



Revolutionary entrapment model of uniformly distributed swarm robots in morphogenetic formation

Chen Wang^a, Zhaohui Shi^a, Minqiang Gu^a, Weicheng Luo^a, Xiaomin Zhu^b, Zhun Fan^{a, c, *}

^a Shantou University, Shantou, Guangdong, China

^b National University of Defense Technology, Changsha, China

^c Key Lab of Digital Signal and Image Processing of Guangdong Province, China

ARTICLE INFO

Article history:

Received 19 April 2022

Received in revised form

13 July 2022

Accepted 22 August 2022

Available online xxx

Keywords:

Swarm intelligence

Revolutionary entrapment

Flocking

Robots

Gene regulatory network

Vicsek-model

Entrapping multiple targets

ABSTRACT

This study proposes a method for uniformly revolving swarm robots to entrap multiple targets, which is based on a gene regulatory network, an adaptive decision mechanism, and an improved Vicsek-model. Using the gene regulatory network method, the robots can generate entrapping patterns according to the environmental input, including the positions of the targets and obstacles. Next, an adaptive decision mechanism is proposed, allowing each robot to choose the most well-adapted capture point on the pattern, based on its environment. The robots employ an improved Vicsek-model to maneuver to the planned capture point smoothly, without colliding with other robots or obstacles. The proposed decision mechanism, combined with the improved Vicsek-model, can form a uniform entrapment shape and create a revolving effect around targets while entrapping them. This study also enables swarm robots, with an adaptive pattern formation, to entrap multiple targets in complex environments. Swarm robots can be deployed in the military field of unmanned aerial vehicles' (UAVs) entrapping multiple targets. Simulation experiments demonstrate the feasibility and superiority of the proposed gene regulatory network method.

© 2022 China Ordnance Society. Publishing services by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

The development of swarm robot system is a challenging endeavor in robotics [1]. Swarm intelligence is based on the concept that individuals, through cooperative interaction, produce collective intelligent behaviors [2,3]. Extensive research on swarm robotics has been conducted in recent years [4–9]. Swarm intelligence has various applications, including industrial manufacturing [10], warehouse logistics [11], disaster response [12], scenario reconstruction [13], and military surveillance [14].

Entrapment is one of the main directions of swarm intelligence research. Entrapment methods are primarily classified as follows: behavior-based approach [15,16], leader–follower model [17,18], virtual structure method [19,20], and biomimetic approach

[21–25]. The key idea of the behavior-based approach is the design of various basic behaviors and an effective behavior coordination mechanism. Basic behaviors include robot collision avoidance, obstacle avoidance, and formation maintenance [15,16]. This method is suitable mostly for simple tasks. For the leader–follower method, an agent in the system is designated as the leader, and its trajectory is controlled [17,18]. The individual agents follow the leader. Meanwhile, the following agents maintain a specific geometric relationship with the ones leading. However, this method cannot enable the agents to generate an adaptive formation suited to the environment according to the environmental information. The virtual structure method considers the system to be a rigid body virtual structure, and each member controls its own behavior according to its position relative to that structure. The process of this method is as follows: (1) The desired dynamic behavior of the virtual structure is specified; (2) The motion of the virtual structure is translated into the desired motion of each robot; and (3) each robot assumes the expected behavior [19,20]. However, when agents are in a complex environment with obstacles, a fixed shape

* Corresponding author. Department of Electronic and Information Engineering, Shantou University, 243 Daxue Road, Shantou, Guangdong, China.
E-mail address: zfan@stu.edu.cn (Z. Fan).

Peer review under responsibility of China Ordnance Society

<https://doi.org/10.1016/j.dt.2022.08.015>

2214-9147/© 2022 China Ordnance Society. Publishing services by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

cannot cope with the environment flexibly. Wang et al. proposed an adaptive algorithm named AGENT for entrapping multiple targets based on Vicsek-model [21]. This method enables swarm robots to avoid obstacles flexibly, and uses adaptive group division for entrapping multiple targets.

Another interesting line is the method based on biological inspiration. The gene regulatory network (GRN) is a robust self-organizing mechanism for simulating biological morphogenetic development. Recently, the GRN method has been used in multi-robot entrapping tasks [22–25]. Jin et al. [26] proposed a hierarchical GRN (H-GRN) that enabled the agents to generate an entrapping formation according to the environmental changes. Oh et al. [27,28] proposed an evolving hierarchical GRN (EH-GRN) for swarm robots to entrap targets in environment with obstacles. Peng et al. [29] proposed a pattern adaptation strategy where robots entrapped each target separately with split sub-patterns. Wu et al. [30] introduced a cooperation-based GRN method assuming that the agent can use sensors to receive information about its neighbors and record it in its own coordinate system. The partner's position was used as input to trigger changes in pattern formation. Building on GRN-based research, Fan et al. [31] proposed an automated design framework, including structures and parameters, with the genetic programming method for automatically generating entrapment patterns. The framework generated GRN models exhibiting better performance compared with those designed by human experts in complex and dynamic environments yet with simpler GRN structures.

Although the previous GRN method provides an effective pattern generation mechanism for robot swarm entrapment, it still has some areas to improve. For example, it lacks an efficient swarm decision-making mechanism that can realize a uniform pattern distribution of robots for entrapping multiple targets. In addition, the robot while entrapping targets may repeatedly jump forward near the generated pattern and have a non-uniform position distribution relative to the pattern, which desires for a better velocity control mechanism. In this paper, we propose a multiple-target uniformly revolving entrapment GRN (RE-GRN) algorithm with a decision-making mechanism and velocity control mechanism to solve these two problems.

The primary contributions of the study are as follows:

- The structure of the RE-GRN method is divided into three layers: Entrapment pattern generation layer, decision-making layer, and motion control layer. This framework can be used to extend the execution of various swarm tasks by modifying the relevant layer.
- A new mechanism to allocate captures points to entrapping patterns is proposed. The number of capture points corresponds to the number of robots and targets. This approach improves the uniformity of the division of robot groups when entrapping multiple targets.
- A decision-making mechanism for each robot in the swarm is proposed that allows it to choose the most suitable capture point in the pattern that revolves about the pattern as the target moves, facilitating more efficient and effective entrapment.
- An improved Vicsek-model to achieve smooth motion of the robots towards their target points mimicking natural population motion is proposed. With this method, the robot can adaptively and smoothly travel through complex obstacles to reach the target point according to the environmental conditions.

To the best of our knowledge, this is the first study where the agents can revolve around targets along the changing entrapping pattern and entrap the targets. It can make the robot maintain a relatively higher entrapping velocity and constantly change the entrapping position, so that the target cannot obtain a static entrapping formation gap, which prevents the target from escaping

through the gap of the entrapping formation and enable the robot swarm to achieve tighter and more flexible entrapment [33–35]. Furthermore, if the swarm robots keep relatively static with the target, they will not have the speed advantage when the target wants to escape. In this case, the speed of the swarm robots revolving around the target will enable them to respond more quickly when the target attempts to escape, which reduces the requirements for the robot to accelerate to a certain speed. Moreover, orbiting the target enables the sensors carried by the swarm robots, such as visual sensors, play a greater role. If other targets appear, they are more easily detected by the patrolling robots, and the formation can be adjusted more responsively when robots have a certain starting speed. In conclusion, the entrapping mechanism proposed in this study is beneficial and worth to be further investigated.

The rest of the paper is organized as follows. Section 2 presents the research question and summarizes the methodology. Section 3 proposes the methods, including the difference equation of the RE-GRN model suitable for robotics to generate entrapping morphologies, the selection criterion of the robot's target points on the entrapment pattern to achieve a uniform distribution, and the maneuvering of the robots to the target point. Section 4 compares the RE-GRN method with the previous GRN as well as AGENT methods, with comprehensive experimental results analyzed. Section 5 summarizes the study's findings, limitations, and future research directions.

2. Problem statement

Gene regulation is the mechanism by which organisms control gene expression. The GRN method is mainly divided into three aspects. First, regulation at the DNA level, transcriptional control, and translational control. Second, microbes adapt their metabolism to the environment through the GRN. Third, multicellular organisms enable cell differentiation, morphogenesis, and ontogeny through biological gene regulation. This study mainly draws on the mechanism by which microorganisms adapt to their environment by changing their metabolism through gene regulation. Microorganisms exhibit a high degree of adaptability to environmental conditions during development and rapidly regulate the expression levels of various genes in response to environmental stimuli.

Previous studies [21–28] utilized GRNs and morphological gradients to realize the morphogenesis of swarm robotic systems. The fundamental idea of applying gene expression mechanisms in biological morphogenesis to swarm robotic control is to establish an analogy between microbes and robots. Microorganisms regulate their own protein production based on environmental information. Changes in protein concentration affect the environmental information received by microorganisms, affecting the microorganisms in return. Microorganisms undergo this regulatory process to better adapt to the environment. The behavior pattern of this type of microorganism is analogous to that of a robot. A robot also establishes the concentration field of the target and obstacles information and adjusts its movement through its own concentration field accordingly, which, to a certain extent, reflects the environmental adaptability of the robot.

In this study, the GRN was used to coordinate a swarm of robots to entrap multiple targets. As illustrated in Fig. 1, the algorithm has a three-layer structure: An upper layer, a transition layer, and a lower layer. The robot maps the target and obstacle position information in the upper layer to generate the intermediate products g_1 , g_2 , g_3 and . These products represent concentration maps with different levels of information about the target and the obstacles (e.g., g_1 is the robot considering the concentration information required to fly close to the target, g_2 is the concentration

information for the robot to fly away from the obstacle, g_3 simultaneously couples the concentration information of the robot flying away from the obstacle and close to the target). The robot generates a concentration information map that adapts to the current environment by regulating g_1 , g_2 , and g_3 . The mechanisms of g_1 , g_2 , and g_3 (including the logical structure and parameters) is automatically evolved by genetic programming. The robot samples the concentration field with a specific concentration value to take the target surrounding form M which considers both the obstacle and target information in the robot's environment.

