Multi-Robot Formation Control using Collective Behavior Model and Reinforcement Learning

Jung-Chun Liu and Tsung-Te Liu
Department of Electrical Engineering
National Taiwan University, Taipei, Taiwan
Email: r10921043@ntu.edu.tw

***Abstract*—A multi-robot system has advantages in complex tasks, where formation control is one of the most critical and fundamental tasks. For small-sized, autonomous, and enduring robots, realizing high energy and area efficiency is extremely important. This paper presents an approach that combines swarm intelligence and reinforcement learning to realize accurate and reliable operations. An area-energy-efficient hardware architecture is proposed to perform formation control in a distributed robotic system. The proposed system demonstrates substantially lower cost and power consumption when compared with the state-of-the-art designs.**

I. Introduction

A multi-robot system is a collection of several robots cooperating to accomplish complex tasks, such as pattern formation, area exploration, and collaborative path planning. Among these tasks, formation control is one of the most fundamental tasks in multi-robot systems. Several distributed robots move toward the goal or perform tasks while maintaining a swarm or a desired pattern.

One promising approach to realize reliable formation control is utilizing swarm intelligence. Swarm intelligence is based on low-level individual intelligence, which can perform high-level collective behavior. It has been used in the implementation of multi-robot systems as swarm robotics. In swarm robotics, robots are usually decentralized and allowed to act as individuals while interacting with others and solving problems as a group. It is more efficient to build multiple simple robots instead of building one robust but costly robot. Therefore, swarm robotic systems are regarded as fault-tolerant and robust multi-robot systems [1].

Inspired by the collective behavior of animal flocks, various computational mathematical models and algorithms have been developed. Boids model was proposed by Reynolds in 1986, which aims to simulate the flocking behavior in birds and other animals [2]. The mathematical model had been extended to apply different flock control mechanisms. Couzin presented a self-organizing formation model and surveyed the spatial dynamics of animal groups in three-dimensional space [3]. However, the agent's movement seems disorderly and unexpected in the basic model, much like a realistic flock in nature. Since security and safety are critical issues in the area of industrial robotics and human-robot interaction, robotic systems are supposed to accomplish tasks accurately. Moreover, although the robot can efficiently compute the next decision in real-time since the algorithm is determined in these model-based methods, the overall performance heavily relies on model architecture, and it is more vulnerable in a real-world environment due to uncertainty [4].

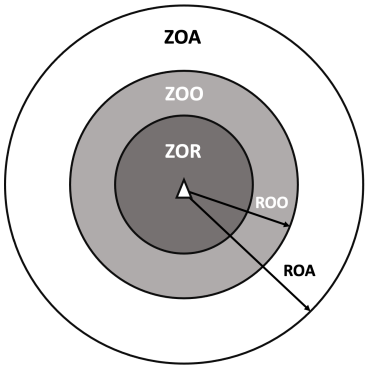


Fig. 1. Metric distance model

On the other hand, learning-based methods allow agents to learn and optimize their policies by interacting with a dynamic environment. In recent years, reinforcement learning has attracted a lot of attention. It is commonly used for solving sequential decision-making problems. Some reinforcement learning methodologies are proposed for the formation control of robots [4] [5]. However, the error of the methods is still significant compared to traditional methods [6] [7].

Finally, energy-efficient hardware designs are critical for small, autonomous swarm robots, which are expected to interact continuously with others in a real-time environment [8]. In this work, we proposed a multi-robot formation control algorithm combining the features of the collective behavior model and reinforcement learning. The proposed approach takes advantage of a model-based method that ensures the accuracy of control with a low error rate while retaining the adaptive characteristics of swarms. Moreover, the proposed rules of the model are hardware-friendly and easy to be implemented at a low cost. Finally, the robots employing the proposed approach are able to adapt to the environment reliably due to the characteristic of the learning-based method with significantly reduced area and power compared to the previous works.

II. Algorithm Design

A. Metric distance model

In the Boids model, collective behavior relies on the local interaction of individual agents. Each agent follows only three simple rules [2]:

- separation: avoid collision with neighbors
- alignment: match speed and direction with neighbors
- cohesion: stay close to neighbors

Couzin used the metric distance model to implement the rules of the Boids model by separating the behaviors into three concentric zones [3]. As shown in Fig.1, the nearest zone to the individual is the zone of repulsion (zor), the middle zone is the zone of orientation (zoo), and the furthest zone to the individual is the zone of attraction (zoa). The radius of repulsion (ror) and the radius of orientation (roo) are the outer radii of zor and zoo. Which rule will emerge depends on the interaction zone where its neighbor is. If there is a neighbor in the zoa, the agent attempts to stay close to the neighbor. Finally, if there is a neighbor in the zoo, the agent will align itself with the neighbor [9].

