A Multi-Agent Control Model Based on Gene Regulatory Network

2023 9th International Conference on Big Data and Information Analytics (BigDIA) | 979-8-3503-3007-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/BigDIA60676.2023.10429364

1st Wenji Li
Department of Electronic Engineering
Shantou University
Guangdong, China
liwj@stu.edu.cn

2nd Pengxiang Ren
Department of Electronic Engineering
Shantou University
Guangdong, China
21pxren@stu.edu.cn

3rd Zhaojun Wang
Department of Electronic Engineering
Shantou University
Guangdong, China
17zjwang@stu.edu.cn

4th Dongliang Wang
Department of Electronic Engineering
Shantou University
Guangdong, China
dlwang@stu.edu.cn

5th Biao xu
Department of Electronic Engineering
Shantou University
Guangdong, China
xubiao@stu.edu.cn

6th Chaotao Guan
Department of Electronic Engineering
Shantou University
Guangdong, China
22ctguan@stu.edu.cn

7th Chuangbin Chen
Department of Electronic Engineering
Shantou University
Guangdong, China
23cbchen@stu.edu.cn

8th Jiafan Zhuang
Department of Electronic Engineering
Shantou University
Guangdong, China
jfzhuang@stu.edu.cn

9th Zhun Fan
Department of Electronic Engineering
Shantou University
Guangdong, China
zfan@stu.edu.cn

***Abstract*—This paper tackles the challenge of collaborative control in multi-agent systems by introducing a motion control strategy based on a hierarchical gene regulatory network (GRN) model. At the GRN model's upper layer, we define an influence area for each agent, aiding them in making autonomous decisions through the dynamics of these influence areas. The lower layer of the GRN incorporates three core behavioral principles: obstacle avoidance, aggregation, and co-directional movement, enabling self-organized, coordinated movement in multi-agent settings. The effectiveness of our approach is validated through experimental simulations in two distinct environments: a forest and a channel. The experimental results demonstrate the superiority of the proposed method in these environments.**

***Index Terms*—Multi-agent system, Gene regulatory network, Self-organized motion**

I. Introduction

In multi-agent systems, collaborative cooperation allows each agent to perform complex tasks beyond their individual capabilities [1], offering benefits in robustness, scalability, and cost-effectiveness. Effective collaboration requires consideration of local perception, decision-making, and control mechanisms [2]. Researchers often look to natural biological systems for inspiration in these areas. Current motion control methods for multi-agent systems mainly include the artificial potential field method, the virtual structure method, and the leader-follower method.

The artificial potential field method, introduced by Khatib [3], models interaction forces among agents using artificial potential fields. Agents respond to attraction fields for desired positioning and repulsion fields for obstacle avoidance. Wu et al. developed an obstacle modeling method for complex environments [4], reducing the impact of local minima issues. However, this method requires complex strategies for balancing competition and collaboration in multi-objective tasks, increasing task complexity. Sang et al. [4] proposed a multi-objective artificial potential field method to prevent trapping in local minima, incorporating task goal switching and path planning. This method, however, faces difficulties in managing agent motion in dynamic environments.

The virtual structure method treats a multi-agent system as a single rigid body, with individual agent movements governed by overarching rules. Abbasi et al. developed an improved group control and fuzzy synchronization strategy based on this concept [7]. This approach allows agents to synchronize using precise position measurements, and visual sensors help prevent collisions by assessing relative positions between agents. Zhou et al. [6] applied a virtual structural framework to control a quad-rotor UAV, enabling multi-agent motion control within a local coordinate system. However, these methods typically rely on centralized communication, which burdens the central node and may compromise the system's fault tolerance and robustness.

In the leader-follower model, some agents act as leaders, and the rest follow [8]. Followers maintain a constant offset from the leader, who moves along a predefined trajectory. Chen et al. [9] introduced a predictive performance control strategy for a leader-follower motion control model that relies on relative position data, ensuring coherence in the multi-agent system. However, this model lacks feedback mechanisms,

979-8-3503-3007-6/23/$31.00 ©2023 IEEE

and a leader's failure can have system-wide effects. Yang improved the leader-follower model for better estimation of relative distances and precise leader tracking by followers [10]. However, this method does not address obstacle avoidance for the agents.