The second layer is the transition layer. Based on the two-layer structure of the original GRN, we constructed the transition layer for the robot's decision-making mechanism. After that, the robots select the most suitable capture points, which are uniformly distributed along the entrapping pattern. In addition, the robot can revolve along the entrapping pattern, creating a flow entrapping effect like the fluidity of cell membranes (the cell membrane also flows to wrap the cell). In this layer, each robot selects its most suitable capture point in real time. In Fig. 1, the robot's decision for the capture point is denoted by D .

The third layer of RE-GRN provides a new velocity control mechanism, in which the robot navigates towards the target position through a modified Vicsek-model (represented by F in Fig. 1). Gabor et al. [32] presented a flocking velocity control model based on the Vicsek-model for real drones. Their experiments demonstrated that the induced swarm behavior remained stable under realistic conditions for large flock sizes. Inspired by this research, we propose an improved Vicsek-model, which considers four velocity action terms: the movement of the robot to the goal capture points (represented by t), obstacle avoidance (represented by o) and collision avoidance between robots and targets (represented by r). Finally, the vector superposition of the separately calculated velocity components determines the predicted speed of the robot at the time. The maximum limit was superimposed on this velocity

for practical considerations of engineering applications. In this way, the robot can achieve a smooth maneuver to its goal capture point, leading to a collective movement behavior of the robot swarm similar to that of a natural population. To the movement pattern of the natural population.

Based on the three-layer mechanism, robots can uniformly entrap multiple targets, revolving in each pattern and tightly entrapping each target. Throughout the movement process, the robot swarm maneuvers as smoothly and naturally as the natural population. The design flowchart of the RE-GRN method is illustrated below.

3. Methods and strategy

3.1. Upper layer: gene regulatory networks generate entrapping morphologies

In the upper layer of the RE-GRN, robots generate the morphologies (patterns) for entrapping multiple targets. These patterns can be automatically generated by conceptualizing automation ideas. The automated design of a GRN can significantly reduce the workload on human designers. In another study [31], we used genetic programming to automatically assemble and optimize the GRN superstructure and parameters. A fitness function and a series of assembly rules were established to generate the optimal structure adapted to the environment. For further details on the design of the upper layer of the RE-GRN method, readers are referred to Fan et al. [31].

After constructing the upper layer of the GRN by genetic programming, we derived the differential equation for the generation mechanism of the concentration of the upper layer of the network. The commonly used mathematical models of the GRN include the directed graph, Boolean network, Bayesian network, differential equation, and difference equation model. Differential equation

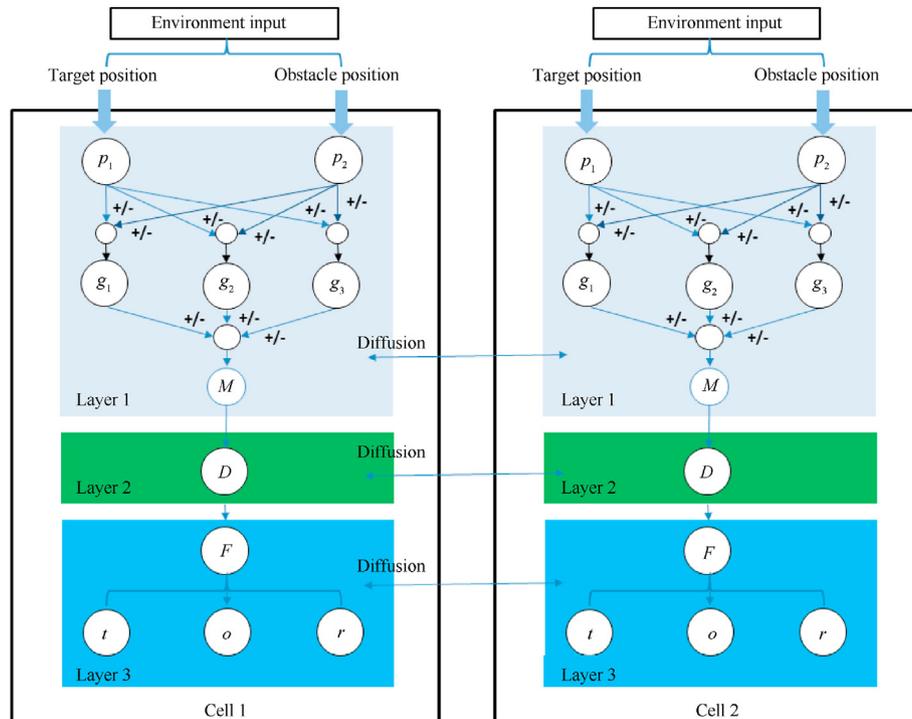


Fig. 1. Flowchart of the multiple-target uniformly revolving entrapment (RE-GRN) algorithm.

models are widely used to construct small and precise GRNs. The upper-level mechanism of the RE-GRN converts the differential equation form of the original GRN into a difference equation one. We assume that the robots can detect the locations of the targets, obstacles, and other robots in the environment during the entrapment task. The robots can communicate with one another to share information about the positions of the targets and obstacles. During the pattern-generation process, as the absolute position information of the target is the same for all robots, they establish the same pattern when applying the same RE-GRN mechanism. Specifically, the upper layer equation of the RE-GRN is as follows:

$$p_j(t+1) = \nabla^2 p_j(t) \quad (1)$$

$$p(t+1) = \sum_{j=1}^{N_t} p_j(t+1) \quad (2)$$

The computation of the concentration gradients of the target and obstacle by the Laplacian operator can be considered as a simulation of the concentration diffusion of proteins in a biological system. By solving the above equation, we obtained the form of $p_j(t)$ as follows:

$$p_j(t) = e^{-v\sqrt{x^2(t)+y^2(t)}} \quad (3)$$

$$p(t+1) = \sum_{j=1}^{N_t} p_j(t+1) \quad (4)$$

where, $x(t)$ and $y(t)$ are the horizontal and vertical coordinate distances of the robot from a target or obstacle. As this distance may vary with time, it can be expressed as a function of time t . The exponent v is the concentration diffusion factor, which can be used to adjust the mapping relationship between distance and concentration. $p_j(t)$ represents the protein concentration generated by the j th target or obstacle location information. p represents the comprehensive protein concentration produced by all the detected targets or obstacles at the current robot position from $j = 1$ to N_t . Among them, N_t is the total number of detected targets or obstacles.

$$g_1(t+1) = \text{sig}(p, \beta_1, k) \quad (5)$$

$$g_2(t+1) = 1 - \text{sig}(p, \beta_2, k) \quad (6)$$

$$g_3(t+1) = \text{sig}(g_1 + g_2, \beta_3, k) \quad (7)$$

$$\text{sig}(r, \beta, k) = \frac{1}{1 + e^{-k(r-\beta)}} \quad (8)$$

The robots construct the concentration information fields g_1 , g_2 , and g_3 on the map after processing the obstacle or target position information. The upper layer of the RE-GRN can take several forms. Therefore, the methods of calculating g_1 , g_2 , and g_3 in the upper layer of different GRNs are not identical. Particularly, in the superstructure of the RE-GRN, g_1 and g_2 process the inputs of the target and obstacle position information separately. The final concentration information field is then coupled by g_3 to form the information map with the highest concentration of robots. r is the distance between the robot and the target (or the obstacle). We assume that β_1 , β_2 , and β_3 are the thresholds of the sigmoid function. Thus, by adjusting β , we can adjust the interval range of the entire generated concentration field. i is the concentration difference adjustment coefficient. The larger the

k , the more evident the concentration difference. A smaller k makes the concentration difference close to linear. The range of k is the concentration difference adjustment coefficient. The larger the k , the more evident the concentration difference. A smaller k makes the concentration difference close to linear. The range of β is $0 - 1$, and the range of k is $0 - 2$

Through this mechanism, the concentration fields of the obstacle and target position information are calculated using Eq. (3) and Eq. (4), respectively. The processed fields are combined into a concentration field (Fig. 2) by using Eq. (5)–Eq. (8) containing both types of information.

The robots sample the specific concentration values on the final concentration map to form a pattern suitable for the current situation. If the robots assemble into this pattern, they can surround the targets while avoiding the obstacles. Thus, the swarm robots only need to aggregate in this pattern and produce the desired swarm behavior.

3.2. Transition layer of RE-GRN: decision making

In Subsection 3.1, the RE-GRN algorithm provides an entrapment pattern that adapts to the environment of the robots. In this section, we attempt to design a reasonable mechanism that can make robots allocated for the capture points on the pattern, to satisfy the condition of the uniform distribution of the swarm robots on the pattern generated by the upper layer of the RE-GRN. To this end, we construct a mechanism to allocate capture points to the swarm robots. The algorithm for this mechanism is as follows:

Algorithm 1. An adaptive uniform allocation framework for the capture points on multiple-target entrapment pattern

Input: Location of the pattern, number of robots, positions of robots;

Output: The capture points on the pattern of robot i .

1. The number of robots is defined as Nr . The number of patterns is Np (patterns will merge if targets are too close; hence, the number of robots should be evenly distributed according to the number of patterns).
2. Calculate the sampling interval for the capture points in the current situation. $m = Nr/Np$.
3. The capture points are numbered 1 to m counterclockwise from the reference point. When the robot is not assigned to a capture point, its flag is 0. If not, it stores the capture point number and changes its flag to the capture point number.
4. If robot i has the highest priority in the swarm, robot i selects capture points as reference points and broadcasts to the swarm. Or robot i receives the reference points through communication.

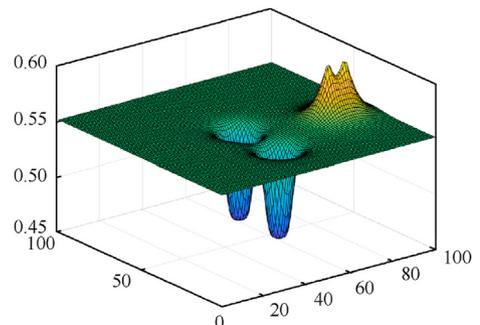


Fig. 2. Concentration field formed by considering the concentration information of both targets and obstacles.

