Time-varying Algorithm for Swarm Robotics

Ligang Hou, Fangwen Fan, Jingyan Fu, and Jinhui Wang

***Abstract*—Ever since the concept of swarm intelligence was brought out, a variety of control algorithms for swarm robotics has been put forward, and many of these algorithms are stable enough and efficient. Most of the researches only take an invariable controller which functions through the whole stage into consideration, the situation in which controller changes over time is rarely taken into account. However, there are limitations for invariable controller dominated algorithms in practical situation, which makes them unable to meet changing environment. On the contrary, variable controller is more flexible and can be able to adapt to complex environment. Considering such advantages, a time-varying algorithm for swarm robotics is presented in this paper. The algorithm takes time as one of the independent variables so that the controller is no longer fixed through the time, but can be changed over time, which brings more choices for the swarm robot system. In this paper, some relevant simulations are designed to test the algorithm. Different control strategies are applied on the same flock during the time, and a more complex, flexible and practical control effect is acquired successfully.**

***Index Terms*—Advanced self-organization behavior, agent-based distributed control system, aggregation, swarm intelligence.**

I. INTRODUCTION

SWARMING behavior is ubiquitous in nature, some species may be capable of achieving extremely complex tasks after their individuals form into swarm following certain orders, and as a result, benefit the migration, hunting, predators avoiding and nests building. The aggregation of individuals brings about quantity advantage, and in addition the intelligence displayed by the flock is far beyond what individual organisms possess, and it can be surprising even for humans. Such natural swarm system is formed with a strong robustness. Increasing or decreasing the number of individuals in a large scale will not affect its function and its self-organization ensure that the system will not break down due to the failure of any "core". For such advantages, more and more researchers began to study and simulate the behavior of nature flocks, and attempted to copy their operating mode to the artificial intelligence system. In 1987, Reynolds [1] built a distributed behavioral model to simulate the behavior of a flock of birds, and proved that simple individual could achieve aggregate motion by dense interaction. This is the beginning of research on nature flock. In 1993, Beni *et al.* [2] suggested that cellular robotic systems were capable of "intelligent behavior", and put forward the concept of swarm intelligence. After that, more and more algorithms for swarm robotics were proposed, typical cases include: Spector *et al.* [3], [4] put forward SWARM and SWARMEVOLVE algorithms based on the work of Reynolds. Gazi *et al.* [5]−[7] used the artificial potential functions and the sliding-mode control technique to achieve swarm aggregation, formation and foraging based on the work of Khatib [8]. Virágh *et al.* [9] designed a flocking algorithm for autonomous flying robots. Reebs [10] tested the possibility of a minority of informed leaders leading a fish shoal to food, and later Couzin *et al.* [11] built a mathematical model to show how leadership affect s animal groups. Instead of proposing new algorithm, some researchers focused on the stability analysis [12]−[14], embedded systems are also involved to test or improving swarm algorithm [15]−[17].

With the rising attention on the swarm intelligence, research in this area kept developing, although the various algorithms described above have been able to achieve flexible motion control, but the invariable controller dominated algorithms usually have limited ability in practical situation. Once the controller is set up, it could not be changed until the task is over or the robot is reconfigured, and also sometimes they are unable to meet the changing environment. On the contrary, algorithm with variable controller is more suitable to describe the operating mode of swarm robot system in practical scenario and has a stronger adaptability to environment: the variation of controller can be arranged along with the robots configuration so that the robot can complete a set of tasks without reconfiguration. Also the controller can be changed at any time if only the users issue relevant command to the robot, and the following actions of the robot will be dominated by new controller. Considering all these advantages and the fact that few researches focus on variable algorithm, this paper proposes a time-varying algorithm. The design of the algorithm and implementation details are discussed in Section II. Two models are built in Section III to establish the practical time-varying controllers and simulations are implemented to test actual control effect of the algorithm. A conclusion is drawn in Section IV.

II. ALGORITHM DESIGN

As mentioned previously, the aim of the algorithm is to make the controller contain time-varied elements, or make it a function of time, when the time changes the control function also changes. In addition, since the swarm may have division of labor and communication delay, it is allowed that

Manuscript received September 29, 2016; accepted November 17, 2016. This work was supported by the Beijing Municipal Natural Science Foundation (4152004) and the National Natural Science Foundation of China (61204040). Recommended by Associate Editor Yanjun Liu. *(Corresponding author: Ligang Hou.)*

Citation: L. G. Hou, F. W. Fan, J. Y. Fu, and J. H. Wang, "Time-varying algorithm for swarm robotics," *IEEE/CAA J. of Autom. Sinica*, vol. 5, no. 1, pp. 217−222, Jan. 2018.

