Su *et al*. *Intell Robot* 2024;4(1):39-60
DOI: 10.20517/ir.2024.03

Intelligence & Robotics

Research Article

Open Access

A collaborative siege method of multiple unmanned vehicles based on reinforcement learning

Muqing Su, Ruimin Pu, Yin Wang, Meng Yu

College of Astronautics, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, Jiangsu, China.

Correspondence to: Prof. Yin Wang, College of Astronautics, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, Jiangsu, China. E-mail: yinwangee@nuaa.edu.cn

How to cite this article: Su M, Pu R, Wang Y, Yu M. A collaborative siege method of multiple unmanned vehicles based on reinforcement learning. *Intell Robot* 2024;4(1):39-60. http://dx.doi.org/10.20517/ir.2024.03

Received: 19 Oct 2023 **First Decision:** 21 Nov 2023 **Revised:** 13 Dec 2023 **Accepted:** 2 Feb 2024 **Published:** 29 Feb 2024

Academic Editor: Simon X. Yang **Copy Editor:** Dong-Li Li **Production Editor:** Dong-Li Li

Abstract

A method based on multi-agent reinforcement learning is proposed to tackle the challenges to capture escaping Target by Unmanned Ground Vehicles (UGVs). Initially, this study introduces environment and motion models tailored for cooperative UGV capture, along with clearly defined success criteria for direct capture. An attention mechanism integrated into the Soft Actor-Critic (SAC) is leveraged, directing focus towards pivotal state features pertinent to the task while effectively managing less relevant aspects. This allows capturing agents to concentrate on the whereabouts and activities of the target agent, thereby enhancing coordination and collaboration during pursuit. This focus on the target agent aids in refining the capture process and ensures precise estimation of value functions. The reduction in superfluous activities and unproductive scenarios amplifies efficiency and robustness. Furthermore, the attention weights dynamically adapt to environmental shifts. To address constrained incentives arising in scenarios with multiple vehicles capturing targets, the study introduces a revamped reward system. It divides the reward function into individual and cooperative components, thereby optimizing both global and localized incentives. By facilitating cooperative collaboration among capturing UGVs, this approach curtails the action space of the target UGV, leading to successful capture outcomes. The proposed technique demonstrates enhanced capture success compared to previous SAC algorithms. Simulation trials and comparisons with alternative learning methodologies validate the effectiveness of the algorithm and the design approach of the reward function.

Keywords: Multi-agent, cooperative capture, soft actor-critic algorithm, attention mechanism, reward function design

1. INTRODUCTION

Multi-agent target capture involves a collection of intelligent agents that work together using collaborative processes to efficiently capture targets by employing specific techniques for optimizing their formation[1]. Collaborative encirclement has the ability to assist various tasks, including search[2], interception, formation transformation, and cooperative transportation. It can also be utilized in areas such as autonomous drone control[3], autonomous vehicle tracking control[4], missiles, and defense systems[5]. Furthermore, it is considered a fundamental aspect in the field of multi-agent collaboration. The solution approaches for multi-agent collaborative pursuit can be categorized into two groups: non-learning and learning techniques[6].

The research on non-learning collaborative pursuit mostly centers around the domain of differential games. This involves converting collaborative pursuit problems into differential game problems in order to facilitate collaboration and cooperation among intelligent agents. Isaacs perceives the encirclement problem as a dynamic game where players do not cooperate. He represents it as a system with decision-making that interacts and evolves over time. To minimize an objective function and find a solution, he employs differential games[7]. Dong *et al.* introduced a hybrid algorithm that combines improved dynamic artificial potential fields (APFs) and differential games to tackle the issues of high computational cost and limited universality in chase and evasion game algorithms[8]. Sun *et al.* utilized differential game theory to devise individualized guidance laws for the motion process of many pursuers, effectively accomplishing the task by dividing the process into two distinct portions[9]. While differential games can be employed to address collaborative pursuit problems, the intricacy of the problem renders the computation of the objective function highly challenging when multiple pursuers are involved. Moreover, the solution becomes intricate and time-consuming due to the rapid expansion of the state space and decision space[10]. The graph search approach[11] investigates efficient techniques for enabling trackers to locate evaders on interconnected graphs. In contrast to the differential game technique, the graph search method does not depend on solving the objective function and instead emphasizes path design and optimization of search tactics. This enhances the resilience of the graph search method when confronted with intricate and ever-changing surroundings, enabling it to promptly adjust to various pursuer behaviors and environmental alterations. It possesses the benefits of being simple, intuitive, and highly scalable. Athanasios converted the encirclement problem into a graph node problem and introduced an iterative greedy node search algorithm based on this approach. This program effectively apprehended fugitives through testing conducted in indoor environments[11]. While this strategy may be effective in apprehending fugitives in specific situations, creating precise graph structures in dynamic, unfamiliar, or intricate settings can be challenging and time-consuming, ultimately influencing the effectiveness and precision of the search. The collaborative capture challenge is utilized to examine team collaboration capture approaches, as it bears a resemblance to biological predation processes. The bio-heuristic technique, in contrast to the differential game method and graph search method, places greater emphasis on team cooperation and cluster behavior. It draws inspiration from biological systems to gain collective knowledge and possesses enhanced robustness and adaptive capabilities. Janosov *et al.* introduced a cluster pursuit approach that draws inspiration from biological predatory systems in nature[12]. Wang *et al.* established the capture circumstances from a biological standpoint and devised an effective control technique by considering operational costs and their associated coefficients[13]. Biological heuristic techniques have limited adaptability and lack universality[14].

The advancement of reinforcement learning theory in recent years has stimulated study in the disciplines of decision-making and planning. Several techniques, including Q-learning, Deep Deterministic Policy Gradient (DDPG), and Proximal Policy Optimization (PPO), have been utilized to address a wide range of intricate challenges. Bilgin *et al.* utilized the algorithm to address the challenge of capturing a solitary target agent on a grid map within the context of reinforcement learning-based capture and escape problems. This approach was employed in situations where two agents attempt to apprehend a solitary agent[15]. Wang *et al.* introduced a decentralized collaborative pursuit technique incorporating a communication feature. The cooperative chase control problem was resolved by implementing a centralized critic and distributed actor structure,

together with a learning-based communication system[16]. Du *et al.* introduced a novel reinforcement learning approach that combines cellular data parameter sharing with curriculum learning. This method allows Unmanned Ground Vehicles (UGVs) to share observation information and acquire effective methods for encircling, attacking, shrinking, and capturing targets[17]. Qu *et al.* employed the Multi-Agent Posthumous Credit Assignment (MA-POCA) algorithm with a centralized training distributed execution architecture to train a pursuit strategy. This method allows numerous unmanned surface vehicles to independently evade barriers and collaboratively capture targets[14]. De Souza *et al.* employed the Twin Delayed DDPG (TD3) algorithm and implemented a group reward system that incentivizes the construction of a good capture control strategy[6]. Zhang *et al.* introduced the Dueling Double Deep Q Network (DDQN)-based Adaptive Cooperative (DACOOP) algorithm, which integrates reinforcement learning and APFs as predefined rules to synchronize the chasing strategy of a group of robots pursuing a fleeing soldier. They demonstrated that the algorithm outperformed APF and Dueling DDQN (D3QN) in terms of success rate in cooperative chasing scenarios[18]. Hüttenrauch *et al.* employed the Trust Region Policy Optimization (TRPO) algorithm to tackle the multi-agent capture problem and embedded mean feature embedding as a state observation to solve the problem of high-dimensional information perception by agents[19]. Liu *et al.* introduced a novel approach that integrates the double fuzzy system with Q-learning and Q-value table fuzzy inference system (QTFIS) to address the challenges of solving problems in continuous space and surpass the limitations of low-dimensional space. The proposed technology aims to tackle the problem of tracking and capturing unmanned aerial vehicles[20]. Zhang *et al.* introduced a method to address the problem of intercepting multiple spacecraft during orbit chasing and escaping. This method involves using deep reinforcement learning (DRL) to generate a capture zone (CZ) embedding strategy and a barrier inverse solution neural network (BISNN) to determine an approximately optimal guidance law within the CZ. This approach was described in detail in their paper[21]. dos Santos *et al.* developed a parallel optimization algorithm to minimize the capture time in the capture and escape problem. They utilized the pruning technique of the pac-dot strategy to decrease the number of states and transitions, thereby reducing the computational resources needed for the game and enhancing the scalability of the technology[22]. Wang *et al.* introduced a collaborative approach for capturing strategies that relies on sensing information instead of communication. This approach is based on the Multi-Agent DDPG (MADDPG) algorithm and employs centralized training and distributed execution to govern the highly cooperative capture agents[23]. Despite the significant progress made in using reinforcement learning to address the roundup problem, there are still obstacles that need to be overcome. For instance, in the majority of research, the effectiveness of rounding is evaluated based on whether the rounding algorithm is able to confine the target intelligence inside the prescribed parameters, notwithstanding the significant constraints imposed by rounding. Information in multi-intelligent systems typically encompasses numerous dimensions, resulting in complex state issues characterized by a high number of dimensions. This exacerbates the intricacy and computational load of the state space, perhaps resulting in the inability to devise an efficient rounding strategy throughout the rounding process. The attention mechanism has garnered significant attention and research in addressing complex spatial training problems with high dimensions. This is because it possesses the ability to concentrate on crucial information within the model, resulting in enhanced model performance. The attention mechanism in neural networks allows for the assignment of varying weights to different parts of the input, enhancing flexibility and accuracy in information processing[24]. Zhang *et al.* integrated the attention mechanism with DDQN to address the path planning and obstacle avoidance challenges faced by UGVs in urban airspace. This integration resulted in a reduction in domino conflict counts and minimized deviation from the reference path[25]. Peng *et al.* proposed DRL-GAT-SA (DRL based on Graph Attention Networks) to tackle the issue of self-driving cars reacting to pedestrians with unpredictable movements. This approach combines Graph Attention Networks with DRL. The system, which incorporates Attention Networks and Simplex Architecture, ensures the safety of vehicle driving and efficiently prevents crashes[26].

