

## Research Article

# Behavior-Based Formation Control of Swarm Robots

**Dongdong Xu, Xingnan Zhang, Zhangqing Zhu, Chunlin Chen, and Pei Yang**

*Department of Control and System Engineering, School of Management and Engineering, Nanjing University, Nanjing 210093, China*

Correspondence should be addressed to Zhangqing Zhu; [zzqing@nju.edu.cn](mailto:zzqing@nju.edu.cn)

Received 25 March 2014; Accepted 22 May 2014; Published 19 June 2014

Academic Editor: Guangming Xie

Copyright © 2014 Dongdong Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Swarm robotics is a specific research field of multirobotics where a large number of mobile robots are controlled in a coordinated way. Formation control is one of the most challenging goals for the coordination control of swarm robots. In this paper, a behavior-based control design approach is proposed for two kinds of important formation control problems: efficient initial formation and formation control while avoiding obstacles. In this approach, a classification-based searching method for generating large-scale robot formation is presented to reduce the computational complexity and speed up the initial formation process for any desired formation. The behavior-based method is applied for the formation control of swarm robot systems while navigating in an unknown environment with obstacles. Several groups of experimental results demonstrate the success of the proposed approach. These methods have potential applications for various swarm robot systems in both the simulation and the practical environments.

## 1. Introduction

With the development of the robot technology, the swarm robot system has a wide application prospect in theoretical research and engineering applications because of its robustness, flexibility, and scalability [1]. Formation control is one of the most important research areas for swarm robot systems and has many applications (e.g., industrial production, aerospace, and military affairs) [2–5]. The control design of swarm robot systems is quite complex because a lot of issues should be considered, such as formation keeping, communication, and coordination between robots. In addition, when the number of the robots reaches a certain degree (hundreds or even more), the computation of the system will be staggering.

During recent years, research on the formation control of swarm robot systems has attracted wide attention. The swarm robots can form or keep various formations, for example, line, orthogon, and triangle [6]. In addition, the formation can transform in order to suit different tasks. There are three main methods to implement formation control, namely, behavior-based formation control [7–9], leader-follower [10–12], and virtual structure methods [13, 14]. The behavior-based formation control method utilizes a set of predefined basic behaviors, such as moving-to-target, keeping-formation, and avoiding-obstacles [15, 16]. This method is more suitable to

distributed multirobot system and every robot is strongly autonomous. But there is no clear definition to group behaviors for swarm robots, and it is difficult to guarantee the stability of a desired formation when the environment is complex. The leader-follower method designates one robot as the leader and the others as followers [17, 18]. The followers keep a certain distance and direction angle from the leader. This method is prone to deadlock of robot collisions. The third one is virtual structure method which regards the swarm robot system as an inflexible entity [19, 20]. The robots are regarded as some points in the system. This method defines the structure of the overall behavior of the robot formations, but the stringent requirements to maintain a virtual structure formation movement limit the application of this method.

On the other hand, most of the existing methods for the swarm robot formation problem are based on small-scale robot formation where the number of robots is around or less than 10 [21–23]. There is little research on formation control of large-scale problems except some theoretical results regarding optimization; for example, Derenick and Spletzer use convex optimization to make one hundred robots form a certain formation [24]. In this paper, we extend the behavior-based navigation method to swarm robot systems and focus on two kinds of important formation control problems (i.e., initial formation and formation control while avoiding obstacles). In particular, for the initial formation problem, we

present an improved algorithm named classification-based searching for initial formation. This algorithm is suitable for large-scale robot formation where the number of robots is more than one hundred. Based on the initial state of the robots, the algorithm will classify robots to different types and use different methods and processing sequences to handle these types, which can significantly reduce the time complexity. These studies have potential applications for practical swarm robot systems.

The rest of this paper is organized as follows. In next section, the problem formulation and behavior design for single-robot systems are presented. Section 3 describes an algorithm of classification-based searching for initial formation control to reduce time complexity. In Section 4, the formation control with obstacle-avoidance is presented and several groups of experimental results are analyzed. Conclusions are given in Section 5.

## 2. Problem Formulation and Behavior Design

**2.1. Problem Formulation.** The ability and efficiency of swarm robot systems depend on the functions of the single robots and the control structure. The control structure of the swarm robot system is categorized as centralized, distributed, and layered structures [25]. The centralized structure contains a control unit and several robots with the ability of perception and execution [26]. The control unit contains the information of the environment and every single robot. The centralized structure is simple and easy to implement, but has low flexibility, fault tolerance, and suitability and can only be applied in the simple and structured environments. Unlike the centralized one, the distributed structure [27] has no central control unit. Each robot is autonomous to make decision according to its own tasks. This method has high fault tolerance and reliability but is more difficult to coordinate between robots. The layered structure is the mixture of centralized and distributed structures, which has the control unit to supervise the whole system and is independent of each robot [28, 29]. The distributed and layered structures can be applied in high dynamic and complex environments.

The swarm robot system studied in this paper consists of dozens to hundreds of single mobile robots. In order to ensure the high flexibility, fault tolerance, and adaptability, the distributed control architecture is chosen and behavior-based control methods are presented for the swarm robot systems.

The model of a single mobile robot is shown as in Figure 1, which contains embedded computer systems, motion control systems, a perceptual system, and a communication system. The perceptual system is a multiultrasonic system to detect obstacles in the environment. The communication system is a wireless communication module. As shown in Figure 1(a), the working environment of the robot is two-dimensional and the robot's coordinate is  $(x, y, \beta)$ , where  $(x, y)$  is the position coordinate and  $\beta$  is the orientation angle. The speed of the robot is  $(v, \omega)$ , where  $v$  is the linear velocity and  $\omega$  is the angular velocity. The model of the multiultrasonic system is shown in Figure 1(b). The detection space is divided into four areas, namely, left one ( $L_1$ ), left two ( $L_2$ ), right one ( $R_1$ ), and right

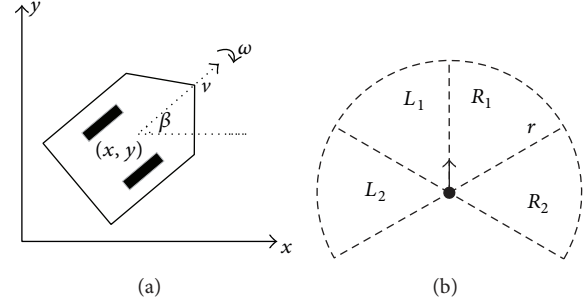


FIGURE 1: Models of a single robot and its sonar sensors.

two ( $R_2$ ), respectively. The central angle of each region is  $60^\circ$ , so in order to detect these four areas, at least 4 sonar sensors are needed and the detecting radius is  $r$ .

**2.2. Behavior Design.** The overall behavior conducted by a robot is the combination of several subbehaviors, that is, moving to the goal, avoiding obstacles, wall-following, avoiding robot, and formation keeping. In practical applications, each behavior of a robot is stated as the form of vector that consists of the magnitude and direction. The weight of the vector (the intensity of behavior) can be changed by adjusting the parameters. Based on the information detected from the surrounding environment, the robot chooses the proper behaviors. Therefore, a movement command will be produced. The overall behavior vector which is the sum of all subbehavior vectors is defined as

$$V_{\text{direction}} = [f_1 \ f_2 \ f_3 \ f_4 \ f_5] \begin{bmatrix} V_{\text{move.to.goal}} \\ V_{\text{avoid.obstacle}} \\ V_{\text{follow.wall}} \\ V_{\text{avoid.robot}} \\ V_{\text{keep.formation}} \end{bmatrix}, \quad (1)$$

where  $\{f_1, f_2, f_3, f_4, f_5\}$  are the controlling parameters regarding the weights of each behavior vectors.