L. G. Hou, F. W. Fan, and J. Y. Fu are with the Laboratory of VLSI and Systems, College of Microelectronics, Beijing University of Technology, Beijing 100124, China (e-mail: houligang@bjut.edu.cn; fanfangwen@emails.bjut.edu.cn; fujingyan@emails.bjut.edu.cn).

J. H. Wang is with the Department of Electrical and Computer Engineering, North Dakota State University, Fargo ND 58102, USA (e-mail: jinhui.wang.1@ndsu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2017.7510685

each individual is to be driven by different functions at the same time to increase the flexibility of the algorithm. The implementation details are as follow:

Set up a swarm system composed of M agents, let S^j be a set of control functions (suppose there are N functions in total), written as:

$$S^j = \{F_1^j, F_2^j, \dots, F_i^j, \dots, F_N^j\}$$
$$1 \le i \le N;\ 1 \le j \le M;\ i, j \in N.$$

Arrange each member of S^j to act on the agent j respectively during $[0, t_1^j]$, $(t_1^j, t_2^j], \dots, (t_{i-1}^j, t_i^j], \dots, (t_{N-1}^j, t_N^j]$. In order to express the control function at any time $t \in [0, t_N^j]$, a function is created as shown below:

$$f^j(t) = \sum_{i=1}^{N} F_i^j \cdot \left[\varepsilon(t - t_{i-1}^j) - \varepsilon(t - t_i^j)\right] \tag{1}$$

where $\varepsilon(t)$ is unit step function written as $\varepsilon(t) = \begin{cases} 1, & t \ge 0 \\ 0, & t < 0 \end{cases}$.

Take all the agents into consideration, there will be functions $f^1(t), f^2(t), \dots, f^j(t), \dots, f^M(t)$ respectively acting on each agent.

Function shown above indicates that there is only one control function acting at any specific time t. When t changes from 0 to t_N^j, the control function also changes from F_1^j to F_N^j, and thus take turns dominating the motion of agent. There are plenty of control functions for F_i^j, flexible and time-varied control effect can be achieved by selecting proper functions to fill set S^j. The control functions in S^j can be scheduled before the swarm system is activated. In addition, adding, removing or replacing functions during working process is also allowed. That means modification of the system can be made at any time to match the changing environment or tasks.

III. Simulation

In order to verify the control effect of the algorithm, two motion models are established and simulated separately based on the time-varying algorithm.

A. Simulation of Interaction Force Model

Supposing that all the agents interact with each other by exerting attractive force or repulsive force, agents can form into a stable formation when the force on each agent reaches a balance. Based on the R-A model [18], the interaction force model can be established as shown below:

$$\vec{F}^j = \frac{C_x^j \cdot \vec{i} + C_y^j \cdot \vec{j}}{|C_x^j \cdot \vec{i} + C_y^j \cdot \vec{j}|} \tag{2}$$

$$C_x^j = \sum c_i(x_i - x)\frac{1}{d_i} \tag{3}$$

$$C_y^j = \sum c_i(y_i - y)\frac{1}{d_i} \tag{4}$$

where $\vec{F}^j$ represents the resultant force on agent j, and C_x^j, C_y^j is the magnitude of its horizontal and vertical components, $\vec{i}$ is horizontal unit vector and $\vec{j}$ is vertical unit vector, x, y are coordinates of agent j, x_i, y_i represent coordinates of agent i which is around agent j (supposing there are K agents around agent j, then $0 \le i \le K$, $i \in N$). d_i is the distance between agent j and agent i, the interaction between the agents gets strengthened when distance between them decreases. c_i is a proportionality coefficient related to the distance between agents, and reflects the direction and the strength of the interaction force, it is defined as follows:

$$c_i = \begin{cases} -c_a, & d_i < a \\ 0, & a \le d_i \le b \\ c_b, & d_i > b \end{cases} \tag{5}$$

where c_a, c_b are always positive constants. The repulsive force dominates when the distance between two agents is less than a, and c_i should be assigned a negative value in such condition; when the distance is larger than a but less than b, agents reach a state of balance, c_i should be assigned 0; The attractive force dominates when the distance between two agents is larger than b, and c_i should be assigned a positive value. Different a, b, c_a, c_b can be selected to achieve different control effect, for variable b, the larger it is, the easier for agents to attract each other, and variable a determines the tightness of the formation, smaller a leads to a tighter formation, c_a, c_b determines the magnitude of attractive force and repulsive force, they should be considered along with variable a and b. Fig. 1 describes the interaction force model.

