Research Article

Design and Analysis of Multi-robot Grouping Aggregation Algorithm

Huailin Zhao[1,*], Zhen Nie[1], Xiaoxing Wang[2]

[1]*School of Electrical and Electronic Engineering, Shanghai Institute of Technology, No. 100, Haiquan Road, Fengxian District, Shanghai 201418, China*
[2]*College of Engineering, University of Massachusetts Amherst, Amherst, MA 01003, USA*

ARTICLE INFO

ABSTRACT

In this paper, the grouping aggaregation problem of multi-robot in simple environment is studied. The grouping algorithm and aggregation algorithm are mainly discussed. Based on *K*-means clustering algorithm, a grouping control algorithm that can determine the number of groups and complete the grouping task autonomously was designed. Aiming at the problem of aggregation, three kinds of aggregation algorithms are proposed. Taking the center aggregation control as an example, the cooperation and control of each group of robots are studied. Finally, the simulation results of MATLAB show that the multi-robot grouping and aggregation algorithm is effective.

1. INTRODUCTION

The multi-robot collaboration system is a frontier topic in the field of artificial intelligence [1]. From the control point of view, the multi-robot collaboration system is a typical distributed control system [2–4], and its goal is to build large complex systems which included software and hardware systems into small, easy-to-manage systems that communicate and coordinate with each other. The research of multi-robot collaboration system involves the knowledge, goals, skills, planning of robots and how to make robots coordinate actions to solve problems, and the idea of swarm intelligence is emphasized. The multi-robot collaboration system forms a complex system through communication, coordination, and cooperation among autonomous robots. The major applications include formation flying of multi-unmanned aerial vehicle (UAV), clusters of warehousing logistics robot, competitions of robot team, large-scale robotic rescue systems, etc. Some examples in nature are dancing fish, flying birds, and so on.

Aggregation is the basic problem of multi-robot system consistency [5,6], and it is also an optimizing measure [7], and the grouping aggregation is a special case of aggregation problem. Group aggregation is a motion planning of the robots randomly distributed in space, which are divided into many groups according to a specific grouping rule, and then each group of robots adjust the motion states through communication and cooperation between each other, finally they gather together, or in the desired area to complete the aggregation. In the end, all the robots formed a multi-group gathering situation, the idea of swarm intelligence is emphasized.

After analyzing the *K*-means algorithms, an efficient grouping aggregation algorithm that can automatically select the number of

*Corresponding author. Email: zhao_huailin@yahoo.com

groups is designed based on *K*-means. The feasibility of multi-robot grouping aggregation is verified by MATLAB (Mathworks Inc., United States) simulation experiment. And the multi-robot distributed coordination and grouping aggregation control are realized.

2. ANALYSIS AND DESIGN OF GROUPING ALGORITHM

Grouping aggregation is divided into two steps, grouping task and aggregation task. The grouping task is to complete the group division of each robot according to a specific grouping rule. This study designed and analyzed a grouping algorithm based on *K*-means algorithm.

The *K*-means algorithm is a clustering algorithm which is the simplest and most commonly used in unsupervised learning. It is a center-based clustering algorithm [8,9]. The *K*-means algorithm completes the grouping task by minimizing the distance between each sample and the center of the cluster. The initial centers of each cluster were randomly selected.

In the Euclid space, a set of *n*-dimensional samples $X = \{x_1, x_2, x_3 \cdots x_i \cdots x_n\}$ (representing *n* robots) is given, where $x_i \in R^n$. The *K*-means algorithm divides all samples into *k* clusters $C = \{c_i, i = 1, 2, \ldots, k\}$. The Euclidean distance is chosen as the criterion for the distance, the equation is expressed as follows.

$$d(x_i, x_j) = \left\| x_i - x_j \right\| \tag{1}$$

where x_i, x_j represents two points in the Euclid space, which are two *n*-dimensional data.

