Carpathian Journal of Electronic and Computer Engineering 14/2 (2021) 1-5

DOI: 10.2478/cjece-2021-0006

A Survey on Automatic Design Methods for Swarm Robotics Systems

1st Alaa Iskandar
Faculty of Mechanical Engineering and Informatics
University of Miskolc
Miskolc city, Hungary
iskandar.alaa@student.uni-miskolc.hu

2nd Béla Kovács
Faculty of Mechanical Engineering and Informatics
University of Miskolc
Miskolc city, Hungary
matmn@uni-miskolc.hu

***Abstract*— Swarm robots are a branch of robotics that draws inspiration from biological swarms to mimic their collective behavior. Automatic design methods are part of swarm engineering, depend on artificial intelligence algorithms to produce the collective behavior of robots. In general, they follow two-approach evolutionary algorithms like practical swarm optimization and reinforcement learning. This paper studies these approaches, illustrating the effect of modifications and enhancements of algorithms for both directions, showing important parameters considered for the best performance of the swarm, and explaining the methods and advantages of using deep learning to reinforcement learning.**

Keywords*— Swarm robotics, Swarm engineering, particles Swarm optimization PSO, Bacterial Foraging BFOA, Reinforcement learning, Deep learning, Policy gradient PG, Deep-Q-network DQN.

I. Introduction

Swarm robotic systems are models that mimic biological beings' social behaviors when they form groups like bees, ants, fish, and others, as shown in fig 1. Building these models aims to take advantage of the benefits of teamwork, and it is well-known that working in a group is better than individual work. They are commonly used in searching and rescue processes, surveillance, and others. However, the single robot is undesired to execute these tasks; it needs more time, and failure will stop the system from working. Swarm robotics is a group of simple and small robots. They have small capabilities and a local sense of the environment. They interact with each other and with the environment to achieve the desired task, fig 1. There are two kinds: homogenous; the robots are similar in structure or heterogeneous when there are differences in the design. Most of the studies focus on the homogenous one [1], [2]. The most crucial feature of swarm systems is decentralized control architecture. For that, robots should be autonomous and make their own decisions by interacting with the environment, modifying it, and cooperating features between them to tackle the task.

The main advantages are obtained are robustness, scalability, flexibility [3]. Robustness is concerned with dealing with losses the individuals, scalability to deal with various group sizes, and flexibility to deal with different environments.

Robots in swarms work together and cooperate to do the required task. The entire behavior of the group is called collective behavior. There are many tasks swarm systems can do; they are grouped in sets[4], as shown in fig. 2.

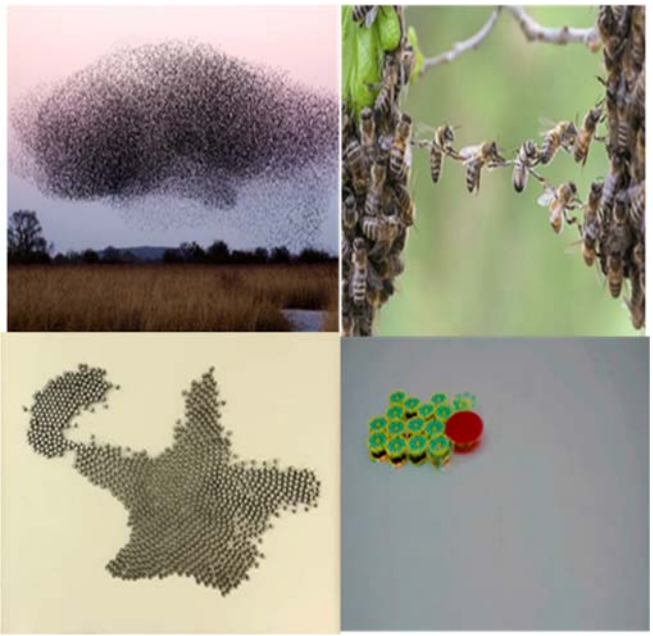

Fig. 1: Biological swarms and Robots swarm.

II. Swarm Engineering

Swarm engineering is the systematic application of scientific and technical knowledge to build a swarm robotic system. Swarm engineering as a term was introduced by Kazadi [5]. Swarm engineering illustrates the stages which we should follow to produce a swarm robotics system. It begins from building the model, specifies requirements, design, realizes, verifies, validates, operates, and maintain a swarm intelligence system [2] [1]. Lots of papers focus on the design and analysis level, fig.3. For analysis level, it goes through three phases. The microscopic phase analysis the performance of every robot individually, the macroscopic phase analysis the performance of the entire system to evaluate the collective behavior, and the final phase is the actual robot analysis in a natural experiment. Design methods are concerned with generating the rules that drive each robot in the swarm to achieve the desired collective behavior. There are two methods: behavior-based design method, the designer put the rules manual according to the experiment, and automatic design methods, which depend on artificial intelligence to produce rules automatically.