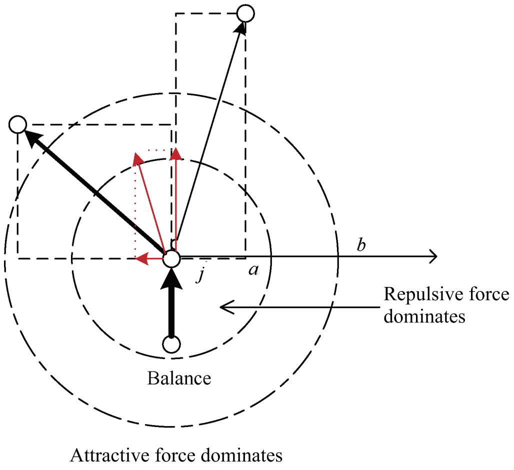


Fig. 1. Interaction force model.

Now consider the situation as follows: in a two-dimensional environment there are randomly placed 10 control agents, agents are in a quiescent state initially, then the control function (under the current circumstances, all agents are dominated by same control function, which means $\vec{F}_1^1 = \vec{F}_1^2 = \cdots = \vec{F}_1^{10}$, so we can unify them and use $\vec{F}_1$ to represent all of the ten functions) can be expressed as:

$$\vec{F}_1 = 0 \cdot \vec{i} + 0 \cdot \vec{j}. \tag{6}$$

All the agents are respectively activated at random time t_1^1, t_1^2, $\dots$, t_1^{10} to bring them together, the control function of current state can be written as:

$$\vec{F}_2 = \frac{[\sum c_i(x_i - x)\frac{1}{d_i}]\cdot\vec{i} + [\sum c_i(y_i - y)\frac{1}{d_i}]\cdot\vec{j}}{\left|[\sum c_i(x_i - x)\frac{1}{d_i}]\cdot\vec{i} + [\sum c_i(y_i - y)\frac{1}{d_i}]\cdot\vec{j}\right|} \tag{7}$$

$$c_i = \begin{cases} -5, & d_i < 20 \\ 0, & 20 \le d_i \le 40 \\ 1, & d_i > 40. \end{cases} \tag{8}$$

The values of c_i mean that when $d_i < 20$ the dominating force between two agents will be repulsive force. When d_i is greater than or equal to 40, agents will attract each other. Repulsive force is set stronger than attractive force to avoid collision. After gathering completes, all agents are made to spread orderly to form a new and looser formation, suppose such behavior begins at time t_2 and ends in t_3, then the control function can be written as

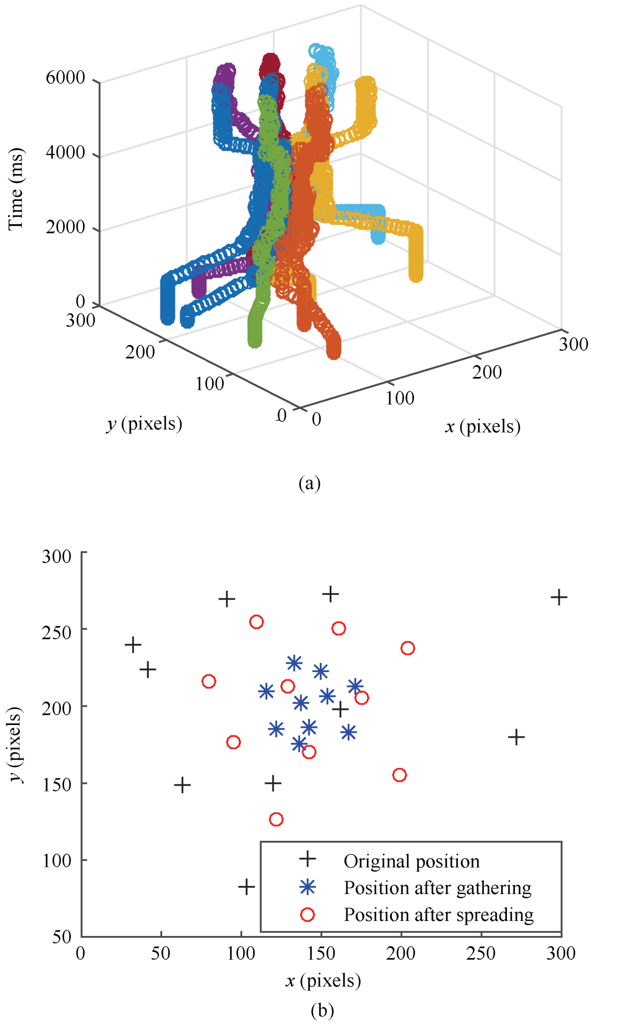
$$\vec{F}_3 = \frac{[\sum c_i(x_i - x)\frac{1}{d_i}]\cdot\vec{i} + [\sum c_i(y_i - y)\frac{1}{d_i}]\cdot\vec{j}}{\left|[\sum c_i(x_i - x)\frac{1}{d_i}]\cdot\vec{i} + [\sum c_i(y_i - y)\frac{1}{d_i}]\cdot\vec{j}\right|} \tag{9}$$