B. Reinforcement Learning

In this work, we use approximate reinforcement learning to cope with the real-world environment. The method consists of five components for its update rule: the current state and action s, a, the immediate reward r, and the next state and action s', a'. The update rule of the method is:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[R(s) + \gamma max_a Q(s',a') - Q(s,a)]$$

The Q function estimates the Q-value of actions. The robot will take the action with the largest Q-value. The parameters of Q function are updated based on the action that the robot takes from the next state with the current policy. Traditional reinforcement learning has dealt with discrete and finite-state/action spaces. This work proposes a method of using function approximation that allows robots to learn with continuous states in real-world situations. In our approach, discrete action space is sufficient due to the limited acceleration for smooth movement.

C. System Integration of the Metric Distance Model and Reinforcement Learning

We propose an algorithm of multi-robot formation control based on the swarm behaviors and approximation reinforcement learning. First, the system calculates the error with the rules in the metric distance model, while linear approximation is used to evaluate Q-values. Then, the Q-learning module finds the best action and adjusts the policy according to Q-values and reward.

We use the metric distance model that is more intuitive for robot sensors. Narrowing the boundaries of zoo improves the accuracy performance on aligning with neighbors at the desired distance. Given a distance between agents D_A, a zoo width δ, ror and roo are $D_A - \delta/2$, $D_A + \delta/2$ respectively.

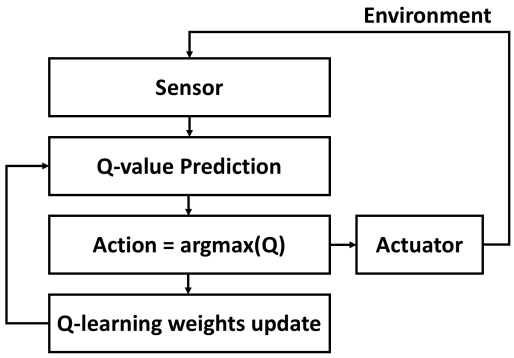


Fig. 2. Reinforcement learning implementation in this work

To promote the swarm moving toward the goal, we add the velocity rule and the goal rule. For the reward and Q-value function, we mathematically formulated the rules for determining actions at the given states:

$$e_v = |V_{ref} - v_i|$$

$$e_{goal} = 1 - cos\phi_{i,goal}(s),\ -\pi < \phi_{i,goal}(s) \leq \pi$$

$$e_{zor,i,j} = \begin{cases} D_A - d_{i,j}(s) & d_{i,j}(s) < D_A - \delta/2 \\ 0 & otherwise \end{cases}$$

$$e_{zoa,i,j} = \begin{cases} d_{i,j}(s) - D_A & d_{i,j}(s) > D_A + \delta/2 \\ 0 & otherwise \end{cases}$$

$$e_{zoo,i,j} = \begin{cases} \phi_{i,j}(s) & D_A - \delta/2 < d_{i,j}(s) < D_A + \delta/2 \\ 0 & otherwise \end{cases}$$

$$e_{obs} = \begin{cases} cos\phi_{i,obs}(s)/d_{i,j}(s) & -\pi/2 < \phi_{i,obs}(s) \leq \pi/2 \\ 0 & otherwise \end{cases}$$

where $d_{i,j}(s)$ denotes the distance between agents i and j at state s, $\phi_{i,j}(s)$ denotes the difference of direction between agents i and j at state s. $\phi_{i,goal}(s)$ denotes the difference angle between agent i toward goal and direction of agent i.

The Q-function is defined as

$$Q(s,a) = w_{zor} \sum e_{zor,i,j} + w_{zoa} \sum e_{zoa,i,j} + w_{zoo} \sum e_{zoo,i,j}(s) + w_v e_v + w_{goal} e_{goal}$$

The algorithm uses the policy gradient method to minimize the function. For every iteration, after receiving an input signal, the agent predicts the Q-function for each action and updates its policy by evaluating the actual error for the last prediction. The RL technique has been illustrated in Fig. 2.