In determining the similarity of the samples, the samples are divided into the clusters with the highest similarity according to the following equation (the similarity is based on the distance between the samples and the centers, and the smaller the distance, the higher the similarity).

$$c_i = \underset{C}{\arg\min} \sum_{i=1}^{k} \sum_{x_j \in X} \left\| x_j - \mu_i \right\|^2 \qquad (2)$$

where μ_i is the center of ith cluster.

$$\mu_i = \frac{1}{N_i} \sum_{x \in c_i} x \qquad (3)$$

where N_i is the number of samples in group i.

The K-means algorithm needs to re-divide the cluster and update the centers iteratively. When the square error sum is converged, the K-means algorithm ends. The equation of square fitting error sum is given by

$$J = \sum_{i=1}^{k} \sum_{j=1}^{n} d_{ij} \left\| x_j - \mu_i \right\|^2 \qquad (4)$$

where $d_{ij} = \begin{cases} 1, & x_j \in c_i \\ 0, & x_j \notin c_i \end{cases}$.

3. IMPROVED GROUPING ALGORITHM

There are two limitations [10,11] of K-means algorithm. Firstly, the value of the k needs to be specified by the users in advance. But the value of k is difficult to estimation, which may result in unreasonable grouping. Secondly, the biggest disadvantage of K-means is that the selection of initial clustering centers has a great influence on the results, which can easily lead to the increase of iteration times and fall into local optimum. Therefore, we have improved the grouping algorithm that can automatically select the number of groups based on the K-means algorithm. For the problem that the k-value is difficult to estimate, we have added two processes of "merging" and "split" in the process of grouping, so that it can automatically adjust the number of groups in the grouping process. Merging means that merge the two groups into one group when the number of samples in one group is too small or the distance between the two groups is too close. Splitting refers to splitting the group into two groups when the variance in one group is too large. For the initial center problem, we select the initial cluster center by the maximum distance.

The grouping algorithm can complete the grouping by seven initial parameters (K, k, θ_N, θ_c, θ_s, L, I). Where K is the expected number of groups. k is the initial number of groups. θ_n is the minimum number of samples in each group. If the samples are less than this number, the group will be removed. θ_s is the maximum standard deviation of the sample features in a group. If it is greater than this value, the group may be split. θ_c is the minimum distance between the centers of the two groups. If it is less than this value, the two groups need to be merged. L is the maximum number of groups that can be merged in a merge operation. I is the number of iterations.

The idea of the grouping algorithm is as follows:

After seven initial parameters are given, K initial clustering centers must be determined first. And the initial clustering centers are selected according to the maximum distance. The specific operation is to calculate the distance between any two points according to formula (1), and find the two points x_1 and x_2 with the largest distance. Then with x_1 and x_2 as references, search for x_3 so that the distance from x_3 to x_1 and x_2 is the largest at the same time. Repeat this operation until k points are selected, and the k points are used as the initial cluster center.

After the initial clustering center is selected, clustering is carried out according to formula (2). If the number of samples in any group is less than θ_n, the group will be cancelled, and the value of k is subtracted by 1. Then the clustering is carried out again. The average distance $\overline{D}_i$ from each group of samples to the group center and the total average distance $\overline{D}$ of all samples were calculated.

$$\overline{D}_i = \frac{1}{N_i} \sum_{x \in c_i} \left\| x - \mu_i \right\|, \quad i = 1, 2, \cdots, k \qquad (5)$$

$$\overline{D} = \frac{1}{n} \sum_{i=1}^{n} N_i \overline{D}_i, \quad i = 1, 2, \cdots, k \qquad (6)$$

where N_i is the number of samples in i^{th} group; μ_i is the center of ith group, and n is the number of all samples.

After each clustering is completed, the obtained clustering results are split and merged according to a given requirement, thereby a new clustering center will be obtained. Split, merge or iterate is according to the following conditions.