$$c_i = \begin{cases} -5, & d_i < 50 \\ 0, & 50 \le d_i \le 80 \\ 1, & d_i > 80. \end{cases} \tag{10}$$

Considering all of the three stages, the control function of each controllable agent during the time period of $[0, t_3]$ can be expressed as

$$\begin{aligned} f^1(t) &= \vec{F}_1\cdot\left[\varepsilon(t) - \varepsilon(t - t_1^1)\right] + \vec{F}_2\cdot\left[\varepsilon(t - t_1^1) - \varepsilon(t - t_2)\right] \\ &\quad + \vec{F}_3\cdot[\varepsilon(t - t_2) - \varepsilon(t - t_3)] \\ f^2(t) &= \vec{F}_1\cdot\left[\varepsilon(t) - \varepsilon(t - t_1^2)\right] + \vec{F}_2\cdot\left[\varepsilon(t - t_1^2) - \varepsilon(t - t_2)\right] \\ &\quad + \vec{F}_3\cdot[\varepsilon(t - t_2) - \varepsilon(t - t_3)] \\ &\vdots \\ f^{10}(t) &= \vec{F}_1\cdot\left[\varepsilon(t) - \varepsilon(t - t_1^{10})\right] + \vec{F}_2\cdot\left[\varepsilon(t - t_1^{10}) - \varepsilon(t - t_2)\right] \\ &\quad + \vec{F}_3\cdot[\varepsilon(t - t_2) - \varepsilon(t - t_3)]. \end{aligned} \tag{11}$$

Carry out simulation based on the model described above, the results are shown in Fig. 2. Fig. 2 (a) shows the changing of all agents' position through simulation time, the vertical axis represents time. Fig. 2 (b) shows the position of all the agents under the two steady states that are achieved by switching control function, and "+" in Fig. 2 (b) shows the initial position of agents. It can be seen in Fig. 2 (a) that all the agents remain static at beginning and their traces are vertically upward which means the position does not change with time. After activated sequentially they start to aggregate and after a period of time the gathering completes, a tight formation is formed as shown in Fig. 2 (b) with "*". Subsequently, due to control function change, the agents disperse, and a looser formation appears, as the "o" in Fig. 2 (b) has shown. In the whole process the control function changed with time for three times, each time the agent took a new motion: remained stationary at first, formed a tight formation later and re-formed into a new formation at last. Each agent changes the state at different time which can be regarded as a simulation of the communication delay.



(a)

(b)

Fig. 2. Simulation results. (a) Agents positions changing with time. (b) Two steady states achieved by switching control function.

B. *Simulation of Target Oriented Model*

The interaction forces model make the agents in the group gather in random positions which depend on the initial position of agents, and this position cannot be specified. In order to realize the directional movement of the group, the following model is considered:

$$\vec{F^j} = \frac{c_t\cdot(\vec{V}_t^j) + \sum_{i=1}^{K} c_i\cdot\vec{V}_i^j}{\left|c_t\cdot(\vec{V}_t^j) + \sum_{i=1}^{K} c_i\cdot\vec{V}_i^j\right|} \tag{12}$$

$$\vec{V}_t^j = \frac{(x_t - x)}{d_t + 1}\cdot\vec{i} + \frac{(y_t - y)}{d_t + 1}\cdot\vec{j} \tag{13}$$