This paper focuses on the problem of capturing in the absence of boundary conditions. The main contributions are as follows: Firstly, we provide the necessary conditions for successfully determining the capturing of

UGVs in the absence of boundary conditions, taking into account practical application scenarios. Secondly, we propose a solution method based on the Soft Actor-Critic (SAC) reinforcement learning framework to address the capturing problem. Additionally, we introduce an attention mechanism in the Critic network to tackle issues arising from the environment and prevent collisions. The attention mechanism addresses the challenges posed by high-dimensional difficulties resulting from environmental factors and changes in the status of intelligences. By separating the incentive function into individual and collaborative rewards in order to optimize both global and local rewards, the UGV is directed to gain a more efficient comprehension of the roundup task and afterward make decisions. Ultimately, the enhanced algorithm proposed in this research is validated by both simulation and experimentation, with the experimental findings confirming the efficacy and applicability of the method presented.

2. DESCRIPTION OF MULTI-UGV ENCIRCLEMENT

The assumption is that the capturing UGV carries out the duty of capturing in a specific area, while the target UGV possesses a certain level of mobility and can evade the capturing UGV by using information about its position and speed. This allows the target UGV to keep a distance from the capturing UGV and reach a safe area.

In a finite two-dimensional area, there exist $n(n >= 3)$ UGVs designed to encircle and capture a target UGV in motion. This mission scenario, as depicted in Figure 1, demonstrates that the three capturing UGVs, including P_1, P_2, and P_3, are positioned within the designated area along with the target UGV including T. They are equipped with sensors that enable them to perceive each other. V_1, V_2, V_3 and V_T represent the velocities of three pursuing autonomous vehicles and the target autonomous vehicle, respectively. $V_i > V_T(i = 1, 2, 3)$. At the start, it is assumed that the capturing UGV and the target are unaware of each other's presence. However, once the capturing UGV recognizes the target, it promptly travels towards the target location. The primary objective of the capturing UGV is to successfully encircle the target UGV by gradually approaching it. Upon detecting the approach of the capturing UGV, the target UGV will employ an escape strategy to distance itself as much as possible from the capturing UGV. The capturing UGV is considered successful in capturing the target UGV when it effectively forms a circle around it through coordination and cooperation. Conversely, it is deemed unsuccessful.

2.1. The kinematics model of an UGV

The study presents a two-wheel differential model for the UGV, considering it to be a rigid body controlled by the rotational speeds of its two wheels. The two-wheel differential model is illustrated in Figure 2. The motion model of the UGV can be expressed by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} cos\theta & 0 \\ sin\theta & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v \\ w \end{bmatrix} \tag{1}$$

where x and y represent the current position of the UGV, and $theta$ is the yaw angle. v and w represent the instantaneous linear velocity and the instantaneous angular velocity, respectively[27].

The model takes in a motion velocity vector comprising linear and angular velocities and generates an output position vector. This position vector determines the subsequent moment's location of the cart model based on the provided velocity and angular velocity information.

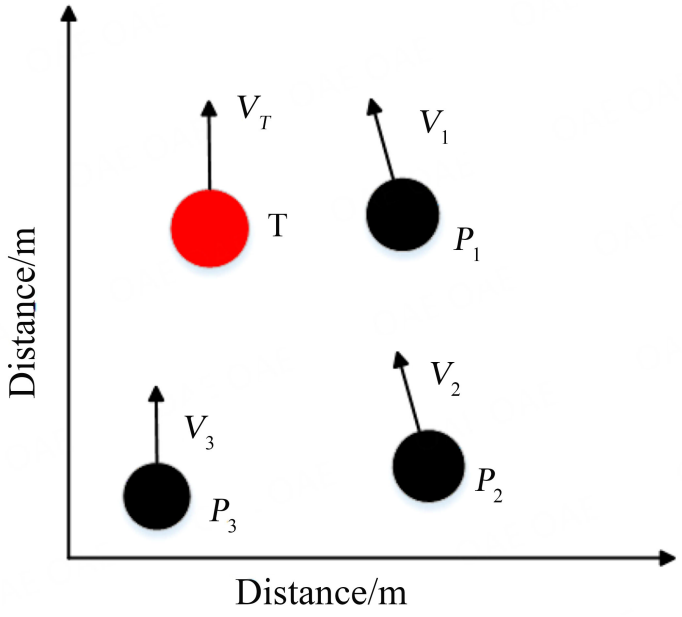

Figure 1. The task scenario description.

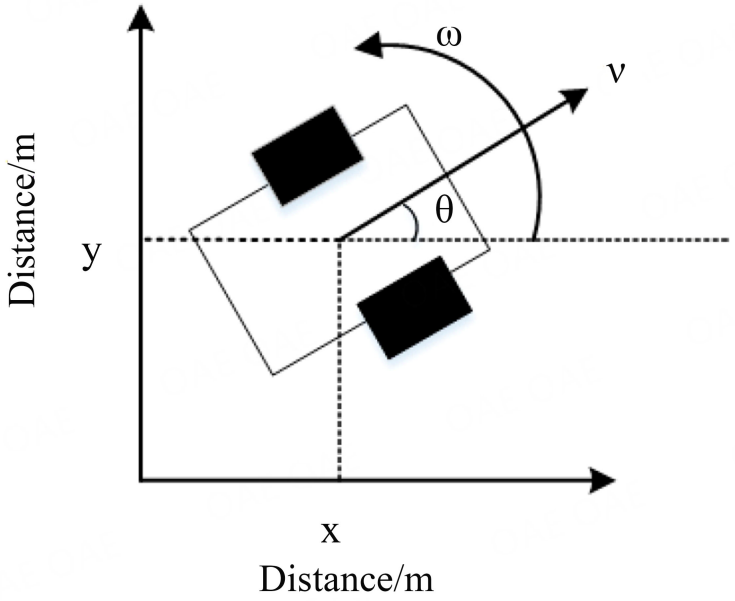

Figure 2. Differential Drive Kinematic Model.

2.2. Strategy for the escape of UGV

The UGV being targeted follows a predetermined path from a specific beginning point to a specific endpoint. When it finds a roundup vehicle, it will employ specific rules to evade capture. When the target vehicle detects that the distance between itself and the surrounding vehicle is 0.5m or less, it evades based on factors such as the distance from the endpoint, the distance from the surrounding vehicle, and the heading angle. Unless there is a requirement to evade, the target vehicle will persist in its initial velocity and angular velocity.

2.3. Explanation of the process of encirclement

This work employs a reinforcement learning control approach to enable the apprehending vehicle to navigate from its starting point toward the target vehicle and successfully apprehend it while avoiding any collisions. An

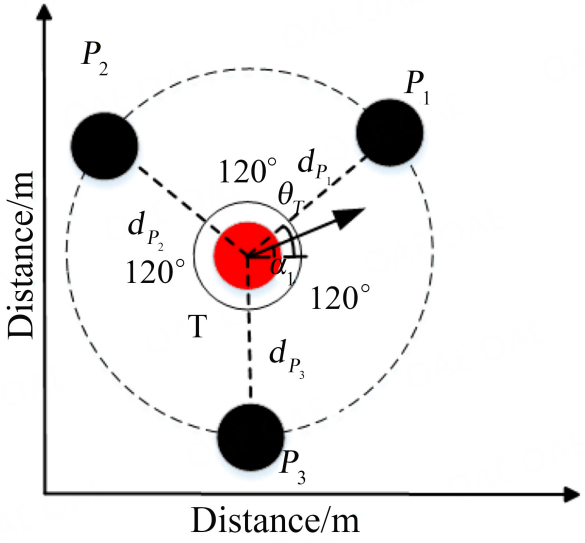

Figure 3. Ideal Capture Circle.

optimal encirclement circle is achieved when the $n(n > 3)$ capturing UGVs called P_i are evenly spaced around the target UGV T. The distance separating the capturing UGVs and the target UGV is d_{P_i}. The relative angle between the capturing UGV, P_i, its clockwise direction UGV, P_i, and the target UGV, T represents the angle produced by the neighboring intelligences, which is called θ_{iTj}. The following prerequisites are required for successful completion: it is necessary to adhere to the criteria for the formation of the capture radius, where d_{min} represents the minimum collision distance, and d_{max} represents the maximum successful capturing radius. This paper uses three capturing UGVs as examples to demonstrate the distribution of capturing UGVs around a target UGV. The goal is to maintain the motion state of the capturing UGV while allowing for a certain angle of deviation. The specific deviation angle can be set accordingly, such as 10°. In conclusion, the roundup angle conditions are as follows: $110° \leqslant \theta_{iTj} \leqslant 130°$. The requirements for rounding distance and angle limitations can be expressed by

$$\begin{cases} lim_{t\to\infty} d_{min} < d_{P_i} < d_{success} (i \in N) \\ \lim_{t\to\infty} |\theta_{iTj} - 120°| \leq 10° (i, j \in N; i \neq j) \end{cases} \tag{2}$$

By taking three pursuit unmanned vehicles P_1, P_2, and P_1 as examples, it is assumed that these vehicles are distributed counterclockwise around the target unmanned vehicle, with P_1 positioned to the upper right of the target unmanned vehicle. The ideal capture encirclement is illustrated in Figure 3, where P_1, P_2 and P_3 represent the capturing UGVs, and T denotes the target UGV. The distance between P_1, P_2, P_3 and T is denoted as d_{P_1}, d_{P_2}, d_{P_3}, respectively, and each falls within the range of d_{min} to $d_{success}$; θ_T refers to the heading angle of the target UGV; α_1 represents the angle between the target line of sight from capturing UGV P_i and T and the x-axis; The ideal angle formed by the vertex of T and P_i and the clockwise direction of called is 120°. The optimal position of P_1 is determined by $(x_T + d_{P_1} \cdot cos\alpha_1, y_T + d_{P_1} \cdot sin\alpha_1)$. The ideal position of P_2 is derived from $(x_T + d_{P_1} \cdot cos(120° - \alpha_1), y_T - d_{P_1} \cdot sin(120° - \alpha_1))$, and the ideal position of P_3 is obtained through $(x_T - d_{P_1} \cdot cos(\alpha_1 - 60°), y_T - d_{P_1} \cdot sin(\alpha_1 - 60°))$. Traditional algorithms are algorithms that rely on established rules and predetermined tactics to solve issues by manual design[8]. These techniques encompass search, optimization, and planning. Nevertheless, conventional algorithms have certain drawbacks: they heavily depend on predetermined rules and strategies, making it challenging to adapt to intricate and unpredictable situations. Additionally, handling high-dimensional problems becomes difficult due to variations in the UGV's state or alterations in the environment. Expressing the cooperative clustering behavior of numerous capturing UGVs using an objective optimization function is challenging.

