methodology and control framework.**

***Index Terms*—Multi-agent systems, flocking control, reinforcement learning, collision avoidance, control barrier function.**

I. Introduction

FLOCKING behavior, describing the group coordination and motion of animals in nature, has great potential for various engineering applications. The self-organizing feature and robust scalability make flocking control suitable for transportation systems and mobile sensor networks [1]. In 1987, Reynolds proposed three famous heuristic rules as the foundation of flocking: cohesion, separation, and alignment [2]. Inter-agent collision avoidance is a fundamental property that helps mobile robot systems avoid incidents and hardware damage. The flocking rules were extended to include obstacle avoidance and leader following later [3]. Moreover, to realize these flocking rules, different flocking control designs were proposed by using potential field methods [3], and other approaches [4], [5]. In addition to theoretical research, other studies have focused on applying flocking control theory to engineering practice, such as considering traffic rules and real-world scenarios that include road boundaries [6] and vehicle spacing [7].

However, model-based flocking control has a potential issue: the control performance heavily relies on accurate models of multi-agent systems, which may not be realistic [8], [9]. An alternative approach is the learning-based method, which does not require accurate knowledge of multi-agent systems and environments. Moreover, learning-based flocking can offer environmental adaptability [10], handle disturbances from a dynamic environment [11], and find near-optimal results [12].

Because of the advantages, various reinforcement learning algorithms were recently applied to achieve different flocking behaviors and applications. For example, reinforcement techniques with both teacher and non-teacher scenarios were used to achieve Vicsek flocking, which only considered velocity consensus [13]. A particle-based flocking algorithm was first developed using Q-learning to achieve Reynolds flocking [14], and a Q-learning flocking algorithm was applied for fixed-wing UAV dynamics [8]. As an extension of Q-learning, a deep Q-Network (DQN) was studied to solve the flocking problem with fixed-wing UAV kinematics [15]. The feasibility of collision-free flocking was achieved via discrete actions in these learning methods [8], [14], [15]. Imitation learning was used to train policies and learn the local controller by mimicking the centralized controller using global information [16]. In addition, graph neural networks (GNNs) were used to address the scaling problem. Deep deterministic policy gradient (DDPG)-based flocking was also recently studied and improved [17], and a brain emotional learning-based flocking control method was applied for UAVs with experimental validation [11].

Implementing machine learning on cyber-physical systems (CPS) in the real world is a great challenge [18]. A popular method is to learn policies in simulation and execute the resulting policies experimentally. However, this approach may not be capable of considering various uncertainties in hardware and environment. Training on CPS can overcome this issue by continuously learning when policies are executed on hardware. In the trial-by-error method of reinforcement learning, because agents learn from misbehavior by receiving a penalty and minimizing the overall penalty scores, collision-free is critical for learning the near-optimal policy during the learning on CPS process. On the other hand, a collision-free condition in a learning process is challenging due to the essential conflict between inter-agent repulsion and the (virtual) leader agent attraction. This conflict makes the training result highly dependent on the reward function design and parameter tuning.

Manuscript received 15 September 2023; revised 24 November 2023; accepted 11 December 2023. Date of publication 28 December 2023; date of current version 22 January 2024. This work was supported in part by the Office of Naval Research (ONR) under Grant N00014-21-1-2403. Recommended by Senior Editor P. Tesi. *(Corresponding author: Yan Chen.)*

The authors are with the Polytechnic School, Arizona State University, Mesa, AZ 85212 USA (e-mail: mingzhe.liu@asu.edu; yanchen@asu.edu).

Digital Object Identifier 10.1109/LCSYS.2023.3347809

In other words, collision-free is not guaranteed in the aforementioned learning methods. Since collisions usually generate bad effects, such as hardware damages or injuries, a guaranteed collision-free mechanism is critical for mobile agents to utilize machine learning methods in real-world applications.

Control barrier function (CBF) is a model-based control design method to guarantee safety constraints in dynamic systems [19], which was recently integrated with flocking control. For instance, a new flocking control scheme was proposed to consider the synchronization of agents' attitudes, in which CBF was used for separation [20]. Recently, CBF-based resilient flocking control was applied to collision and obstacle avoidance of ground robots with experimental validations [21]. Furthermore, CBF was integrated into flocking control to control a dynamic obstacle and prevent the flocking group from entering the protection zone, which demonstrated the capability of CBF for obstacle avoidance [22]. CBF was also integrated with learning-based control, though not for flocking. A CBF-based guiding control with reinforcement learning, including DDPG and TRPO, was proposed to achieve safe and efficient learning [18]. A matrix-variate Gaussian process was used to learn dynamic uncertainties, which were incorporated into robust multi-agent CBF to prevent collisions [23]. Inspired by the aforementioned works, the research gap is identified to achieve safety (collision-free) guaranteed constraints by using CBF in learning-based flocking for multi-agent systems.