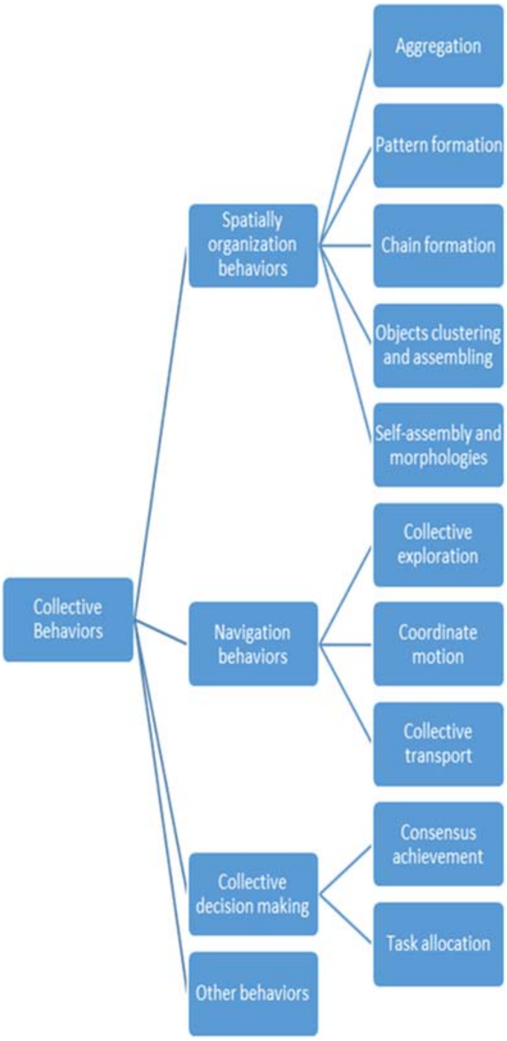

Fig. 2: Collective behaviors types.

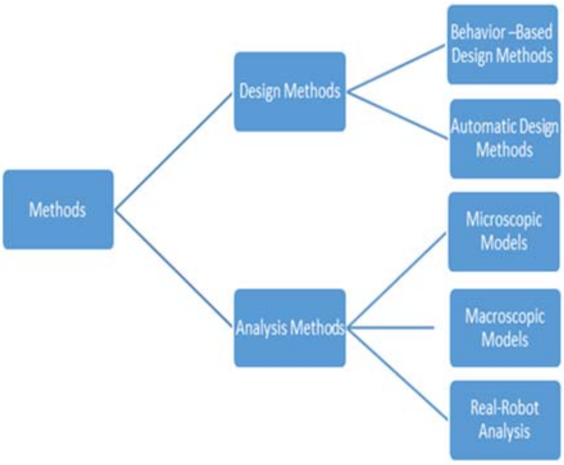

Fig. 3: Swarm engineering, design and an analysis.

III. Automatic Design Methods

Automatic methods use intelligent algorithms to produce the behaviors for all system entities without the developer's explicit interior. The most used approaches are evolutionary algorithms like practical swarm optimization PSO and reinforcement learning algorithms which are a field of machine learning. Furthermore, it is helpful to depend on artificial intelligence algorithms to reduce developers' computations and efforts. This paper will illustrate the most used algorithms in automatic design methods for swarm robotics systems and show the modifications on these algorithms and how they affect the collective behavior of the swarm.

A. Particles swarm optimization PSO :

Particles swarm optimization is one of the well-regarded algorithms in the literature of optimization and has been widely used in various science and industry fields. The PSO mimics the navigation and foraging of a flock of birds or schools of fishes. PSO was proposed by Eberhart and Kennedy [6]. Particles search randomly in an area for the best solution, every particle is initialized by a random one, and then they will search for the best by updating their solutions during every iteration; every particle has two basic values that should be recorded during the process:

pBest: The best fitness value particle has at the moment. gBest: The best fitness value among the swarm.