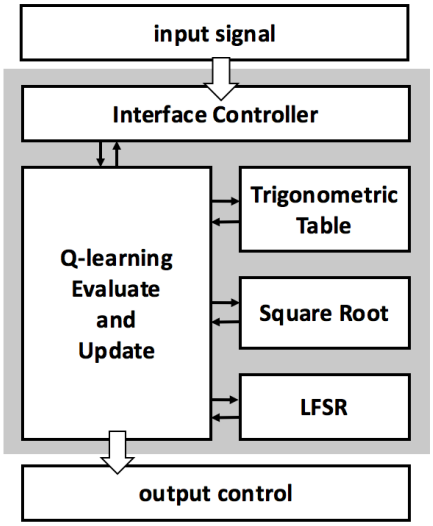


Fig. 3. System architecture

III. System Architecture

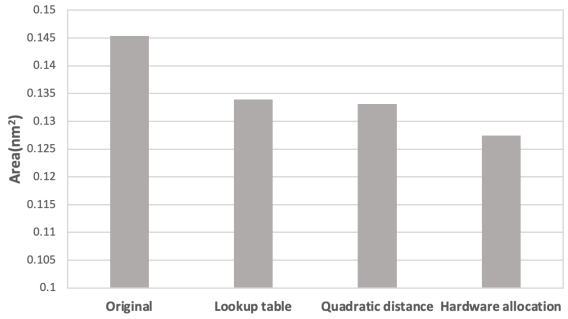
The proposed system architecture is shown in Fig.3. The system uses 16-bit integer resolution for input and output computation. Input signals from sensors include self velocity, relative position and orientation to other robots, and the desired goal. During an evaluation, the proposed system simulates each action and calculates Q-value. Next, the system chooses $\arg\max_a(Q)$ or a random action according to a ϵ-greedy policy. Finally, it updates weights by the gradient descent method.

The proposed algorithm is intuitive and friendly to hardware implementation. It requires only low-complexity linear arithmetic operations for the primary components of computation. The trigonometric and square root modules are employed for the calculation of simulated distance. The trigonometric module is implemented using a lookup table and the square root module uses a bit-wise binary search method, both of which help minimize the overall latency. To further lower the cycle time,$e_{zor,i,j}$ and $e_{zoa,i,j}$ can be modified to quadratic function as:

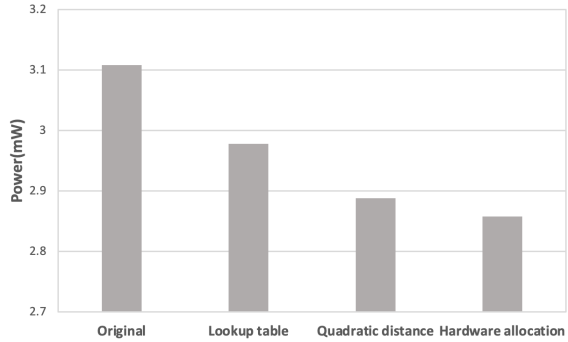
$$e_{zor,i,j} = (D_A - d_{i,j}(s))^2, \quad d_{i,j}(s) < D_A - \delta/2$$

$$e_{zoa,i,j} = (d_{i,j}(s) - D_A)^2, \quad d_{i,j}(s) > D_A + \delta/2$$

This modification eliminates the need for complex square root operation, which significantly saves average cycle time by 60% while maintaining overall accuracy and performance. Finally, since each neighbor meets only one condition of three rules, the arithmetic modules can be dynamically allocated and shared to further reduce the area cost by 11%. Fig. 4 summarizes the corresponding area and power reduction resulting from different optimization techniques, respectively.



(a) Area



(b) Power

Fig. 4. Hardware optimization

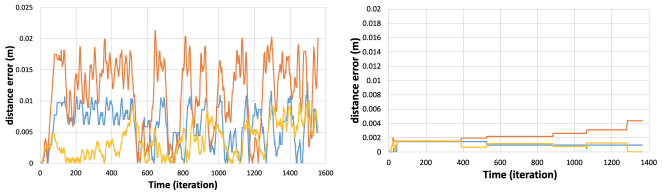
IV. Result

A. System verification

In the experiments, the distance between agents is 1 m, and the maximum velocity is 1.2m/s. The agents use their optimal policy during performance analysis. In the first experiment, we compare the original Boids model and our algorithm with the metric distance model. The original Boids mathematical model is unstable because of the difficulty in finding a balance between the rules. Our method realizes a state-of-the-art average error rate lower than 1%, which achieves higher accuracy than other deep learning methods [6] [10]. The result is shown in Fig. 5. In the second and third experiments, we compare the performance improvements of using RL method for the tasks of triangular and square formation. The respective results are shown in Fig. 6, and Fig. 7. The significant performance improvement clearly demonstrates that the proposed learning-based method is essential for the robotic system to optimize the policy. The proposed model is substantially more stable and accurate to reach the goal.