(1) If the number of iterations is odd or $k \le \frac{1}{2} K$, that is, the number of cluster centers is less than half of the expected number of groups, splitting the existing clusters;

(2) If the number of iterations is even or $k \ge 2K$, that is, the number of cluster centers is more than twice the number of expected groups, the existing clusters are not split.

The split operation is as follows:

First, the standard deviation of all components in each group (also known as features of each dimension) is calculated, which is defined as formula (7):

$$\sigma_{in} = \sqrt{\frac{1}{N_i} \sum_{x_i \in c_i} \left(x_{\ln} - \mu_{in} \right)^2} \qquad (7)$$

where N_i is the number of samples in i^{th} group, $x_{\ln}$ is the nth component of the lth in the group c_i, $\mu_{\ln}$ is the nth component of the center of the group c_i.

Find the largest standard deviation of component in each group: $\delta_{i\max}$, $i = 1, 2, 3, \dots, k$, if $\delta_{i\max} > \theta_s$, and the group satisfy one of the following two conditions, split the group into two groups.

(1) $\overline{D}_j > \overline{D}$ and $N_i > 2\theta_n + 1$

(2) $k < \frac{K}{2}$

The center after splitting is recorded as μ_i^+ and μ_i^-, μ_i^+ and μ_i^- are given by

$$
\begin{aligned}
\mu_i^+ &= \mu_i + h\delta_i \\
\mu_i^+ &= \mu_i - h\delta_i
\end{aligned}
\quad (8)
$$

where $0 < h < 1$.

The merge operation is as follows:

Firstly, the distance D_{ij} of each cluster center is calculated. The D_{ij} is expressed as follows.

$$
D_{ij} = \left\| \mu_i - \mu_j \right\| \quad (9)
$$

where μ_i is the center of the ith group and μ_j is the center of the jth group.

Comparing D_{ij} with θ_c, and arranging all D_{ij} that smaller than θ_c from small to large. Then merging the group c_i and c_j start from who has the smallest D_{ij}. The new center is expressed as follows:

$$
\mu_l = \frac{1}{N_i + N_j}(N_i\mu_i + N_j\mu_j), \, l = 1, 2, \dots L \quad (10)
$$

where N_i and N_j are the number of samples of the groups c_i and c_j.

4. ANALYSIS AND DESIGN OF AGGREGATION ALGORITHM

As mentioned earlier, the aggregation problem is that multi-robots randomly distributed in the space adjust the motion states through communication and cooperation between each other, finally they gather together, or in the desired area to complete the aggregation. Aiming at the aggregation task requirements of this topic, three aggregation control algorithms were designed and analyzed in this paper:

(1) Selecting a robot from each group as leaders, as shown in Figure 1a. The basic idea of this algorithm is to pre-set a leader robot, and other robots are called followers [12,13]. In the process of aggregation, the leader controls the motion path, and the followers keep a certain angle and distance with the leader, finally all robots aggregate at the position of the leader. After careful analysis, the aggregation has some defects. On the one hand, because the algorithm does not make full use of the advantages of distributed systems, when the leader robot goes wrong, it will cause all robots fail to aggregate. On the other hand, because the control of the system is relatively simple, the follower or leader robot may be lost in the process of motion
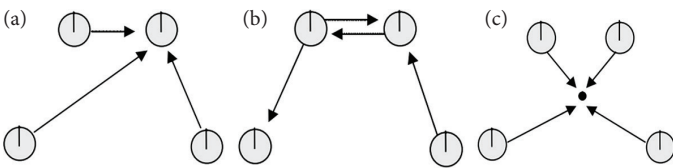

Figure 1 | Three different aggregation algorithms: (a) Selecting a robot from each group as leaders. (b) Selecting the neighboring robot as the reference point. (c) Selecting the center of group as the reference point.

in the complicated reality environment. The aggregation task failed.

(2) Selecting the neighboring robot as the reference point, as shown in Figure 1b. Each robot finds its own target position in the whole aggregation movement through the position information of neighboring robots. The limitation of this method is that a large amount of communications is required, and each robot needs to spend more time for communication, and the result of the formation control is not satisfactory.