Thus, we propose a safety-guaranteed framework for learning-based flocking to ensure collision-free operations of multi-agent systems. The main contributions of this letter are summarized as follows:

- A leader-following flocking scheme is realized through a deep Q-network (DQN), in which the phenomena of inter-agent collisions are unavoidable and analyzed.
- A safety-guaranteed learning-based framework is proposed to employ a pairwise CBF to achieve collision-free flocking behavior.
- The proposed framework can introduce additional benefits in easier or simpler reward design and tuning, thereby enhancing learning efficiency, which was demonstrated through online training and simulation results.

The remainder of this letter is organized as follows. Section II provides the background of DQN and formulates the flocking problem as the DQN. Section III introduces the CBF and shows how it is formulated and integrated into the learning strategy. Section IV presents the simulation results and discusses the effectiveness and benefits of the proposed method. Conclusions are drawn in Section V.

II. Learning-Based Flocking

A. Background: Deep Q-Network

The deep Q-network (DQN) [24] is a reinforcement learning method that extends Q-learning by employing a neural network (NN) to predict Q-values rather than relying on tabular reward-action updates. This extended feature provides several advantages. First, DQN can effectively handle continuous state inputs and large state spaces. Second, DQN exhibits strong generalization capabilities and performs well in complex situations where traditional learning methods struggle. The utilization of a neural network for approximating Q-values enhances the flexibility and efficiency of the learning process, making DQN a favorable tool for learning-based multi-agent systems, which usually require massive amounts of state inputs.

B. DQN-Based Flocking

A decentralized multi-agent learning scheme based on DQN is proposed to achieve learning-based flocking. Leader-following can be formulated as a reinforcement learning (RL) problem within a Markov decision process (MDP) framework. In the multi-agent system, a set of RL agents indexed as $N = \{1, 2, \ldots, n\}$ are trained to achieve virtual leader following and desired flocking behaviors. To represent the dynamics of the RL agents, a particle-based double integrator model, which is commonly used in the flocking control and multi-agent systems literature, such as [2], [3], [4] and [25], was adopted.

$$\begin{cases} \dot{\mathbf{q}}_i = \mathbf{p}_i \\ \dot{\mathbf{p}}_i = \mathbf{u}_i \end{cases}, \tag{1}$$

where each RL agent $i \in N$ is characterized by its position $\mathbf{q}_i \in \mathbb{R}^2$, velocities $\mathbf{p}_i \in \mathbb{R}^2$, and control input $\mathbf{u}_i \in \mathbb{R}^2$.

The generated trajectories or references based on the point-mass model (1), although simplified, could be applied to tracking control of CPS through a hierarchical control architecture, in which nonlinear model dynamics and model uncertainties of connected and automated vehicles were handled in the low-level control layer [1].

The virtual leader follows a predefined trajectory, while the RL agents aim to learn how to track the leader with the Reynolds' flocking rules of maintaining separation, alignment, and cohesion within the flock.

1) Action Space: To provide flexibility in velocity control and enable different speed profiles, three magnitudes of acceleration are utilized: 0 (maintaining velocity), a low acceleration denoted as a_L (low acceleration), and a high acceleration denoted as a_H (high acceleration). The action space for selecting acceleration directions is discretized into intervals of 30 degrees, starting from the X direction. Consequently, the action space *Action* can be described as a matrix in $\mathbb{R}^{2\times 25}$, as shown in (2), where each column represents a specific direction and magnitude of acceleration,

$$Action = \begin{bmatrix} \begin{bmatrix} 0 \\ 0° \end{bmatrix} & \begin{bmatrix} a_L \\ 0° \end{bmatrix} & \begin{bmatrix} a_L \\ 30° \end{bmatrix} & \begin{bmatrix} a_L \\ 330° \end{bmatrix} & \cdots \\ & \begin{bmatrix} a_H \\ 0° \end{bmatrix} & \begin{bmatrix} a_H \\ 30° \end{bmatrix} & \cdots & \begin{bmatrix} a_H \\ 330° \end{bmatrix} \end{bmatrix}. \tag{2}$$

2) Observation Representation: In the context of multi-agent systems, the states of RL agent i, denoted as s_i, is defined as the concatenation of its global position $\mathbf{q}_i$ and velocity $\mathbf{p}_i$, represented as $s_i = \begin{bmatrix} \mathbf{q}_i & \mathbf{p}_i \end{bmatrix}^T$. To facilitate information exchange among the RL agents, inter-agent communication and sensor networks are employed. Additionally, the state of the virtual leader, denoted as $s_\gamma = \begin{bmatrix} \mathbf{q}_\gamma & \mathbf{p}_\gamma \end{bmatrix}^T$, is broadcast to all RL agents. With knowledge of the states of all RL agents and the virtual leader, the observation S_i for agent i is defined in order as follows,

$$S_i = \begin{bmatrix} s_i & s_1 & \cdots & s_{i-1} & s_{i+1} & \cdots & s_n & s_\gamma \end{bmatrix}. \tag{3}$$