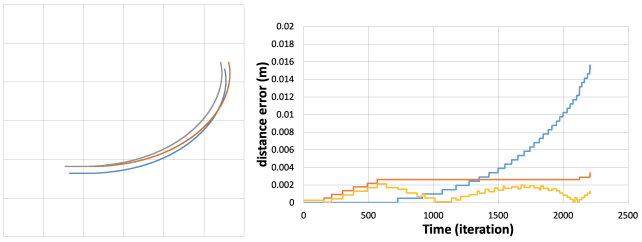
B. Hardware implementation

The proposed system is implemented in a 130-nm CMOS process. The synthesis results show that the system area is $0.13mm^2$ and the power consumption is 2.8mW. As shown in Table I, the proposed design substantially outperforms other works in the area and power performance, when compared to the state-of-the-art accelerator designs performing similar tasks [11] [12] [13]. The design [11] consumes lower power but has substantially larger area and lower resolution.

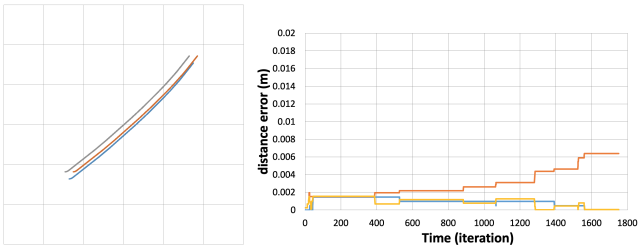


(a) (b)

Fig. 5. Distance error (a) Original Boids model (b) With the metric distance model

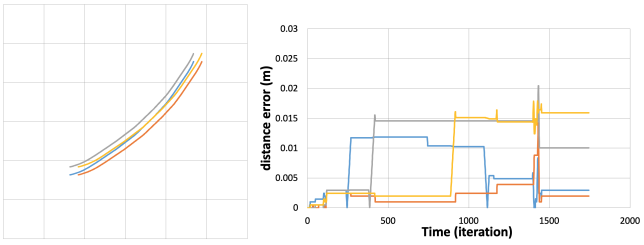


(a) Motion path (b) Distance error

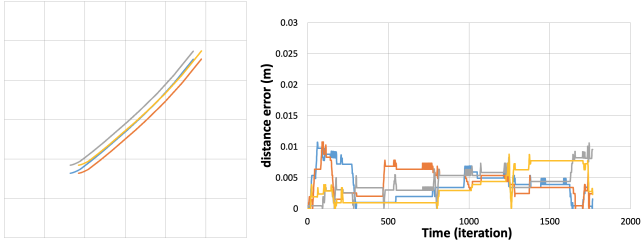


(c) Motion path (d) Distance error

Fig. 6. Triangle formation (a)(b) before RL (c)(d) after RL



(a) Motion path (b) Distance error



(c) Motion path (d) Distance error

Fig. 7. Square formation (a)(b) before RL (c)(d) after RL

TABLE I
COMPARISON WITH THE OTHER HARDWARE ACCELERATORS

	This Work	JSSC 2019	ISSCC 2018	ISSCC 2016
Technology	$130nm$	$55nm$	$65nm$	$65nm$
Area	$0.13mm^2$	$3.4mm^2$	$16mm^2$	$16mm^2$
Resolution	16b	6b	16b	16b
Power	2.8mW	690μW	6.57mW	45mW
Frequency	50MHz	67.5MHz	10-100MHz	Not Reported

TABLE II
COMPARISON WITH THE OTHER GENERAL PROCESSORS

	This Work	Raspberry Pi 3B	i7-7700 CPU
Power	2.8mW	1.4W	14W
Task	Swarm Robot	Leader Follower	Swarm Robot Fish

The proposed system is significantly more power-efficient than other designs based on general processors, as shown in Table II. Robots that use i7-7700 CPU consume 14 Watts [14], while robots equipped with Raspberry Pi 3B consume at least 1.4 Watts [15].