(3) Selecting the center of group as the reference point, as shown in Figure 1c. This selection method is called center algorithm. During the aggregation process, multiple robots form a geometric figure. According to the geometric figure, the geometric center is calculated. Each robot uses takes the center as a reference point to adjust the direction and speed, and move to the reference point. During the aggregation movement, the system constantly updates the reference points, and finally all robots reach a same point. The aggregation task is end. This algorithm is simple and practical. The system is convergent. The advantages of the distributed control system are fully reflected.

Comparing the three aggregation algorithms, this study intends to use the center algorithm for aggregation control. The center calculation equations are expressed as follows:

$$
x_c = \sum_{i=1}^{n} \frac{x_i}{n}, \quad y_c = \sum_{i=1}^{n} \frac{y_i}{n} \quad (11)
$$

where (x_i, y_i) is the 2-D coordinate of the current position for the ith robot, and (x_c, y_c) is the center coordinate.

During the aggregation process, the system periodically updates the center position and notifies all robots. Each robot makes motion planning according to its current position and new center position. The motion planning equation is given by:

$$
X_i(k+1) = X_i(k) + v_i(k+1) \cdot T \quad (12)
$$

where T is the sampling period, X_i is the motion state of ith robot, v_i is the velocity of ith robot.

In the later stage of aggregation, the robot is close to each other and collision may occur. So the dynamic obstacle avoidance of robots is studied. When the robot finds that other robots may affect its motion, the robot produces an obstacle area according to the current position and predicted position of the obstacle. Then the robot turns left or right at a certain angle to avoid collision. The output vector formula of robot avoiding dynamic obstacles is defined as follows:

$$
v_{\text{avoid}} = \begin{bmatrix} \cos(\pm(\theta_m + \alpha)) - \sin(\pm(\theta_m + \alpha)) \\ \sin(\pm(\theta_m + \alpha))\cos(\pm(\theta_m + \alpha)) \end{bmatrix}^T \begin{bmatrix} x_d \\ y_d \end{bmatrix} \quad (13)
$$

where θ_m is the angle between the motion direction of the robot and the line connecting the robot and the obstacle robot. $\pm$ indicates the direction of the robot's rotation, which is determined according to the predicted direction of obstacle motion and the relative position of obstacle area. $\begin{bmatrix} x_d \\ y_d \end{bmatrix}$ is the direction of obstacle motion predicted by the robot, and α is a given constant.

5. CALCULATION PROCESS OF GROUPING AGGREGATION ALGORITHM

As mentioned earlier, to complete the grouping aggregation task, the first thing should be done is grouping operation, and the next is the aggregation operation. The specific process is as follows.

According to the grouping requirements, the robot updates the position of the center of the group in real time, and grouping according to the splitting merging principle mentioned. The specific algorithm flow is as follows:

(1) Set the seven control parameters of grouping, and select k initial centroids according to the principle of maximum distance.

(2) The Euclidean distance between each robot and the group center are calculated, and assign them to the group with the closest center according to the nearest distance principle.

(3) Splitting: If the samples distribution in the same group is too dense or the number of samples is too small, the group needs to be split. If the sample distance in the same group exceeds a given limit, it will be split, otherwise it will be retained. After the splitting, a new center is obtained, and then repeat from step 2.

(4) Merging: If the two groups are too close to each other, or the number of samples in a group is too small, the groups are merged according to the merge rule. And then repeat from step 2.

(5) Repeated grouping, judgment, splitting or merging until the expected grouping effect is reached, or the number of operations has reached the upper limit.

After the grouping operation is completed, the aggregation operation is started. According to the requirements of center algorithm, the center position of each group should be real-time updated and notified each robot in the group. Each robot adjusts the speed and direction according to its current position and new center position, then moves to the center. The specific algorithm flow is as follows:

(1) Each robot initializes its position, and reads and saves the current position.