5. Robot i samples each pattern into N points with N/m as the sampling interval, and obtain m points as the capture points of the robots. N is a self-designed integer multiple of m .
6. The robot i traverses all capture points in its list and finds the nearest one. The robot broadcast its own flag and the distance to its capture point.
7. **for** $j = 1$ to Nr
8. **If** Robot i 's flag is same with robot j 's flag (that is, the two robots have selected same capture point; here, it is assumed to be capture point e . The distance between the robot i and the capture point $e <$ the distance from the robot j to the capture point e . **then**
9. Establish the mapping relationship between the robot i and the capture point e .
10. **else**
11. Establish the mapping relationship between the robot j and the capture point e . The robot i removes the capture point e from its list. **Go to 6**
12. **end if**
13. **end for**
14. The robot i uses the Vicsek-model to approach its capture point.

To ensure that all robots in the distributed system obtain the same information about the capture points, the robots need to determine the reference point (the first capture point) of all capture points through communication. The robot with the highest priority in the swarm (preset by the system) determines the reference point by random in the pattern and constantly broadcasts the position of the reference point to the other swarm robots. If the robot with the highest priority in the system is damaged, the robot with the second highest priority will assume its role again and so on. All robots in the swarm will receive the position information of reference point. Meanwhile the pattern is divided into segments to obtain the list of capture points. In the capture point allocation mechanism, the list of capture points calculated by all robots is the same. Every robot autonomously selects the capture point most suited (considering the distance factor and the other robots' position) to their current position. Consequently, the robots achieve a reasonable distribution of capture points on the pattern and choose the most suitable capture point as the destination.

As mentioned earlier, the robot's revolution around the target reduces the chance that the target accelerates away if the target does not want to collide with the robot. We propose a mechanism to help the robots revolve in the entrapment pattern, called reference point change. Here, the change of the reference point (the first capture point) forces all capture points to change, thus leading the robots to revolve in the pattern. In order to express this mechanism more clearly, we model the circular pattern in the ideal environment. The pattern will change its position with the movement of targets. After the robot has confirmed the reference point and used it to determine all capture points, the reference point $e1$ on the pattern will change to the closest capture point $e2$ in the pattern at the next moment, which becomes the new reference point (The overall position of the pattern will change as the target moves, and accordingly the positions of all the capture points on the pattern), and the process is repeated. Fig. 3 depicts a standard circular pattern as an example (the entrapment pattern in a free obstacle environment with one target is a standard circle) to model the revolution mechanism. Thus, the robots exhibit revolutionary motion. The figure has two straight lines: the connecting line between the positions of the reference point and the target, and the straight line with the target's direction of motion. We define the angle between these two straight lines as $\theta(t)$. The angle between two lines at the next moment is $\theta(t+1)$.

However, under this behavioral mechanism, the revolution behavior of the robot is not constant. In the case of constant target motion, if the reference point moves to the intersection ($s1$) of the reverse extension line of the target's motion direction and the pattern, the reference point $s1$ is closest to the reference point $s2$ in the new pattern and the direction of movement of the robot from $s1$ to $s2$ is parallel to the direction of target motion. At this time, the robot stops revolving in the pattern, while the target continues to travel in the same direction. To sustain this revolution behavior, robots limit the change of the reference point position to one cycle ($c1 - c2$). For representation, as illustrated in Fig. 4, the target is placed at the center of the circle, the direction of velocity of the target is the 0° reference axis, and the reference point is (counterclockwise) between $\frac{\pi}{2} - \pi \cdot \frac{Np}{Nr}$ and $\frac{\pi}{2} + \pi \cdot \frac{Np}{Nr}$ within the angular range. Here, Np is the number of patterns, and Nr is the number of robots. Initially, the robot in swarm with highest priority randomly selected a capture point as the reference point. As this reference point moves with the target to $c2$ (revolving $(\frac{\pi}{2} + \pi \cdot \frac{Np}{Nr})^\circ$ counterclockwise from the 0° reference axis), the reference point is reset to the capture point near $c1$ (revolving $(\frac{\pi}{2} - \pi \cdot \frac{Np}{Nr})^\circ$ counterclockwise from the 0° reference axis). This process is repeated continuously to realize the periodic transformation of the reference point. The following equation calculates the revolutionary velocity of the robot as a function of time, as the robot encounters a circular pattern.

$$v_r(t+1) = r - \sqrt{r^2 + v^2 - 2rv \cos(\theta(t))} \quad (9)$$

$$\theta(t+1) = \pi - \arccos\left(\frac{v - r \cos(\theta(t))}{r^2 + v^2 - 2rv \cos(\theta(t))}\right) \quad (10)$$

where, $v_r(t+1)$ and $\theta(t+1)$ are the magnitude and direction of the revolutionary velocity of the robot in the pattern, respectively. Notably, the equation is fully applicable when θ is an acute angle. At time $t+1$, the revolutionary velocity component of the robot along the pattern is deduced as follows. For an obtuse included angle, the target is assumed to have reversed its direction. Thus, it becomes an acute angle according to the definition of θ , and the equations are still applicable. Particularly, its velocity when θ is a right angle is as follows.

$$v_r(t+1) = \sqrt{r^2 + v^2} - r \quad (11)$$

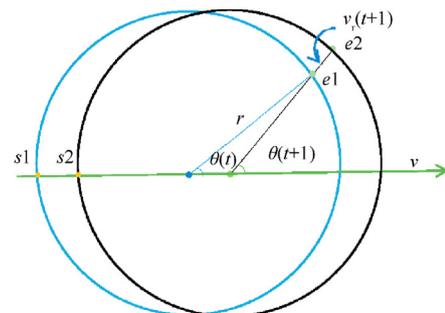


Fig. 3. Schematic of the scenario of a robot switching capture points in a standard circular pattern.

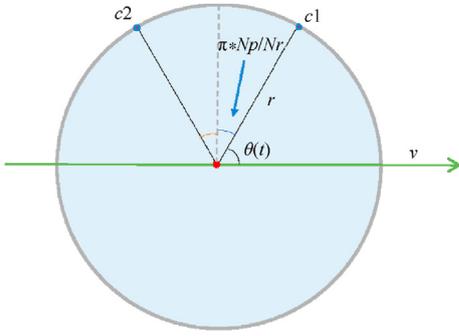


Fig. 4. Schematic of the scenario of a robot switching the reference point in a standard circular pattern.

3.3. Lower layer of RE-GRN: motion control

The first two layers (upper and transition layers) of the RE-GRN were introduced in the last two sections. In the upper layer, the robots generate the concentration field, while in the transition layer, they sample the field to generate a pattern and adaptively allocate capture points to uniformly entrap multiple targets. Each robot must move to the corresponding position on the target's entrapping pattern. To this end, we proposed a flocking motion velocity controller based on an improved Vicsek-model. Although the upper layer of the RE-GRN already considers the position information of obstacles and targets, the robots may still be relatively too close to the obstacles or targets as they form the pattern. Therefore, the purpose of the lower layer algorithm is to allow the robots to smoothly navigate to the capture points calculated in the first two layers of the RE-GRN without colliding with the neighboring robots, targets, or obstacles. Therefore, this section is divided into three parts, including repulsion, obstacle avoidance, and maneuvering toward the target.

3.3.1. Repulsion

As the robot maneuvers, it may occasionally tread too close to other robots. For safe navigation, we introduced a mutually exclusive speed between robots in the Vicsek-model [32]. r_{arep} is the distance at which the local repulsion is activated, and larger values create sparser flocks with fewer collisions. If the distance between the agents is less than r_{arep} , the robots will generate a repulsion velocity to maintain a safe distance from their neighbors.

$$\mathbf{v}_{ij}^{rep} = \begin{cases} p_a^{rep} (r_{arep} - r_{ij}) \cdot \mathbf{r}_{ij} & \text{if } (r_{ij} < r_{arep}) \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

where, $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$ is the distance between robot i and robot j . $\mathbf{r}_{ij} = \frac{\mathbf{r}_i - \mathbf{r}_j}{r_{ij}}$ represents the direction(unit direction vector) of velocity of robot j toward robot i . p_a^{rep} is the linear coefficient of the repulsion velocity between robots.

The robots may also maneuver too close to the target as they approach the capture point. To prevent the robots from colliding with the target, we set a unidirectional repulsion velocity between the robot and the target (in the swarm robot system, the target is considered to be unaffected by the robot). If the distance between a robot and its target is less than r_{trep} , the robot will generate a velocity away from the target.

$$\mathbf{v}_{itarget}^{rep} = \begin{cases} p_t^{rep} (r_{trep} - r_{it}) \cdot \mathbf{r}_{it} & \text{if } (r_{it} < r_{trep}) \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

Similarly, $r_{it} = |\mathbf{r}_i - \mathbf{r}_t|$ is the distance between robot i and its targets. $\mathbf{r}_{it} = \frac{\mathbf{r}_i - \mathbf{r}_t}{r_{it}}$ represents the direction(unit direction vector) of repulsion velocity of the target toward robot i . p_t^{rep} is the linear coefficient of the repulsion velocity between a robot and its targets. The algorithm in each robot should calculate the repulsion velocity term for all the targets.