V. Conclusion

In this work, we present a biologically inspired approach for distributed multi-robot formation control. We introduce an algorithm based on the swarm metric distance model and reinforcement learning. The design maintains the advantage of the model-based method, which demonstrates high accuracy while realizing the high adaptability from the learning-based method. The experiments results show that the agents move smoothly with a low error rate. The hardware implementation results demonstrate high area and power efficiency, which offers a promising solution to emerging multi-robot systems.

Acknowledgment

This work was supported by the Ministry of Science and Technology, Taiwan.

References

[1] A. F. T. Winfield and J. Nembrini, "Safety in numbers: Fault-tolerance in robot swarms," International Journal of Modelling, Identification and Control, vol. 1, no. 1, p. 30, 2006.

[2] C. W. Reynolds, "Flocks, herds, and schools," Seminal graphics, pp. 273–282, 1998.

[3] I. Couzin, J. Krause, & R. James, & G. Ruxton, & N. Franks, "Collective memory and spatial sorting in animal groups," Journal of Theoretical Biology, vol. 218, no. 1, pp. 1–11, 2002.

[4] C. Jiang, Z. Chen, and Y. Guo, "Multi-robot Formation Control: A comparison between model-based and learning-based methods," Journal of Control and Decision, vol. 7, no. 1, pp. 90–108, 2019.

[5] G. Zuo, J. Han, and G. Han, "Multi-robot formation control using reinforcement learning method," Lecture Notes in Computer Science, pp. 667–674, 2010.

[6] Y. Zhou, F. Lu, G. Pu, X. Ma, R. Sun, H.-Y. Chen, and X. Li, "Adaptive leader-follower formation control and obstacle avoidance via deep reinforcement learning," 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019.

[7] P. Zhu, W. Dai, W. Yao, J. Ma, Z. Zeng, and H. Lu, "Multi-robot flocking control based on Deep Reinforcement Learning," IEEE Access, vol. 8, pp. 150397–150406, 2020.

[8] N. Cao, M. Chang, and A. Raychowdhury, "A 65nm 1.1-to-9.1 tops/W hybrid-digital-mixed-signal computing platform for accelerating model-based and model-free Swarm Robotics," 2019 IEEE International Solid-State Circuits Conference (ISSCC), 2019, pp. 222-224.

[9] J. Delcourt, N. W. Bode, and M. Denoël, "Collective vortex behaviors: Diversity, proximate, and ultimate causes of circular animal group movements," The Quarterly Review of Biology, vol. 91, no. 1, pp. 1–24, 2016.

[10] Z. He, L. Dong, C. Sun, and J. Wang, "Reinforcement learning based multi-robot formation control under Separation Bearing Orientation Scheme," 2020 Chinese Automation Congress (CAC), 2020.

[11] A. Amaravati, S. B. Nasir, J. Ting, I. Yoon, and A. Raychowdhury, "A 55-nm, 1.0–0.4V, 1.25-PJ/Mac time-domain mixed-signal neuromorphic accelerator with stochastic synapses for reinforcement learning in Autonomous Mobile Robots," IEEE Journal of Solid-State Circuits, vol. 54, no. 1, pp. 75–87, 2019.

[12] J. Sim, J.-S. Park, M. Kim, D. Bae, Y. Choi, and L.-S. Kim, "14.6 a 1.42tops/w deep convolutional neural network recognition processor for intelligent IOE Systems," 2016 IEEE International Solid-State Circuits Conference (ISSCC), 2016, pp. 264–265.

[13] S. Choi, J. Lee, K. Lee, and H.-J. Yoo, "A 9.02MW CNN-stereo-based real-time 3D hand-gesture recognition processor for smart mobile devices," 2018 IEEE International Solid-State Circuits Conference (ISSCC), 2018, pp.220–222.

[14] D. Xu, L. Yu, Z. Lv, J. Zhang, D. Fan, and W. Dai, "Energy Consumption Optimization for the Formation of Multiple Robotic Fishes Using Particle Swarm Optimization," Energies, vol. 11, no. 8, p. 2023, Aug. 2018.

[15] J. N. Yasin, H. Mahboob, M.-H. Haghbayan, M. M. Yasin, and J. Plosila, "Energy-efficient navigation of an autonomous swarm with adaptive consciousness," Remote Sensing, vol. 13, no. 6, p. 1059, 2021.