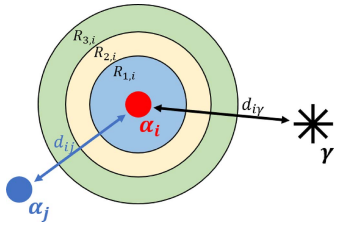

Fig. 1. Spatial relations between different types of agents.

3) Reward Scheme: The reward scheme in the learning setup is designed based on the Reynold's flocking rules plus the leader following requirements. Fig. 1 illustrates the spatial relations where an ego α-agent has a sensing range R_3. For agent α_i, any other α-agents within the sensing range of $R_{3,i}$ are considered as neighbors. The sensing range covers a 360-degree field of view and forms a cyclic coverage area.

The inter-agent distance d_{ij} between agents α_i and α_j is categorized in different ranges to determine the corresponding rewards in Fig. 1. When d_{ij} is small (less than $R_{1,i}$ in Fig. 1), a negative reward is assigned to discourage close proximity. When d_{ij} is within the sensing range $R_{3,i}$ and the agents maintain a safe distance (larger than $R_{1,i}$), a positive reward is given to encourage the α-agents to remain together. Moreover, a desired distance zone ($R_{1,i} < d_{ij} \leq R_{2,i}$) is introduced to encourage agents to maintain a compact group. Based on the design, the reward function r_{ij}^d between two agents, α_i and α_j, is formulated as follows,

$$r_{ij}^d = \begin{cases} -5, & d_{ij} \leq R_1 \text{ (Separation)} \\ 10, & R_1 < d_{ij} \leq R_2 \text{ (Desired zone)} \\ 5, & R_2 < d_{ij} \leq R_3 \text{ (Cohesion)} \\ 0, & d_{ij} > R_3 \text{ (No interaction)} \end{cases}. \tag{4}$$

Note that the values of r_{ij}^d in (4) are given as examples, which can be tuned based on the learning performance. Furthermore, for the case of multiple neighbors, the accumulating distance reward r_i^d of agent α_i can be defined as,

$$r_i^d = \sum_{j \in N_j} r_{ij}^d, \quad N_j = \{j \in N : i \neq j\}. \tag{5}$$

where r_{ij}^d is shown in (4).

In addition to position rewards described in (4) and (5), the reward of alignment or velocity consensus for the neighbors of an α-agent also need to be defined. An alignment reward $r_i^a \in [0, 1]$ is introduced based on the standard deviation of velocities in the neighbor group, which is expressed in (6).

$$r_i^a = \begin{cases} \dfrac{1}{1+\sigma_{i,v}}, & d_{ij} \leq R_3 \\ 0, & d_{ij} > R_3 \end{cases}, \tag{6}$$

where $\sigma_{i,v}$ is the standard deviation of the velocity among all the α-agents in the neighbor group of α_i agent. Note that only α-agents in the communication range R_3 are considered neighbors.

Furthermore, a straightforward reward for the leader following scheme is designed for α_i, which is dependent on the distance $d_{i\gamma}$ between α_i and the virtual leader γ-agent as shown in Fig. 1.

$$r_i^\gamma = -d_{i\gamma}. \tag{7}$$

Finally, the total reward r_i for α-agent i is expressed as,

$$r_i = c_d r_i^d + c_a r_i^a + c_\gamma r_i^\gamma, \tag{8}$$

where c_d, c_a, and c_γ are the weights for distance reward, alignment reward, and leader-following reward, respectively.

III. Collisions-Free Guaranteed Learning-Based Flocking Framework

A. Background: Control Barrier Function

This section presents a concise overview of CBF, which forms the basis of the proposed approach for ensuring collision-free learning among the agents. The CBF concept serves as a vital tool in control theory for enforcing safety constraints on dynamic systems. A mathematical framework was presented to design controllers that maintain system states within predefined safety bounds, see details in [19]. By incorporating CBF constraints in the control design, the proposed method can enable the agents to navigate without collisions and enhance overall safety and maneuverability.