$$\vec{V}_i^j = \frac{(x_i - x)}{d_i}\cdot\vec{i} + \frac{(y_i - y)}{d_i}\cdot\vec{j} \tag{14}$$

where $\vec{F^j}$ represents resultant force of agent j, $\vec{V}_t^j$ represents the vector from the agent to the target position, x_t, y_t are the coordinates of target position and d_t represents the distance between agent j and target position. Assume that there are K agents around agent j, and agent i is one of these K agents, then $\vec{V}_i^j$ represents the repulsive force between agent

j and agent i. x, y represent the coordinates of agent j, x_i, y_i represent the coordinates of agent i and d_i is the distance between agent i and j. c_t, c_i are proportional coefficients, c_t should be assigned a positive value and c_i is represented as follows:

$$c_i = \begin{cases} -c_r, & d_i < r \\ 0, & d_i \geq r \end{cases} \tag{15}$$

where r represents radius of repulsive force and c_r is a positive constant. This model is shown in Fig. 3.

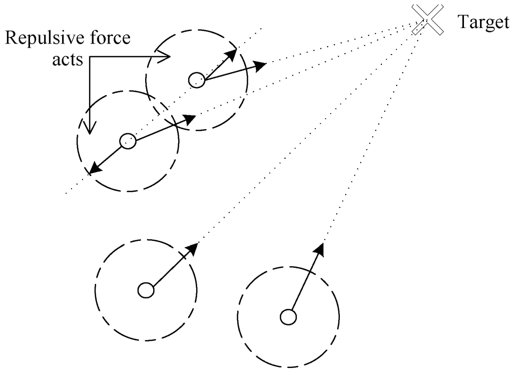


Fig. 3. Target oriented model.

Assume that there are 20 agents with random initial positions, before they are activated at random time $t_1^1, t_1^2, \ldots, t_1^{20}$, the control function on each agent should be

$$\vec{F_1} = 0 \cdot \vec{i} + 0 \cdot \vec{j}. \tag{16}$$

After the agents' activation, the target point can be set to the center of gravity of all agents to make them aggregate:

$$\begin{cases} x_t = \dfrac{\sum\limits_{i=1}^{20} x_i}{20} \\ y_t = \dfrac{\sum\limits_{i=1}^{20} y_i}{20}. \end{cases} \tag{17}$$

Set $c_t = 3$, $c_i = \begin{cases} -10, & d_i < 40 \\ 0, & d_i \geq 40 \end{cases}$, the control function of this period can be written as

$$\vec{F_2} = \frac{3\left(\frac{x_t - x}{d_t + 1} \cdot \vec{i} + \frac{y_t - y}{d_t + 1} \cdot \vec{j}\right) + \sum\limits_{i=1}^{K} c_i \left(\frac{x_i - x}{d_i} \cdot \vec{i} + \frac{y_i - y}{d_i} \cdot \vec{j}\right)}{\left|3\left(\frac{x_t - x}{d_t + 1} \cdot \vec{i} + \frac{y_t - y}{d_t + 1} \cdot \vec{j}\right) + \sum\limits_{i=1}^{K} c_i \left(\frac{x_i - x}{d_i} \cdot \vec{i} + \frac{y_i - y}{d_i} \cdot \vec{j}\right)\right|}. \tag{18}$$

Later, two agents (for example agent 1 and 11) are chosen to be fixed at t_2. Meanwhile, the rest of the agents are divided into two groups, agents $2, 3, \ldots, 10$ head to their target position $(50, 450)$, agents $12, 13, \ldots, 20$ head to $(450, 300)$. The control function on the fixed agents is

$$\vec{F_3^1} = \vec{F_3^{11}} = 0 \cdot \vec{i} + 0 \cdot \vec{j}. \tag{19}$$

$c_t = 3$, $c_i = \begin{cases} -10, & d_i < 40 \\ 0, & d_i \geq 40 \end{cases}$, the control function on the rest of the agents is expressed as