**2.2.1. Behavior of Moving to the Goal.** As shown in Figure 2(a), if there are no obstacles in front of the robot, the robot will move to the target at maximum speed. In most occasions, the robot maintains the behavior of moving to the goal. Assuming the current coordinates of the robot  $R_i$  are  $(x_i, y_i)$  and the coordinates of the goal are  $(x_t, y_t)$ , the behavior vector of moving to the goal is

$$V_{\text{move.to.goal}} = \frac{1}{\sqrt{(x_t - x_i)^2 + (y_t - y_i)^2}} \begin{bmatrix} x_t - x_i \\ y_t - y_i \end{bmatrix}, \quad (2)$$

The controlling parameter of this behavior is defined as

$$f_1(d_m) = \begin{cases} a_m, & d_m \in (b_m, +\infty) \\ a_m \frac{d_m}{b_m}, & d_m \in [0, b_m], \end{cases} \quad (3)$$

where  $d_m$  is the distance between the robot and target point and  $a_m$  and  $b_m$  are the adjustable parameters.  $b_m$  is related to

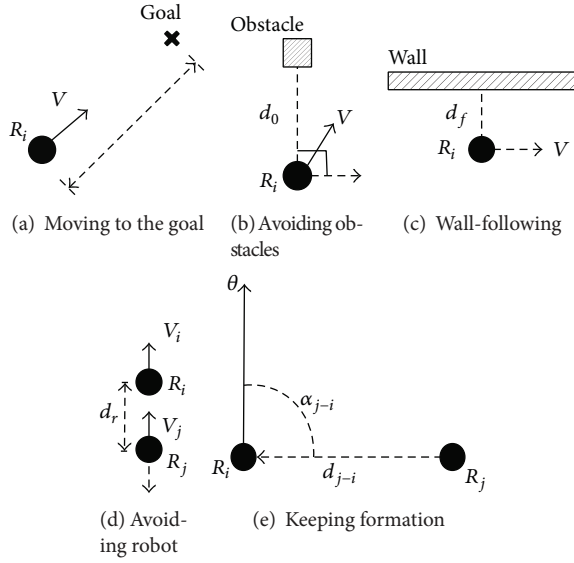


FIGURE 2: Behavior design of single robot.

the robot's movement step and the robot will slow down when the distance between the robot and target point is less than  $d_m$ .

**2.2.2. Behavior of Avoiding Obstacles.** The robot will trigger the behavior of avoiding obstacles when any obstacle around the robot is detected. The traditional way is to make the robot move backward, which will cause moving deadlock. In order to solve this problem, we change the direction of movement with an angle of 90 degrees to the obstacle (as shown in Figure 2(b)). The virtual force of the obstacle-avoiding behavior will be enhanced when the robot is close to the obstacles.

Assume that the current coordinate of the robot is  $(x_i, y_i)$  and the coordinate of the nearest obstacle is  $(x_o, y_o)$ , then the vector of avoiding obstacles is

$$V_{\text{avoid\_obstacle}} = \frac{1}{\sqrt{(x_o - x_i)^2 + (y_o - y_i)^2}} \begin{bmatrix} \pm (y_o - y_i) \\ \mp (x_o - x_i) \end{bmatrix}, \quad (4)$$

where the sign  $\pm$  is determined by the relationship between the moving direction of the robot and the obstacles. When the obstacle are on the left, set it  $+$ . On the contrary, when the obstacle is on the right, set it  $-$ . The control parameter is set as

$$f_2(d_0) = \begin{cases} 0, & d_0 \notin [b_f, b_0] \\ a_0 \left( \frac{d_0}{b_f - b_0} + \frac{b_0}{b_0 - b_f} \right), & d_0 \in [b_f, b_0], \end{cases} \quad (5)$$

where  $d_0$  is the distance between the robot and the obstacle.  $a_0$ ,  $b_0$ , and  $b_f$  are the adjustable parameters. As shown in Figure 3(a),  $b_0$  is the distance between the robot and the obstacle when the behavior of avoiding obstacle is triggered.

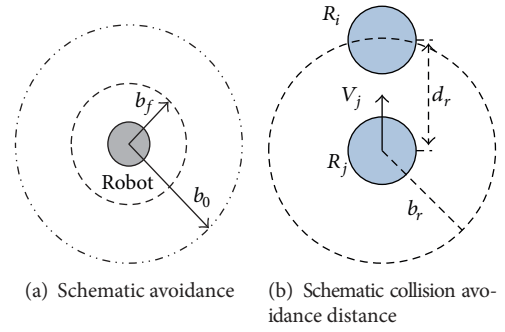


FIGURE 3: Schematic distance.

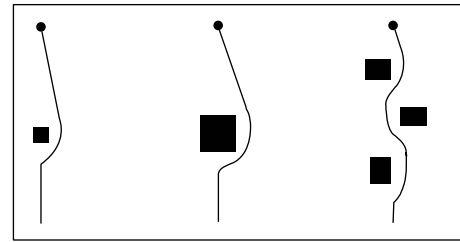


FIGURE 4: Navigation control of a single robot in three simple environments.

TABLE 1: Control parameters of robot in simple environments.

$a_m$	$b_m$	$a_0$	$b_0$	$b_f$
1.0	5.0	1.0	5.0	2.0

$b_f$  is the minimum distance between the robot and the obstacle, which is the precondition that triggers the behavior of wall-following. For simple environments where there are only one or several small obstacles, we can use only the behaviors of moving to the goal and avoiding obstacles for navigation control of a single robot. Equation (1) is then reduced as

$$V_{\text{direction}} = [f_1(\cdot) \quad f_2(\cdot)] \begin{bmatrix} V_{\text{move\_to\_goal}} \\ V_{\text{avoid\_obstacle}} \end{bmatrix}. \quad (6)$$

Some simulated experiments are demonstrated in Figure 4 and the parameter settings are shown in Table 1. The results show that these two behaviors work well for navigation control in simple environments.

**2.2.3. Behavior of Wall-Following.** When the environment becomes more complex, the robot could not complete the task just relying on the above two behaviors. As shown in Figure 2(c), in order to avoid large obstacles, we design the behavior of wall-following. This behavior is activated when the distance between the robot and the obstacle reaches a certain value. If the target point and the obstacles are on the left side of the robot, choose the behavior of following the left wall. If the target point and the obstacles are on the right side of the robot, choose the behavior of following the right wall. Assuming the current coordinate of the robot is  $(x_i, y_i)$  and

the coordinate of the nearest obstacle is  $(x_o, y_o)$ , the vector of wall-following is

$$V_{\text{follow\_wall}} = \frac{1}{\sqrt{(x_o - x_i)^2 + (y_o - y_i)^2}} \begin{bmatrix} \pm (y_o - y_i) \\ \mp (x_o - x_i) \end{bmatrix}, \quad (7)$$

where  $\pm$  is determined by the relationship between the moving direction of the robot and the coordinate of the obstacles. When the obstacle is on the left, set it + and trigger the behavior of following left wall. On the contrary, when the obstacle is on the right, set it - and trigger the behavior of following right wall. The control parameter is

$$f_3(a_f) = \begin{cases} 0, & \text{wall-following off} \\ a_f, & \text{wall-following on,} \end{cases} \quad (8)$$

where  $a_f$  is an adjustable parameter.

The robot will end the behavior of wall-following when (1) the target point is within  $3/2\pi$  range in front of the robot and (2) there is no obstacle within  $1/4\pi$  range in front of the robot. The behavior vector in the environments with large obstacles is

$$V_{\text{direction}} = [f_1(\cdot) \quad f_2(\cdot) \quad f_3(\cdot)] \begin{bmatrix} V_{\text{move\_to\_goal}} \\ V_{\text{avoid\_obstacle}} \\ V_{\text{follow\_wall}} \end{bmatrix}. \quad (9)$$

Some examples of wall-following to avoid large obstacles are shown in Figure 5 using the combination of the above designed behaviors.

**2.2.4. Behavior of Avoiding Robot.** In order to prevent collision between robots, the behavior of avoiding robot will be triggered when the robot is too close to the one in front of it (as shown in Figure 2(d)). The movement direction of this behavior is opposite to the robot in front of it. The virtual force of the avoiding behavior will be enhanced when the robot moves close to the other one. Assume the current coordinate of the robot  $R_j$  is  $(x_j, y_j)$  and the current coordinate of the robot  $R_i$  is  $(x_i, y_i)$ , then the vector of the behavior of avoiding robot is

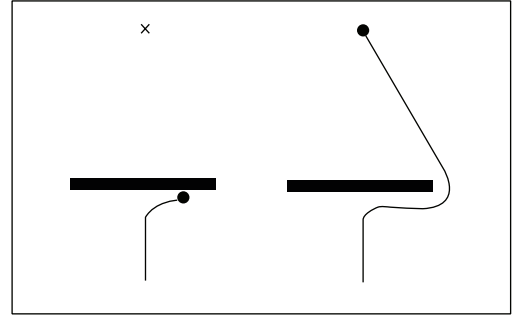
$$V_{\text{avoid\_robot}} = \frac{1}{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}} \begin{bmatrix} \pm (x_j - x_i) \\ \pm (y_j - y_i) \end{bmatrix}. \quad (10)$$

The control parameters of the behavior are

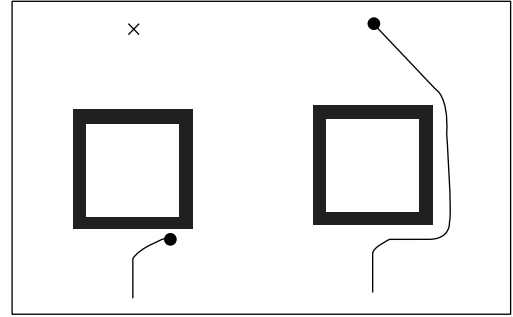
$$f_4(d_r) = \begin{cases} 0, & d_r \in (b_r, +\infty) \\ a_r d_r, & d_r \in [2r_0, b_r], \end{cases} \quad (11)$$

where  $d_r$  is the distance between the robot and the robot in front of it,  $a_r$  and  $b_r$  are adjustable parameters,  $b_r$  is the critical distance between these two robots, and  $r_0$  is the radius of the robot (as shown in Figure 3(b)).