The CBF approach is designed to guarantee system safety by imposing safety constraints on the system behaviors. These safety constraints are carefully defined to prevent undesired behavior or violations of safety limits. Consider a control system represented in an affine form,

$$\dot{x} = f(x) + g(x)u, \tag{9}$$

where $x \in \mathbb{R}^n$, and the control input is represented as $u \in U \subset \mathbb{R}^m$. The functions f and g are assumed to be locally Lipschitz continuous. Let $\mathscr{C} \subset \mathbb{R}^n$ be a safe set that all agents need to maintain. A controller u can guarantee the forward invariance of $\mathscr{C}$ if, for every initial state $x_0 \in \mathscr{C}$, the state trajectory $x(t)$ remains within $\mathscr{C}$ for all $t \geq 0$. In other words, regardless of the initial condition, the system's solutions will always remain in the invariant set $\mathscr{C}$. Assuming that the invariant set $\mathscr{C}$ can be defined as the level set of a control barrier function,

$$\mathscr{C} = \{x \in \mathbb{R}^n | h(x) \geq 0\}. \tag{10}$$

Definition 1 [19]: Assuming the existence of a dynamic system (9) and a safety set (10). Let $h : \mathbb{R}^n \to \mathbb{R}^n$ be a continuously differentiable function. Suppose there exists a locally Lipschitz extended class κ function α and a set $\mathscr{C} \subseteq \mathscr{D} \subset \mathbb{R}^n$ such that for all $x \in \mathscr{D}$,

$$\sup_{u \in U} \left[L_f h(x) + L_g h(x) u + \alpha(h(x)) \right] \geq 0, \tag{11}$$

where the first-order Lie derivative of the system (9) is expressed as $\dot{h}(x) = \frac{\partial h}{\partial x}(f(x) + g(x)u) = L_f h(x) + L_g h(x)u$, the function $h(x)$ is considered a Zeroing Control Barrier Function (ZCBF) defined on $\mathscr{D}$.

In the ZCBF $h(x)$, the set of feasible solutions as the control input is,

$$K(x) = \{u \in U | L_f h(x) + L_g h(x) u + \alpha(h(x)) \geq 0\}. \tag{12}$$

B. CBF Formulation for Pairwise Double Integrator Agents

Each α-agent $i \in N = \{1, 2, \ldots, n\}$ is modeled by a double integrator dynamics in (1). The acceleration of α-agent i is bounded by $\|\mathbf{u}_i\|_\infty \leq a_{\max}$, and the velocity is limited by

$\|\mathbf{p}_i\|_\infty \leq \beta_{\max}$. $\Delta\mathbf{q}_{ij} = \mathbf{q}_i - \mathbf{q}_j$ and $\Delta\mathbf{p}_{ij} = \mathbf{p}_i - \mathbf{p}_j$ represent the relative position and velocity, respectively, between α-agent i and agent j.

In this letter, we adopt the pairwise CBF proposed in [25]. The inter-agent distance requires maintaining a safe distance d_s, which considers the normal component of the relative velocity $\Delta\bar{\mathbf{p}}_{ij} = \|\Delta\dot{\mathbf{q}}_{ij}\| = \frac{\Delta\mathbf{q}_{ij}^T}{\|\Delta\mathbf{q}_{ij}\|}\Delta\mathbf{p}_{ij}$ and the maximum deceleration between α-agents i and j in (13),

$$\|\Delta\mathbf{q}_{ij}\| - \frac{\left(\Delta\bar{\mathbf{p}}_{ij}\right)^2}{4a_{\max}} \geq d_s, \forall i \neq j, \tag{13}$$

where $a_{\max}$ can be practically understood as the saturated actuation capabilities of actuators.

The ZCBF candidate $h_{ij}(\mathbf{q}, \mathbf{p})$ is formulated in (14) based on the constraint in (13).

$$h_{ij}(\mathbf{q}, \mathbf{p}) = \sqrt{4a_{\max}\left(\|\Delta\mathbf{q}_{ij}\| - d_s\right)} + \frac{\Delta\mathbf{q}_{ij}^T}{\|\Delta\mathbf{q}_{ij}\|}\Delta\mathbf{p}_{ij}, \tag{14}$$

and $h_{ij}(\mathbf{q}, \mathbf{p})$ is the level set function of the pairwise safe set $\mathscr{C}_{ij} = \{(\mathbf{q}_i, \mathbf{p}_i) \in \mathbb{R}^4 | h_{ij}(\mathbf{q}, \mathbf{p}) \geq 0\}, \forall i \neq j$. Furthermore, the constraint in (15) is obtained by combining (12) and (14).

$$\begin{aligned} -\Delta\mathbf{q}_{ij}^T\Delta\mathbf{u}_{ij} \leq \gamma h_{ij}^3\|\Delta\mathbf{q}_{ij}\| - \frac{\left(\Delta\mathbf{p}_{ij}^T\Delta\mathbf{q}_{ij}\right)^2}{\|\Delta\mathbf{q}_{ij}\|^2} + \|\Delta\mathbf{p}_{ij}\|^2 \\ + \frac{2a_{\max}\Delta\mathbf{p}_{ij}^T\Delta\mathbf{q}_{ij}}{\sqrt{4a_{\max}\left(\|\Delta\mathbf{q}_{ij}\| - d_s\right)}}, \quad \forall i \neq j. \end{aligned} \tag{15}$$