After finding previous values, particles update the velocity and the position according to the equations (1) and (2), fig.4 illustrates the producers of PSO [7]:

$$v_{i,t+1}^{d} = \omega * v_{i,t}^{d} + c_1 * rand_i() * \left(P_{i,t}^{d} - x_{i,t}^{d}\right) + c_2 * rand_i() * \left(p_g^{d} - x_{i,t}^{d}\right) \quad (1)$$

$$x_{i,t+1}^{d} = x_{i,t}^{d} + v_{i,t+1}^{d} \quad (2)$$

$v_{i,t+1}^{d}$: Velocity of particle i at t+1 in d dimensions.

$v_{i,t}^{d}$: Velocity of particle i at t.

$P_{i,t}^{d}$: The best fitness value particle has at the moment.

p_g^{d} : The best fitness value among the swarm.

Ω: Inertia weight.

c_1: Cognitive learning factor.

c_2: Social learning factor.

$x_{i,t}^{d}$: position of particle i at t moment.

To recap, PSO has three repetitive steps: Calculation of the fitness value for each particle i, update pBest and gBest, and update the velocity for each particle.

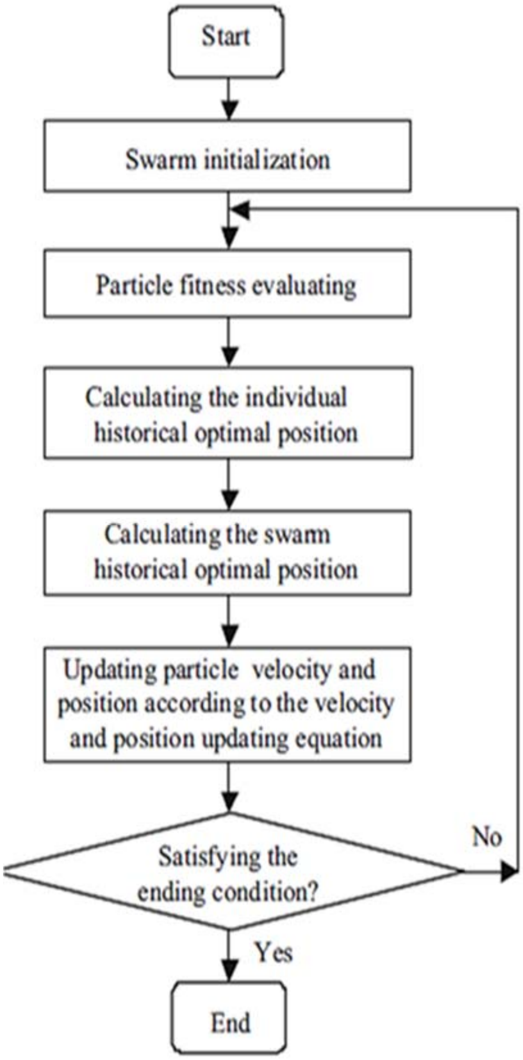
Start
Swarm initialization
Particle fitness evaluating
Calculating the individual historical optimal position
Calculating the swarm historical optimal position
Updating particle velocity and position according to the velocity and position updating equation
Satisfying the ending condition?
No
Yes
End

Fig. 4: PSO Flowchart.

PSO in swarm robotic systems:

Table 1 shows a mapping between PSO terms and swarms robotic system for applying PSO to the system. Synchronous particle swarm optimization SPSO means updating pBest, gBest after evaluating fitness value for all particles. In another way, we can update these values for every particle after its fitness evaluation. This method is called asynchronous particles swarm optimization ASPSO. It is better to use it when some information is missed for neighbors. In paper [8], after applying the two methods SPSO, APSO on the same swarm system, it is noticeable that there are similarities in results. Still, APSO is slower than SPSO for convergence. However, it is not a disadvantage because it allows the particles to discover the search space. In addition to that, APSO is preferred for swarm systems. It will enable efficient usage of the robot's PSO is considered an efficient evolutionary algorithm to plan the path for swarm robots; for example, in paper [9], fuzzy logic is used to update the velocity of particles, making the system more reliable and faster to reach the target. Another modification was implemented in [10] to find the shortest path. It used another evolutionary algorithm called Bacterial Foraging BFOA, introduced by Passino [11] besides PSO.

Table 1: Mapping between PSO and swarm robots system.