**2.2.5. Behavior of Keeping Formation.** In order to keep a desired formation while moving, each robot should maintain



(a) The process of avoiding large obstacles (type 1)



(b) The process of avoiding large obstacles (type 2)

FIGURE 5: Performance with the wall-following behavior to avoid large obstacles.

a certain angle and distance to other robots. As shown in Figure 2(e), the robots  $R_i$  and  $R_j$  are moving forward side by side.  $R_j$  maintains its relative position in the formation by referencing to  $R_i$ .  $\alpha_{j-i}$  is the angle between formation's direction and the line of  $R_j$  and  $R_i$ .  $d_{j-i}$  is the distance between  $R_j$  and  $R_i$ . The ideal location of  $R_j$  in the formation is determined by that of  $R_i$ , the moving direction of the formation,  $\alpha_{j-i}$  and  $d_{j-i}$ . Assume the current coordinate of  $R_i$  is  $(x_i, y_i)$  and  $\theta$  is the moving direction of the formation. Then the ideal coordinate of  $R_j$  is

$$\begin{bmatrix} x_{jf} \\ y_{jf} \end{bmatrix} = \begin{bmatrix} x_i + d_{j-i} \cos(\alpha_{j-i} + \theta) \\ y_i + d_{j-i} \sin(\alpha_{j-i} + \theta) \end{bmatrix}. \quad (12)$$

When the location of  $R_j$  is not the same as the ideal coordinate, that is,  $\sqrt{(x_{jf} - x_j)^2 + (y_{jf} - y_j)^2} > \varepsilon$ , the behavior of keeping formation is activated and we have

$$V_{\text{keep\_formation}} = \frac{1}{\sqrt{(x_{jf} - x_j)^2 + (y_{jf} - y_j)^2}} \begin{bmatrix} x_{jf} - x_j \\ y_{jf} - y_j \end{bmatrix}. \quad (13)$$

The control parameter of the formation keeping behavior is

$$f_5(d_k) = \begin{cases} 0, & d_k \in (0, \varepsilon) \\ a_k d_k, & d_k \in (\varepsilon, +\infty), \end{cases} \quad (14)$$

where  $d_k = \sqrt{(x_{jf} - x_j)^2 + (y_{jf} - y_j)^2}$  is the distance to the ideal location and  $a_k$  is an adjustable parameter.

- (1) Initialize  $\Omega_R = \{R_0, R_1, \dots, R_{n-1}\}$ ,  $\Omega_T = \{T_0, T_1, \dots, T_{n-1}\}$ , set  $PRI = n$ , which represents the priority and when  $n = 0$ , the priority is highest.
- (2) **repeat** (for each robot in the formation)
- (3) Search the target position nearest to  $R_i \in \Omega_R$  and denote the distance as  $d_i$ . If there are several  $d_i$  with the same value, the target position will be chosen randomly.
- (4) Set  $d_m = \max d_i$ ,  $PRI = PRI - 1$ . Assume that  $d_m$  is the distance between  $R_j \in \Omega_R$  and  $T_a \in \Omega_T$ . Then  $T_a$  is the target position of  $R_j$  with the priority of  $PRI$ .
- (5) Set  $\Omega_R = \Omega_R - \{R_j\}$ ,  $\Omega_T = \Omega_T - \{T_a\}$ .
- (6) **until**  $\Omega_R = \emptyset$ ,

ALGORITHM 1: The main steps of the target searching algorithm.

### 3. Classification-Based Searching for Initial Formation Control

Initial formation is one of the most important tasks for various kinds of formation control. In this section, a classification-based target searching algorithm [30] is presented for effective initial formation control of the swarm robot systems.

**3.1. Target Searching Method.** Generally, there is a main control unit to implement the computing tasks of initial formation for the swarm robot system. Assume there are  $n$  robots in the formation, denoted as  $R_0, R_1, \dots, R_{n-1}$ , and their target positions are denoted as  $T_0, T_1, \dots, T_{n-1}$ , respectively.  $\Omega_R$  is the set of robots that have not got its target position.  $\Omega_T$  is the set of target positions that have not been assigned to any robot. The priority of the robots is denoted as  $PRI$ . The main steps of the target searching method [31] are shown as in Algorithm 1.

The main idea of the target search algorithm is that each robot chooses its target position as near as possible. If two or more robots choose the same target position, the farthest robot will get the highest priority. When two or more robots have the same distance with the same target position, the robot will get the highest priority randomly. The algorithm can reduce the possibility of the collision between the robots. An example with 4 robots is shown in Figure 6.

However, when the number of the robots tends to be hundreds, it will expose some shortcomings by using the standard target search algorithm. If all the robots are far away from the target formation, most of the robots will always share the same target position in the searching process. It means that most of the robots have to keep searching for the new nearest target positions. In addition, the algorithm distributes only one robot each time and has to compare all the remaining robots to find out the referred one. These defects reduce the efficiency of the algorithm in large-scale systems. Therefore we provide an improved method (classification-based target searching algorithm), which is more effective for swarm robots.

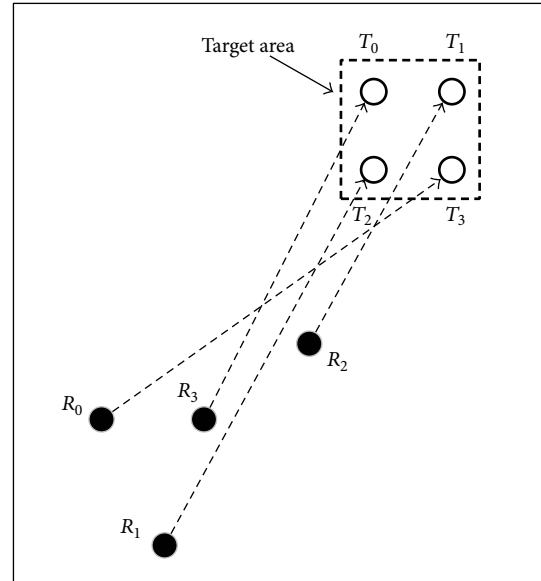


FIGURE 6: Schematic of initial formation with the target searching method.

**3.2. Classification-Based Target Searching Algorithm.** Assume that each robot moves in a two-dimensional workspace without static barriers. The position of each robot is denoted as  $P = [x, y]^T$ . To avoid high computation complexity, we divide the workspace into  $r \times s$  square cells. Let  $\alpha$  denote the acreage of each cell.  $\alpha$  should be greater than or equal to 25 times of a single robot's acreage so that the robots in a cell will not be too dense.  $\alpha$  has positive relationship with the acreage of the whole workspace and has negative relationship with the number of robots. Each cell is signed as  $C_{ij}$  denoting that the cell is in the  $i$ th column and  $j$ th row. Then we define the targets area as all the cells containing target positions. Figure 7(a) shows an example with 100 robots, where dots represent robots and the  $x$  represents a target position. The workspace is divided into  $8 \times 8$  cells. The shape of required formation is square in the center of the 2D space.