In order to get all the repulsive forces on the robot i , we calculated the vectorial sum of the interaction terms of repulsion:

$$\mathbf{v}_i^{rep} = \sum_{j \neq i} \mathbf{v}_{ij}^{rep} + \sum_{target} \mathbf{v}_{itarget}^{rep} \quad (14)$$

3.3.2. Obstacle avoidance

A robot may encounter obstacles as it approaches the capture points on the pattern. To allow the robot to maneuver smoothly as it circumvents an obstacle, the obstacle avoidance speed should be gradually attenuated at a greater distance from the obstacle. Function $D(r, a, p)$ can provide a smooth velocity decay curve for the velocity change between the robot and the desired stopping point, where r is the distance between the robot and the desired stopping point, p determines the crossover point between the two phases of deceleration, and a is the robot's preferred acceleration [32].

$$D(r, a, p) = \begin{cases} 0 & \text{if } r \leq a/2p \\ \frac{rp}{\sqrt{2ar - a^2/p^2}} & \text{if } a/2p < rp \leq a/p \\ \text{otherwise} & \end{cases} \quad (15)$$

To help the robot safely avoid collision with an obstacle in the environment, we assume the presence of a virtual agent at the point closest to the robot on the boundary of the obstacle [36]. Each robot has a corresponding virtual agent as it approaches an obstacle, and the position of the virtual agent changes with the position of the robot.

$$\mathbf{v}_{id}^{obs} = \begin{cases} 0 & \text{if } (r_{id} > r_{obs}) \\ c^d \cdot [v_{id} - D(r_{id} - r_{obs}, a^d, p^d)] \cdot \mathbf{v}_{id} & \text{otherwise} \end{cases} \quad (16)$$

where, c^d is the strength (velocity coefficient) of the distance from the obstacles, and r_{obs} is the safe distance between the robot and the obstacle. Larger values of this distance prompt the robot to start braking at greater distances from the obstacle. $r_{id} = |\mathbf{r}_i - \mathbf{r}_d|$ is the actual distance of the robot i from its nearest obstacle. $v_{id} = |\mathbf{v}_i - \mathbf{v}_d|$, where \mathbf{v}_d is the velocity of the virtual agent pointing perpendicularly inward to the boundary of the obstacle in the arena. $\mathbf{v}_{id} = \frac{\mathbf{v}_i - \mathbf{v}_d}{v_{id}}$, \mathbf{v}_{id} is the unit direction vector of the velocity vector difference between the robot i and its virtual agent. This obstacle avoidance method helps the robot avoid falling into the local minimum. a^d is the maximum allowed acceleration on the optimal braking curve. Higher values of this acceleration imply faster braking by the robot. Excessively high values would result in the failure of the robot to react to excessively large velocity gradients in time, thus causing collisions. p^d is the linear gain of the optimal braking curve, which is used to determine the maximum allowable velocity difference. Large values of this gain approximate the braking curve to the

constant acceleration curve, while small values elongate the final part of the braking (at a small velocity) with decreasing acceleration and smoother stops.

3.3.3. Move toward the goal position

Another topic of interest is enabling the robot to approach the goal capture point with a gentle velocity. As introduced earlier, the function a^d establishes a mapping relationship between the velocity and the distance to the desired goal capture point, forming a smooth velocity decay curve affected by the distance factor. In this section, we employ the following function to construct a new navigation speed function for the goal capture point.

$$\mathbf{v}_{ig} = \left[\mathbf{v}_f + C^g \cdot D(r_{ig}, a^g, p^g) \right] \cdot \mathbf{r}_{gi} \quad (17)$$

where, the initial velocity \mathbf{v}_f of the agent is preset, and C^g is the preferred common traveling velocity coefficient for all agents to approach the goal position. $r_{ig} = |\mathbf{r}_i - \mathbf{r}_g|$ is the distance between the agents a^g and its goal position. a^g is the maximum allowable acceleration on the optimal braking curve. p^g is the linear gain of the optimal braking curve, which is used to determine the maximum allowable velocity difference. a^g and p^g are the same as a^d and p^d , but for wall alignment interactions. $\mathbf{r}_{gi} = \frac{\mathbf{r}_g - \mathbf{r}_i}{r_{ig}}$, \mathbf{r}_{gi} is the direction(unit direction vector) in which the agent points toward the goal position. It is worth noting that r_{it} and r_{ig} are different. r_{it} is the distance from the robot i to the target, and r_{ig} is the distance from the robot i to the position of the goal entrapping point on the entrapping pattern.

3.3.4. Final equation of desired speed

The robot needs to consider both these velocity effects. We calculate the vectorial sum of all the interaction terms:

$$\mathbf{v}_i^{\text{desire}} = \mathbf{v}_i^{\text{rep}} + \mathbf{v}_{id}^{\text{obs}} + \mathbf{v}_{ig} \quad (18)$$

For practical applications, we set a cutoff velocity $\mathbf{v}_{\text{limit}}$ to compensate for the motion restraint. If the velocity $\tilde{\mathbf{v}}_i^{\text{desire}}$ exceeds the limit, then maintaining the direction of the desired velocity while reducing its magnitude yields the following equation:

$$\tilde{\mathbf{v}}_i^{\text{desire}} = \frac{\tilde{\mathbf{v}}_i^{\text{desire}}}{|\tilde{\mathbf{v}}_i^{\text{desire}}|} \cdot \min \left\{ |\tilde{\mathbf{v}}_i^{\text{desire}}|, \mathbf{v}_{\text{limit}} \right\} \quad (19)$$

4. Experiments and index evaluation

4.1. Simulation experiments

In this section, the performance of the proposed RE-GRN method is verified using simulation experiments based on MATLAB and CoppeliaSim simulation platform. Simulations based on MATLAB are used for proof-of-concept given global information. We used GRN method, AGENT method and RE-GRN method to do the comparative experiment in five scenarios to verify the validity and robustness of these methods. In the RE-GRN simulations based on MATLAB, the robots obtain the same obstacle and target position information in the same coordinate system (with global information known), so the entrapping shape calculated by them is identical. The robots can communicate with each other to determine the location of the reference point to synchronize the locations of all

the capture points. The following five scenarios are considered.

- (1) Obstacle-free environment. Two targets are in circular motion around a fixed point in the arena and are being trapped by agents (Scenario 1).
- (2) Fifteen robots trap two targets. The environment contains two obstacles (Scenario 2).
- (3) Ten robots trap one target. The unknown environment contains several obstacles. The robot may encounter an unforeseen obstacle around the next corner (Scenario 3).
- (4) Fourteen robots trap two targets. The environment has many barbed obstacles (Scenario 4).
- (5) Twelve robots trap two targets. The environment has several square obstacles in the environment. Two targets move arbitrarily in the environment (Scenario 5).

As demonstrated by these scenarios (Figs. 5–17), the RE-GRN performs best among the three methods can realizing that the robot swarm orbits the target while entrapping the target. The previous GRN method has an evident problem of the irregular distribution of robots in a pattern. The robots jump back and forth into the pattern when approaching it but do not completely settle down on it. The RE-GRN method overcomes this drawback. The robot applying AGENT method achieves a better entrapping effect than the GRN method, but the entrapping effect is not as ideal as the RE-GRN method, which is reflected in the fact that AGENT method cannot enable a revolving entrapping effect, and the uniformity of entrapping is not as good when trapping multiple targets.

Notably, the method demonstrates distinct advantages over the existing method when entrapping multiple targets with swarms. As observed in Fig. 15–Fig. 17, the two targets perform a Lévy flight [37] above the field (a random movement step generation mechanism when some living creatures forage), and the robots apply the GRN method, AGENT method and RE-GRN method to respectively entrap the two targets. The division of groups in the RE-GRN method is evidently more uniform than that in the GRN method and AGENT method when trapping multiple wandering targets. In the RE-GRN method, the robots are uniformly distributed on the pattern, and constantly adjust their motions as the pattern changes. The robots maintain a safe distance from the nearby targets, obstacles, and other robots, avoiding collisions.

To further verify the effectiveness of the RE-GRN algorithm, we conducted more experiments on CoppeliaSim. In this simulation, the location of the target was known. The drones could detect other drones and obstacles within the range of the sensor and broadcast their detected obstacle position information in the swarm. This ensures that all the robots in the swarm eventually have the same concentration map and entrapping pattern. In this way, all robots can synchronize the list of capture points (see Section 3 for specific mechanism). The experimental effect of drones entrapping multiple targets is shown in Fig. 18.

Fig. 18 shows that, in the case of free movement of multiple targets, the deployment of the RE-GRN algorithm in the drone swarm can deal with multiple targets, achieve adaptive grouping, and uniformly entrap multiple targets. The drone swarm can revolve along the entrapping pattern to tightly encircle the target. The drones' revolving entrapment effect can be observed in the video.

4.2. Index evaluation

By the definition of entrapment, the method must consider distribution of robots in several directions around the target.

Therefore, we calculate the occupancy rate of entrapment circle, i.e. the presence of robots in six directions (we divided the 360° area around the target into six sectors) to evaluate the trap quality. Furthermore, the robots must efficiently trap the targets, which may be reflected by the distance the robots traveled to trap those targets. In addition, the robot should also pay attention to its own safety. Therefore, their velocity should not change significantly over time as that could be detrimental to their stability. In summary, this study uses the following evaluation indicators to digitize the experimental results:

- (1) The uniformity of the number of robot groups when trapping multiple targets (Fig. 19).
- (2) The occupancy rate of each target in the entrapment circle (Fig. 20).
- (3) The average distance traveled by robots to entrap (Encirclement occupancy = 6/6) the targets (Fig. 21).
- (4) The velocity correlation [32] ψ^{corr} of the robot at consecutive moments (Figs. 22–24).

$$\psi^{\text{corr}} = \frac{\mathbf{v}_i \cdot \mathbf{v}_{i+1}}{|\mathbf{v}_i| \cdot |\mathbf{v}_{i+1}|} \quad (20)$$

Eq. (20) gives the calculation formula of velocity correlation.

These indicators demonstrate the enhanced performance of the RE-GRN method for multi-target entrapment over the previous method.