Acknowledging the limitations of the aforementioned approaches, this paper presents a new motion control method for multi-agent systems, based on a hierarchical gene regulatory network (GRN) model. In this model, the upper layer network equips each agent with an influence area that integrates environmental information, facilitating autonomous decision-making. The lower layer network manages the agents' movements, ensuring the system's integrity, preventing collisions with obstacles, and enabling coordinated motion within the multi-agent system.

The remainder of this paper is structured as follows: Section II provides a detailed description of the proposed cooperative motion control method with GRN model. Section III demonstrates the effectiveness of our proposed model in various scenarios. Finally, Section IV summarizes the findings and conclusions of this work.

II. GENE REGULATORY NETWORK MODEL

In biology, GRN models are used to simulate the intricate regulatory interactions among genes within a cell. These models illustrate how the protein concentration produced by one gene can activate or inhibit the expression of others. Such interactive linkages create a complex network that regulates gene expression in an organism, thereby influencing behaviors and functions at the cellular level. Drawing inspiration from this concept, this paper introduces a coordinated motion control method for multi-agent systems, utilizing the principles of GRN models.

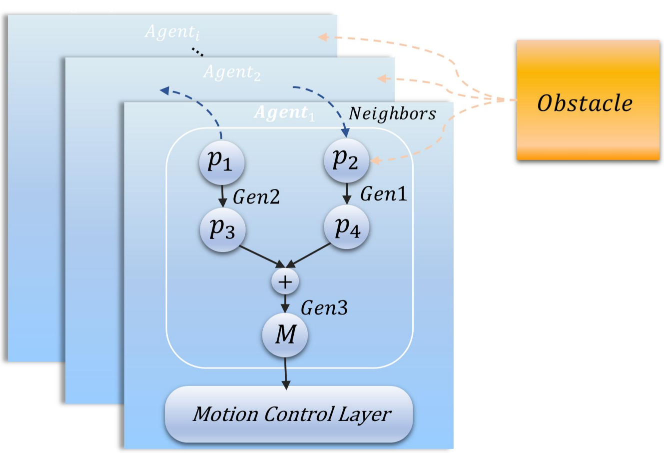


Fig. 1. Hierarchical GRN Model. Each intelligent agent is equipped with a dual-layer GRN model, designed to facilitate both autonomous decision-making and motion control. The model processes input data in the form of the agent's own protein concentration field, denoted as $p1$, and the protein concentration fields of neighboring agents, denoted as $p2$.

The virtual protein concentration fields in a multi-agent system are explained as follows. Each intelligent agent in the system is analogous to a biological cell. The agent's position is represented as a concentration value, $p1$, within a virtual protein concentration field, as shown in Fig. 1. The positions of neighboring agents and obstacles are similarly converted into concentration fields, labeled as $p2$. These fields are received and processed by the agents. When neighbors or obstacles fall within an agent's sensing range, the concentration fields $p1$ and $p2$ activate Genes $Gen1$ and $Gen2$, respectively. This activation leads to the production of new concentration fields $p3$ and $p4$. The combined effect of $p3$ and $p4$ drives the expression of Gene $Gen3$, resulting in the creation of the final concentration field M. Based on the final field M, the agent determines its influence area and makes autonomous decisions, drawing from the state information of this influence area. The decision information is relayed to the lower layer of the network, enabling the agent to adapt its motion in response to the complexities of the environment. The proposed GRN model features a two-layer structure. The following two sections will delve into the specific functions and roles of each layer within this framework.