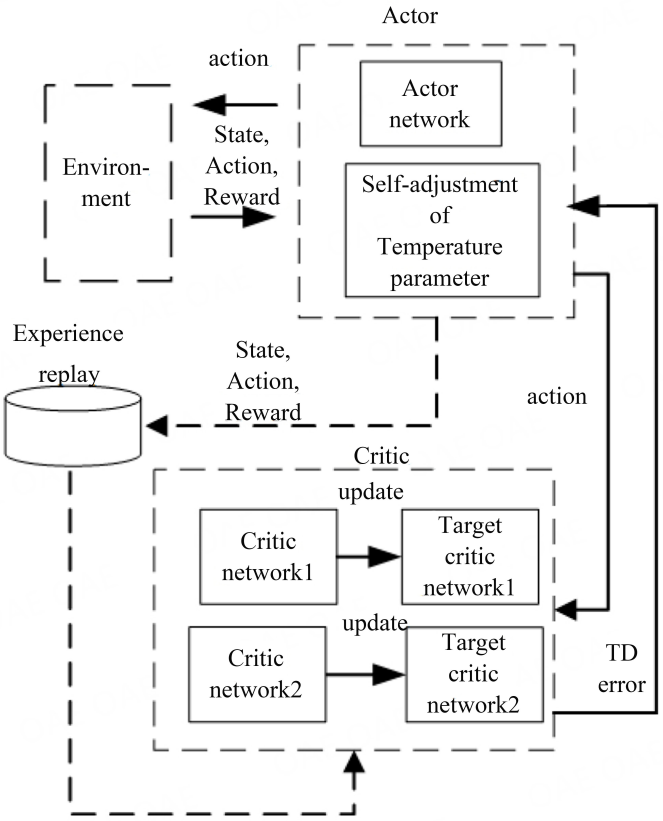

Figure 4. SAC Structure Diagram.

Reinforcement learning involves actively learning from interactions with the environment to optimize a specific objective by maximizing the desired reward. Typically, reinforcement learning algorithms are trained in simulated environments to minimize the need for computational resources. These algorithms have the capacity to learn independently and continuously improve. By interacting with the environment, the agent can adjust its strategy based on feedback signals to adapt to changes and uncertainties. Reinforcement learning algorithms can be applied in various situations by interacting with the environment to accommodate environmental changes. Additionally, they are capable of effectively dealing with state spaces that are high-dimensional, continuous, and complex. This study utilizes reinforcement learning methods to address the constraints of conventional algorithms in the context of capturing.

3. MULTI-UGV ROUNDUP BASED ON SAC

SAC is an algorithm that utilizes strategy gradient and incorporates the concept of maximum entropy reinforcement learning. Unlike other algorithms that rely on value function, SAC aims to enhance the ability to explore by maximizing the entropy of the strategy. This allows the strategy to effectively explore the environment and prevent getting stuck in local optimal solutions, thereby improving algorithm convergence. The schematic diagram of the SAC is depicted in Figure 4.

The SAC algorithm typically employs a configuration consisting of one Actor network, two Critic networks, and two Target Critic networks. The Actor network is responsible for generating the action strategy by selecting an action based on the current state and updating its parameters using the gradient descent method. The Critic network is used to estimate the Q-value function by estimating the Q-value of the action based on the current state and selected action. It calculates the Temporal Difference (TD) error based on the Bellman equation and

uses this error to calculate the gradient of the Critic network. The parameters of the Critic network are then updated using the gradient descent method. The gradient of the Critic network is computed, and its parameters are modified using the gradient descent technique. A target Critic network is employed to provide a consistent target Q-value, minimizing fluctuations during training, and updating the parameters through a soft update strategy. SAC utilizes a maximum entropy objective function to optimize the strategy. The primary objective of this function is to optimize the expected return while maintaining an exploratory strategy.

3.1. Designing algorithms

SAC algorithm incorporates the entropy of the strategies into the learning objective of the strategy network. This allows for the exploration of new strategies while maximizing the expected cumulative payoff[28]. By doing so, SAC avoids the limitations of a global strategy generation approach and enhances the algorithm's ability to explore and generalize effectively. As the number of intelligences grows, the multi-intelligence environment becomes more intricate and unpredictable, potentially resulting in dimensional explosion or failure to converge. SAC struggles to handle the issue of high-dimensional state space. The conventional Critic network of SAC is not sufficiently accurate in handling state features, particularly when there are numerous state features. This is because it treats all state features equally, which can result in over-estimation of the Critic. This paper introduces an attention mechanism to the Critic network in the SAC algorithm. This allows the network to prioritize important state features and selectively process different features. As a result, the onlooker intelligences can focus on the behaviors and positions of other intelligences, leading to improved cooperation and better pursuit of the target intelligences. To improve the efficiency of the algorithm, it is important to accurately capture the critical information of the task and estimate the true value function accurately. This will help avoid unnecessary activities or wasteful situations. Additionally, it is necessary to adaptively adjust the attention weights based on changes in the environmental state. This will enable better adaptation to changes in the behavior and state of the UGV. This, in turn, mitigates the over-estimation problem, thereby reducing the problems associated with high dimensional state spaces, boosting the efficiency and performance of learning and enhancing robustness.

The attention mechanism involves transforming the input matrix $\mathbf{x}$ into a novel representation that incorporates contextual information $\mathbf{y}$. The differential parameter matrix, denoted as $\mathbf{W}_q$, $\mathbf{W}_k$, $\mathbf{W}_q$, is utilized to create a representation of the input data by incorporating contextual information $\mathbf{y}$, which can be expressed by

$$\mathbf{y} = softmax(\frac{(\mathbf{x}^T\mathbf{W}_q)(\mathbf{x}^T\mathbf{W}_q)^T}{\sqrt{d_k}}) \tag{3}$$

where $\sqrt{d_k}$ is the normalization coefficient; d_k is the dimension of $\mathbf{W}_q$ (and $\mathbf{W}_k$). Figure 5 illustrates the functioning of the attention mechanism.

In Figure 5, x_i denotes the contribution of UGV i, as given in

$$x_i = \sum_{j \neq i} \alpha_i v_j = \sum_{j \neq i} \alpha_i h(g_j(o_j, a_j)) \tag{4}$$

where f_L, g_L indicate neural networks. v_j can be denoted by g_j. h is the activation function. α_j can be expressed by

$$\alpha_j \propto exp((\mathbf{W}_k g_j(o_j, a_j))^T \mathbf{W}_q g_i(o_i, a_i)) \tag{5}$$

The strategy entropy is represented as $H(\pi)$, as given in

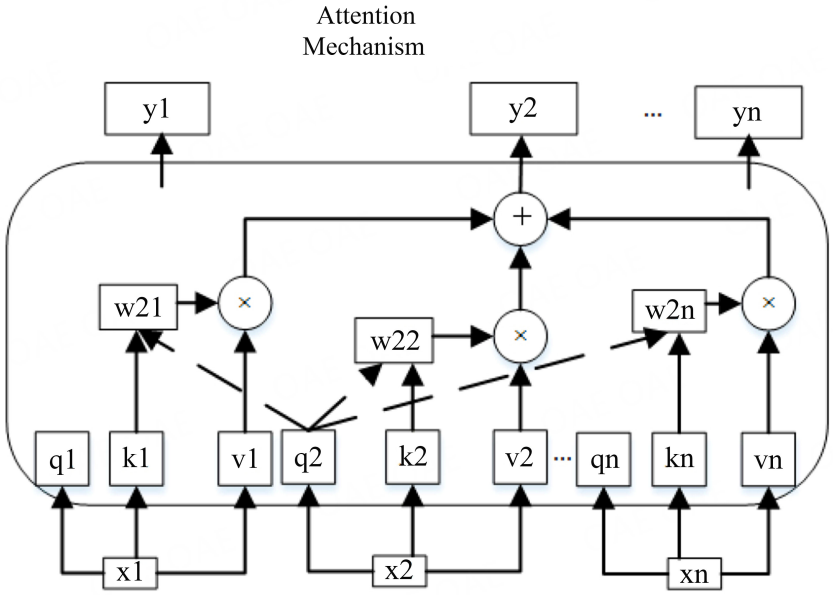

Figure 5. The working principle of the attention mechanism.

$$H(\pi) = -E_{\pi}[log(\pi)] \tag{6}$$

The policy of the SAC algorithm can be expressed by

$$\pi^{*} = arg\max_{\pi} E_{\tau\sim\pi}[\sum_{t=0}^{\infty}\gamma^{t}(r(s_t,a_t) + \alpha H(\pi(a_t,s_t)))] \tag{7}$$

This paper employs the attention mechanism in the Critic network of SAC to prioritize the features associated with the Q-value. By disregarding or minimizing less relevant states, the Critic network can enhance its learning capabilities and more efficiently approximate the value function. Consequently, this approach reduces computational complexity and accelerates the learning process. Modify the $Q - function$ of the SAC algorithm according to

$$J_Q(\theta) = E_{(s_t,a_t)\sim D}[\frac{1}{2}Q_{\theta}^{\psi}(s_t,a_t) - \hat{Q}(s_t,a_t))^2] \tag{8}$$

where $Q_{\theta}^{\psi}(s_t,a_t)$ represents the Q value obtained by the Critic network with attention mechanism when receiving states and actions. It may be computed using Equation (4). On the other hand, $\hat{Q}(s_t,a_t)$ refers to the Q value generated by the Target Critic network; D is the empirical cache. θ adopts gradient descent update, as defined in

$$\nabla J_Q(\theta) = \nabla_{\theta}Q_{\theta}(a_t,s_t) - r(s_t,a_t)(Q_{\theta}(s_t,a_t) - \gamma V_{\psi}(s_t)) \tag{9}$$

The strategy parameters are updated by minimizing the KL-divergence between the parametric strategy and the Boltzmann distribution, as derived in

$$J_\pi(\phi) = E_{s_t \sim D}[D_{KL}(\pi_\phi(\cdot|s_t)||\frac{exp(Q_\theta(s_t,\cdot))}{Z_\theta(s_t)})] \tag{10}$$