As depicted in Fig. 19, the robots can be grouped as they entrap multiple targets. The number of robots allocated to each target was calculated and plotted. The data reflect the stability of the RE-GRN method in terms of group division. In Fig. 20, the robot forms a tight siege around the target. In most cases, the occupancy rate of the encirclement is 6/6 which confirms the superior entrapping performance of the RE-GRN method. In addition, Fig. 21 illustrates the average distance traversed by the robots to trap the targets (Encirclement occupancy = 6/6) under the three methods. The RE-GRN method requires the robots to travel a significantly shorter distance to trap the targets. We have carried out the significance test of difference, and the test results show that the method

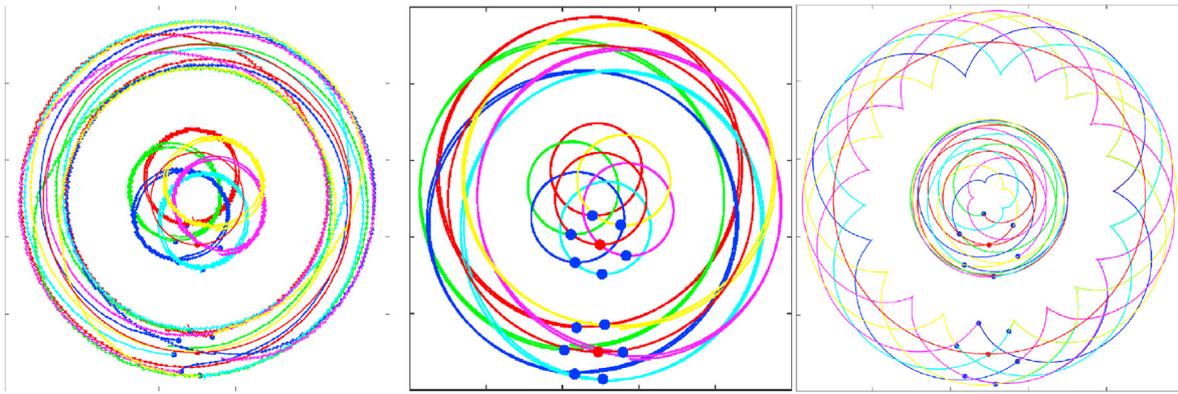


Fig. 5. Twelve robots trap two targets in an obstacle-free environment with the GRN (left), AGENT(middle) and RE-GRN methods (right), and the scenario illustrates their trajectories (Scenario 1).

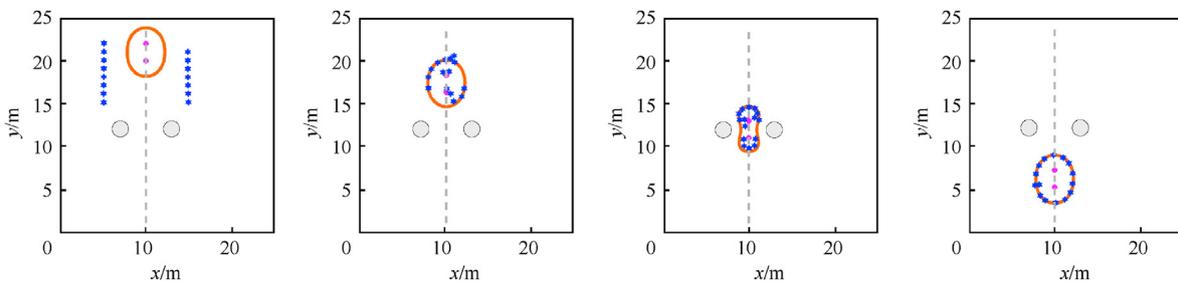


Fig. 6. Fifteen robots trap two targets in the environment of two obstacles with the previous GRN method (Scenario 2).

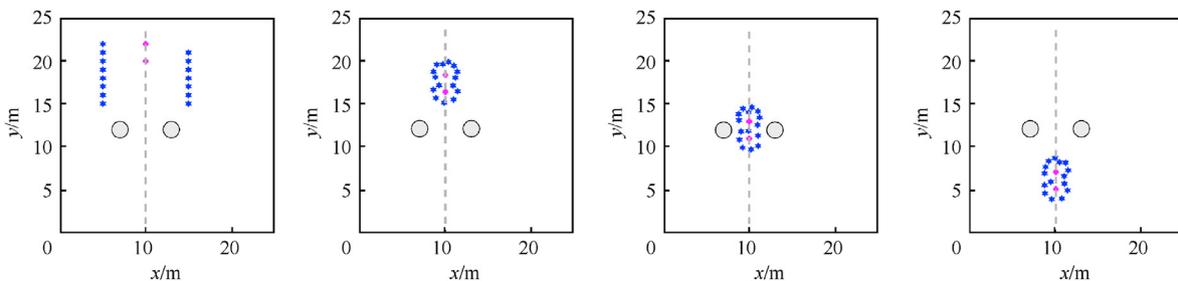


Fig. 7. Fifteen robots trap two targets in the environment of two obstacles with the AGENT method (Scenario 2).

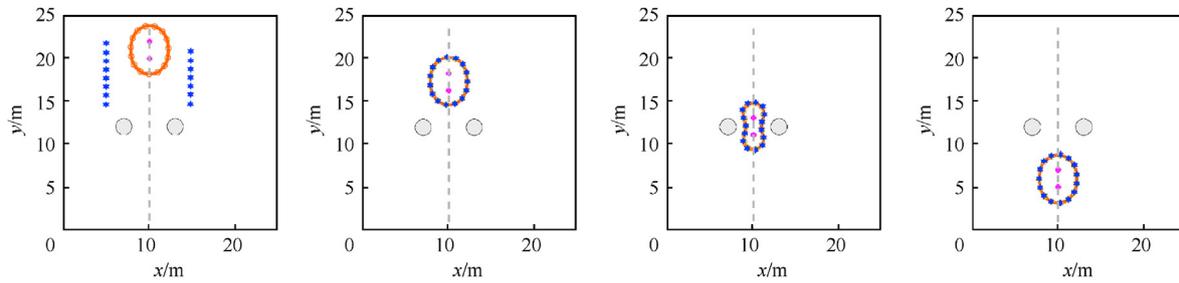


Fig. 8. Fifteen robots trap two targets in the environment of two obstacles with the RE-GRN method (Scenario 2).

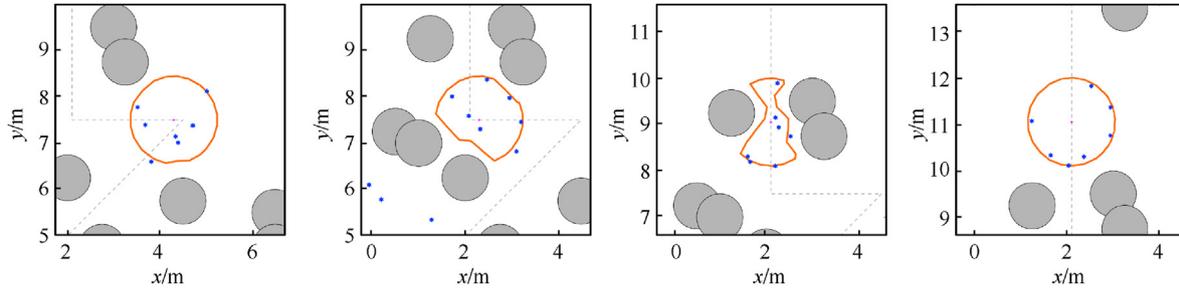


Fig. 9. Ten robots trap one target in the environment with many obstacles with the previous GRN method (Scenario 3).

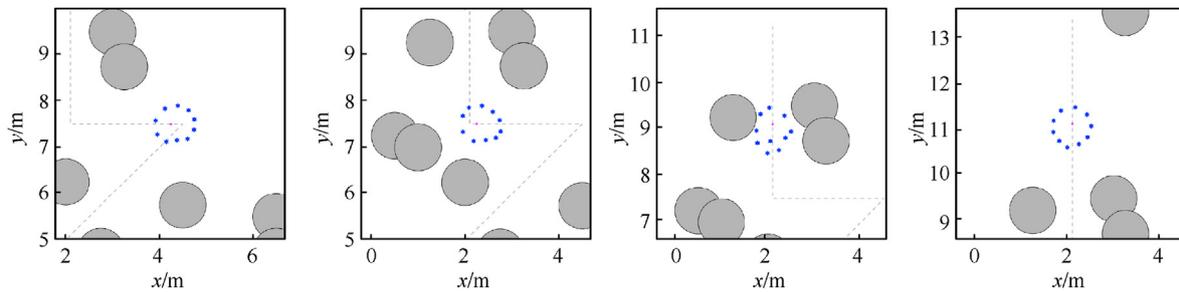


Fig. 10. Ten robots trap one target in the environment with many obstacles with the AGENT method (Scenario 3).

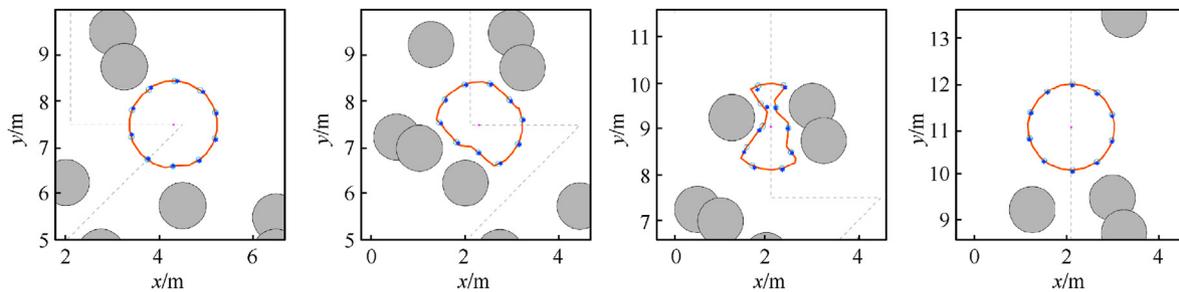


Fig. 11. Ten robots trap one target in the environment with many obstacles with the RE-GRN method (Scenario 3).

proposed in this study has significant advantages compared with the two comparison methods.