Finally, the pairwise safety barrier constraints are represented in the form of $A_{ij}\mathbf{u} \leq b_{ij}$, where

$$A_{ij} = \left[0, \ldots, \underbrace{-\Delta\mathbf{q}_{ij}^T}_{\text{agent } i}, \ldots, \underbrace{\Delta\mathbf{q}_{ij}^T}_{\text{agent } j}, \ldots, 0\right], \tag{16}$$

and

$$\begin{aligned} b_{ij} = \gamma h_{ij}^3\|\Delta\mathbf{q}_{ij}\| - \frac{\left(\Delta\mathbf{p}_{ij}^T\Delta\mathbf{q}_{ij}\right)^2}{\|\Delta\mathbf{q}_{ij}\|^2} + \|\Delta\mathbf{p}_{ij}\|^2 \\ + \frac{2a_{\max}\Delta\mathbf{p}_{ij}^T\Delta\mathbf{q}_{ij}}{\sqrt{4a_{\max}\left(\|\Delta\mathbf{q}_{ij}\| - d_s\right)}}. \end{aligned} \tag{17}$$

C. RL-CBF Flocking Architecture

The architecture of the multi-agent flocking system incorporating reinforcement learning and CBF is depicted in Fig. 2. In this architecture, RL agent i learns a policy based on the observation S_i in (3), and its output is represented by the predefined nominal control $\hat{\mathbf{u}}_i$, which serves as one desired control input of a CBF controller. The CBF controller utilizes the desired input to generate the actual control command for α_i agent, ensuring adherence to safety constraints and achieving collision-free flocking behavior.

Inspired by [18] and [25], a quadratic programming (QP) controller is employed in the architecture. This controller

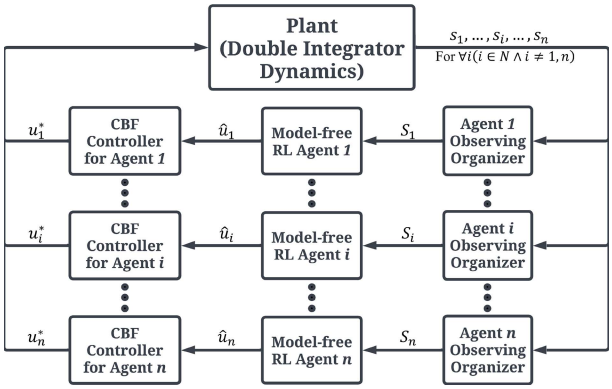

Fig. 2. Control architecture combining the DQN controller for flocking and the CBF controller for guaranteed free collisions during flocking.

TABLE I
SIMULATION PARAMETERS

Symbol	Parameter value	Symbol	Parameter value
a_1	3 m/s^2	d_s	0.4 m
a_2	5 m/s^2	$a_{\max}$	5 m/s^2
R_1	0.8 m	C_d	1
R_2	1.2 m	C_a	10
R_3	1.5 m	C_γ	20

aims to minimize the discrepancy between the actual control command $\mathbf{u}_i$ and the nominal control command $\hat{\mathbf{u}}_i$.

$$\begin{aligned} \mathbf{u}^* = \underset{\mathbf{u}\in\mathbb{R}^{2N}}{\text{argmin}} \; J(\mathbf{u}) = \sum_{i=1}^{N}\|\mathbf{u}_i - \hat{\mathbf{u}}_i\|^2 \\ \text{s.t.} \quad A_{ij}\mathbf{u} \leq b_{ij}, \quad \forall i \neq j \\ \|\mathbf{u}_i\|_\infty \leq a_i, \quad \forall i \in N. \end{aligned} \tag{18}$$

As a result, the resulting control command $\mathbf{u}^*$ is equal to the nominal control $\hat{\mathbf{u}}_i$, when the system is in a safe state without any predicted collisions. However, when a collision is anticipated based on the states of the agents, the QP controller triggers an essential behavior modification. In such cases, the QP controller adjusts the control command to ensure collision avoidance and maintain the safety of the multi-agent flocking system. Ultimately, $\mathbf{u}_i^*$ is substituted into (1) as the control input.

IV. SIMULATION RESULTS AND DISCUSSIONS

A ten-agent system was simulated with MATLAB/Simulink and Reinforcement Learning Toolbox, utilizing the proposed framework in Fig. 2. A comparison study was conducted to compare the outcomes of a pure RL method and the RL method with CBF. The simulation results and advantages of the proposed framework are discussed and analyzed in this section.