Z_θ is the distribution function of the standard distribution. The gradient of Actor can be expressed as

$$\nabla_\phi J_\pi(\phi) = \nabla_\phi log\pi_\phi(a_t|s_t) + (\nabla_{a_t} log\pi_\phi(a_t|s_t) - \nabla_{a_t} Q(s_t,a_t))\ \nabla_\phi\ f_\phi(\varepsilon_t;s_t) \tag{11}$$

The state variables can be expressed by

$$S = \left\{d_{P_1}, d_{P_2}, d_{P_3}, d_{12}, d_{13}, d_{23}, \theta_{1T2}, \theta_{1T3}, \theta_{2T3}, v_{T_1}, v_{T_2}, v_{T_3}, \theta_1, \theta_2, \theta_3, \theta_T\right\} \tag{12}$$

where d_{P_1}, d_{P_2}, d_{P_3} denote the distance between the capturing UGV and the target UGV; d_{12}, d_{13}, and d_{23} denote the distance between the three capturing UGVs; θ_{1T2}, θ_{1T3}, and θ_{2T3} denote the angle formed by the capturing UGV with its clockwise UGV and the target UGV; v_{T_1}, v_{T_2}, and v_{T_3} denote the difference between the speeds of the capturing UGV and the target UGV; θ_1, θ_2, θ_3, and θ_T denote the yaw angle of the capturing UGV and the target UGV. Action variables: they refer to the speed and angular velocity of the UGV during capturing, including $A = \{v_1, \omega_1, v_2, \omega_2, v_3, \omega_3\}$. The speed range is $[0m/s, 0.5m/s]$, and the angular velocity range is $[-1.0rad/s, 1.0rad/s]$.

In this research, we design an architecture for the Actor network that includes a feature input layer to receive un-normalized observations. The architecture consists of two fully connected layers with 256 and 128 neurons, respectively. The ReLU activation function is used in both layers. Within the branch responsible for calculating the average value of the action distribution, we implemented a fully connected layer consisting of 64 neurons. This was followed by the ReLU activation function and another completely connected layer with the same number of neurons as the action dimension. The purpose of this design was to represent the mean value of the action distribution. Furthermore, we incorporate a separate component for the standard deviation of the action distribution. This component comprises a fully connected layer with the identical number of neurons as the action dimension. It is then followed by a ReLU activation function and a softplus activation function, which guarantees that the standard deviation remains positive.

This research introduces a feature input layer in the Critic network to receive un-normalized observations. The layer consists of two fully connected layers with 256 and 128 neurons, respectively, and utilizes a ReLU activation function. The output of the action path is additionally linked to the output of the observation value path. Within the observation value path, two supplementary fully connected layers are incorporated. These layers consist of 128 and 64 neurons, respectively, and employ the ReLU activation function. Furthermore, an attention mechanism was incorporated by including a completely linked layer with 128 neurons and another fully connected layer with 64 neurons, both utilizing the ReLU activation function. The attention weights in relation to the attention mechanism were computed using a fully connected layer consisting of 16 neurons. Subsequently, the attention weights are multiplied with the observation path output to create weighted observations. The weighted observations are linked to the output of the action path and thereafter undergo processing via the fully connected layer. Ultimately, a completely linked layer with a single neuron is employed to represent the Q-value of the output. Table 1 displays the parameters associated with the algorithm. The training framework diagram, which is based on the algorithm developed in this paper, is depicted in Figure 6. The neural network architecture is depicted in Figure 7.

Table 1. Algorithm parameter

Neural Network/Training Parameter	Value
Entropy Weight /w	0.1
Target Smooth Factor/	5e-4
Experience Buffer Length/N	1e7
Policy Update Frequency/K	2
Critic Update Frequency/M	2

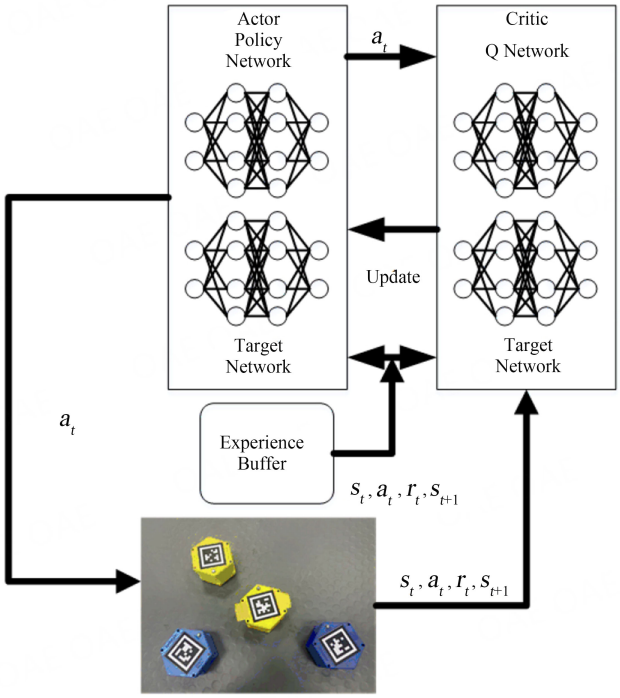

Figure 6. The algorithm training framework diagram proposed in this paper.

3.2. Configuration of the reward function

The selection of the reward function is crucial in the field of reinforcement learning. It has a direct impact on the performance and efficacy of the UGV during the learning phase. Hence, it is imperative to devise a rational incentive mechanism that can steer the autonomous vehicle toward the intended path of learning. This paper proposes a method of decoupling the reward function, which involves separating the reward function into individual and collaborative rewards. The goal is to maximize both the global reward and the local reward. The individual reward focuses on bringing multiple UGVs closer to the target UGV. The collaborative reward aims to minimize the action space of the target UGV by promoting cooperative behavior among the surrounding UGVs. The ultimate objective is to form an optimal encircling circle and capture the target UGV. The reward function is formulated in the following manner in this paper.

3.2.1. The reward function based on distance

During the pursuit process, the distance between the pursuing UGV and the target UGV should generally be less than the maximum successful capture distance yet greater than the collision distance. In the reinforcement learning training process, it is necessary to assign rewards and penalties to the pursuing UGV. Therefore, within this reward function, when the distance between the pursuing UGV and the target UGV is less than the maximum successful capture distance (set to 0.3m in this simulation), a positive reward is granted. Conversely, when this distance exceeds the maximum successful capture distance (i.e., 0.3m), a penalty is applied, and the larger the distance, the greater the penalty, as validated by

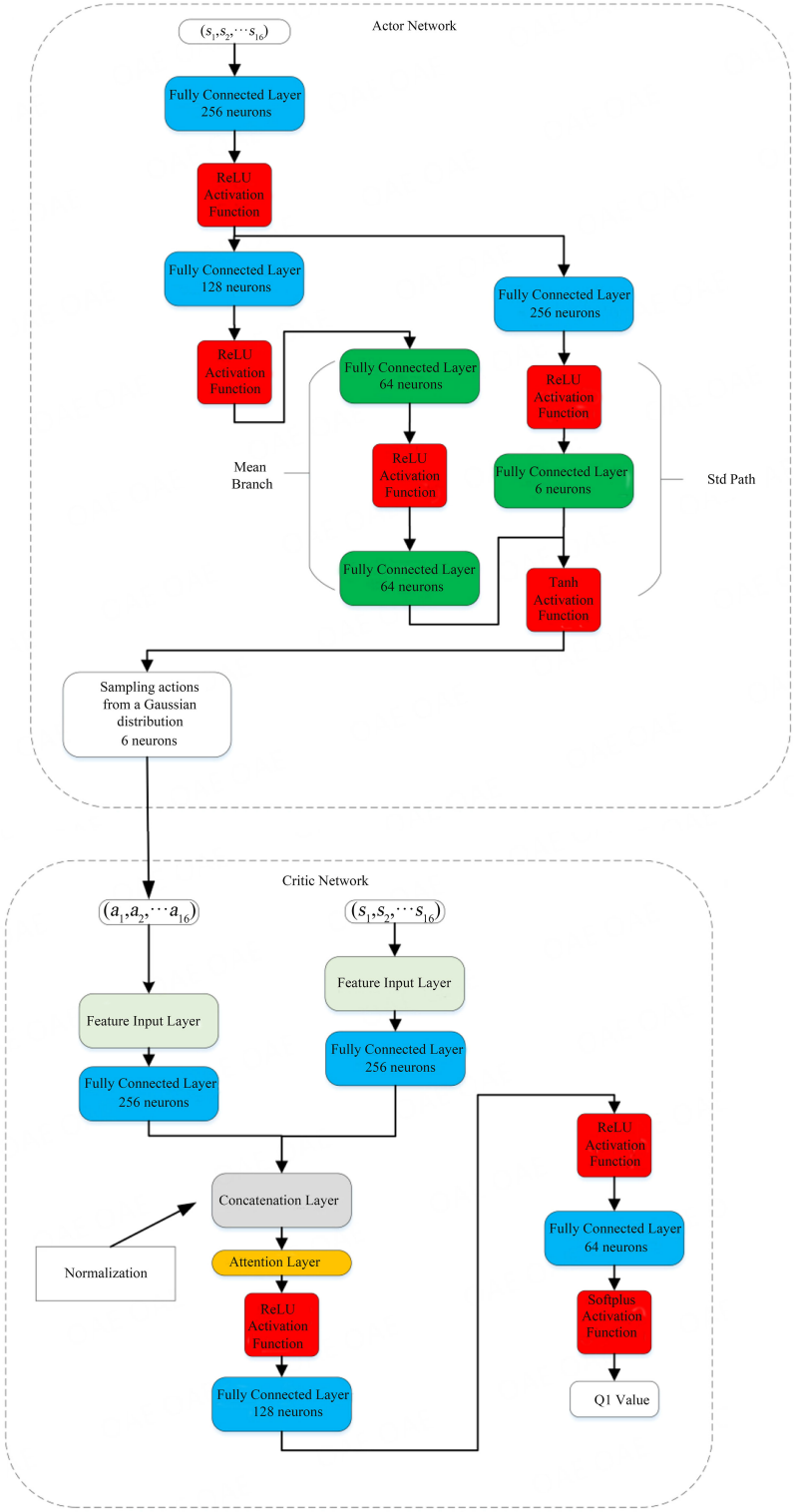

Figure 7. Algorithm neural network architecture diagram in this paper.