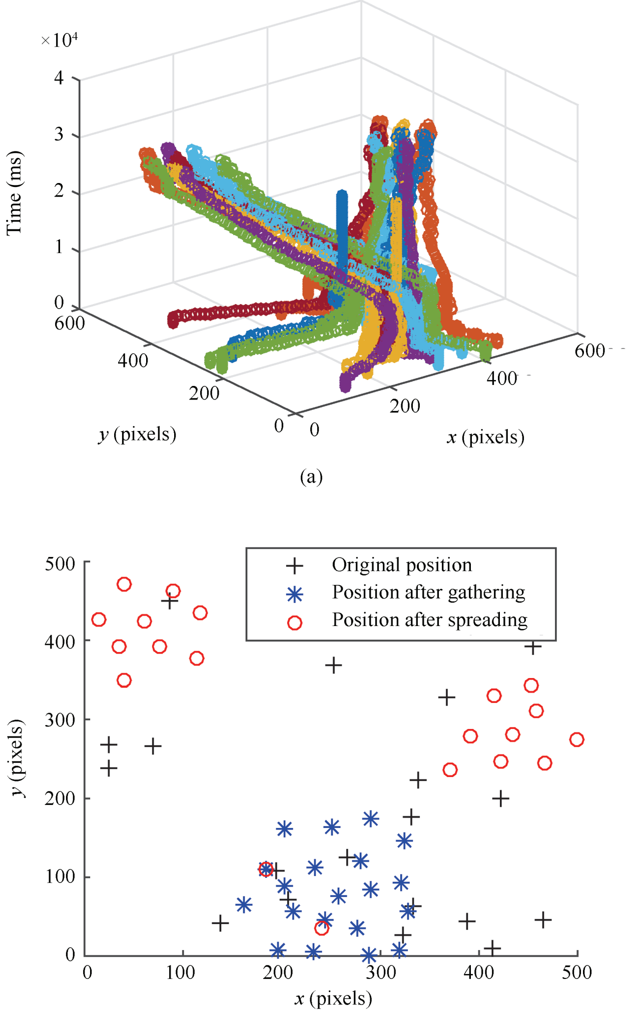
$$\begin{aligned} \vec{F_3^2} &= \vec{F_3^3} = \cdots = \vec{F_3^{10}} \\ &= \frac{3\left(\frac{50 - x}{d_t + 1} \cdot \vec{i} + \frac{450 - y}{d_t + 1} \cdot \vec{j}\right) + \sum\limits_{i=1}^{K} c_i \left(\frac{x_i - x}{d_i} \cdot \vec{i} + \frac{y_i - y}{d_i} \cdot \vec{j}\right)}{\left|3\left(\frac{50 - x}{d_t + 1} \cdot \vec{i} + \frac{450 - y}{d_t + 1} \cdot \vec{j}\right) + \sum\limits_{i=1}^{K} c_i \left(\frac{x_i - x}{d_i} \cdot \vec{i} + \frac{y_i - y}{d_i} \cdot \vec{j}\right)\right|} \end{aligned} \tag{20}$$

$$\begin{aligned} \vec{F_3^{12}} &= \vec{F_3^{13}} = \cdots = \vec{F_3^{20}} \\ &= \frac{3\left(\frac{450 - x}{d_t + 1} \cdot \vec{i} + \frac{300 - y}{d_t + 1} \cdot \vec{j}\right) + \sum\limits_{i=1}^{K} c_i \left(\frac{x_i - x}{d_i} \cdot \vec{i} + \frac{y_i - y}{d_i} \cdot \vec{j}\right)}{\left|3\left(\frac{450 - x}{d_t + 1} \cdot \vec{i} + \frac{300 - y}{d_t + 1} \cdot \vec{j}\right) + \sum\limits_{i=1}^{K} c_i \left(\frac{x_i - x}{d_i} \cdot \vec{i} + \frac{y_i - y}{d_i} \cdot \vec{j}\right)\right|}. \end{aligned} \tag{21}$$

When the simulation is over at t_3, the control function of each agent within $[0, t_3]$ can be expressed as

$$\begin{aligned} f^1(t) &= \vec{F_1} \cdot \left[\varepsilon(t) - \varepsilon(t - t_1^1)\right] + \vec{F_2} \cdot \left[\varepsilon(t - t_1^1) - \varepsilon(t - t_2)\right] \\ &\quad + \vec{F_3^1} \cdot \left[\varepsilon(t - t_2) - \varepsilon(t - t_3)\right] \\ f^2(t) &= \vec{F_1} \cdot \left[\varepsilon(t) - \varepsilon(t - t_1^2)\right] + \vec{F_2} \cdot \left[\varepsilon(t - t_1^2) - \varepsilon(t - t_2)\right] \\ &\quad + \vec{F_3^2} \cdot \left[\varepsilon(t - t_2) - \varepsilon(t - t_3)\right] \\ &\vdots \\ f^{20}(t) &= \vec{F_1} \cdot \left[\varepsilon(t) - \varepsilon(t - t_1^{20})\right] + \vec{F_2} \cdot \left[\varepsilon(t - t_1^{20}) - \varepsilon(t - t_2)\right] \\ &\quad + \vec{F_3^{20}} \cdot \left[\varepsilon(t - t_2) - \varepsilon(t - t_3)\right]. \end{aligned} \tag{22}$$