A. The Upper Layer of the GRN

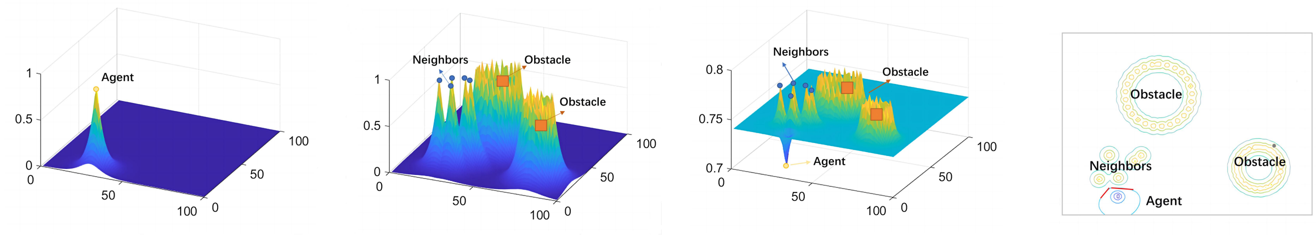
In the upper layer of the GRN, agents create a virtual 3D protein concentration field based on their location. They assign specific concentration values to define their individual influence areas, which are generally depicted as circles centered around each agent. When these agents receive positional information about nearby agents and obstacles (considering obstacle boundaries as equivalent to virtual agents), they calculate their corresponding virtual concentration fields. These calculated fields are then integrated into the upper layer of the GRN. The process for determining the virtual concentration field within the detection range is defined by the following formula:

$$\frac{dp_j}{dt} = \nabla^2 p_j + \gamma_j - p_j \tag{1}$$

$$p = \sum_{j=1}^{N} p_j \tag{2}$$

In this framework, p_j represents the concentration produced by neighbors and obstacles, with j ranging from 1 to N, denoting the number of neighbors and obstacles within the detection range. The expression $\frac{dp_j}{d_t}$ signifies the rate at which the virtual concentration field changes over time t. Notably, ∇^2 is the Laplacian operator which is used to model the spatial diffusion of protein concentration. The term p encompasses the overall concentration field, which is the sum of all neighboring concentration fields.

As depicted in Figure 2, within the upper layer of the GRN model, the concentration fields of agents and their neighbors undergo gene regulation. This gene regulation modifies the state of their respective influence areas. These alterations in the state of the influence areas are then conveyed to the lower layer of the network as input. Such modifications in the states



(a) (b) (c) (d)

Fig. 2. Virtual protein concentration fields. (a). The concentration field of an agent. (b). The concentration fields of neighbors and obstacles. (c). The final concentration field after gene regulation. (d). A 2D plane visualization of the final concentration field. In this representation, blue circles depict the agent's influence area, while a red line signifies areas where the influence is compressed or altered due to external factors.

of the influence areas enable autonomous decision-making by the agents, thereby facilitating the self-organized, coordinated movement of the entire multi-agent system. The formula for this gene regulation is detailed below:

$$\frac{dp_3}{dt} = -p_3 + sig(p_1, \theta_1, k) \tag{3}$$

$$\frac{dp_4}{dt} = -p_4 + [1 - sig\left(p_2, \theta_2, k\right)] \tag{4}$$

$$\frac{dM}{dt} = -M + sig(p_3 + p_4, \theta_3, k) \tag{5}$$

In this model, the agent's concentration field p_1 triggers the expression of Gene $Gen1$, resulting in the generation of concentration field p_3. Simultaneously, the concentration field p_2, originating from neighboring agents, activates Gene $Gen2$, creating concentration field p_4. The final virtual concentration field, represented as M, is produced by the combined effects of p_3 and p_4. In this equation, θ and k serve as adjustment parameters for the sigmoid function, with θ ranging from 0 to 1 and k from 0 to 2. In related research by Fan et al. [12], genetic programming was used to automate the design of GRN topology and the optimization of its structural parameters. As shown in Figure 2(d), the agent's influence area experiences a compression-like transformation. Unlike traditional multi-agent motion control models that depend on absolute distance information, this approach enables autonomous decision-making through the morphological changes within the agent's influence area.