$$R_{P_i} = \begin{cases} -(d_{P_i} - 0.232) & d_{P_i} \le 0.232m \\ -(d_{P_i} - 0.232) \times (d_{P_i} - 0.3) & 0.232m \le d_{P_i} \le 0.3m \\ e^{d_{Pi}-0.3} - 1 & d_{P_i} > 0.3m \end{cases} \tag{13}$$

3.2.2. The reward function based on Angular

In order for the roundup to be successful, it is necessary to round up UGVs where the distance between the target UGV and the rounding UGV is less than 0.3m. Additionally, it is important to ensure that the rounding UGVs are evenly distributed around the target UGV, forming a circular arrangement. In this study, the desired number of UGVs for the roundup is 3. Therefore, the expected range of the rounding angle is as follows: $[110°, 130°]$. When the angle θ_{iTj} falls within the specified range, a reward of 1 is given; when the angle exceeds 130° or is less than 110°, a penalty is imposed, and the penalty rises as the angle increases. The angle reward function is defined by

$$R_{angle_i} = \begin{cases} -(\theta_{iTj} - 110)^2 & 0° \le \theta_{iTj} \le 110° \\ 1 & 110° \le \theta_{iTj} \le 130° \\ -(\theta_{iTj} - 130)^2 & \theta_{iTj} > 130° \end{cases} \tag{14}$$

3.2.3. Creating an optimal circular enclosure incentive

Typically, the proximity and orientation of the UGV can only indicate that the capturing UGV is near the target UGV, but they cannot serve as the primary criteria for the success of the enclosure. Due to the inherent randomness of the training process, it is not feasible to reliably determine the effectiveness of rounding based solely on the distance and angle parameters. Thus, using a rounding circle reward system can establish a distinct objective, enhance collaboration among the rounding UGVs, and facilitate their comprehension of the notion of capturing. More precisely, when a circular formation is created among UGVs for the purpose of rounding, supplementary incentives can be provided as the primary criterion for accomplishing successful rounding. This can enhance the collaboration between the capturing UGVs and raise the likelihood of successful capturing. Simultaneously, the reward function can incorporate the distance and angle between the rounding UGVs to holistically account for the collaborative synergy between the UGVs and effectively guide their training. The reward for forming the capture circle is defined by

$$R_{circle} = 30 \quad 0.232 < d_{P_i} < 0.3; 110° < \theta_{iTj} \le 130° (i, j \in 3; i \ne j); T \in \triangle P_1 P_2 P_3 \tag{15}$$

3.3. Penalty functions

A penalty is a punitive consequence employed to indicate to an UGV that a particular behavior is undesirable and should not be executed. Punishment can be administered to the UGV when it exhibits poor performance during roundups, such as making incorrect decisions, disregarding limitations, or causing unfavorable outcomes. Through the implementation of suitable penalties and halting of training, the UGV can acquire the ability to refrain from carrying out undesirable activities, hence enhancing its overall performance and efficacy. This document outlines the penalties imposed in the event of a collision involving an UGV when the vehicle deviates significantly from the target UGV or when it exceeds the designated boundaries. The penalty function is formulated in the following manner in this paper.

3.3.1. Collision penalty

This paper sets the collision distance to 0.232m. The distance between the capturing UGVs is determined as $d_{ij}(i \ne j)$ when the capturing UGV collides with the target UGV, resulting in punishment and training termination. Similarly, when there is a collision between the capturing UGVs, punishment is given and training is stopped. The collision punishment function can be defined by

$$P_{collision} = -10 \quad d_{P_i} \le 0.232; d_{ij} \le 0.232 \tag{16}$$

3.3.2. Distance penalty

In this research, we establish a condition where training is halted and a penalty is imposed when the distance between the capturing UGV and the target UGV exceeds 2m. The mathematical representation of this penalty is given by

$$P_{longdistance} = -10 \quad d_{P_i} \geq 0.232 \tag{17}$$

To enhance the UGV's learning and adaptability to the task while preventing it from getting stuck in an unproductive cycle or failing to finish the mission, a negative reward term of -0.1 is included in the reward function. The overall reward function can be defined by

$$Re = 0.6 \times \sum_{1}^{3} R_{P_i} + 0.4 \times \sum_{1}^{3} R_{Angle_i} + R_{circel} + P_{collision} + P_{longdistance} - 0.1 \tag{18}$$

4. EXPERIMENT AND ANALYSIS

The SAC algorithm is employed to train an UGV for capturing. The objective is for the UGV to move to a designated endpoint while following specific obstacle avoidance rules, thus creating a dynamic simulation scenario. During the training process, the UGV is given a reward value for successfully capturing, and the resulting effect diagram is analyzed to compare the impact of different reward functions on the training outcome. Once the training is completed, the trained strategy will be applied to a physical object for validation.

4.1. Simulation validation and analysis

This research introduces a borderless multi-UGV fencing environment in a two-dimensional continuous space. The experiment involves three fencing UGVs and one target UGV. The duration of each simulation step is 30 seconds, and there are a total of 300 steps every round. Figure 8 displays the starting position and initial heading angle of the UGVs. The blue UGV represents the target UGV, while the red ones represent the three rounding UGVs, namely UGVs 1, 2, and 3. The initial positions of the target UGV and the capturing UGVs are respectively given as (0.51m, 0.48m), (0.91m, 1.0m), (-0.12m, 0.52m), and (0.0m, 0.0m).

Figure 9 displays the average reward for 20,000 training rounds. The blue curve represents the average reward function curve obtained by combining the reward function proposed in this paper with the SAC algorithm. On the other hand, the yellow curve demonstrates the average reward function obtained by using the traditional reward function combined with the TD3 algorithm. Upon comparing the proposed algorithm and the SAC algorithm with the proposed reward function to the TD3 method, it is evident that the proposed one exhibits superior exploratory capabilities. It effectively guides the UGV to consistently explore new policies while maintaining stability and comparable exploratory performance to the SAC algorithm without the enhancement. Furthermore, the average reward achieved by the suggested algorithm, utilizing the reward function outlined in this research, surpasses that of both the SAC algorithm without any enhancements using the conventional reward function and the TD3 method employing the reward function devised in this paper. Furthermore, the reward function curve of the suggested algorithm can achieve stability at a faster rate compared to the SAC algorithm.

In order to further verify the performance of the algorithm provided in this work, the SAC algorithm and the method proposed in this paper are simulated and compared. The trajectory diagram produced by the suggested method is depicted in Figure 10. The SAC algorithm generates a trajectory diagram, which is depicted in Figure 11. Based on Figure 10 and Figure 11, it is clear that the three Pursuing UGVs move closer to the target

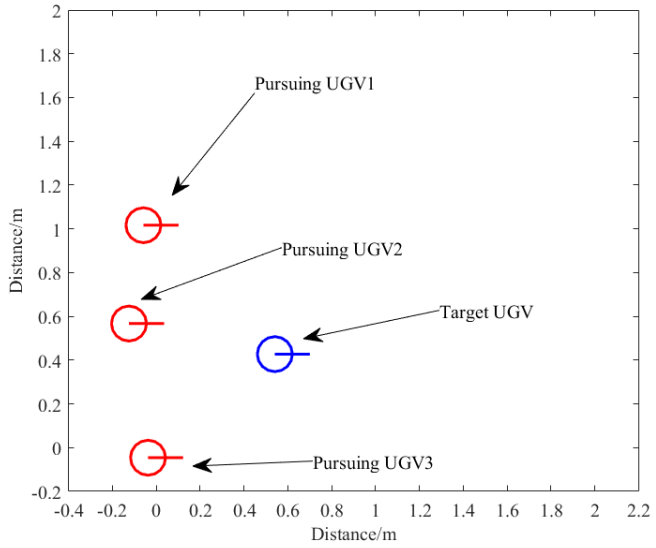

Figure 8. Initial UGV position.

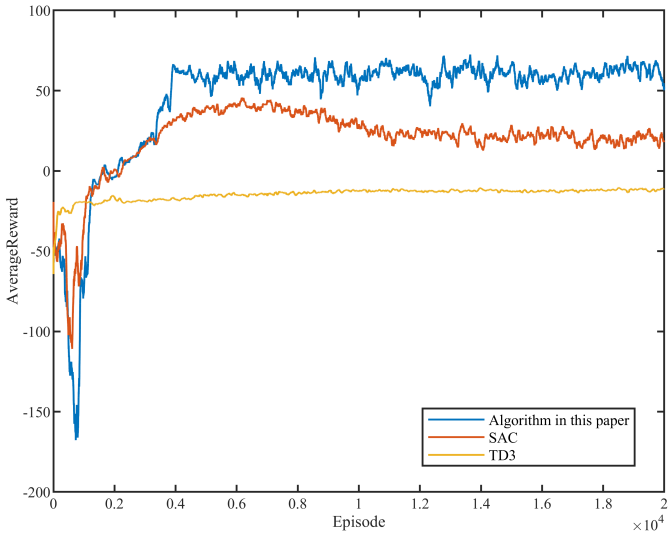

Figure 9. Average reward curve during the training process.

UGV from their starting positions. Pursuing UGVs 2, 3, and 4 effectively encircle the target UGV from three different directions: above, to the left, and below. The black solid line represents the triangular area formed by Pursuing UGVs 1, 2, and 3. During the progression of the roundup, the three Pursuing UGVs gradually move closer to the target UGV and create a circular formation for the roundup. This circular formation, similar to the triangular shape depicted in the algorithm provided in Figure 10, is smaller, thus capable of constraining the movement of the target UGV. Furthermore, the trajectories followed by the UGVs are depicted in Figure 11. The UGVs utilizing the algorithm presented in this study exhibit a higher rate of change and a reduced turning amplitude, suggesting that they possess greater responsiveness and the ability to make strategic decisions with increased speed.