As shown in Figure 7(b), we have six typical types of cells, each of which represents different kinds of quantity

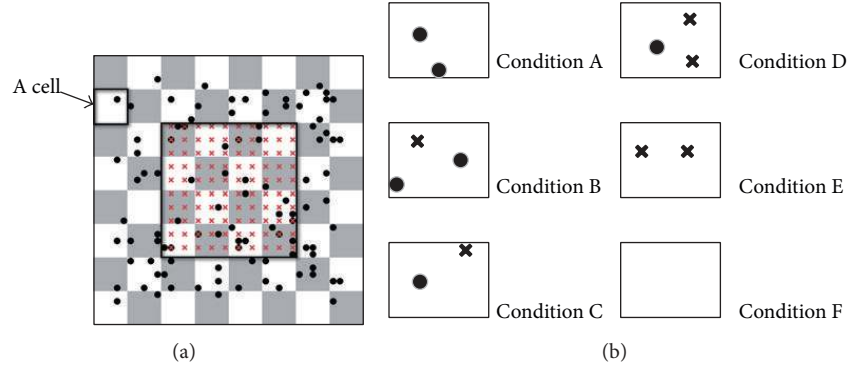


FIGURE 7: An example of the initial state of 100 robots before formation and six types of cells.

Initialize the robots  $R_0, R_1, \dots, R_{n-1}$ , the target positions  $T_0, T_1, \dots, T_{n-1}$ , the cells  $C_{ij}$ , ( $i = 1, 2, \dots, r, j = 1, 2, \dots, s$ ),  $\Omega_{ijR}, \Omega_{RA}, \Omega_{RB}, \Omega_{RC} = \emptyset$ .

- (1) **repeat** (for each robot in the formation)
  - repeat** (for each robot in the cells of type A)
    - (2) Search the robots in the cells of type A and put them into  $\Omega_{RA}$ .  
Use the target search algorithm to assign  $R_K \in \Omega_{RA}$  to the cells with  $t_{ij} \neq 0$ . Assume that  $R_m$  is assigned to  $C_{pq}$ . Put  $R_m$  into  $\Omega_{pqR}$ .  $PRI = PRI - 1, t_{pq} = t_{pq} - 1, \Omega_{RA} = \Omega_{RA} - \{R_m\}$ . Then continue to assign next  $R_K \in \Omega_{RA}$ .
  - (3) **until**  $\Omega_{RA} = \emptyset$ .
  - repeat** (for each robot in the cells of type B)
    - (4) Search the cells in type B, and put one of the robots in a cell of type B into  $\Omega_{RB}$ .  
Use the target search algorithm to assign  $R_K \in \Omega_{RB}$  to other cells with  $t_{ij} \neq 0$ . Assume that  $R_m$  is assigned to  $C_{pq}$ . Put  $R_m$  into  $\Omega_{pqR}$ .  $t_{pq} = t_{pq} - 1, \Omega_{RB} = \Omega_{RB} - \{R_m\}$ . Then continue to assign next  $R_K \in \Omega_{RB}$ .
  - (5) **until**  $\Omega_{RB} = \emptyset$ .
- (6) **until**  $\Omega_{RA} = \emptyset \ \& \ \Omega_{RB} = \emptyset$ .
- (6) Put  $R_k$  that has not be assigned in  $C_{ij}$  into  $\Omega_{ijR}$ .  
Use the target search algorithm to assign  $R_i \in \Omega_{ijR}$  to the targets in  $C_{ij}$ .

ALGORITHM 2: The classification-based search algorithm.

relationships between the robots and the target positions in a cell. Our approach is to provide the robots in different cell types with different assignment priorities. The robots in type A have no target position in their own cells. They are farther from the required formation than the robots in other types. So they have the highest assignment priority to move. The number of robots is more than that of target positions in type B, so the excess robots in type B will not be assigned to the target positions in their own cells. They get the second priority level. When types A and B have all been settled, only type C cells contain robots. The remaining robots will go to the target positions in their own cells. According to the above rules, it is clear that if all the cells of types A, B, and C are completely processed, the cells of types D, E, and F will be resolved or will disappear. Therefore, we can classify the robots into only three

categories (i.e., types A, B, and C), which can greatly decrease the computation complexity.

For the initial formation of  $n$  robots, let  $R_0, R_1, \dots, R_{n-1}$  denote the robots and let  $T_0, T_1, \dots, T_{n-1}$  denote the target positions. The workspace is divided into  $r \times s$  square cells.  $C_{ij}$  denotes the cell in column  $i$  and row  $j$  ( $i = 1, 2, \dots, r$  and  $j = 1, 2, \dots, s$ ).  $t_{ij}$  is the number of the robots assigned to  $C_{ij}$  and  $\Omega_{ijR}$  is the set of the robots assigned to  $C_{ij}$ . The distance between the robot and a cell is defined as the distance between the robot and the center of the cell.  $\Omega_{RA}$  is the set of the robots in type A.  $\Omega_{RB}$  is the set of the robots which are excess in the cells in type B.  $\Omega_{RC}$  is the set of the robots in type C. At first, all robots in  $\Omega_{RA}, \Omega_{RB}, \Omega_{RC}$  have not got their target positions yet. The classification-based search algorithm is summarized as in Algorithm 2.

TABLE 2: Control parameters of robot behaviors for initial formation.

$a_m$	$b_m$	$a_0$	$b_0$	$a_r$	$b_r$
1.0	4.0	1.0	4.0	2.0	3.0

If there is more than one cell or target position to a robot, this robot can choose any of them to be its cell or target position randomly.

**3.3. Experimental Results.** In order to verify the reliability of the classification-based target searching algorithm, which can effectively reduce the complexity of the algorithm, a number of simulation experiments are carried out. For the initial formation of a swarm robot system consisting of  $n$  robots, the complexity of calculating the distance between the robot and the target position is

$$n^2 + (n-1)^2 + \dots + 1 = \frac{n(n+1)(2n+1)}{6}, \quad (15)$$

and the complexity of comparison time of the distance between robots and the target positions is

$$(n+1)(n-1) + n(n-2) + \dots + 3 \cdot 1. \quad (16)$$

Therefore, the time complexity is  $O(n^3)$ . The time complexity reduces to  $O(n^2)$  if the robot has memory function, which is implemented in algorithm. As for classification-based target searching algorithm, the time complexity remains  $O(n^2)$ . However, the coefficient of the time complexity drops significantly according to the experimental results. Several groups of simulated experiments are demonstrated to show the performance of the proposed classification-based searching method regarding the computation complexity. The reliability of this algorithm is shown by whether the swarm robot system can quickly form a target formation without collision. All the movement of each robot is accomplished through the behavior-based control method and the control parameters of the robot behaviors are shown in Table 2.

**3.3.1. Performance Comparison of Systems with Different Numbers of Robots.** As shown in Figure 8, the robots and their target positions are distributed randomly. They are matched with target searching method and classification-based target searching algorithm, respectively, which is performed 10 times in each experiment. This map includes  $8 \times 8$  cells, the cell size is  $5 \times 5$  and each cell contains about 1.56 target positions averagely. For this group of experiments, the number of robots is set at 20, 100, and 200, respectively.

As shown in Table 3,  $n$  is the number of robots,  $n_a$  is the density of robots in the map,  $sd1$  is the distance calculation time of the target search method, and  $sd2$  is the distance calculation time of the classification-based target searching algorithm.  $sm1$  is distance comparison time of the target search method and  $sm2$  is the distance comparison time of the classification-based target searching algorithm. These results show that the classification-based target searching algorithm is superior to the standard target searching algorithm regarding the computation complexity, especially for large scale swarm robot systems.

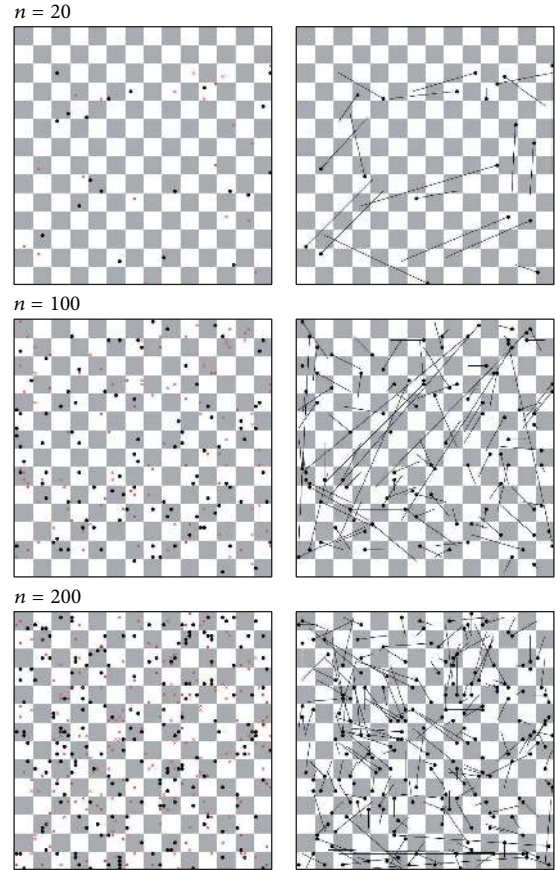


FIGURE 8: Performance of initial formation with different numbers of robots.