(2) The robots in the group communicate with each other to inform the position information. The center coordinates are calculated according to the position coordinates of all robots. Then each robot calculates its own speed and direction of motion based on its own coordinate and center coordinate. Finally, the robots move to the center of the group.

(3) Repeat from step 1 until all robots in each group enter a small range.

6. SIMULATION ANALYSIS

According to the grouping aggregation control algorithm, the research group carried out simulation analysis on the MATLAB. In order to facilitate the observation of the simulation results and analysis, the simulation environment which is a 2-D with a length of 100 and a width of 100 was first configured. About 60 robots were randomly set up in the grid. The initial parameters of the group are configured. The specific parameters are as shown in Table 1. The specific 2-D grid and the initial position of the robot are shown in Figure 2.

After the simulation starts, the multi-robot system automatically groups according to the grouping algorithm we designed before. The grouping process is shown in Figures 3a and 3b. Figure 3a shows the results of the first grouping, in which the stars show the first four initial centers selected according to the principle of maximum distance, and the diamonds show the new centers after the first grouping is completed. Figure 3b shows the result after the grouping is completed. The stars show the centers of each group. In each grouping process, splitting, merging, or iteration is judged according to the given parameter, and the centers are updated in real time during the grouping process.

The aggregation movement begins after the grouping is completed. In the process of aggregation control, the robots of each group moved from the initial position to the center of each group. During the aggregation motion, the centers were updated in each sampling period (the sampling period T was set 1 s). Figures 4a–4e correspond to a sampling period in the aggregation process, respectively. Figure 4d shows that the robots have gathered within a small range. Figure 4e shows that the motion state of the robots was slightly adjusted relative to Figure 4d, but it was still within a small range, so that the aggregation task was completed.

Table 1 | The robot group initial parameters

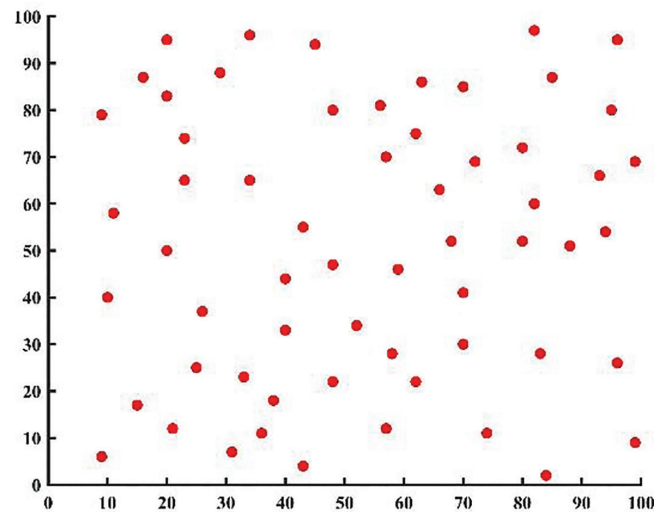|	
The expected number of clusters: K	4
The number of initial cluster centers: k	4
The minimum number of samples in each group: θ_n	3
The maximum standard deviation among each sample characteristics: θ_s	1
The minimum distance between each group: θ_c	10
The number of groups that can be merged in one merge process: L	1
The maximum number of iterations: I	200