Tab. I provides detailed information about the parameters of the simulations. Decentralized training using DQN was conducted for all α-agents in the multi-agent system. Each α-agent consisted of two fully connected layers, with 32 neurons in each layer. The training contained 1000 episodes, and each episode had 30 seconds duration. For both cases (with and without CBF), the initial positions of α-agents $i \in N$, were set as $[-1\ 4], [-1\ 2], [-1\ 0], [-1\ -2], [-1\ -4], [0\ 4], [0\ 2], [0\ 0], [0\ -2]$, and $[0, -4]$,

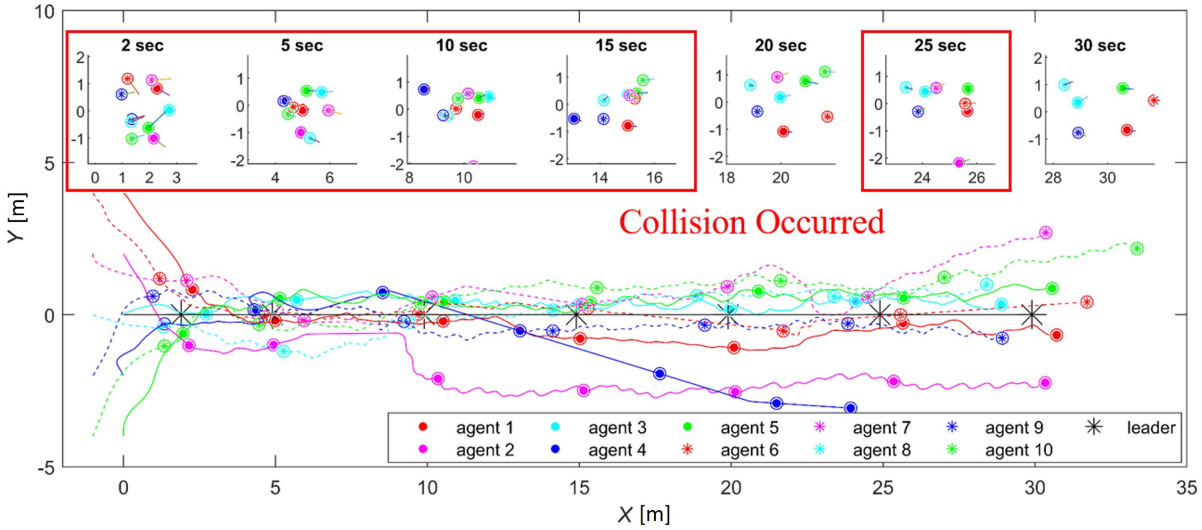

Fig. 3. Simulation result of a pure DQN flocking at the 1000^{th} episode. From the time-stamp section and the trajectories of the α-agents, collisions occurred during the flocking training, highlighted in the red boxes. The circles around the agents represent the safe range with a radius of d_s

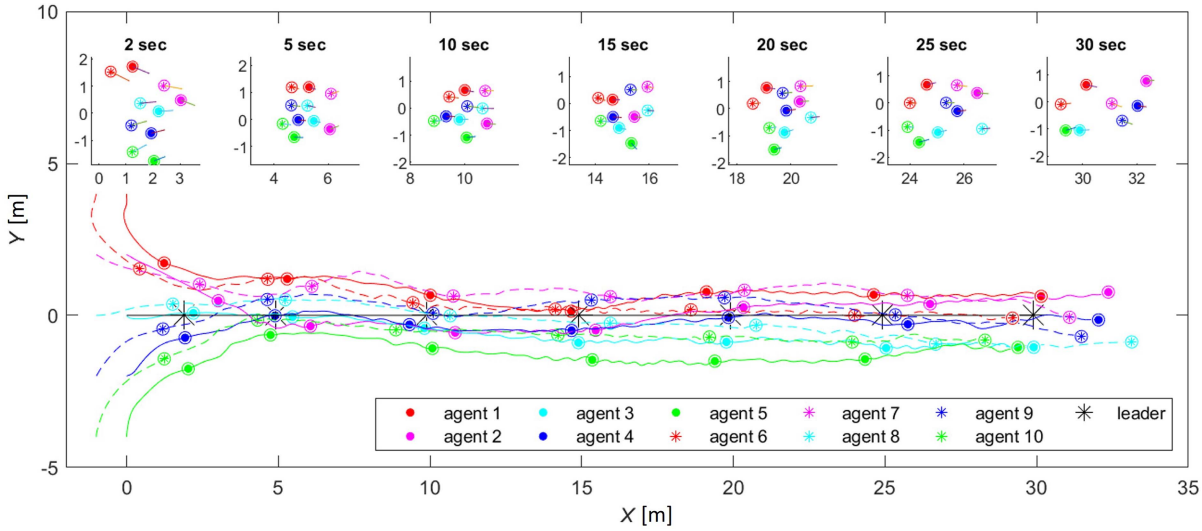

Fig. 4. Simulation result of the DQN flocking with CBF constraints at the 1000^{th} episode. The flocking motion was maintained collision-free during the entire journey. The circles around the agents represent the safe range with a radius of d_s.

respectively. The virtual leader traveled along the y-axis at a constant speed of 1 m/s, starting from the origin.