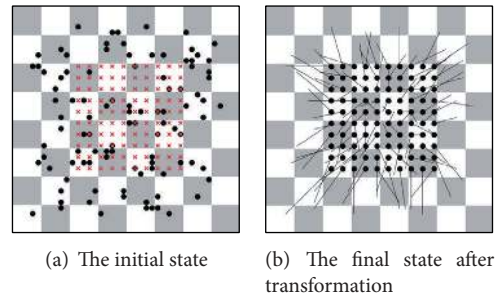


FIGURE 9: Initial formation control of a square shape.

**3.3.2. Time Complexity Comparison in Different Formation Transformation Tasks.** As shown in Figure 9, the robots are distributed on the map randomly and the target formation is a square. The map includes  $8 \times 8$  cells and the cell size is  $5 \times 5$ , each cell contains about 6.25 target point averagely. As for another case study shown in Figure 10, the number of the robots is 121, the initial formation shape is a square in the lower left of the map, while the target formation is a circle in top right. This map includes  $8.5 \times 8.5$  cells, the cell size is  $7 \times 7$  and each cell contains about 7.56 target point averagely. It is clear that both the target searching method and the classification-based target search method work well for the

TABLE 3: Performance comparison of the target searching method and the classification-based target searching method for different numbers of robots.

$n$	$n_a$	sd1	sd2	$sd2/sd1 \times 100\%$	sm1	sm2	$sm2/sm1 \times 100\%$
20	0.31	689	458	66.68%	833	508	61.09%
100	1.56	35021	5483	15.81%	39462	5830	14.87%
200	3.13	292355	9613	3.35	310116	11018	3.60%

TABLE 4: Performance comparison of the target searching method and the classification-based target searching method for different shapes of formation.

$n$	$n_a$	sd1	sd2	$sd2/sd1 \times 100\%$	sm1	sm2	$sm2/sm1 \times 100\%$
100	6.25	97405	2854	2.95%	100957	4965	4.95%
121	7.56	232851	12588	5.41%	237311	19084	8.04%

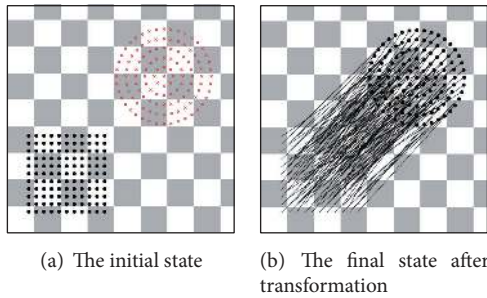


FIGURE 10: Initial formation control of a circular shape.

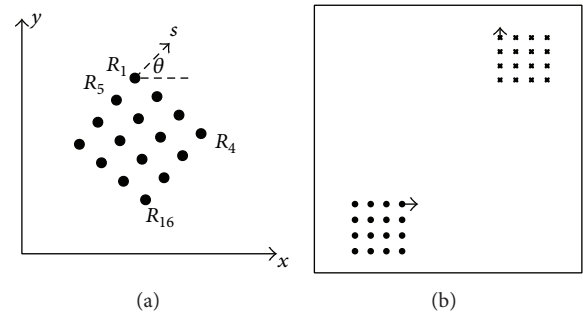


FIGURE 11: Basic model of a square formation and the problem of navigation without obstacle-avoidance.

initial formation problem, but regarding the computation complexity the classification-based searching method performs much better. More comparison details are listed in Table 4.

#### 4. Formation Control with Obstacle-Avoidance

In this section, we focus on the other important and challenging task of navigation in an unknown environment while keeping a desired formation. First, the proposed behavior-based method is applied for the navigation control without obstacle-avoidance, where the swarm robots (after initial formation) move to a desired target position in group while maintaining this formation. Then this method is applied to the navigation control problem with obstacle-avoidance. Several groups of simulated experiments are carried out to demonstrate the performance of the proposed method.

*4.1. Navigation without Obstacle-Avoidance.* As shown in Figure 11(a), a basic model of a square formation contains 16 robots ( $R_1, R_2, \dots, R_{16}$ ) and the angle  $\theta$  is the orientation of the formation (as marked with an arrow).

The problem of navigation without obstacle-avoidance is described in Figure 11(b). The initial location of the swarm robot system is on the lower left corner with the orientation  $\theta = \theta_0$  and the target position is on the right corner with the orientation  $\theta = \theta_t$ . The objective is to move the swarm robot system to the target position with  $\theta = \theta_t$  while maintaining the desired formation.

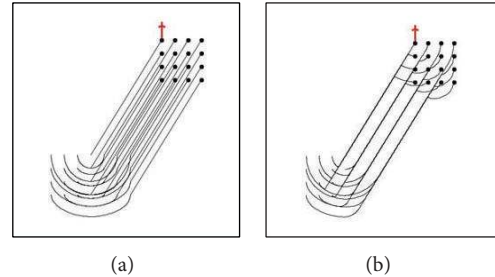


FIGURE 12: An example of navigation without obstacle-avoidance using traditional centralized approach.

An example of formation control while navigating without obstacle-avoidance using the simple centralized approach is shown in Figure 12, where the whole task is divided into two subtasks named straight movement and rotation, and the sequence of these two subtasks is adjustable. For example, firstly, the formation rotates with an angle to make  $\theta = \theta_t$  and then move straight (Figure 12(a)). Or the formation adjusts the angle to face the target point at first and then move straight, rotates with an angle to make  $\theta = \theta_t$  (Figure 12(b)). This centralized control method is suitable to the task with strong centrality (e.g., the task of collective transportation). However, this method is not applicable to the complex environments or distributed tasks and its performance is not good regarding the cost of general movement. In order to complete the task of formation control in distributed and complex



TABLE 5: Control parameters of robot behavior for the problem of navigation without obstacle-avoidance.

$a_m$	$b_m$	$a_r$	$b_r$	$a_k$
1.0	4.0	3.0	2.0	1.0

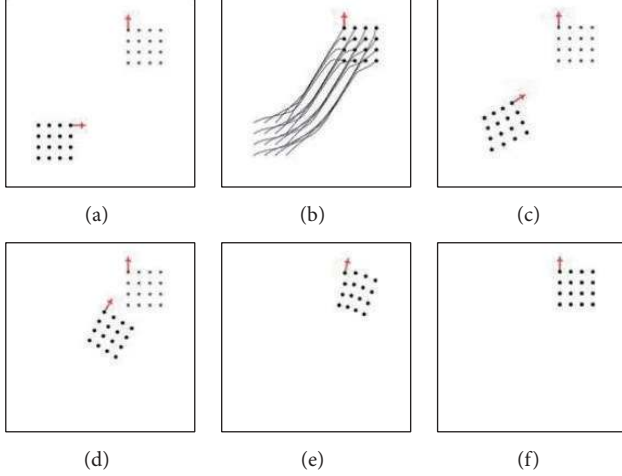


FIGURE 13: Performance of formation control of 16 robots in a square shape while navigating without obstacle-avoidance.

environments for swarm robots, the behavior-based control method of formation control is applied. The robot has the ability of formation-maintaining, obstacle-avoiding, and wall-following. The behavior vector for each robot is given by

$$V_{\text{direction}} = [f_1(\cdot) \quad f_4(\cdot) \quad f_5(\cdot)] \begin{bmatrix} V_{\text{move\_to\_goal}} \\ V_{\text{acioid\_robot}} \\ V_{\text{keep\_formation}} \end{bmatrix}. \quad (17)$$

With the parameter settings shown as in Table 5, the performance for the problem of formation control of 16 robots without obstacle-avoidance is shown in Figure 13. It is clear that the rotation task is accomplished while moving toward the target position. Thus, the formation will be slightly deformed in the process of rotation, but the whole system still maintains the basic shape in general.