Figure 2 | The specific two-dimensional grid and the initial position of the robot.
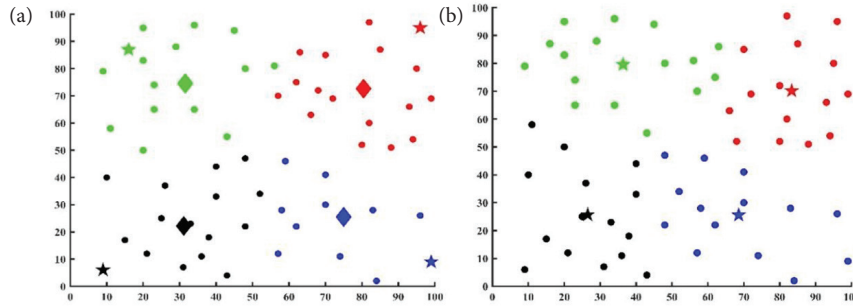
Figure 3 | The simulation of grouping process: (a) The results of the first grouping. The initial centers are shown as stars and the new centers after the first grouping is completed are shown as diamonds. (b) The result after the grouping is completed.
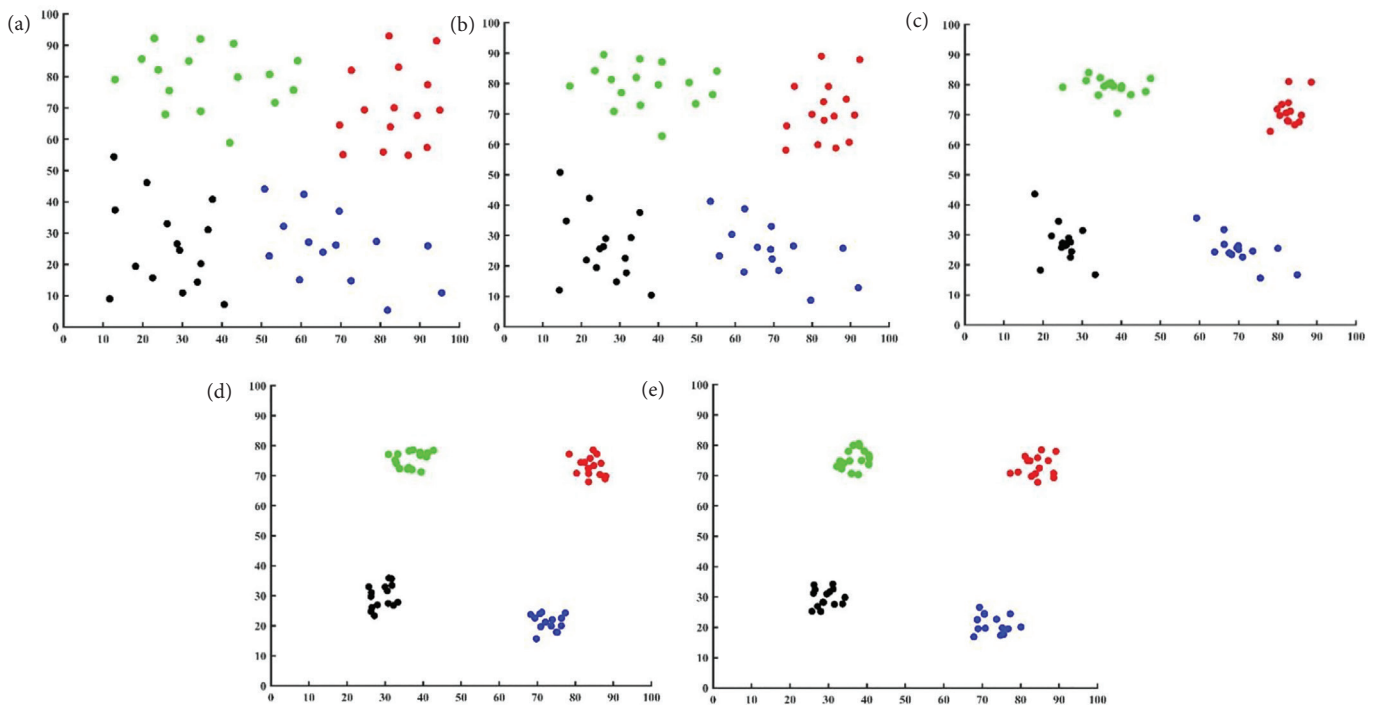

Figure 4 | The simulation of aggregation process: (a) The initial position of four groups of robots. (b) and (c) Two processes of robot aggregation. (d) The robots have aggregated within a small range. (e) The result of aggregation.