B. The Lower Layer of the GRN

In the lower layer of GRN model, the motion of agent primarily involves three fundamental behavioral principles: obstacle avoidance, aggregation, and co-directional motion.

1) Obstacle Avoidance: When agents are significantly distant from neighbors or obstacles, their influence area maintains a circular shape, centered on the agent. However, as agents get closer to neighbors or obstacles, genetic regulation leads to deformations in their concentration field, which in turn changes the shape of the influence area. To avoid collisions, it is essential for agents to move in the direction opposite to where the deformation occurs. The exact formula for this process is detailed below:

$$v_c = \sum_{k=1}^{N_k} u_{ki} * \left(sig\left(d_{min}, d_{pt}^{ki}, k_1\right) - 1\right) \tag{6}$$

In this model, v_c is defined as the velocity required for collision avoidance. N_k represents the total number of points within the agent's influence area that are subject to compression. The term d_{min} indicates the shortest distance between the agent's influence area and any nearby entity, and it is set to 1. d_{pt}^{ik} refers to the distance from agent i to a specific compressed point k within the influence area. The parameter k_1 serves as an adjustment factor in this context. Furthermore, u_{ki} represents the direction for obstacle avoidance, which is the directional vector leading from the compressed point k within the influence area back to agent i.

2) Aggregation Motion: In situations where the influence area state of the agents remains unchanged, maintaining the multi-agent system's overall integrity becomes crucial. We define the maximum allowable distance between agents as d_{max} and require that d_{max} be less than the agent's sensor range, which is set to 2 in this study. If the distance between an agent and its nearest neighbors exceeds d_{max}, the agent is obliged to adjust its position to be closer to its nearest neighbor, while staying within the sensor's range. The specific formula for this adjustment is as follows:

$$v_a = \sum_{j=1}^{N_j} u_{ij} * \left(sig\left(d_{pt}^{ij}, d_{max}, k_2\right) - 1\right) \tag{7}$$

where v_a represents the aggregation velocity of an agent. N_j indicates the number of neighbors within the sensing range of the agent. d_{pt}^{ij} refers to the distance between agent i and its neighbor j. k_2 is a tuning parameter in the system. u_{ij} denotes the directional vector pointing from agent i to its neighbor j.

3) Co-directional Motion: To achieve synchronized movement in the multi-agent system, a virtual leader is designated at the forefront of the group. The agents follow the movement pattern of this virtual leader. In cases where agents are unable to detect the virtual leader directly, they are instructed to move towards their nearest neighbor, thus ensuring cohesive, co-directional motion throughout the system. The specific formula governing this co-directional motion is as follows:

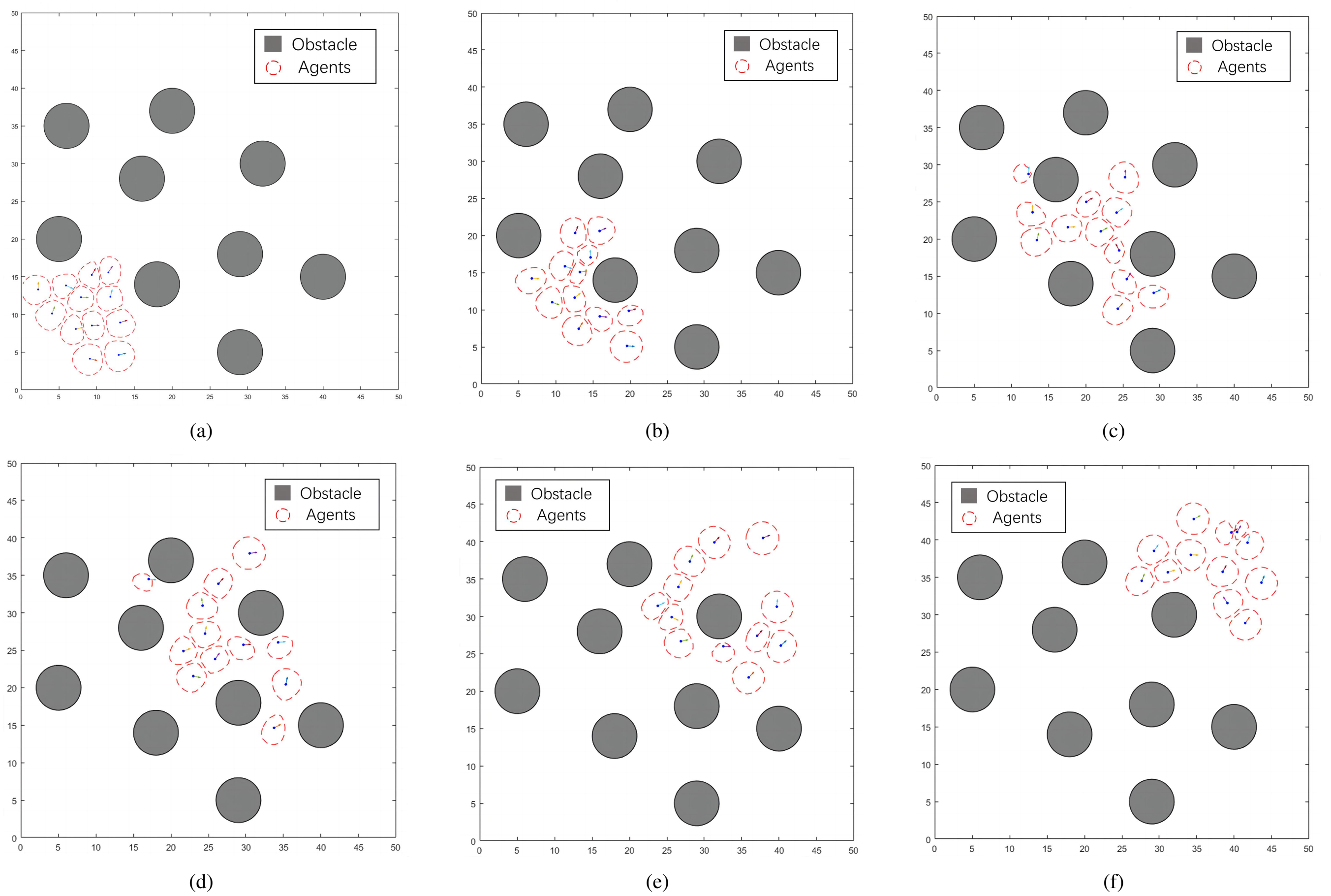


Fig. 3. Motion process of the multi-agent system in a forest-like environment

$$v_p = v * u_{il} \tag{8}$$

where v represents the current velocity of the agent, and $u_{il} = u_i - u_l$ denotes the directional vector from agent i to the virtual leader l.

The proposed motion control model for the GRN lower layer can be summarized as follows:

$$v_f = v_c + v_a + v_p \tag{9}$$

III. EXPERIMENTAL STUDY AND ANALYSIS

To evaluate the effectiveness of the MACM-GRN method introduced in this study, we carried out experiments in two distinct simulated settings: a forest-like environment and a channel-like environment. We then compared the performance of MACM-GRN with that of MACM-III [11].

A. Experimental Setup

The experimental environment is set up with dimensions of 50 × 50 m^2, where 12 agents are randomly positioned in the lower left quadrant of the area. The movement speed of these agents is set at 0.25 m/s, and their sensor detection range is 3 m. Parameters for the upper layer of the GRN are defined as follows: θ_1=0.295, θ_2=0.326, θ_3=0.545, and k=1.

To evaluate the effectiveness of collaboration within the system, a stability metric was introduced. This metric assesses the system's ability to maintain an organized state. The formula for calculating this metric is given as:

$$f_{sta} = \sqrt{\frac{\sum_{i=1}^{N} (d_i - d_{avg})^2}{N}} \tag{10}$$

where N is the number of agents, d_i represents the distance of agent i from the cluster's centroid, and d_{avg} is the average distance of all agents from the centroid. The parameter f_{sta} measures the dispersion of agent positions within the system, with a lower value indicating higher system stability.