The simulation results are shown in Fig. 4. All of the agents were initially static. When they were activated, they moved to the center of gravity position of all agents. After their arrival, a relatively stable state was achieved because of the balance between attractive force and repulsive force. In Fig. 4 (b), "*" marked the position of these agents. Later, the control functions on the agents changed, two agents (agent 1 and agent 11) turned into stationary, and the rest of the agents are divided into two groups. They moved toward opposite side of the map. Because of the repulsive and attractive interaction between the agents, they could still maintain relatively stable formation and moved together without overcrowding or collision. Eventually two groups of agents arrived at the specified locations, which are shown as "o" in Fig. 4 (b). Fig. 4 (a) shows the changing of all agents position through simulation time. In the whole process, the control function of the population was changed for three times, each was stationary, directed aggregation and group division. Some agents in the third stage remain stationary to simulate the failure phenomenon of agents in a group. Due to the robustness of the cluster system, partial failure of the agents will not affect the group behavior. As shown in this simulation, although there are two agents which remain in place, the rest of the group can still finish the transfer and move unaffected.



(a)

(b)

Fig. 4. Simulation results. (a) Agents positions changing with time. (b) Two steady states achieved by switching control function.

IV. Conclusion

This paper discusses the limited ability of invariable controller dominated algorithms to control the swarm robot system, and hence a time-varying algorithm is proposed. Since the controller can be changed over time, the algorithm brings more flexible control effect. To test the actual effect of the algorithm, an interaction force model and a target oriented model are established, the controllers of the two models are put into the algorithm respectively and simulations are carried out. The simulation results show that the time-varying algorithm achieves not only aggregation, formation and directional movement as invariable controller dominated algorithms do, but also formation change, direction change, flock division and so on could be achieved by switching state during the simulation, which indicates that the time-varying algorithm is more flexible and practical, and makes the system be able to accomplish more complex assignment. The implementation of new hardware architecture, especially net based dynamic reconfigurable CPU that is suitable for this algorithm is our ongoing program. Unlike traditional architecture, the new one can be compiled on the net and is able to boot from net directly, therefore the system function can be modified in real time. Together with the time-varying algorithm, the performance of swarm intelligence can be improved to a great extent.

References

[1] C. W. Reynolds, "Flocks, herds and schools: a distributed behavioral model," *ACM SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 25–34, Jul. 1987.

[2] G. Beni and J. Wang, "Swarm intelligence in cellular robotic systems," in *Robots and Biological Systems: Towards a New Bionics*, P. Dario, G. Sandini, and P. Aebischer, Eds. Berlin Heidelberg, Germany: Springer, 1993, pp. 703–712.

[3] L. Spector and J. Klein, "Evolutionary dynamics discovered via visualization in the BREVE simulation environment," *Proc. 8th Int. Conf. Simulation and Synthesis of Living Systems, Artificial Life VIII*, Sydney, Australia, 2002, pp. 163–170.

[4] L. Spector, J. Klein, C. Perry, and M. Feinstein, "Emergence of collective behavior in evolving populations of flying agents," *Genet. Progr. Evol. Mach.*, vol. 6, no. 1, pp. 111–125, Mar. 2005.

[5] V. Gazi and K. M. Passino, "Stability analysis of swarms," *IEEE Trans. Autom. Control*, vol. 48, no. 4, pp. 692–697, Apr. 2003.

[6] V. Gazi and K. M. Passino, "A class of attractions/repulsion functions for stable swarm aggregations," *Int. J. Control*, vol. 77, no. 18, pp. 1567–1579, Jan. 2004.

[7] V. Gazi, "Swarm aggregations using artificial potentials and sliding-mode control," *IEEE Trans. Robot.*, vol. 21, no. 6, pp. 1208–1214, Dec. 2005.