To evaluate the performances regarding formation-keeping and movement smoothness, we define the ratio of deformation  $\delta_j$  for robot  $R_j$  as

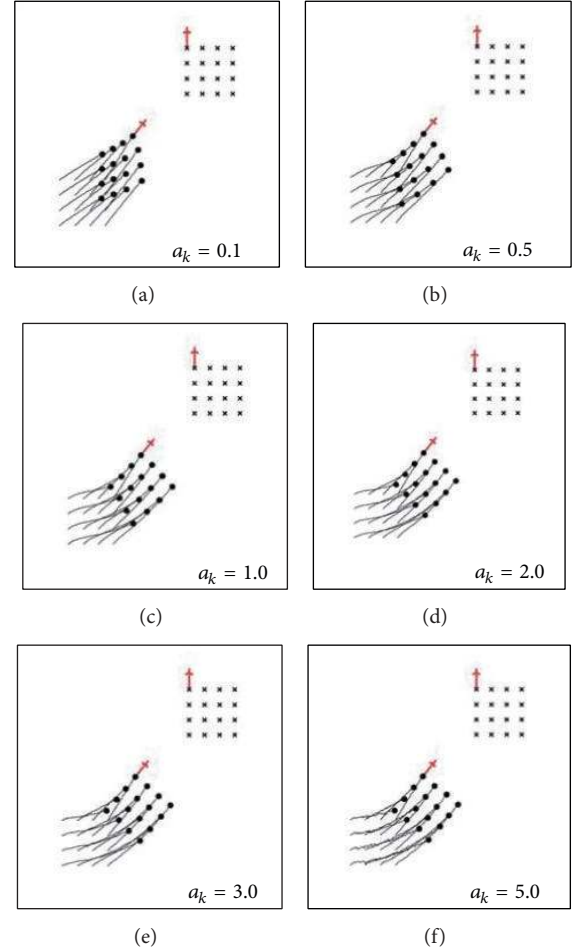
$$\delta_j = \frac{2d_{kj}}{d_{j-i}}, \quad (18)$$

where  $d_{kj}$  is the distance between robot  $R_j$  and its ideal location in the formation.  $d_{j-i}$  is the fixed distance which should be maintained between robot  $R_j$  and its reference robot  $R_i$ . The maximum ratio of deformation is

$$\delta_{\max} = \max_{1 \leq j \leq n} \delta_j. \quad (19)$$

 TABLE 6: Performances of formation control regarding  $\delta_{\max}$  for different  $a_k$ .

$a_k$	0.1	0.2	0.5	0.7	0.8	1.0	1.5	2.0	3.0	4.0	5.0
$\delta_{\max}$	1.97	1.73	1.14	1.01	0.9	0.8	0.65	0.57	0.52	0.61	0.75


 FIGURE 14: Performances of formation control for  $a_k = 0.1, 0.5, 1.0, 2.0, 3.0, 5.0$ , respectively.

In order to maintain the basic formation,  $\delta_{\max}$  must be less than 1, which is much related with the control parameter  $a_k$  for the behavior of keeping formation. The larger  $a_k$  is the greater the intensity of the formation-keeping behavior will be. But larger  $a_k$  may decrease the smoothness of the movement trajectory. Figure 14 shows a group of simulated experiments with different  $a_k$  and more details are listed in Table 6. It is clear that, to maintain a good performance of formation control while navigating without obstacle-avoidance, the value of  $a_k \in [1.0, 3.0]$  is acceptable.

Another group of experimental results for the formation control of 100 robots in a square shape is shown as in Figure 15. The initial and target orientation are  $\theta_0 = -3\pi/4$  and  $\theta_t = \pi/2$ . We set  $a_k = 1.5$  and the results show that this

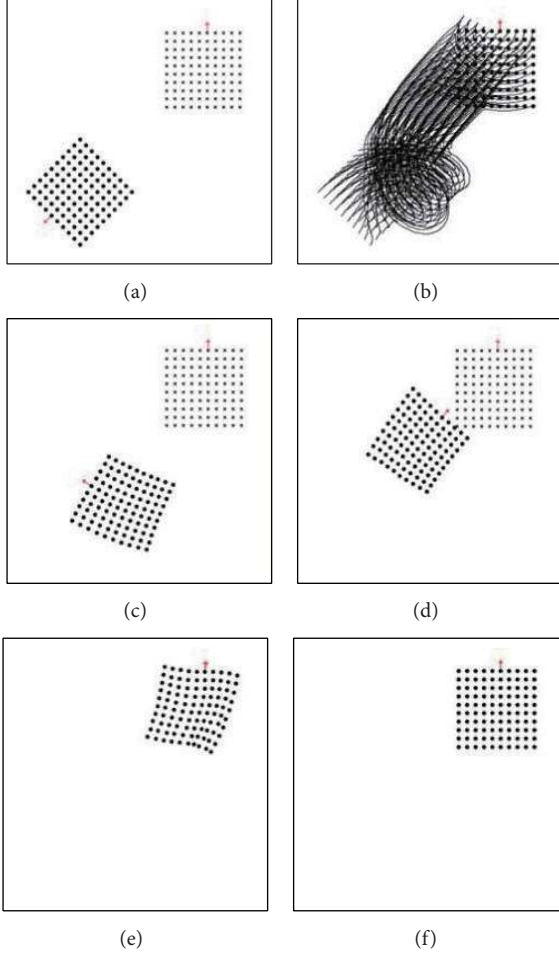


FIGURE 15: The performance of formation control of 100 robots in a square shape while navigating without obstacle-avoidance.

swarm robot system reaches the target location in the desired shape very successfully.

**4.2. Navigation with Obstacle-Avoidance.** For the simple environments with a few distributed obstacles as shown in Figure 16, the behavior-based formation control method does not need to adopt the behavior of wall-following. The behavior vector for every single robot is

$$V_{\text{direction}} = [f_1(\cdot) \quad f_2(\cdot) \quad f_4(\cdot) \quad f_5(\cdot)] \begin{bmatrix} V_{\text{move\_to\_goal}} \\ V_{\text{avoid\_obstacle}} \\ V_{\text{avoid\_robot}} \\ V_{\text{keep\_formation}} \end{bmatrix}. \quad (20)$$

When any obstacles are detected, the behavior of avoiding obstacles will be triggered. To ensure the success of avoiding obstacles, the behavior of formation-keeping will be weakened when there is any obstacle detected. Only after leaving the obstacle region, the desired shape of formation will be

TABLE 7: The control parameters of the behavior-based method in situation of obstacles.

$a_m$	$b_m$	$a_0$	$b_0$	$a_r$	$b_r$	$a_k$
1.0	3.0	1.0	4.0	2.0	2.0	1.0

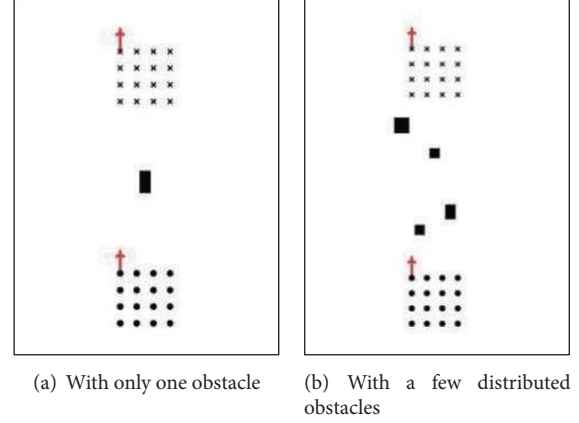


FIGURE 16: Simple environments with a few distributed obstacles.

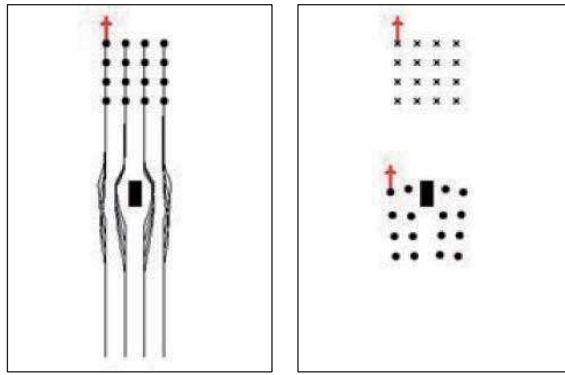
recovered. The control parameters of formation-keeping are modified as

$$f_5(d_5) = \begin{cases} 0, & d_k \in [0, \varepsilon] \\ a_k d_k, & d_k \in (\varepsilon, +\infty) \text{ and no obstacles} \\ a_k \ln d_k, & d_k \in (\varepsilon, +\infty) \text{ and with obstacles.} \end{cases} \quad (21)$$

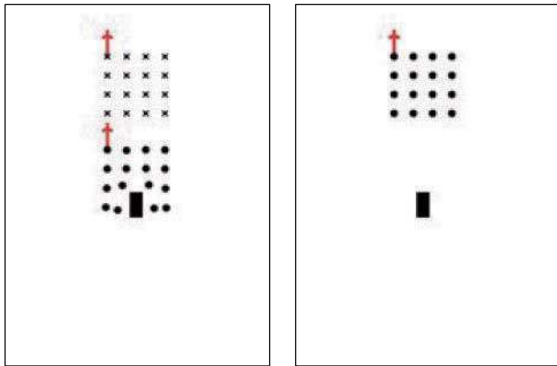
Figures 17 and 18 show the performance of formation control while avoiding obstacles, where a single obstacle is considered in Figure 18. The control parameters of the behavior-based method in situation of obstacles are listed in Table 7. Both of these two groups of experiments show the success of the proposed method for formation control while navigating with simple obstacle-avoidance.