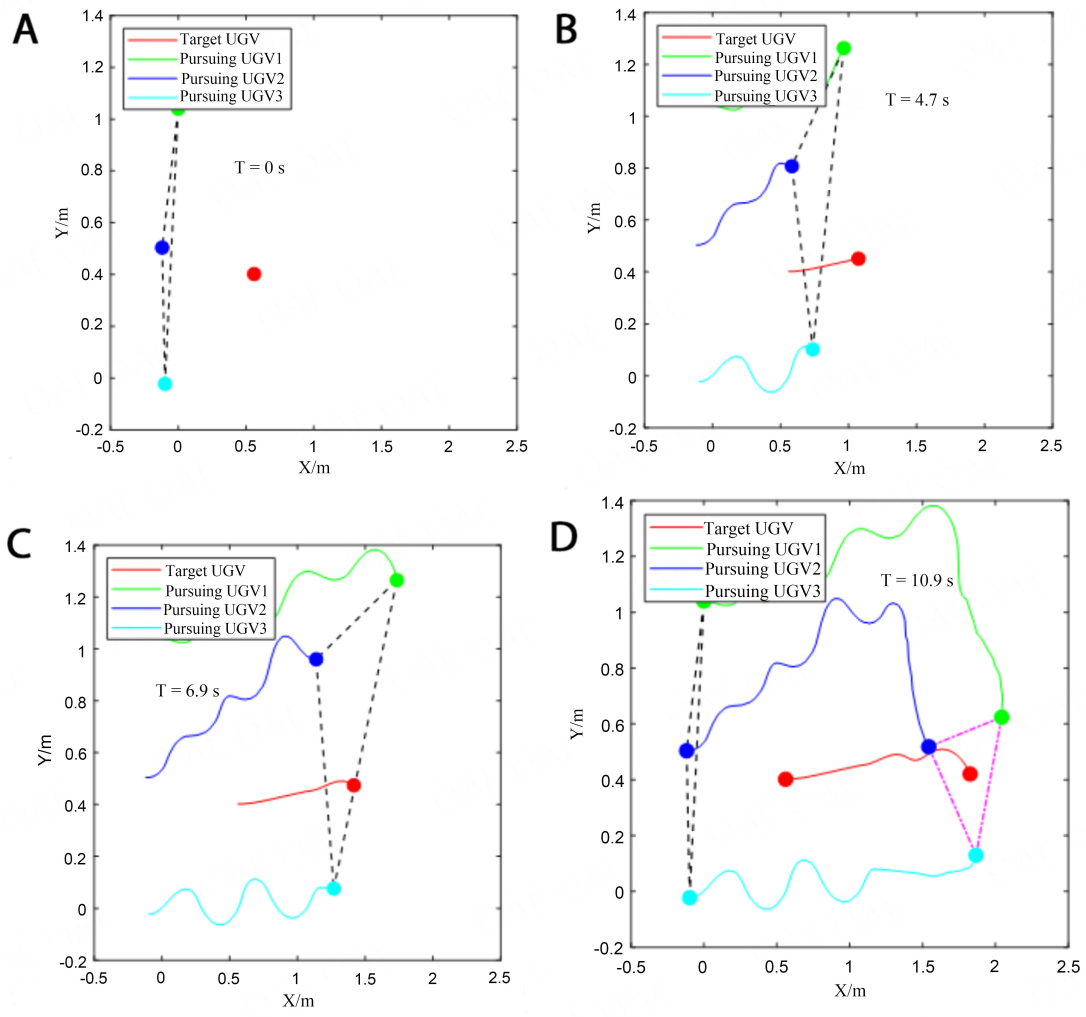

Figure 10. trajectory maps of pursuit for the Algorithm: (A) Encirclement Trajectories at 0s; (B) Encirclement Trajectories at 4.7s; (C) Encirclement Trajectories at 6.9s; (D) Encirclement Trajectories at 10.9s.

Table 2. Different algorithm experimental results

Algorithm	The average distance/m	Consumption Time/s	Capture Formation Time/s	Success rate
Traditional	3.81	20.23	8.9	66%
SAC	3.59	11.7	7.4	83%
Ours	3.28	10.8	7.0	92%

Figure 12 illustrates the change in distance between the captured UGV and the target UGV. It is evident that as the capturing process progresses, the distance gradually decreases and converges to 0.3m. The proposed algorithm in this paper results in a faster reduction in distance compared to the SAC algorithm, ultimately achieving a smaller distance between the captured UGV and the target UGV. Figure 13 illustrates the variation in angles between the UGV and its adjacent UGVs during the process of capturing the UGVs. The figure demonstrates that the suggested method guarantees a uniform distribution of surrounding UGVs around the target UGVs, forming a circular enclosure. This is in contrast to the SAC approach. To assess the stability and generalizability of the proposed approach, we conducted 100 experiments. The results, comparing the performance of various methods in the task of capturing, are presented in Table 2. The initial speed parameters of the four UGVs are identical, and their speed ranges are within a specified range.

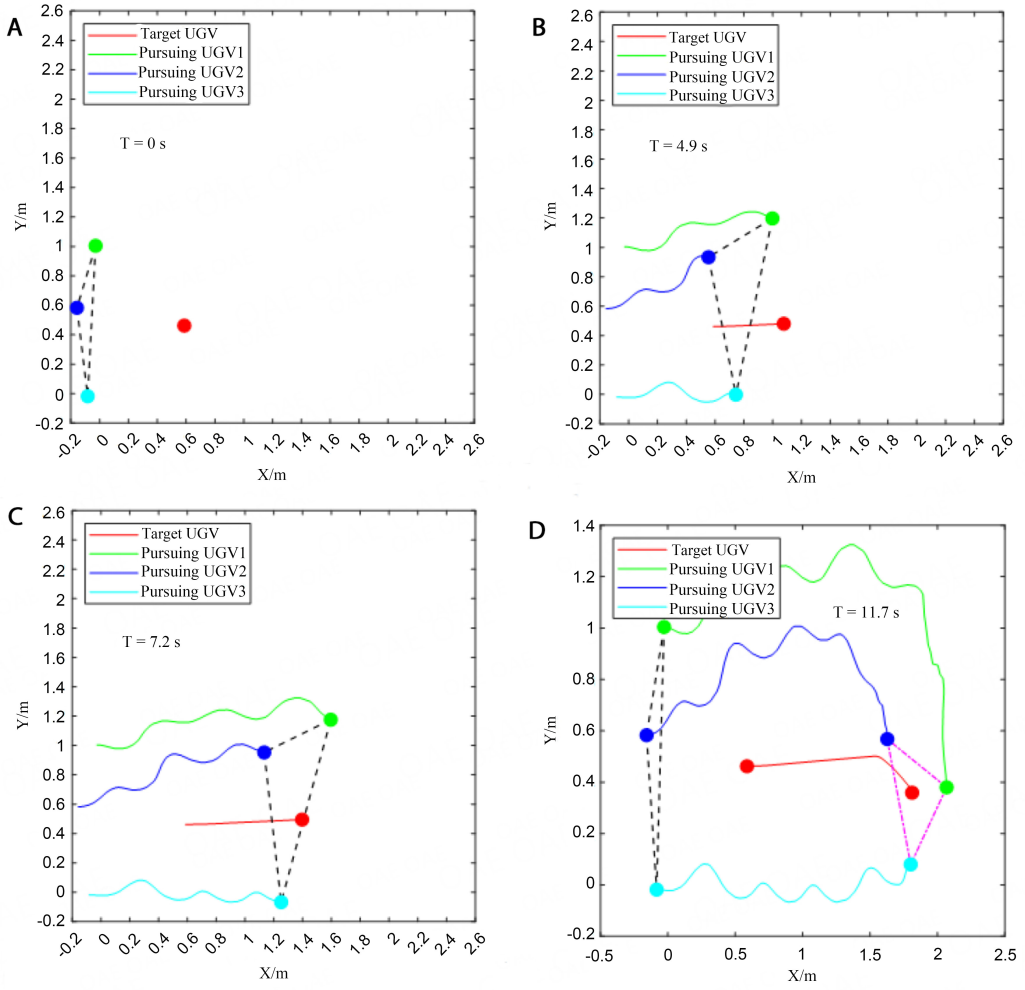

Figure 11. Trajectory maps of pursuit for SAC: (A) Encirclement Trajectories at 0 s; (B) Encirclement Trajectories at 4.9 s; (C) Encirclement Trajectories at 7.2 s; (D) Encirclement Trajectories at 11.7 s.

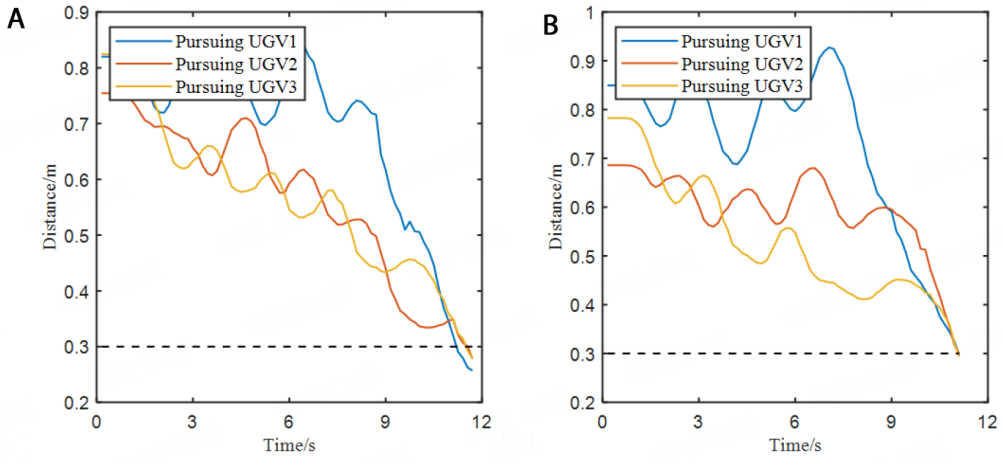

Figure 12. The distance change curve between the pursuing vehicle and the target vehicles: (A) the distance change curve between the pursuing vehicle and the target vehicles by SAC; (B) the distance change curve between the pursuing vehicle and the target vehicles by the algorithm in this paper.