As the robots maneuver, the change in velocity at two consecutive moments must be sufficiently small to meet practical requirements. Figs. 22–24 show that in the previous GRN method, the robot's continuous motion correlation is very low, which is

close to -1 most of the time, which means that there may be vibration in the robot's motion. In addition, in the AGENT method, the robot's continuous motion correlation is very high, which is close to 1 most of the time, which means that the robot does not change the direction frequently, and does not have the effect of revolution around the target. In the RE-GRN method, the motion correlation of

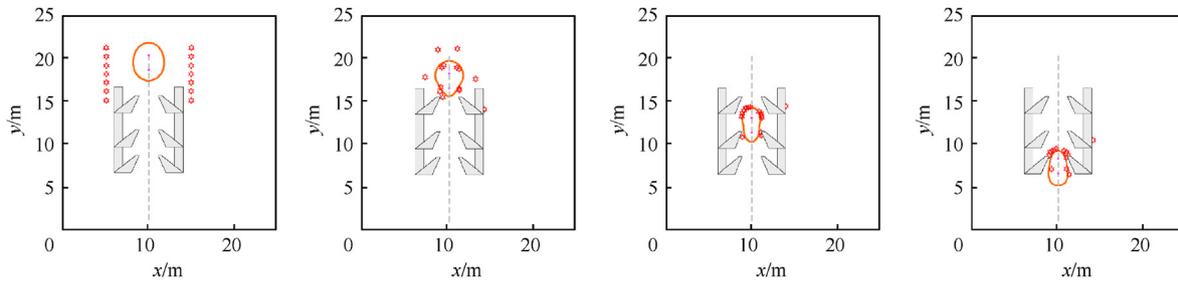


Fig. 12. Fourteen robots trap two targets in the environment with barb obstacles with the previous GRN method (Scenario 4).

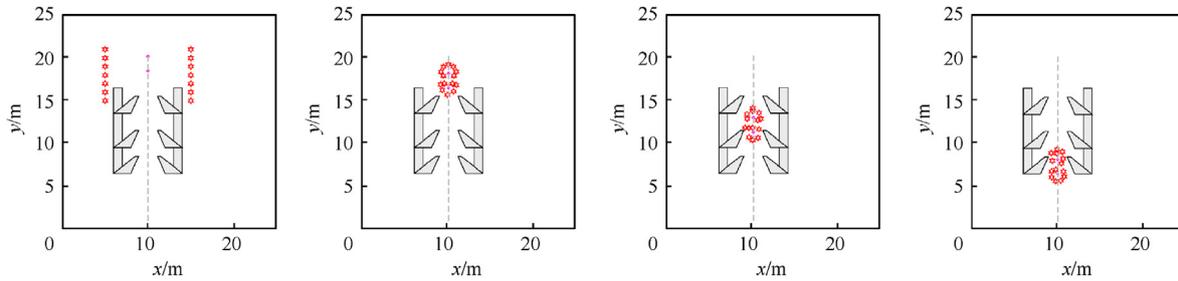


Fig. 13. Fourteen robots trap two targets in the environment with barb obstacles with the AGENT method (Scenario 4).

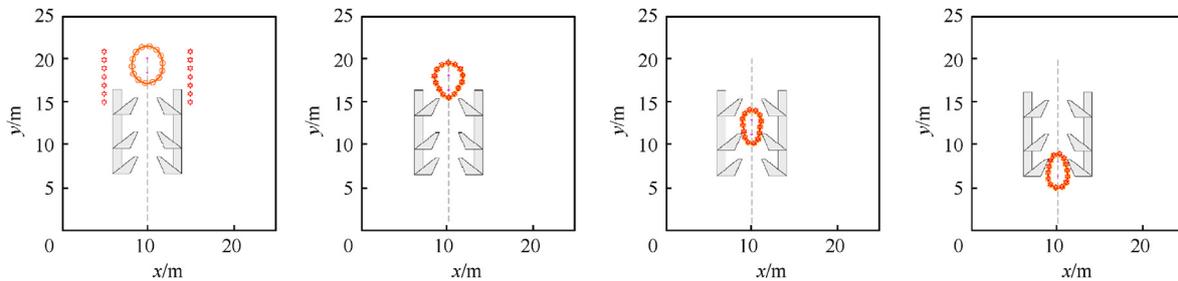


Fig. 14. Fourteen robots trap two targets in the environment with barb obstacles with the RE-GRN method (Scenario 4).

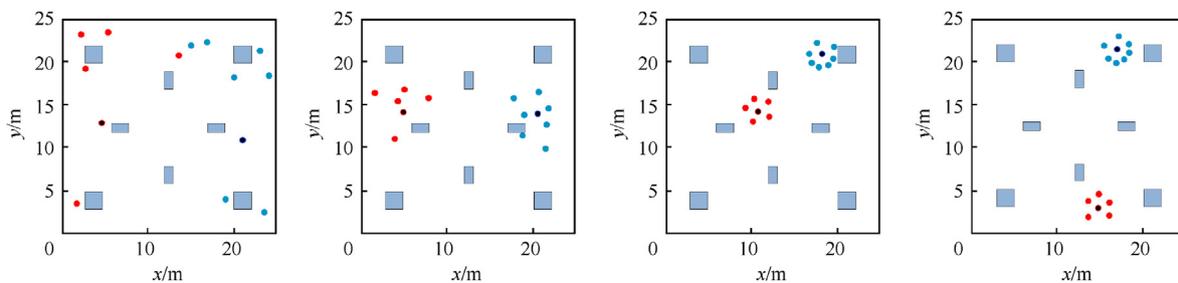


Fig. 15. Twelve robots trap two wandering targets in complex obstacle environment with the previous GRN method (Scenario 5).

the robot at continuous time is almost evenly distributed between $[-1, 1]$, which indicates that the robot changes the motion direction continuously and smoothly in most of the time, reflected by the fact that the robot orbits the target.

From the experimental results shown in the video, we can also observe that the robots by applying RE-GRN algorithm can autonomously make decisions when encountering multiple targets and execute the swarm behaviors of grouping while trapping multiple

targets. With this method, robots can adapt to the environmental conditions and accordingly change the entrapping pattern without colliding into the neighboring robots or obstacles. The experimental indicators of this study prove the potential of the RE-GRN model in target entrapping applications. The entire system can be deployed on robots in a distributed manner and can be safely applied in industrial and military applications.

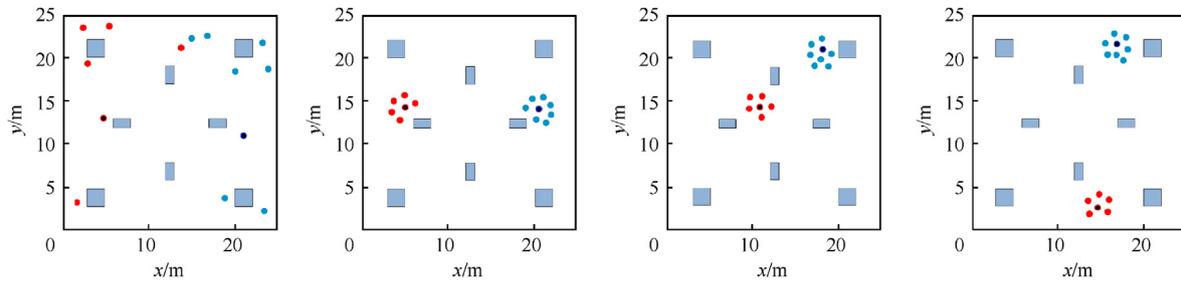


Fig. 16. Twelve robots trap two wandering targets in complex obstacle environment with the AGENT method (Scenario 5).

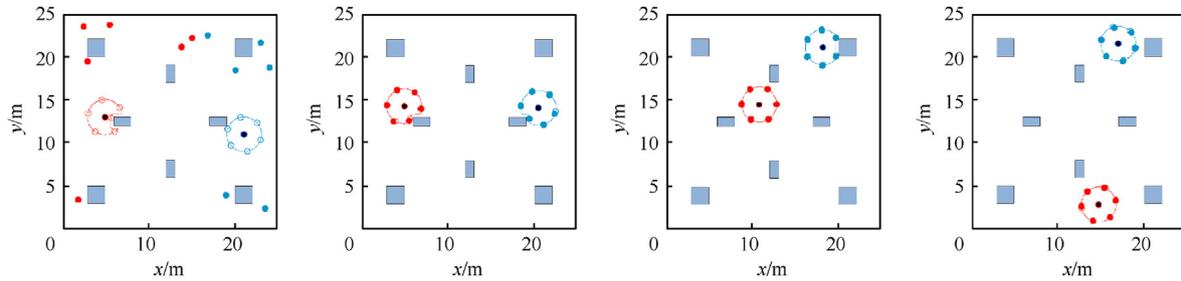


Fig. 17. Twelve robots trap two wandering targets in complex obstacle environment with the RE-GRN method (Scenario 5).