Additionally, for the effective operation of the proposed control model, we assume that:

- The multi-agent system is equipped with GPS positioning sensors, which enable agents to rapidly and accurately obtain their own location data as well as the distance to nearby neighbors or obstacles.
- Each agent has a limited communication range, allowing for the exchange of positional and velocity information with adjacent agents.

B. Forest Environment

Figure 3 demonstrates the movement dynamics of a multi-agent system using the MACM-GRN method within a simulated forest environment. In this environment, characterized by dense and complex obstacles, the multi-agent system often fragments into subgroups, which then regroup after successfully navigating the obstacles. This dynamic is evidenced in Fig. 4, where we observe an initial increase in the system's stability during forest traversal, followed by a decrease in stability after the agents have navigated past the obstacles.

In contrast, the MACM-III approach employs an artificial potential field for obstacle avoidance and a virtual leader to guide the co-directional movement of the group. While effective in maintaining low system instability, MACM-III tends to reach local optima in complex scenarios, which can impede the completion of tasks. Fig. 4 further illustrates that agents operating under MACM-III face challenges in finding an exit in environments surrounded by obstacles, highlighting the limitations of this approach in more intricate settings.

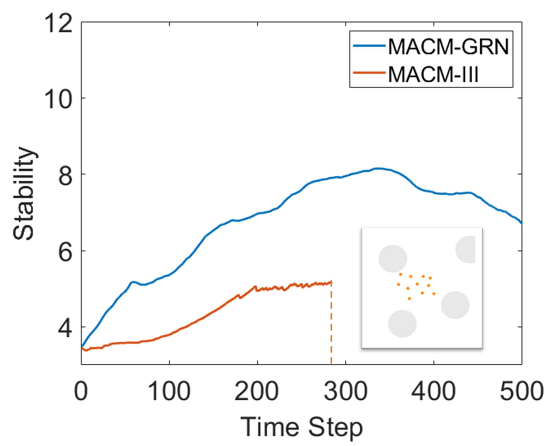


Fig. 4. Stability indicators of MACM-GRN and MACM-III methods in the simulated forest environment

C. Channel Environment

Figure 6 depicts the trajectory of a multi-agent system navigating through a channel environment. Initially positioned at the channel's entrance, the agents are tasked with moving towards the upper right corner to exit. The figure clearly shows that the multi-agent system manages to navigate the channel successfully, reaching their destination without any collisions.

Fig. 5 presents the stability value change curve for the MACM-GRN method. This curve reveals an increase in the system's stability during the initial time steps (0-100), a period when the agents are maneuvering around the channel's corner. The change in the environment contributes to this increase in stability. Subsequently, as the agents enter the straight portion of the channel, the system's stability value becomes more consistent, as indicated in Figs. 6(a) to 6(c). A notable resurgence in stability is observed between time steps 200-300, attributed to the channel's corners becoming narrower, yet the system's stability soon stabilizes once more.

Compared to the MACM-III method, our approach demonstrates improved system stability. This is particularly evident during the navigation of channel corners and in straight channel segments, indicating that the MACM-GRN method better adapts to varying environmental conditions and maintains a more stable configuration of the multi-agent system.

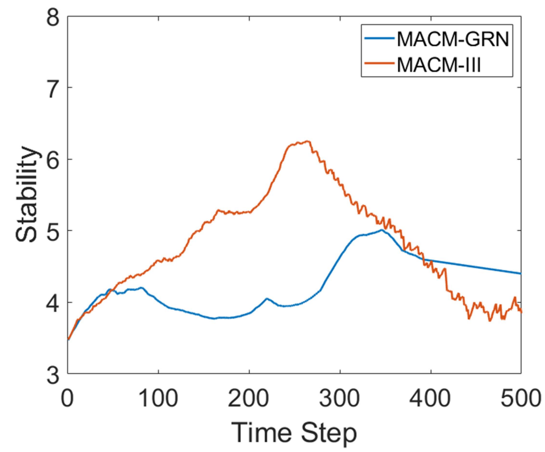


Fig. 5. Stability indicators of MACM-GRN and MACM-III methods in simulated channel environment