The simulation results of the last episode using the DQN training without and with CBF are presented in Fig. 3 and Fig. 4, respectively. To make a fair comparison, the DQN reward design (4)-(8) and the configurations of parameters in Tab. I are identical in both cases. In both simulations, the virtual leader γ attracted all α-agents in the early phase (before 10 second), causing them to converge within a small range. As highlighted/labeled in Fig. 3, there are various collisions during the flock training. In contrast, by incorporating CBF into the learning scheme, the flocking members can successfully avoid inter-agent collisions from the beginning. The CBF was continuously applied, which allowed the α-agents to avoid inter-agent collisions in a continuous action space and remedy the limitation of discrete control commands from the DQN controller.

Another significant distinction between the two simulations lies in the overall flocking performance of achieving the desired flocking rule with smooth trajectories. Because the introduced CBF from the beginning of the learning process will influence both the collision results and the overall flocking performance, the two simulations trained separately with random sampling processes gave different trajectories of α-agents, as shown in Fig. 3 and Fig. 4. The training with CBF was able to achieve the desired leader-following flocking behaviors with a smaller number of training episodes. As a result, all α-agents followed the virtual leader γ, maintained a safe distance, and prevented fragmentation. In contrast, the training without CBF exhibited inferior performance. Specifically, α_4 and α_7 deviated from the agent group. Additionally, α_4 struggled to track the virtual leader with a significantly large gap. The flocking group also tended to adopt diverse trajectories toward the end of the simulation. Furthermore, due to the reduced penalties incurred for undesired small inter-agent distances,

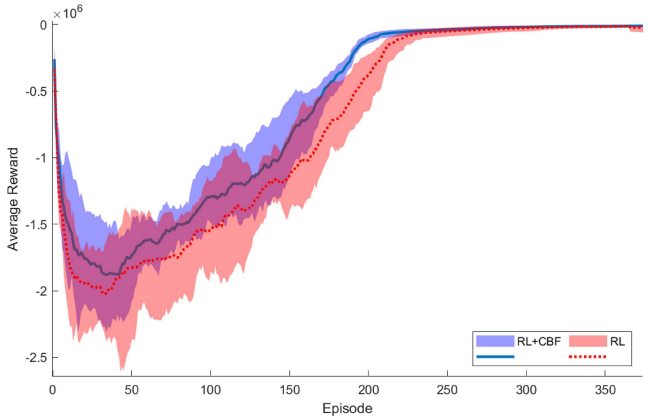

Fig. 5. Comparison of average reward for both cases, when the score averaging window length is 30 episodes. The lines indicate the mean value of 10 α-agents' average reward, and the patch areas represent the range of the average reward.

DQN with CBF trained flocking behaviors with smoother trajectories than those of the pure DQN approach. It is worth pointing out that the neural network setup and parameter settings remain consistent across both simulations. From this perspective, integrating CBF into the RL-based flocking not only ensures collision-free but also relaxes parameter tuning and the complexity of reward function design. Unlike other flocking control methods (e.g., [3]), no specific formation (e.g., hexagon shapes corresponding to the minimization of the Hamiltonian function) emerges among the α-agents during the flocking process.

An additional benefit of integrating CBF into reinforcement learning for flocking control is the enhanced training efficiency. Fig. 5 presents a comparison of the rewards with a moving average filter between the two simulations. By effectively mitigating the high penalties arising from inter-agent distances falling below the safety threshold and minimizing over-repulsion due to close inter-agent distances, the reinforcement learning integrated with CBF exhibits superior reward progression throughout the training process. The advantage was demonstrated by a higher reward over time and faster convergence to the 'fine-tuning' training stage, as shown in the solid line and the corresponding patch area in Fig. 5.

V. Conclusion

This letter developed a new learning-based flocking control framework that guarantees free collisions among interacting agents. In addition to guaranteeing safety during the learning process and outcomes, the introduced CBF in the DQN-based framework can also enhance the training efficiency and relax the reward design to achieve desired flocking behaviors. Simulation results demonstrated the effectiveness of the proposed framework for flocking motions satisfying three Reynolds' flocking rules and also the leader-following rule.

References

[1] F. Wang and Y. Chen, "A novel hierarchical flocking control framework for connected and automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 4801–4812, Aug. 2021.