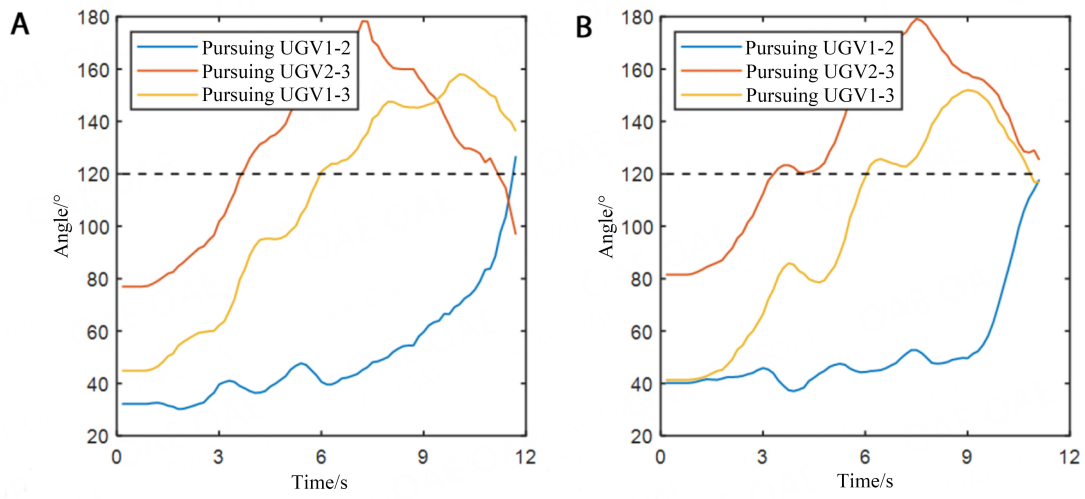

Figure 13. The angle change curve between the pursuing vehicle and its adjacent vehicles: (A) the absolute angle change curve between the pursuing vehicle and its adjacent vehicles by SAC; (B) the absolute angle change curve between the pursuing vehicle and its adjacent vehicles by the algorithm in this paper.

Table 3. Specifications of the UGV

Type	Parameters
Shape and size(mm)	119.75 ×105.01×79.07
Wheelbase(mm)	84.67
Wheel diameter(mm)	60.5
Drive mode	Dual-wheel differential
Speed range(m/s)	0-0.5

4.2. Physical experiment verification and analysis

In order to confirm the practicality and implementation of the method described in this research, the collaborative fencing algorithm suggested is used for a multi-UGV experimental platform. The UGV platform comprises XX, and the positioning system uses the positioning camera to acquire data on the UGV's position, velocity, and azimuth by identifying the QR code on the vehicle's body. Table 3 displays the specifications of the UGV.

Figure 14 depicts the footage captured by the visual positioning camera during the encirclement and capture process. It is evident from the images that the three capturing UGVs initiate movement from their initial position and successfully encircle the target UGVs, ultimately achieving the objective of rounding them up.

Figure 15 depicts the trajectory of UGVs during the cooperative capture process, implemented with the improved algorithm proposed in this paper. From the figure, it is evident that the three capturing vehicles start from their initial positions and move towards the target unmanned vehicle to accomplish the capture task ultimately. The black solid line depicts the extent of the triangle formed by UGVs 1, 2, and 3. From the graphic, it is evident that as the roundup advances, the three Pursuing UGVs progressively converge toward the target UGV and establish a circular formation. The precise coordinates of the UGVs, after capturing, are as follows: UGV 1 at (1.19, 1.85), UGV 2 at (1.05, 2.11), UGV 3 at (0.96, 1.66), and UGV 4 at (1.50, 1.88). The distances between the target UGV and the surrounding UGVs 1, 2, and 3 can be calculated to be 0.29 m, 0.29 m, and 0.30 m, respectively. These distances are all less than or equal to the rounding distance of 0.3 m. The angles formed between UGV 1 and the target UGV, UGV 2 and the target UGV, and UGV 3 and the target UGV are 124.12, 118.32, and 117.56 degrees, respectively. All of these angles fall within the specified range.

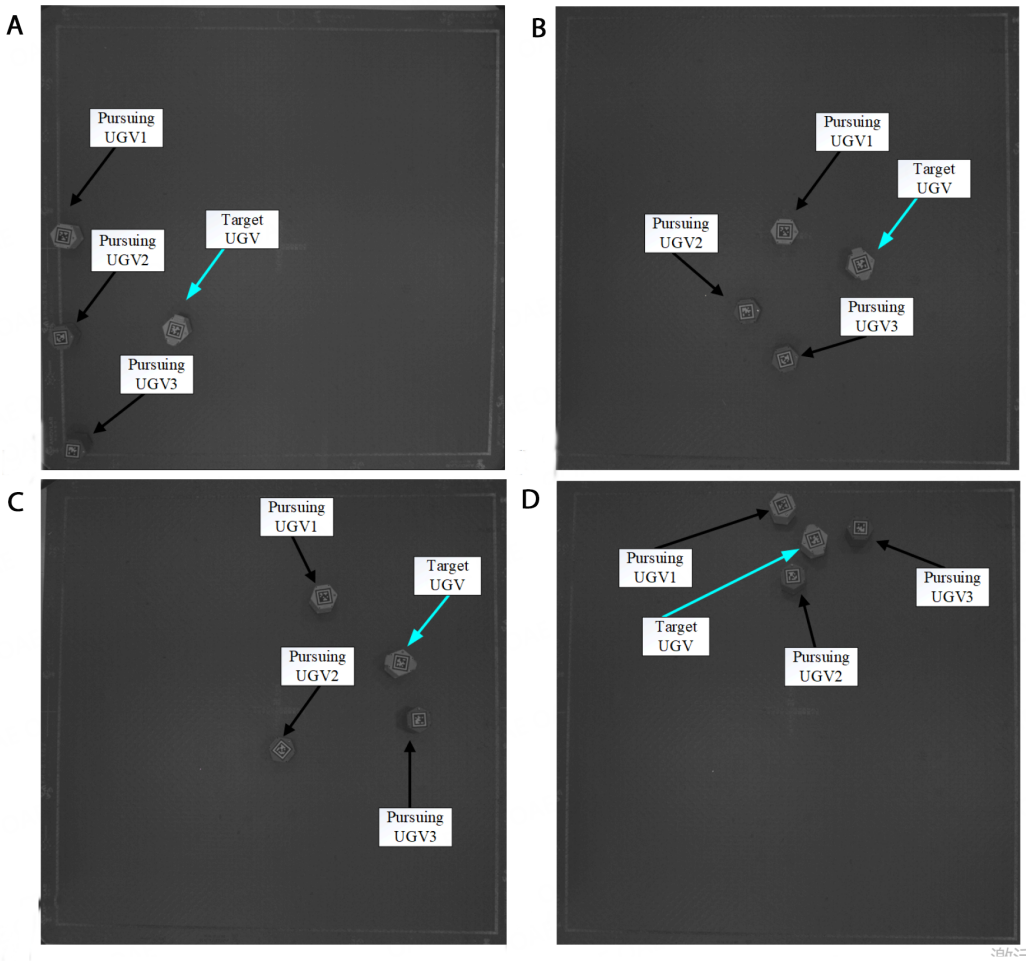

Figure 14. Footage of the trapping process using visual positioning cameras: (A) initial trapping location; (B) Trapping process location at 8 seconds; (C) Successful trapping location at 12 seconds; (D) Successful trapping location at 15.4 seconds.

Figure 16 illustrates the curve depicting the change in distance and angle of the target UGV during the capturing process. The figure shows that as the capturing process progresses, the distance between the capturing UGV and the target UGV gradually decreases and converges to 0.3m. However, the distance between the capturing UGV 2 and the target vehicle is greater than 0.3m, with a difference of 0.01m from the expected value of 0.3m, which may be due to the size of the physical objects and the delay in information transmission. This figure illustrates that the relative angles of the two fence UGVs are approaching 120 degrees. This demonstrates the algorithm's stability and capacity to migrate.

5. CONCLUSIONS

The paper addresses the issue of cooperative capture among unmanned vehicles, considering practical task requirements and constraints, and establishes a kinematic model while defining conditions for direct capture success. Within the SAC framework, an attention mechanism is integrated into the Critic network to tackle the issue of network instability caused by high dimensions, constructing a training framework. To align with task requirements, state and action spaces of reinforcement learning are optimized, introducing a reward function strategy that combines individual and collaborative rewards. This reward function is divided into two

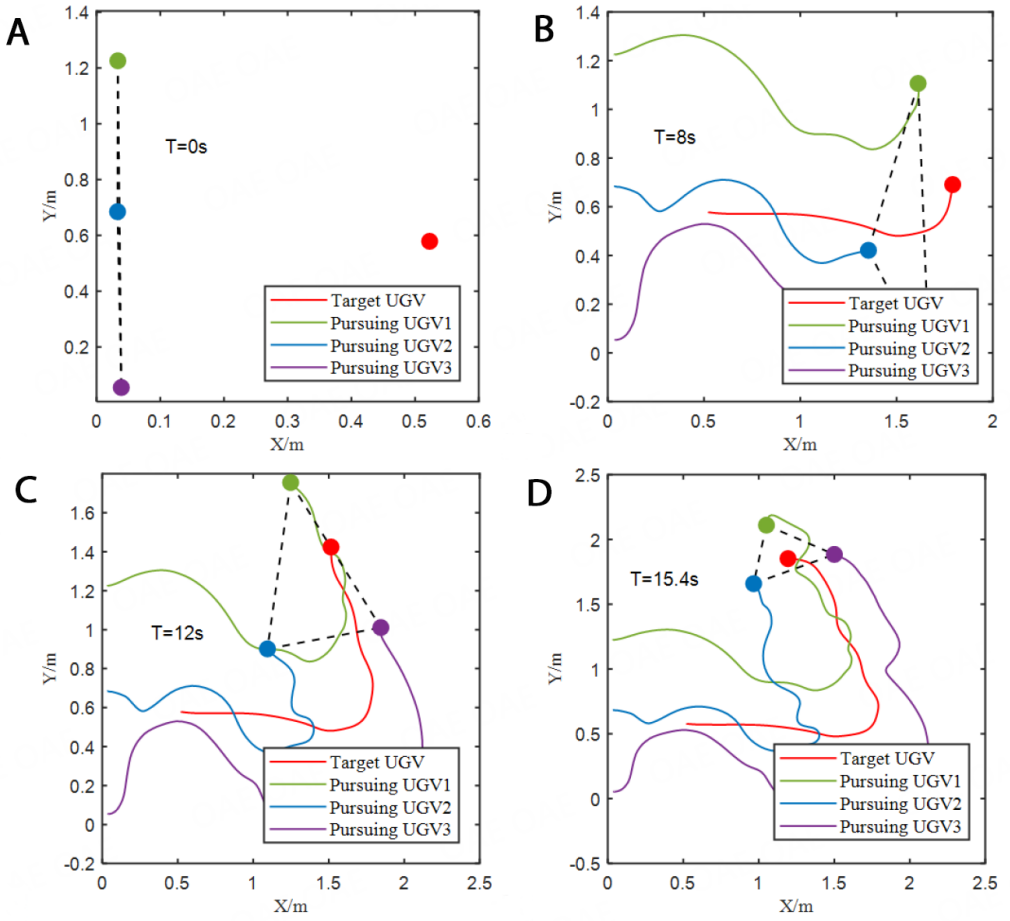

Figure 15. Trajectory maps of pursuit for the Algorithm: (A) Encirclement Trajectories at 4 seconds; (B) Encirclement Trajectories at 8 seconds.; (C) Encirclement Trajectories at 12 seconds; (D) Encirclement Trajectories at 15.4 seconds.