Swarm Robotic Search	PSO
Robots	Particles
Signal detection	Fitness evaluation
Relative localization	Absolute localization
Path planning	Particle update
Continuous control	Iteration
Local communication-based on physical distance	Global communication-based on fixed neighborhood

B. Reinforcement Learning RL:

As mentioned before, according to the developments in machine learning and deep learning field, it is efficient to apply these techniques in swarm robotic systems, especially reinforcement learning and deep neural networks, to reduce the number of needed computations. Many approaches are used to make the robot learn the behavior in its swarm by trail-and-error by receiving negative or positive feedback from the environment to evaluate the behavior. For example, as shown in fig. 5, each robot gets a state from the environment by sensing St. According to this state, the robot will do an action At to move the environment to a new state, St+1. The action will be evaluated by giving the robot reward or punishment R. The robot will learn its behavior by taking the behavior which achieves the highest cumulative rewards.

So RL aims to maximize the expected cumulative rewards Eq.3. In other words, the aim is to map between states and actions. This is called Policy π. Moreover, it is deterministic, which means there is one action for every state, or it is stochastic. Instead of one action for each state, there is a distributed probabilities function of actions [12].

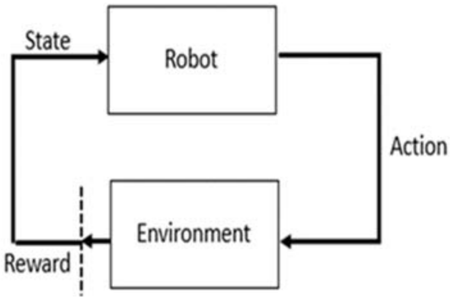

Fig. 5: Reinforcement learning approach.

The process defined by S, A, P, r, γ, T.
Policy π: S×A   R⩾0
Expected Reward $E_{\tau} = \sum_{t=0}^{T} \gamma^{t}\, r(s_t, a_t)$ (3)
S: State, A: Action.
P: probabilities of transition.
r: Reward.
T: number of the taken actions during one episode.

There are two ways to obtain optimal policy: policy-based methods and value-based methods. For policy-based methods, there is a function policy that maps directly between actions and states. In general, the neural network is used to produce policy functions. It takes states as input and actions as output. This algorithm is called Policy Gradient PG, as shown in fig 6. In Value-based methods, a value function calculates the Q value for each state-action and then chooses actions that lead to the highest value states. It is common to use a neural network to train the robot to calculate the Q value, so the neural network takes the state as input, and the output is the Q value. The RL algorithm chooses the best actions according to calculated Q values. This architecture is called deep Q network DQN, as shown in fig 7. [13], [14], [15].

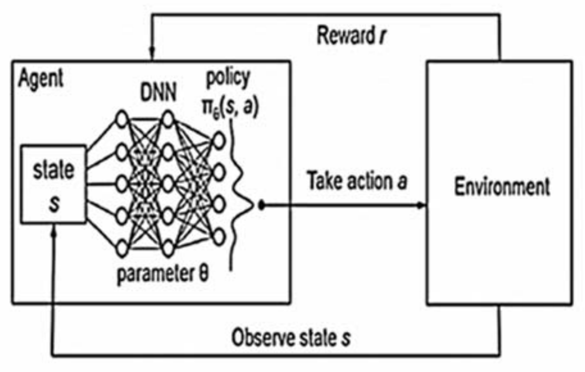

Fig. 6: Policy gradient architecture.

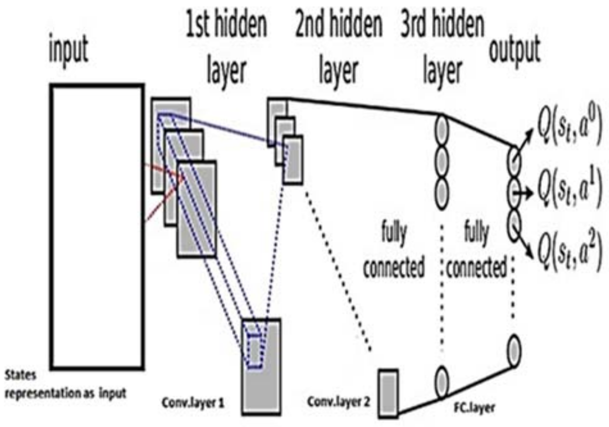

Fig. 7: Deep-Q network architecture.

DEEP REINFORCEMENT LEARNING IN SWARM ROBOTIC SYSTEMS.