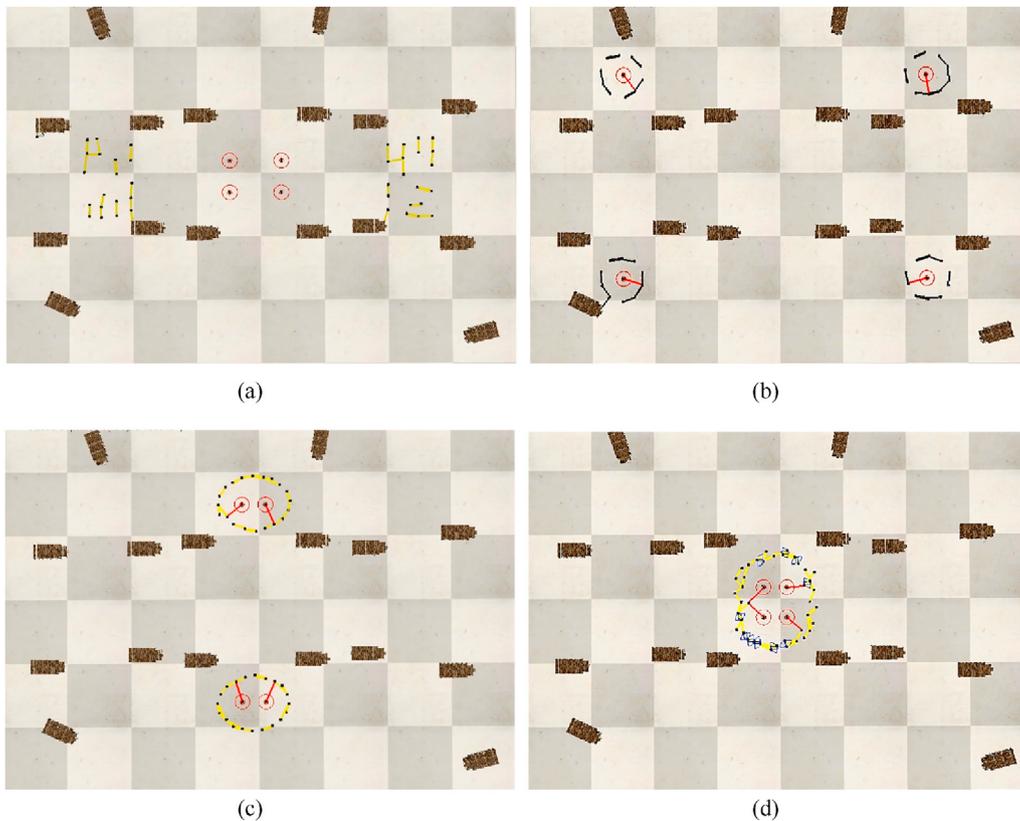


Fig. 18. Forty drones trap four targets in the Coppeliasim simulation: (a) Initial scenario; (b) Drones are divided into four groups to trap targets; (c) Drones are divided into two groups to trap targets; (d) Drones form a large circle to trap the four targets.

5. Conclusions

The RE-GRN method proposed in this paper gives a novel idea of orbiting multiple targets while entrapping them in the military's

application of the UAV swarm system. In this study, a three-layer algorithm framework is put forward to carry out environmental information transformation, adaptive decision making, and motion control respectively. Firstly, the concentration field of the

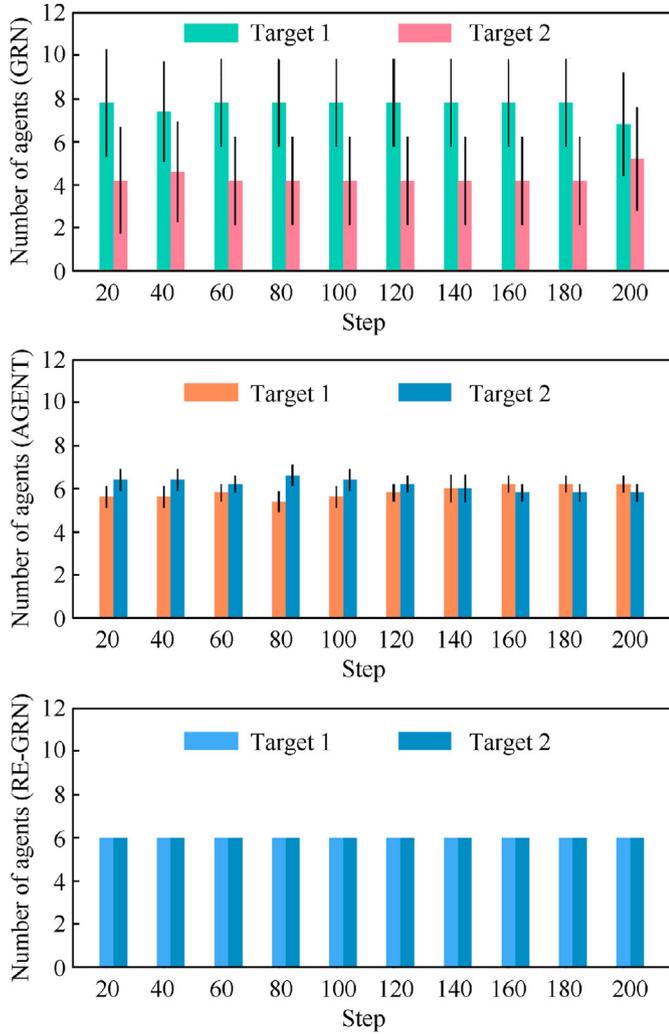


Fig. 19. Uniformity of the division of robot groups.

information on the targets and obstacles is constructed to obtain the entrapping pattern. The designed decision mechanism then samples the entrapping pattern to generate capture points to form a uniform distribution of robots around the entrapping targets. The robots' orbiting the targets enables them to entrap the targets more tightly and effectively. Moreover, this study improves Vicsek-model to make robots more smoothly approach capture points. During this process, the robot can maintain the effect of collision-free flocking and obstacle avoidance.

According to the five experimental scenes in MATLAB experiments, robots by applying RE-GRN method can flexibly deal with varying complex obstacle scenarios, and change entrapping patterns adaptively to trap multiple targets while orbiting them. This study designed evaluation indexes considering entrapping task completion efficiency and the flocking effect. As shown in the evaluation indicators, the RE-GRN method outperforms the GRN method and AGENT method in the task of entrapping multiple targets, with a better encircling and revolving effect. Furthermore, we carried out simulation on the CoppeliaSim Simulation platform, which demonstrates that when entrapping multiple targets, the UAV swarm can effectively divide into groups and orbit the targets with a success.

However, the proposed method relies on global communication,

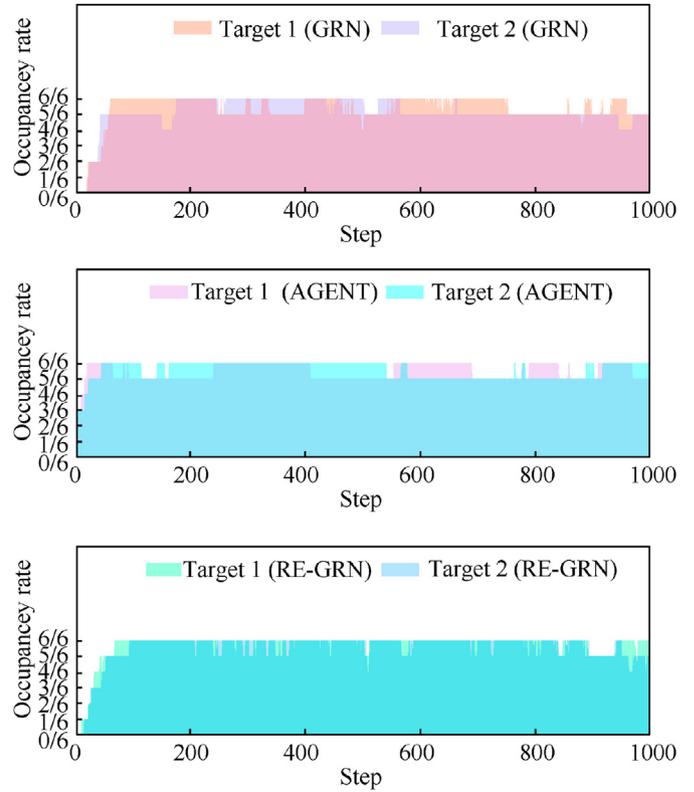


Fig. 20. Robot occupancy rate of entrapment circle for each target.

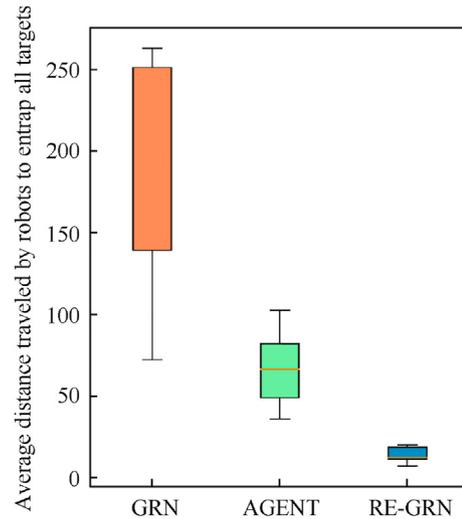


Fig. 21. Average distance traveled by the robots to trap all the targets.

which is not conducive to the deployment of drones in non-communication zones in military applications. Therefore, future studies aim to reduce the reliance of the method on communication. It is also very worthwhile to conduct physical experiments to validate the proposed RE-GRN algorithm. For this purpose, we have designed the prototype of individual swarm robots for 2D experiments and are developing UAV system for 3D experiments. This ongoing work will form the major part of the next step research.

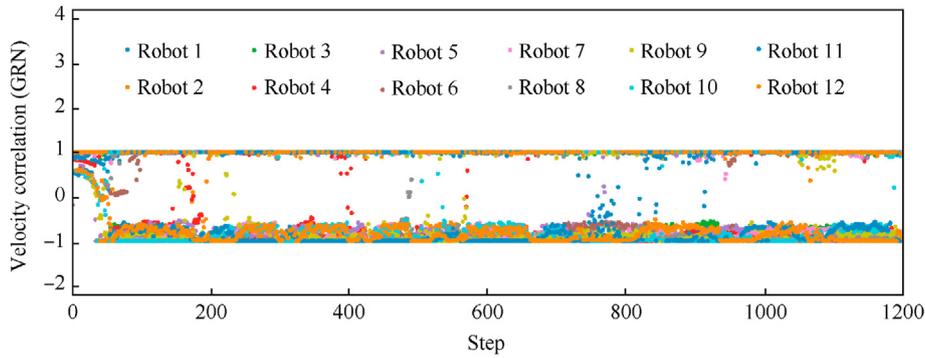


Fig. 22. Velocity correlation of robots at consecutive moments (GRN method).