**4.3. Navigation in Complex Environment with Obstacle-Avoidance.** For more complex environments where there are large obstacles or even local minima (e.g., as shown in Figure 19), it is difficult to avoid the large obstacles by only using the behavior-based method and a communication mechanism will be necessary to help the swarm robot system avoid the large obstacles as an entity. The basic idea can be described as follows: according to the limited information about the obstacles detected by each robot, we design an appropriate information integration and decision-making mechanism to enable the unit to determine the existence of large obstacles and bypass the obstacles under the guidance of local sensing information.

The local range sensing information is modeled as shown in Figure 1(b) with  $\{L_1, L_2, R_1, R_2\}$ . For example, if any obstacle is detected in the  $L_1$  area, we denote  $L_1 = 1$ , otherwise  $L_1 = 0$ . Hence if more than  $M$  robots in the front line of the formation detect obstacles in front of them (i.e.,  $L_1 = R_1 = 1$ ), it is considered that the whole formation encounters



(a) Overall path of obstacle-avoidance (b) Step 1: avoid a single obstacle

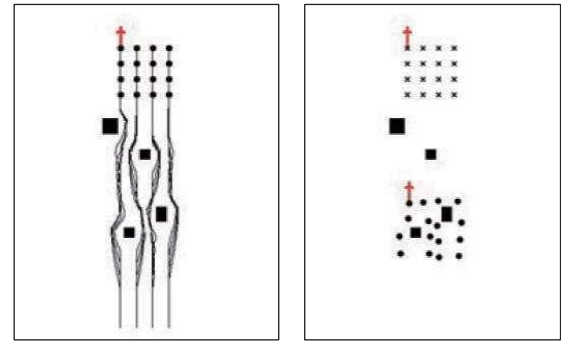


(c) Step 2: avoid a single obstacle (d) Step 3: arrive at the destination

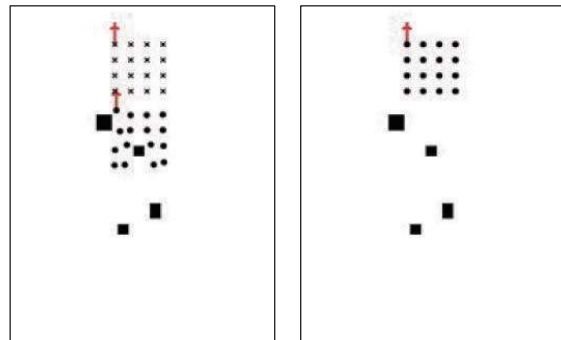
FIGURE 17: Performance of formation control while avoiding a single obstacle.

a large obstacle or clustered multiobstacle.  $M$  is an adjustable parameter related to the formation size and the number of robots. In this paper, when the number of robots is 16, the diameter of the robot is 1 meter and the spacing between robots is 3 meters, we set  $M = 2$ . Then we have the method of avoiding large obstacles as follows. (1) If there are more than  $M$  robots in the front line that encounter obstacles (whose sensing information  $L_1 = R_1 = 1$ ), the overall obstacle-avoiding will be triggered and an integrated moving direction will be given; (2) the whole formation takes this integrated movement until the sensor information is  $L_1 = R_1 = 0$  for all robots in the front line; (3) the whole formation goes forward until no robot on both sides detects any obstacle (i.e.,  $L_2 = R_2 = 0$ ). Then the behavior-based formation control mechanism will be recovered.

Two groups of experimental results are shown in Figures 20 and 21, respectively. In both of these two complex environments, the swarm robot system can avoid large obstacles successfully. In particular, as shown in Figure 21, after detecting the large obstacles, the whole system finds that the obstacle does not block in front of the formation completely, so it decides to move left to avoid the large obstacle. After bypassing the large obstacle, the swarm robot system turns to the basic behavior-based formation control mechanism and continues to navigate with obstacle-avoiding.

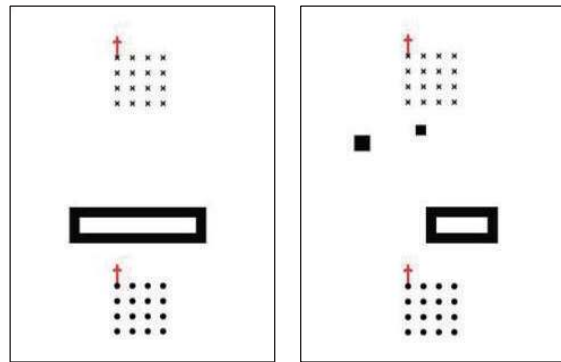


(a) Overall path of obstacle-avoidance (b) Step 1: avoid obstacles



(c) Step 2: avoid obstacles (d) Step 3: arrive at the destination

FIGURE 18: Performance of formation control while avoiding multiple obstacles.



(a) Only with large obstacles (b) Complex environments

FIGURE 19: Complex environments with large obstacles.

### 5. Conclusions

In this paper, a behavior-based formation control approach for swarm robots is proposed by extending the behavior-based navigation method for single robots. Several behaviors are defined and studied for two typical formation control problems, that is, initial formation and formation control while navigating with obstacle-avoidance. For the initial formation problem, a classification-based searching method for any arbitrary shape of formation generating is presented. For

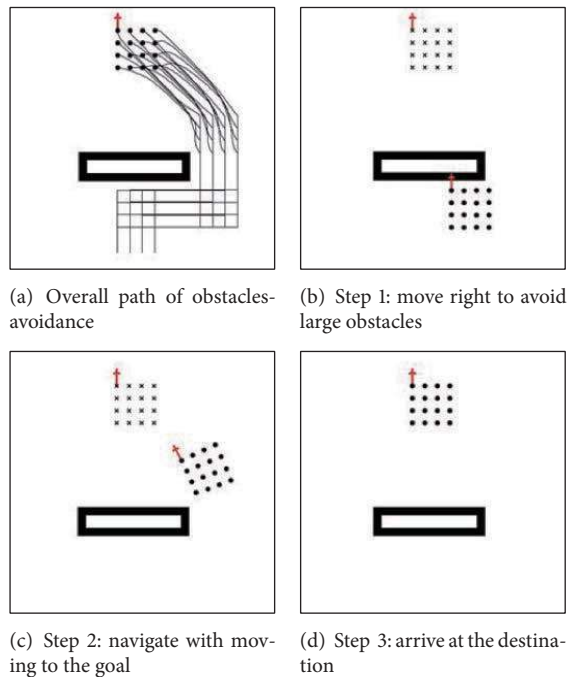


FIGURE 20: Performance of avoiding a large obstacle.

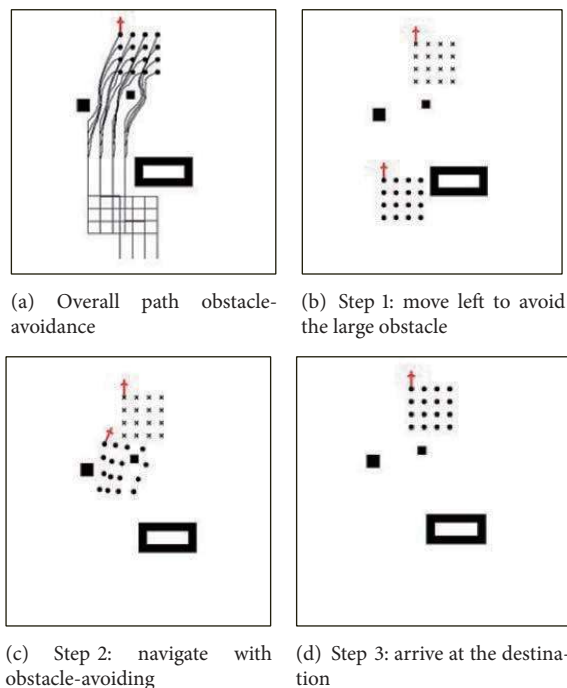


FIGURE 21: Performance of avoiding multiple obstacles including large obstacles.

the navigation problem, different situations are comprehensively studied, which includes navigation without obstacle-avoidance, navigation with obstacle-avoidance, and navigation in complex environments. All the simulated experimental results demonstrate the success of the proposed behavior-based formation control approach for swarm robots. Our

future work will focus on robust control of swarm robots with uncertainties [32] and more practical issues regarding real swarm robotic systems.

### Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

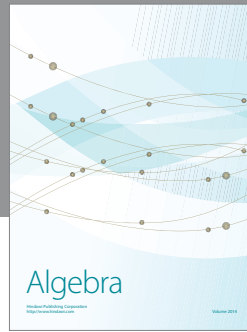
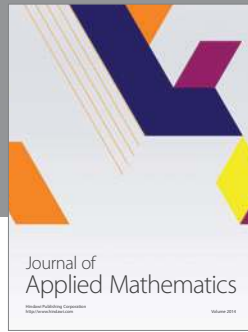
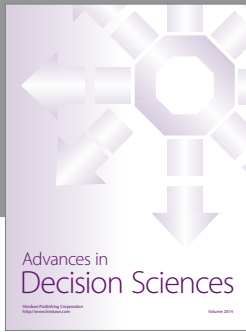
### Acknowledgment

This work was supported by the Natural Science Foundation of China under Grant no. 61273327.

### References

- [1] A. Marino, L. E. Parker, G. Antonelli, and F. Caccavale, "A decentralized architecture for multi-robot systems based on the null-space-behavioral control with application to multi-robot border patrolling," *Journal of Intelligent & Robotic Systems: Theory and Applications*, vol. 71, no. 3-4, pp. 423-444, 2013.
- [2] B. Ranjbar-Sahraei, F. Shabaninia, A. Nemati, and S.-D. Stan, "A novel robust decentralized adaptive fuzzy control for swarm formation of multiagent systems," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 8, pp. 3124-3134, 2012.
- [3] J.-W. Kwon and D. Chwa, "Hierarchical formation control based on a vector field method for wheeled mobile Robots," *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1335-1345, 2012.
- [4] B. Fidan, V. Gazi, S. Zhai, N. Cen, and E. Karatas, "Single-view distance-estimation-based formation control of robotic swarms," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 12, pp. 5781-5791, 2013.
- [5] Y. Zhang, L. Zeng, Y. Li, and Q. Liu, "Multi-robot formation control using leader-follower for MANET," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO '09)*, pp. 337-342, December 2009.
- [6] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926-939, 1998.
- [7] L. L. He and C. L. Xiao, "Study on the formation control methods for multi-agent based on geometric characteristics," *Advanced Materials Research*, vol. 765, pp. 1928-1931, 2013.
- [8] R. M. K. Chetty, M. Singaperumal, and T. Nagarajan, "Behavior based multi robot formations with active obstacle avoidance based on switching control strategy," *Advanced Materials Research*, vol. 433-440, pp. 6630-6635, 2012.
- [9] Z. Linan, Y. Peng, C. Lingling, Z. Xueping, and T. Yantao, "Obstacle avoidance of multi mobile robots based on behavior decomposition reinforcement learning," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO '07)*, pp. 1018-1023, December 2007.
- [10] J. Chen, D. Sun, J. Yang, and H. Chen, "Leader-follower formation control of multiple non-holonomic mobile robots incorporating a receding-horizon scheme," *International Journal of Robotics Research*, vol. 29, no. 6, pp. 727-747, 2010.
- [11] T. Dierks and S. Jagannathan, "Control of nonholonomic mobile robot formations: backstepping kinematics into dynamics," in *Proceedings of the 16th IEEE International Conference on Control Applications (CCA '07)*, pp. 94-99, October 2007.

- [12] J. Ghommam, H. Mehrjerdi, and M. Saad, "Leader-follower based formation control of nonholonomic robots using the virtual vehicle approach," in *Proceedings of the IEEE International Conference on Mechatronics (ICM '11)*, pp. 516–521, April 2011.
- [13] G. L. Mariottini, F. Morbidi, D. Prattichizzo, G. J. Pappas, and K. Daniilidis, "Leader-follower formations: uncalibrated vision-based localization and control," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '07)*, pp. 2403–2408, 2007.
- [14] A. Benzerrouk, L. Adouane, L. Lequievre, and P. Martinet, "Navigation of multi-robot formation in unstructured environment using dynamical virtual structures," in *Proceedings of the 23rd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '10)*, pp. 5589–5594, October 2010.
- [15] T. Balch and M. Hybinette, "Behavior-based coordination of large-scale robot formations," in *Proceedings of the 4th IEEE International Conference on MultiAgent Systems*, pp. 363–364, 2000.
- [16] W. Li and W. Shen, "Swarm behavior control of mobile multi-robots with wireless sensor networks," *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1398–1407, 2011.
- [17] N. Basilico, N. Gatti, and F. Amigoni, "Leader-follower strategies for robotic patrolling in environments with arbitrary topologies," in *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '09)*, vol. 1, pp. 57–64, International Foundation for Autonomous Agents and Multiagent Systems, 2009.
- [18] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques, "Leader-follower formation control of nonholonomic mobile robots with input constraints," *Automatica*, vol. 44, no. 5, pp. 1343–1349, 2008.
- [19] A. Essghaier, L. Beji, M. A. El Kamel, A. Abichou, and J. Lerbet, "Co-leaders and a flexible virtual structure based formation motion control," *International Journal of Vehicle Autonomous Systems*, vol. 9, no. 1-2, pp. 108–125, 2011.
- [20] A. Sadowska, H. Huijberts, D. Kostić, N. van de Wouw, and H. Nijmeijer, "Formation control of unicycle robots using the virtual structure approach," in *Proceedings of the 15th IEEE International Conference on Advanced Robotics (ICAR '11)*, pp. 365–370, IEEE, June 2011.
- [21] W. Xu and X. Chen, "Artificial moment method for swarm robot formation control," *Science in China F: Information Sciences*, vol. 51, no. 10, pp. 1521–1531, 2008.
- [22] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques, "A geometric characterization of leader-follower formation control," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '07)*, pp. 2397–2402, IEEE, April 2007.
- [23] R. Dougherty, V. Ochoa, Z. Randles, and C. Kitts, "A behavioral control approach to formation-keeping through an obstacle field," in *Proceedings of the IEEE Aerospace Conference*, vol. 1, March 2004.
- [24] J. C. Derenick and J. R. Spletzer, "Convex optimization strategies for coordinating large-scale robot formations," *IEEE Transactions on Robotics*, vol. 23, no. 6, pp. 1252–1259, 2007.
- [25] J. P. Desai, J. Ostrowski, and V. Kumar, "Controlling formations of multiple mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 4, pp. 2864–2869, May 1998.
- [26] T. A. Vidal-Calleja, C. Berger, J. Solà, and S. Lacroix, "Large scale multiple robot visual mapping with heterogeneous landmarks in semi-structured terrain," *Robotics and Autonomous Systems*, vol. 59, no. 9, pp. 654–674, 2011.
- [27] A. Martinoli, F. Mondada, N. Correll et al., Eds., *Distributed Autonomous Robotic Systems*, vol. 83 of *Springer Tracts in Advanced Robotics*, 2013.
- [28] J. Witt, B. Annighöfer, O. Falkenberg, and U. Weltin, "Design of a high performance quad-rotor robot based on a layered real-time system architecture," in *Intelligent Robotics and Applications*, S. Jeschke, H. Liu, and D. Schilberg, Eds., vol. 7101 of *Lecture Notes in Computer Science*, pp. 312–323, Springer, Berlin, Germany, 2011.
- [29] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, pp. 14–23, 1986.
- [30] X. Zhang, F. Ouyang, and C. Chen, "A classification-based search method for generating large-scale robot formation," in *Proceedings of the 6th Chinese Conference on System Simulation Technology and Application (SSTA '11)*, pp. 1029–1033, Huangshan, China, August 2011.
- [31] Y. Cai and S. X. Yang, "An improved PSO-based approach with dynamic parameter tuning for cooperative multi-robot target searching in complex unknown environments," *International Journal of Control*, vol. 86, no. 10, pp. 1720–1732, 2013.
- [32] S. Chen and C. Chen, "Probabilistic fuzzy system for uncertain localization and map building of mobile robots," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 6, pp. 1546–1560, 2012.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