Foraging task means that swarm of robots should transport items from sources to a sink, DQN is used to learn swarm of robots to perform foraging task. In [16] used this approach and compared the results after adding many modifications in the architecture of DQN, like freezing the target network. To how freezing is done depends on the error function during the learning process. The error function is used in the neural network by calculating the square of the difference between the target and Q values. These two values change simultaneously, so that will cause a high variance, so it is better to copy the parameters for the current network to the target network every specific time step. Another modification is to use two neural networks to reduce the overestimation. It is called double DQN DDQN; also, it is possible to use future rewards in the N-step Q network NSQ [16]. This shows that freezing the target network and adding NSQ with DDQN will maximize the expected reward and improve and fasten the learning process. Another technique used to enhance the performance of the swarm robotic system is to use shared memory for all robots, as shown in fig 8. All robots store their experiences in the memory, and all of them can access this memory and invest it during the training process. It is noticeable that increasing the number of robots who share this memory will increase learning speed and reduce the variance [17], [18].

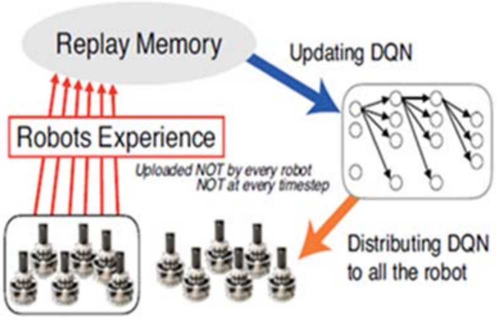

Fig. 8: Sharing memory for swarm robots.

The most crucial parameter in RL algorithms is the reward choosing R affects the swarm's collective behavior. For example, authors in [19] studied swarm behavior for a swarm consisting of blue and yellow robots. Every group should learn how to reach its destination without any collision, the study was conducted with three sets of R as shown in table 2, and the environment has two cases, free and with obstacles, as shown in fig. 9.

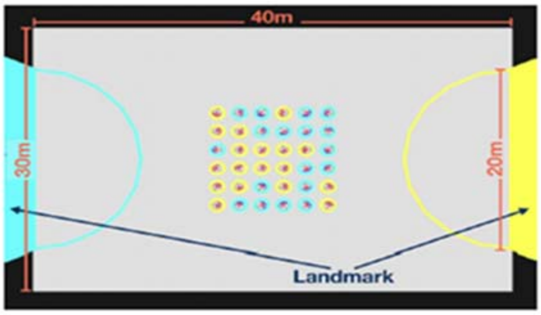

Fig. 9: Proposed swarm system to study R.

Table 2: Rewards sets.

Reward set	*Des.r_d*	*Lan.r_l*	*Fel.r_f*	*Col.r_c*
A	*5*	*5*	*1*	*-1*
B	*5*	*1*	*5*	*-1*
C	*5*	*1*	*1*	*-5*

Des.rd: When the robot arrives at its destination.

Lan.rl: The captured image by camera contains more pixels related to the destination than the previous one.

Fel.rf: The captured image contains more pixels, indicating that the robot is with his group.

Col.rc: punishment for collision with other robots or with obstacles.

The study shows that when robots take severe punishment, that will affect the path planning to reach their destination because robots will focus on avoiding obstacles more than getting their goal. Although, in general, the R-value should be chosen to motivate the swarm entities to reach their destination, lots of studies [18] [16] neglect the punishment by adding R equal to zero. Many improvements are added to traditional RL algorithms like the Sequential Q-learning algorithm based on knowledge sharing to increase the number of robots [20].Multi-agent deep deterministic policy algorithm with a proposed experience sample optimizer for exploration task [21]. Using GQ (λ) Reinforcement Learning for formation control for swarm robots [22]. Bilateral Biased Neighbors-Sharing Cooperative Reinforcement Learning (BBNS-CRL) method is presented to accelerate the learning process by integrating the neighboring robots' knowledge with local knowledge [23].

IV. Conclusion

In this survey, swarm engineering term is introduced by focusing on automatic design methods that depend on artificial intelligence to produce the required behavior for each swarm robot, making it learn from the environment. In general, automated methods have two directions, evolutionary methods like PSO, Bacterial foraging, and others, and reinforcement learning RL, which is more reliable and needs less data to make swarm learn. Lots of modifications were applied to the two approaches. For example, fuzzification of some parameters of PSO algorithms enhances the learning speed. Furthermore, they were using deep learning in RL to reduce the number of computations. Other enhancements were found in the neural network architecture to solve the problems of variance and overestimation and illustrate how modifications affect the swarm behavior. Most of them give a good speed of learning process and a reliable swarm system, but there is no precise formula to model swarm. It still depends on the experiment. Further, this study can be extended for the development of swarm robots algorithms to reduce the problems and obstacles to achieving the swarm's collective behavior.