[2] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 25–34, Aug. 1987.

[3] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. Autom. Control*, vol. 51, no. 3, pp. 401–420, Mar. 2006.

[4] H. Tanner, A. Jadbabaie, and G. Pappas, "Stable flocking of mobile agents part i: Dynamic topology," in *Proc. 42nd IEEE Int. Conf. Decision Control*, 2003, pp. 2016–2021.

[5] M. Arcak, "Passivity as a design tool for group coordination," *IEEE Trans. Autom. Control*, vol. 52, no. 8, pp. 1380–1390, Aug. 2007.

[6] F. Wang and Y. Chen, "Flocking control of multi-agent systems with permanent obstacles in strictly confined environments," *J. Auton. Vehicles Syst.*, vol. 1, no. 2, Sep. 2021, Art. no. 021005, doi: 10.1115/1.4052161.

[7] G. Wang, M. Liu, F. Wang, and Y. Chen, "A novel and elliptical lattice design of flocking control for multi-agent ground vehicles," *IEEE Control Syst. Lett.*, vol. 7, pp. 1159–1164, 2023.

[8] S.-M. Hung and S. N. Givigi, "A Q-learning approach to flocking with UAVs in a stochastic environment," *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 186–197, Jan. 2017.

[9] W. Wang, L. Wang, J. Wu, X. Tao, and H. Wu, "Oracle-guided deep reinforcement learning for large-scale multi-UAVs flocking and navigation," *IEEE Trans. Veh. Technol.*, vol. 71, no. 10, pp. 10280–10292, Oct. 2022.

[10] J. Xiao, G. Wang, Y. Zhang, and L. Cheng, "A distributed multi-agent dynamic area coverage algorithm based on reinforcement learning," *IEEE Access*, vol. 8, pp. 33511–33521, 2020.

[11] M. Jafari and H. Xu, "A biologically-inspired distributed fault tolerant flocking control for multi-agent system in presence of uncertain dynamics and unknown disturbance," *Eng. Appl. Artif. Intell.*, vol. 79, pp. 1–12, Mar. 2019.

[12] J. Blumenkamp, S. Morad, J. Gielis, Q. Li, and A. Prorok, "A framework for real-world multi-robot systems running decentralized GNN-based policies," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, 2022, pp. 8772–8778.

[13] M. Durve, F. Peruani, and A. Celani, "Learning to flock through reinforcement," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 102, Jul. 2020, Art. no. 012601.

[14] K. Morihiro, T. Isokawa, H. Nishimura, and N. Matsui, "Characteristics of flocking behavior model by reinforcement learning scheme," in *Proc. SICE-ICASE Int. Joint Conf.*, 2006, pp. 4551–4556.

[15] C. Yan, X. Xiang, and C. Wang, "Fixed-wing UAVs flocking in continuous spaces: A deep reinforcement learning approach," *Robot. Auton. Syst.*, vol. 131, Sep. 2020, Art. no. 103594.

[16] E. Tolstaya, F. Gama, J. Paulos, G. Pappas, V. Kumar, and A. Ribeiro, "Learning decentralized controllers for robot swarms with graph neural networks," in *Proc. Conf. Robot Learn.*, 2020, pp. 671–682.

[17] P. Zhu, W. Dai, W. Yao, J. Ma, Z. Zeng, and H. Lu, "Multi-robot flocking control based on deep reinforcement learning," *IEEE Access*, vol. 8, pp. 150397–150406, 2020.

[18] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, pp. 3387–3395.

[19] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3861–3876, Aug. 2017.

[20] T. Ibuki, S. Wilson, J. Yamauchi, M. Fujita, and M. Egerstedt, "Optimization-based distributed flocking control for multiple rigid bodies," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1891–1898, Apr. 2020.

[21] M. Cavorsi, B. Capelli, L. Sabattini, and S. Gil, "Multi-robot adversarial resilience using control barrier functions," *IEEE Trans. Robot., Sci. Syst.*, early access, Dec. 12, 2023, doi: 10.1109/TRO.2023.3341570.

[22] J. Grover, N. Mohanty, C. Liu, W. Luo, and K. Sycara, "Noncooperative herding with control barrier functions: Theory and experiments," in *Proc. IEEE 61st Conf. Decision Control (CDC)*, 2022, pp. 80–86.

[23] R. Cheng, M. J. Khojasteh, A. D. Ames, and J. W. Burdick, "Safe multi-agent interaction through robust control barrier functions with learned uncertainties," in *Proc. 59th IEEE Conf. Decis. Control (CDC)*, 2020, pp. 777–783.

[24] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[25] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Trans. Robot.*, vol. 33, no. 3, pp. 661–674, Jun. 2017.