IV. Conclusion

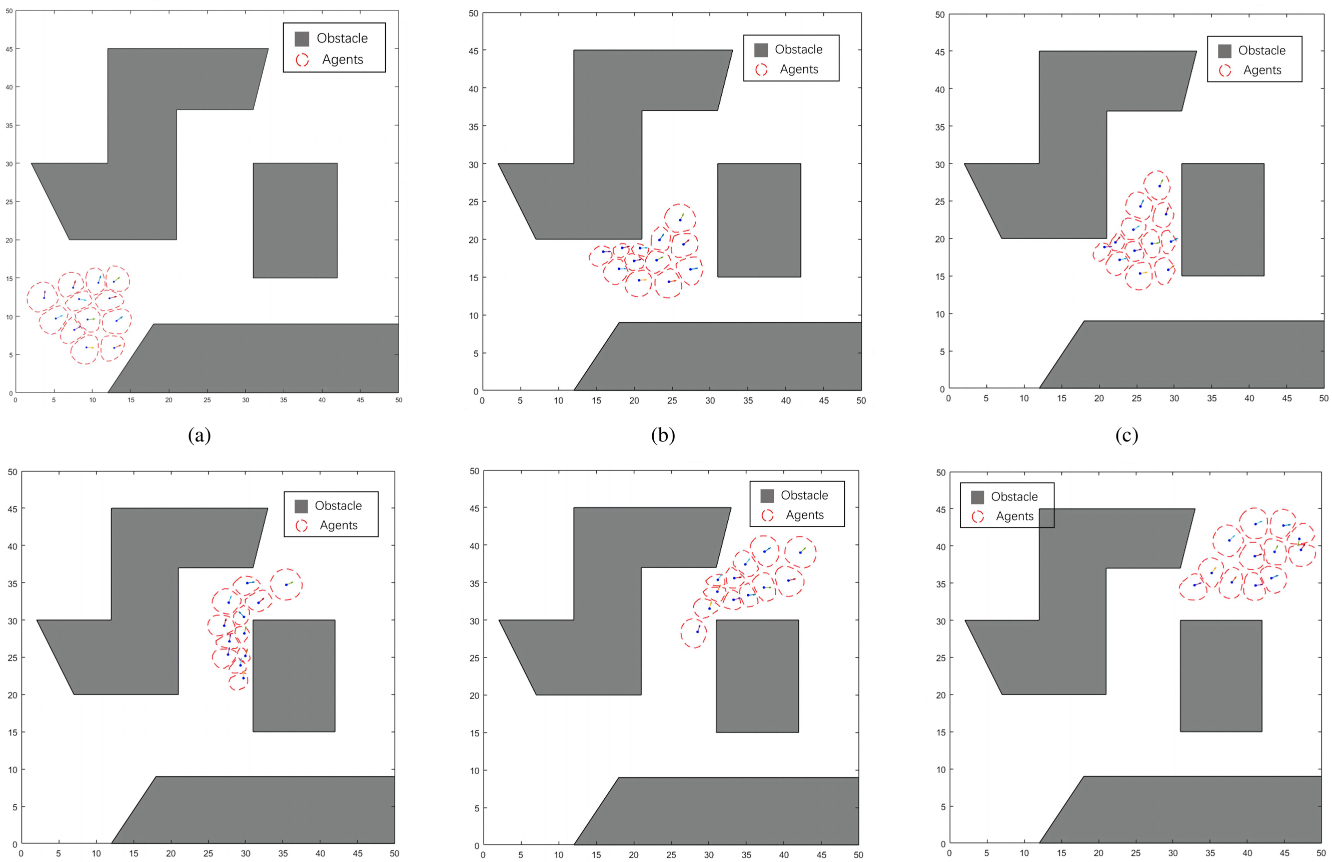
This paper introduces a novel control model for multi-agent systems, inspired by GRNs, designed to facilitate self-organized, coordinated motion in complex environments. Our approach begins with an exploration of the structure and function of GRNs, applying these principles to enable autonomous decision-making and motion control in intelligent agents. We then validate the efficacy of this control method within multi-agent systems. The results of the simulation experiments demonstrate that the controller, grounded in the GRN model, is highly effective and adaptable, successfully achieving coordinated motion control in multi-agent Systems.