References

[1] A. S. A. B. H. V. Seeja Ga, "A Survey on Swarm Robotic Modeling, Analysis and Hardware," in International Conference on Robotics and Smart Manufacturing (RoSMa2018), Chennai,India, 2018.

[2] M. U. ,. M. S. a. W. E. Melanie Schranz, "Swarm Robotic Behaviors and current applications," Frontiers in Robotics and AI, 2020.

[3] E. F. M. B. &. M. D. Manuele Brambilla, "Swarm robotics: a review from the swarm engineering perspective," Swarm Intelligence volume , pp. 1-41, 2013.

[4] L. Bayındır, "A review of swarm robotics tasks," Neurocomputing, pp. 292-321, 2016.

[5] S. T. Kazadi, "Swarm engineering," California Institute of Technology, California,USA, 2000.

[6] R. E. J. Kennedy, "Particle swarm optimization," in Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, 1995.

[7] [7] D. T. &. L. L. Dongshu Wang, "Particle swarm optimization algorithm: an overview.," Soft Computing, pp. 387-408, 2018.

[8] N. A. a. Z. I. Ab Aziz, "Asynchronous particle swarm optimization for swarm robotics," Procedia Engineering, pp. 951-957, 2012.

[9] G. K., L. L. ,& S. D. "Collective robotic search using hybrid techniques :Fuzzy logic and swarm intelligence inspired by nature," Engineering Applications of Artificial Intelligence, pp. 431-441, 2009

[10] M. M. S. B. Dibyendu Roy1, "Study of Formation Control and Obstacle Avoidance of Swarm Robots using Evolutionary Algorithms," in 2016 IEEE International Conference on Systems, Man, and Cybernetics • SMC, Budapest,Hungary, 2016.

[11] K. K. S. A. A. R. V. G. B. G. H. Volodymyr Mnih, "Human-level control through deep reinforcement," Nature, pp. 529-533, 2015.

[12] A. G. B. Richard S. Sutton, Reinforcement learning: An introduction, MIT press, 2018.

[13] K. K. S. A. A. R. V. G. B. G. H. Volodymyr Mnih, "Human-level control through deep reinforcement," Nature, pp. 529-533, 2015.

[14] X. N. M. H. K. O. Z. C. Yufei Wei, "Developing End-to-End Control Policies for Robotic Swarms Using Deep Q-learning," Journal of Advanced Computational Intelligence and Intelligent Informatics, 2019.

[15] T. Zhang and H. Mo, "Reinforcement learning for robot research: A comprehensive review and open issues," International Journal of Advanced Robotic Systems, 2021.

[16] Y. L. ,. Z. H. K. O. Boyin Jin, "Generating collective foraging behavior for robotic swarm using deep reinforcement learning," Artificial Life and Robotics, pp. 588-595, 2020.

[17] T. Y. Kazuhiro Ohkura, "Collective Behavior Acquisition of Real Robotic Swarms using Deep Reinforcement Learning," in 2018 Second IEEE International Conference on Robotic Computing, 2018.

[18] T. Yasuda and a. K. Ohkura, "Sharing Experience for Behavior Generation of Real Swarm Robot Systems Using Deep Reinforcement Learning," Journal of Robotics andMechatronics, 2019.

[19] Y. Wei, X. Nie, M. Hiraga, K. Ohkura and Z. Car, "Developing End-to-End Control Policies for Robotic Swarms Using Deep Q-learning," Journal of Advanced Computational Intelligence Vol.23 No.5, 2019.

[20] Y. Song, Y. Li, X. Wang, X. Ma and J. Ruan, "An Improved Reinforcement Learning Algorithm for Cooperative Behaviors of Mobile Robots," Journal of Control Science and Engineering, 2014.

[21] Y. Huang, S. Wu, Z. Mu, X. Long, S. Chu and G. Zhao, "A Multi-agent Reinforcement Learning Method for Swarm Robots in Space Collaborative Exploration," in International Conference on Control, Automation and Robotics (ICCAR), Singapore, 2020.

[22] M. Knopp, C. Aykın, J. Feldmaier and H. Shen, "Formation Control using GQ(λ) Reinforcement Learning," in IEEE International Workshop on Robot and Human Communication (ROMAN), Lisbon, Portugal, 2017.

[23] J. Wu, C. Fan and G. Wu, "Multi-agent Reinforcement Learning with Biased Experience Sharing in Swarm-robotics Domain," in IEEE International Conference on Real-time Computing and Robotics , Irkutsk, Russia, 2019.