7. CONCLUSION

This topic takes multi-robot system as the research object, and takes grouping aggregation as the main task, and realizes the grouping and aggregation control of multi-robot in simple environment. Based on the *K*-means algorithm, the research group designed a grouping algorithm that can automatically complete the grouping, and designed the centroid aggregation control algorithm. The MATLAB simulation analysis shows the detailed process of grouping operation and aggregation operation, which verifies the feasibility of the grouping aggregation algorithm. The algorithm embodies the swarm intelligence in a certain degree.

CONFLICTS OF INTEREST

There is no conflicts of interest.

REFERENCES

[1] H. Zhao, Y. Guo, M. Sugisaka, Design of the multi-car collaboration system, The 2017 International Conference on Artificial Life and Robotics (ICAROB), A Life Robotics Corporation Ltd, Seagaia Convention Center, Miyazaki, Japan, 2017, pp. 186–190.

[2] W. Tingting, Research on the surround control problem of multi-agent system, University of Electronic Science and Technology of China, dissertation, 2014.

[3] S. Xiaoxing, Research on formation and coordination of group robots, Lanzhou University of Technology, dissertation, 2011.

[4] D.V. Dimarogonas, K.J. Kyriakopoulos, Connectedness preserving distributed swarm aggregation for multiple kinematic robots, IEEE Trans. Robot. 24 (2008), 1213–1223.

[5] H. Zhao, W. Ren, Consensus problem of distributed multi-agent systems, The 2015 International Conference on Artificial Life and Robotics (ICAROB), A Life Robotics Corperation Ltd, Japan, 2015, pp. 201–206.

[6] H. Zhao, Z. Wu, X. Wang. A multi-robot rescuing system, The 2018 International Conference on Artificial Life and Robotics (ICAROB), A Life Robotics Corperation Ltd, Japan, 2018, pp. 582–586.

[7] Z. Guoqing, Study on Control on strategy for self-organized formation of swarm robotics systems, Shandong University, dissertation, 2017.

[8] Q. Wang, C. Wang, Z.Y. Feng, Y.E. Jin-Feng, Review of *K*-means clustering algorithm, Electron. Des. Eng. 7 (2012), 21–24.

[9] P. Yang, C. Guo, C. Li, Swarm robots aggregation formation control based on *K*-means clustering algorithm, Mod. Mach. Tool Autom. Manuf. Technol. 8 (2012), 89–92.

[10] Z. Wang, G. Li, Z. Han, X. Zheng, Research and implementation based on *K*-medoids clustering algorithm, Silicon Valley 1 (2011), 80–81.

[11] N.X. Xia, Y.D. Su, X. Qin, Efficient *K*-medoids clustering algorithm, Appl. Res. Comput. 27 (2010), 4517–4519.

[12] W.M. Wang, S. Zhou, Research on multi-robot cooperative control and formation method, Netinfo Security 6 (2014), 59–66.

[13] L. Jing, L. Chenglin, L. Fei, Pilot-following formation and obstacle avoidance control of multi-mobile robots, CAAI Trans. Intell. Syst. 12 (2017), 202–212.

Authors Introduction

Dr. Huailin Zhao

He received his PhD from Oita University, Japan in 2008. He is a professor in the School of Electrical and Electronic Engineering, Shanghai Institute of Technology, China. His main research interests are robotics, multi-agent system and artificial intelligence. He is the member of both IEEE and Sigma Xi.

Mr. Xiaoxing Wang

He graduated from Shanghai Insitute of Technology in 2017. He is now a graduate student of University of Massachusetts Amherst for the master of science electrical and computer engineering degree. His main research interests include embedded system, internet of things and multi-agent system.

Mr. Zhen Nie

He received the B.S. degree from Shanghai Institute of technology, Shanghai, China, in 2017. He is a master course student at Shanghai Institute of Technology, and his major is bionic equipment and control engineering. His main research interests are robotics and multi-agent system.