Acknowledgment

This work was supported by the National Key Research and Development Program of China (grant numbers 2021ZD0111501 and 2021ZD0111502), the Science and Technology Planning Project of Guangdong Province of China (grant numbers 180917144960530, 2022A1515110660, STKJ2021176 and STKJ2021019), the State Key Lab of Digital Manufacturing Equipment & Technology (grant number DMETKF2019020), STU Scientific Research Foundation for Talents (grant numbers NTF21001 and NTF21052), and the National Natural Science Foundation of China (grant number 62176147),STU Scientific Research Foundation for Talents (grant numbers NTF22030).

References

[1] W. Li, Z. Wang, R. Mai, P. Ren, Q. Zhang, Y. Zhou, N. Xu, J. Zhuang, B. Xin, L. Gao *et al.*, "Modular design automation of the morphologies, controllers, and vision systems for intelligent robots: a survey," *Visual Intelligence*, vol. 1, no. 1, p. 2, 2023.



(a) (b) (c)

(d) (e) (f)

Fig. 6. Motion process of the multi-agent system in a channel-like environment

[2] M. Wu, X. Zhu, L. Ma, W. Bao, Z. Fan, and Y. Jin, "Multi-robot target entrapment using cooperative hierarchical gene regulatory network," *Swarm and evolutionary computation*, vol. 80, p. 101310, 2023.

[3] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The international journal of robotics research*, vol. 5, no. 1, pp. 90–98, 1986.

[4] Z. Wu, G. Hu, L. Feng, J. Wu, and S. Liu, "Collision avoidance for mobile robots based on artificial potential field and obstacle envelope modelling," *Assembly Automation*, vol. 36, no. 3, pp. 318–332, 2016.

[5] H. Sang, Y. You, X. Sun, Y. Zhou, and F. Liu, "The hybrid path planning algorithm based on improved a* and artificial potential field for unmanned surface vehicle formations," *Ocean Engineering*, vol. 223, p. 108709, 2021.

[6] D. Zhou, Z. Wang, and M. Schwager, "Agile coordination and assistive collision avoidance for quadrotor swarms using virtual structures," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 916–923, 2018.

[7] Y. Abbasi, S. A. A. Moosavian, and A. B. Novinzadeh, "Formation control of aerial robots using virtual structure and new fuzzy-based self-tuning synchronization," *Transactions of the Institute of Measurement and Control*, vol. 39, no. 12, pp. 1906–1919, 2017.

[8] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015.

[9] F. Chen and D. V. Dimarogonas, "Leader–follower formation control with prescribed performance guarantees," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 1, pp. 450–461, 2020.

[10] Z. Yang, S. Zhu, C. Chen, G. Feng, and X. Guan, "Leader-follower formation control of nonholonomic mobile robots with bearing-only measurements," *Journal of the Franklin Institute*, vol. 357, no. 3, pp. 1628–1643, 2020.

[11] W. Li, Q. Zhang, Z. Wang, P. Ren, N. Xu, D. Wang, and Z. Fan, "A swarm control model based on individual information interaction," in *2022 8th International Conference on Big Data and Information Analytics (BigDIA)*. IEEE, 2022, pp. 61–66.

[12] Z. Fan, Z. Wang, W. Li, X. Zhu, B. Hu, A.-M. Zou, W. Bao, M. Gu, Z. Hao, and Y. Jin, "Automated pattern generation for swarm robots using constrained multi-objective genetic programming," *Swarm and Evolutionary Computation*, p. 101337, 2023.