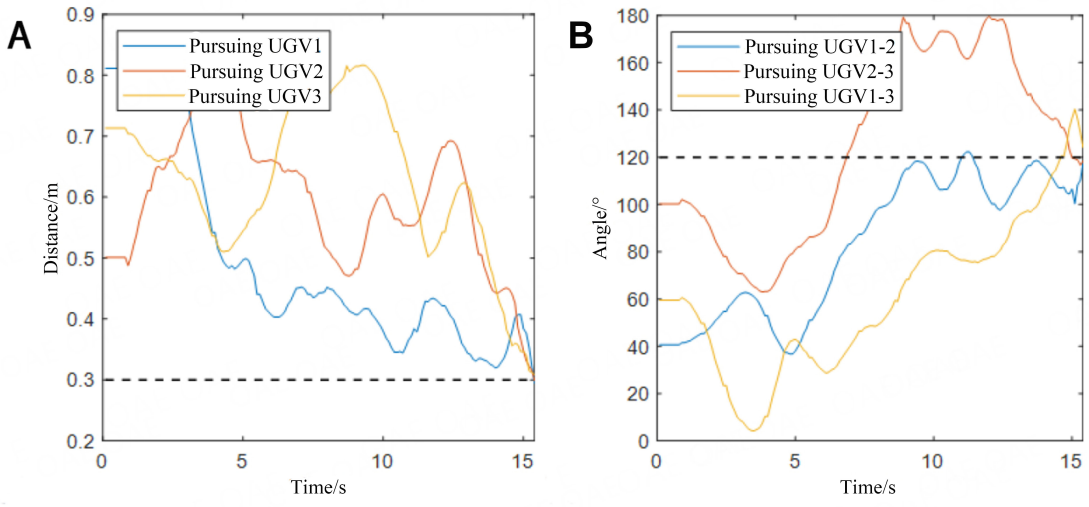

Figure 16. The distance and angle change curve between the pursuing vehicle and the target vehicle:(A) the distance change curve between the pursuing vehicle and the target vehicle; (B) the absolute angle change curve between the pursuing vehicle and its adjacent UGVs.

components: firstly, continuous individual rewards prompt vehicles to swiftly approach the target, facilitating effective capture and enhancing model training efficiency. Secondly, sparse collaborative rewards guide

mutual cooperation among vehicles, culminating in a formation for capture and task achievement. Finally, a simulation environment is developed and utilized to train the cooperative capture strategy. Simulation experiments demonstrate that the proposed algorithm, in comparison to SAC, exhibits higher average rewards during training, faster convergence rates, and an ability to swiftly form a cooperative encirclement posture and accomplish cooperative capture in the context of unmanned vehicle cooperative capture tasks. Moreover, it displays shorter vehicle travel paths, reduced capture time, and higher capture success rates. Compared to the SAC algorithm, it demonstrates a 10.8 % enhancement in task success rates and a 7.6% reduction in capture time.

DECLARATIONS

Authors' contributions

Made substantial contributions to the conception and design of the study and performed data analysis and interpretation: Su M, Wang Y

Performed data acquisition and provided administrative, technical, and material support: Pu R, Yu M

Availability of data and materials

Not applicable.

Financial support and sponsorship

None.

Conflicts of interest

All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

REFERENCES

1. Garcia E, Casbeer DW, Von Moll A, Pachter M. Multiple pursuer multiple evader differential games. *IEEE Trans Automat Contr* 2020;66:2345–50. DOI
2. Yu D, Chen CLP. Smooth transition in communication for swarm control with formation change. *IEEE Trans Ind Inf* 2020;16:6962–71. DOI
3. Camci E, Kayacan E. Game of drones: UAV pursuit-evasion game with type-2 fuzzy logic controllers tuned by reinforcement learning. In: 2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). IEEE; 2016. pp. 618–25. DOI
4. Vidal R, Rashid S, Sharp C, et al. Pursuit-evasion games with unmanned ground and aerial vehicles. In: Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation. IEEE; 2001. pp. 2948–55. DOI
5. Turetsky V, Shima T. Target evasion from a missile performing multiple switches in guidance law. *J Guid Control Dyn* 2016;39:2364–73. DOI
6. de Souza C, Newbury R, Cosgun A, Castillo P, Vidolov B, Kulić D. Decentralized multi-agent pursuit using deep reinforcement learning. *IEEE Robot Autom Lett* 2021;6:4552–59. DOI
7. Lopez VG, Lewis FL, Wan Y, Sanchez EN, Fan L. Solutions for multiagent pursuit-evasion games on communication graphs: Finite-time capture and asymptotic behaviors. *IEEE Trans Automat Contr* 2019;65:1911–23. DOI
8. Dong J, Zhang X, Jia X. Strategies of pursuit-evasion game based on improved potential field and differential game theory for mobile robots. In: 2012 Second International Conference on Instrumentation, Measurement, Computer, Communication and Control. IEEE; 2012. pp. 1452–56. DOI

9. Sun Q, Chen Z, Qi N, Lin H. Pursuit and evasion conflict for three players based on differential game theory. In: 2017 29th Chinese Control And Decision Conference (CCDC). IEEE; 2017. pp. 4527–31. DOI
10. Haslegrave J. An evasion game on a graph. *Discrete Math* 2014;314:1–5. DOI
11. Kehagias A, Hollinger G, Singh S. A graph search algorithm for indoor pursuit/evasion. *Math Comput Modell* 2009;50:1305–17. DOI
12. Janosov M, Virágh C, Vásárhelyi G, Vicsek T. Group chasing tactics: how to catch a faster prey. *New J Phys* 2017;19:053003. DOI
13. Wang J, Li G, Liang L, Wang D, Wang C. A pursuit-evasion problem of multiple pursuers from the biological-inspired perspective. In: 2021 40th Chinese Control Conference (CCC). IEEE; 2021. pp. 1596–601. DOI
14. Qu X, Gan W, Song D, Zhou L. Pursuit-evasion game strategy of USV based on deep reinforcement learning in complex multi-obstacle environment. *Ocean Eng* 2023;273:114016. DOI
15. Bilgin AT, Kadioglu-Urtis E. An approach to multi-agent pursuit evasion games using reinforcement learning. In: 2015 International Conference on Advanced Robotics (ICAR). IEEE; 2015. pp. 164–69. DOI
16. Wang Y, Dong L, Sun C. Cooperative control for multi-player pursuit-evasion games with reinforcement learning. *Neurocomputing* 2020;412:101–14. DOI
17. Du W, Guo T, Chen J, Li B, Zhu G, Cao X. Cooperative pursuit of unauthorized UAVs in urban airspace via Multi-agent reinforcement learning. *Trans Res Part C Emerg Technol* 2021;128:103122. DOI
18. Zhang Z, Wang X, Zhang Q, Hu T. Multi-robot cooperative pursuit via potential field-enhanced reinforcement learning. In: 2022 International Conference on Robotics and Automation (ICRA). IEEE; 2022. pp. 8808–14. DOI
19. Hüttenrauch M, Šošić A, Neumann G. Deep reinforcement learning for swarm systems. *J Mach Learn Res* 2019;20:1–31. Available from: https://www.jmlr.org/papers/volume20/18-476/18-476.pdf. [Last accessed on 13 March 2024]
20. Liu S, Hu X, Dong K. Adaptive double fuzzy systems based Q-learning for pursuit-evasion game. *IFAC-PapersOnLine* 2022;55:251–56. DOI
21. Zhang J, Zhang K, Zhang Y, Shi H, Tang L, Li M. Near-optimal interception strategy for orbital pursuit-evasion using deep reinforcement learning. *Acta Astronaut* 2022;198:9–25. DOI
22. dos Santos RF, Ramachandran RK, Vieira MAM, Sukhatme GS. Parallel multi-speed Pursuit-Evasion Game algorithms. *Robot Auton Syst* 2023;163:104382. DOI
23. Wang S, Wang B, Han Z, Lin Z. Local sensing based multi-agent pursuit-evasion with deep reinforcement learning. In: 2022 China Automation Congress (CAC). IEEE; 2022. pp. 6748–52. DOI
24. Liu Z, Qiu C, Zhang Z. Soft-actor-attention-critic based on unknown agent action prediction for multi-agent collaborative confrontation. In: 2023 3rd International Conference on Neural Networks, Information and Communication Engineering (NNICE). IEEE; 2023. pp. 527–35. DOI
25. Zhang M, Yan C, Dai W, Xiang X, Low KH. Tactical conflict resolution in urban airspace for unmanned aerial vehicles operations using attention-based deep reinforcement learning. *Green Energy Intell Trans* 2023;2:100107. DOI
26. Peng Y, Tan G, Si H, Li J. DRL-GAT-SA: Deep reinforcement learning for autonomous driving planning based on graph attention networks and simplex architecture. *J Syst Archit* 2022;126:102505. DOI
27. Kim Y, Singh T. Energy-time optimal control of wheeled mobile robots. *J Franklin Inst* 2022;359:5354–84. DOI
28. Kathirgamanathan A, Mangina E, Finn DP. Development of a soft actor critic deep reinforcement learning approach for harnessing energy flexibility in a large office building. *Energy and AI* 2021;5:100101. DOI