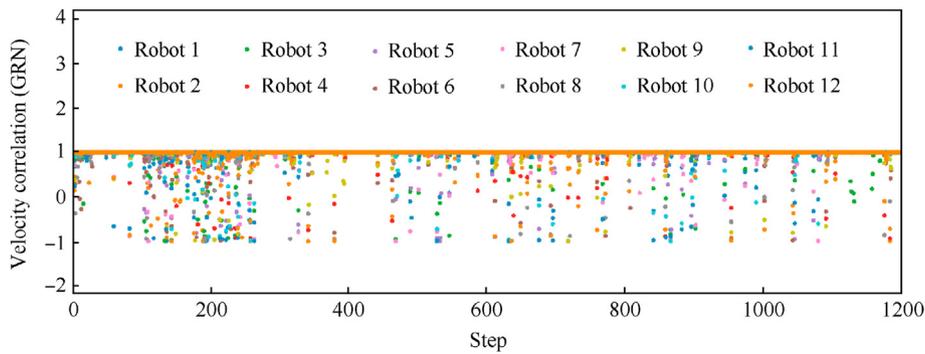


Fig. 23. Velocity correlation of robots at consecutive moments (AGENT method).

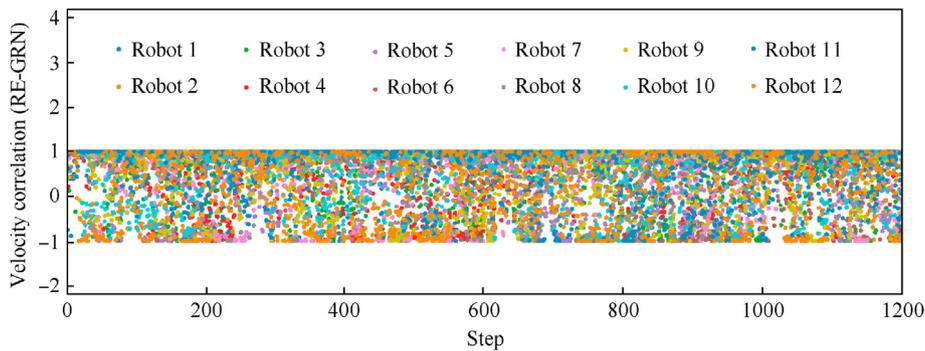


Fig. 24. Velocity correlation of robots at consecutive moments (RE-GRN method).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to thank the Key Laboratory of Digital Signal and Image Processing of Guangdong Province, Shantou University, Guangdong Province, China, for the resources provided by them. This research is also funded by the National Natural Science Foundation of China (62176147), the Science and Technology Planning Project of Guangdong Province of China, the State Key Lab of Digital Manufacturing Equipment and Technology

(DMETKF2019020), and by the National Defense Technology Innovation Special Zone Project(193-A14-226-01-01).

Appendix

Matlab experiment video: https://www.bilibili.com/video/BV1eF41137Nm?spm_id_from=333.999.0.0.

CoppeliaSim simulation video: https://www.bilibili.com/video/BV1XF411W75e?spm_id_from=333.999.0.0.

References

- [1] Yang G-Z, Bellingham J, Dupont PE, Fischer P, Floridi L, Full R, et al. The grand challenges of science robotics. *Science robotics* 2018;3(14):eaar7650.
- [2] Reynolds CW. Flocks, herds and schools: a distributed behavioral model. In: *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*; 1987. p. 25–34.

- [3] Vicsek T, Zafeiris A. Collective motion. *Phys Rep* 2012;517(3–4):71–140.
- [4] M'ehes E, Vicsek T. Collective motion of cells: from experiments to models. *Integrative biology* 2014;6(9):831–54.
- [5] Sousselier T, Dreoj J, Sevaux M. Line formation algorithm in a swarm of reactive robots constrained by underwater environment. *Expert Syst Appl* 2015;42(12):5117–27.
- [6] de S'a AO, Nedjah N, de Macedo Mourelle L. Distributed efficient localization in swarm robotics using min–max and particle swarm optimization. *Expert Syst Appl* 2016;50:55–65.
- [7] Madridano A, Al-Kaff A, Mart'in D, de la Escalera A. Trajectory planning for multi-robot systems: methods and applications. *Expert Syst Appl* 2021;173: 114660.
- [8] Tang H, Sun W, Lin A, Xue M, Zhang X. A gwo-based multi-robot cooperation method for target searching in unknown environments. *Expert Syst Appl* 2021;186:115795.
- [9] Wei H-X, Mao Q, Guan Y, Li Y-D. A centroidal voronoi tessellation based intelligent control algorithm for the self-assembly path planning of swarm robots. *Expert Syst Appl* 2017;85:261–9.
- [10] Glorieux E, Riazi S, Lennartson B. Productivity/energy optimisation of trajectories and coordination for cyclic multi-robot systems. *Robot Comput Integrated Manuf* 2018;49:152–61.
- [11] Farinelli A, Zanutto E, Pagello E, et al. Advanced approaches for multi-robot coordination in logistic scenarios. *Robot Auton Syst* 2017;90:34–44.
- [12] Zhao W, Meng Q, Chung PW. A heuristic distributed task allocation method for multivehicle multitask problems and its application to search and rescue scenario. *IEEE Trans Cybern* 2015;46(4):902–15.
- [13] Dong S, Xu K, Zhou Q, Tagliasacchi A, Xin S, Nießner M, Chen B. Multi-robot collaborative dense scene reconstruction. *ACM Trans Graph* 2019;38(4):1–16.
- [14] Zhang Y, Zeng L, Li Y, Liu Q. Multi-robot formation control using leader-follower for manet. In: 2009 IEEE international conference on robotics and biomimetics (ROBIO), IEEE; 2009. p. 337–42.
- [15] Xu D, Zhang X, Zhu Z, Chen C, Yang P. Behavior-based formation control of swarm robots, *Mathematical Problems in Engineering*. 2014.
- [16] Baizid K, Giglio G, Pierri F, Trujillo MA, Antonelli G, Caccavale F, Viguria A, Chiaverini S, Ollero A. Behavioral control of unmanned aerial vehicle manipulator systems. *Aut Robots* 2017;41(5):1203–20.
- [17] Yu X, Ma J, Ding N, Zhang A. Cooperative target enclosing control of multiple mobile robots subject to input disturbances. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 2019;51(6):3440–9.
- [18] Yang Z, Chen C, Zhu S, Guan X, Feng G. Distributed entrapping control of multiagent systems using bearing measurements. *IEEE Trans Automat Control* 2020;66(12):5696–710.
- [19] Benzerrouk A, Adouane L, Lequievre L, Martinet P. Navigation of multi-robot formation in unstructured environment using dynamical virtual structures, iros'10, ieee. In: RSJ international conference on intelligent robots and systems. vols. 18–22; 2010.
- [20] Zhou D, Wang Z, Schwager M. Agile coordination and assistive collision avoidance for quadrotor swarms using virtual structures. *IEEE Trans Robot* 2018;34(4):916–23.
- [21] C. Wang, M. Gu, W. Kuang, AGENT: an adaptive grouping entrapping method of flocking systems. arXiv preprint arXiv:2206.14614.
- [22] Jin Y, Meng Y, Guo H. A morphogenetic self-organization algorithm for swarm robotic systems using relative position information. In: 2010 UK workshop on computational intelligence (UKCI), IEEE; 2010. p. 1–6.
- [23] Guo H, Meng Y, Jin Y. Swarm robot pattern formation using a morphogenetic multi-cellular based self-organizing algorithm. In: 2011 IEEE international conference on robotics and automation. IEEE; 2011. p. 3205–10.
- [24] Meng Y, Guo H, Jin Y. A morphogenetic approach to flexible and robust shape formation for swarm robotic systems. *Robot Auton Syst* 2013;61(1): 25–38.
- [25] Guo H, Meng Y, Jin Y. Analysis of local communication load in shape formation of a distributed morpho- genetic swarm robotic system. In: IEEE congress on evolutionary computation, IEEE. vols. 1–8; 2010.
- [26] Jin Y, Guo H, Meng Y. A hierarchical gene regulatory network for adaptive multirobot pattern formation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 2012;42(3):805–16.
- [27] Oh H, Jin Y. Evolving hierarchical gene regulatory networks for morphogenetic pattern formation of swarm robots. In: 2014 IEEE congress on evolutionary computation (CEC), IEEE; 2014. p. 776–83.
- [28] Oh H, Jin Y. Evolving h-grns for morphogenetic adaptive pattern formation of swarm robots. *Evolutionary Computation in Gene Regulatory Network Research* 2016:327–61.
- [29] Peng X, Zhang S, Lei X. Multi-target trapping in constrained environments using gene regulatory network-based pattern formation. *Int J Adv Rob Syst* 2016;13(5):1729881416670152.
- [30] Wu M, Zhou Y, Zhu X, Ma L, Yuan Y, Fang T, Wang J, Bao W, Fan Z. Cooperation-based gene regulatory network for target entrapment. In: International conference on swarm intelligence. Springer; 2019. p. 60–9.
- [31] Z. Fan, Z. Wang, X. Zhu, B. Hu, A. Zou, D. Bao, An automatic design framework of swarm pattern formation based on multi-objective genetic programming, arXiv preprint arXiv:1910.14627.
- [32] V'as'arhelyi G, Vir'agh C, Somorjai G, Nepusz T, Eiben AE, Vicsek T. Optimized flocking of autonomous drones in confined environments. *Science Robotics* 2018;3(20):eaat3536.
- [33] Antonelli G, Arrichiello F, Chiaverini S. The entrapment/escorting mission for a multi-robot system: theory and experiments. In: 2007 IEEE/ASME international conference on advanced intelligent mechatronics. IEEE; 2007. p. 1–6.
- [34] Mas I, Li S, Acain J, Kitts C. Entrapment/escorting and patrolling missions in multi-robot cluster space control. In: 2009 IEEE/RSJ international conference on intelligent robots and systems. IEEE; 2009. p. 5855–61.
- [35] Mas I, Kitts C. Object manipulation using cooperative mobile multi-robot systems. In: Proceedings of the world congress on engineering and computer science; 2012. p. 1.
- [36] Han J, Li M, Guo L. Soft control on collective behavior of a group of autonomous agents by a skill agent. *J Syst Sci Complex* 2006;19(1):54–62.
- [37] Viswanathan GM, Afanasyev V, Buldyrev SV, et al. Lévy flights in random searches[J]. *Phys Stat Mech Appl* 2000;282(1–2):1–12.