[8] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proc. 1985 IEEE Int. Conf. Robotics and Automation*, St. Louis, MO, USA, 1985, pp. 396–404.

[9] C. Virágh, G. Vásárhelyi, N. Tarcai, T. Szörényi, G. Somorjai, T. Nepusz, and T. Vicsek, "Flocking algorithm for autonomous flying robots," *Bioinsp. Biomimet.*, vol. 9, no. 2, Article ID: 025012, May 2014.

[10] S. G. Reebs, "Can a minority of informed leaders determine the foraging movements of a fish shoal," *Anim. Behav.*, vol. 59, no. 2, pp. 403–409, Feb. 2000.

[11] I. D. Couzin, J. Krause, N. R. Franks, and S. A. Levin, "Effective leadership and decision-making in animal groups on the move," *Nature*, vol. 433, no. 7025, pp. 513–516, Feb. 2005.

[12] L. Yang, K. M. Passino, and M. Polycarpou, "Stability analysis of one-dimensional asynchronous swarms," *IEEE Trans. Autom. Control*, vol. 48, no. 10, pp. 1848–1854, Oct. 2003.

[13] N. Cai, J. X. Xi, and Y. S. Zhong, "Brief paper swarm stability of high-order linear time-invariant swarm systems," *IET Control Theory Appl.*, vol. 5, no. 2, pp. 402–408, Jan. 2011.

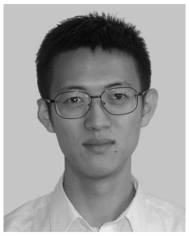
[14] I. I. Viksnin, A. L. Drannik, R. A. Iureva, and I. I. Komarov, "Flocking factors' assessment in case of destructive impact on swarm robotic systems," in *Proc. 2016 18th Conf. Open Innovations Association and Seminar on Information Security and Protection of Information Technology (FRUCT-ISPIT)*, St. Petersburg, Russia, 2016, pp. 357–363.

[15] M. Rubenstein, C. Ahler, and R. Nagpal, "Kilobot: a low cost scalable robot system for collective behaviors," in *Proc. 2012 IEEE Int. Conf. Robotics and Automation (ICRA)*, Saint Paul, MN, USA, 2012, pp. 3293–3298.

[16] J. Faigl, T. Krajník, J. Chudoba, L. Přeučil, and M. Saska, "Low-cost embedded system for relative localization in robotic swarms," in *Proc. 2013 IEEE Int. Conf. Robotics and Automation (ICRA)*, Karlsruhe, Germany, 2013, pp. 993−998.

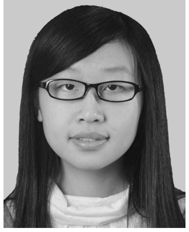
[17] B. Fidan, V. Gazi, S. H. Zhai, N. Cen, and E. Karataş, "Single-view distance-estimation-based formation control of robotic swarms," *IEEE Trans. Ind. Electron.*, vol. 60, no. 12, pp. 5781−5791, Dec. 2013.

[18] I. D. Couzin, J. Krause, R. James, G. D. Ruxton, and N. R. Franks, "Collective memory and spatial sorting in animal groups," *J. Theoret. Biol.*, vol. 218, no. 1, pp. 1−11, Sep. 2002.



Ligang Hou graduated from Shandong University (SDU), China, in 2001. He received the B.S. degree from SDU in 2001. He received the Ph.D. degree from Beijing University of Technology (BJUT), China, in 2007. He is currently an Assistant Professor at the Laboratory of VLSI and Systems, College of Microelectronics, BJUT. His research interests include VLSI design and swarm control.

Fangwen Fan is a master student at Beijing University of Technology. He received the B.S. degree from BJUT in 2015. He is currently a student at the Laboratory of VLSI and Systems, College of Microelectronics, BJUT. His research interests include swarm control and machine learning.



Jingyan Fu is a master student at Beijing University of Technology. She received the B.S. degree from BJUT in 2014. She is currently a student at the Laboratory of VLSI and Systems, College of Microelectronics, BJUT. Her research interests include 3D IC CAD and swarm control.

Jinhui Wang received the B.S. degree from Hebei University, China, in 2004. He received the Ph.D. degree from Beijing University of Technology, China, in 2010. He is currently an Assistant Professor in the Department of Electrical and Computer Engineering, North Dakota State University, USA. His research interests include VLSI design, big data, and data mining.