A Survey on Decentralized Flocking Schemes for a Set of Autonomous Mobile Robots[1]

(Invited Paper)

Naixue Xiong[1, 2], Jing He[2], Yan Yang[3], Yanxiang He[4], Tai-hoon Kim[5], Chuan Lin[6]
[1]School of Computer Science & Technology, Wuhan University of Science and Engineering, China
Email: nxiong@cs.gsu.edu
[2]Department of Computer Science, Georgia State University, Atlanta, GA, USA
Email: jhe9@cs.gsu.edu
[3]Department of Computer Science, Western Illinois University, IL, USA
Email: Y-Yang2@wiu.edu
[4]School of Computer, Wuhan University, 430079, Wuhan, China
Email: parkjonghyuk1@hotmail.com
[5]Division of Multimedia Engineering, Hannam University, Korea
Email: taihoonn@empal.com.
[6]School of Mathematics and Statistics, Wuhan University, China
Email: chlin@whu.edu.cn

***Abstract*—Recently, control and coordination of a set of autonomous mobile robots has been paid a lot of attentions, because the cooperation of simple robots offers several advantages, such as redundancy and flexibility, and allows performing hard tasks that could be impossible for one single robot. There are a lot of interesting applications of multiple robots, such as satellite exploration and surveillance missions.**

The characteristic of simplicity of mobile robots brings potential wide applications; however this characteristic also lead to crash with higher probability during cooperation, especially in harsh environment. Surprisingly, only few researches consider the fault tolerance of mobile robots, especially for dynamic coordination application---robot flocking.

In this paper, we summarize the existed flocking algorithms and discuss their characteristics. Then we briefly described our fault tolerant flocking algorithms in different models. Finally we proposed the potential future research directions for dynamic flocking of a group of mobile robots. In all, this work can provide a good reference for the researchers working on dynamic cooperation of robots in distributed system.

***Index Terms*—Mobile robots, fault tolerance, flocking, formation generation, collision avoidance**

I. INTRODUCTION

Mobile robots cooperating in groups offer several advantages, e.g., redundancy and flexibility, and can sometimes perform tasks that would be impossible for one single robot. Recent advances in robotics have started making it feasible to deploy large numbers of inexpensive robots for tasks. For instance, among many other applications, it is becoming increasingly attractive to consider a group of mobile robots as a way to provide support for rescue and relief during or after a natural catastrophe (e.g., earthquake, tsunami, cyclone, volcano eruption). As a result, research on mechanisms for coordination and self-organization of mobile robot systems is beginning to attract considerable attention (e.g, [1, 6, 7, 15, 31-34]) (see Fig. 1).

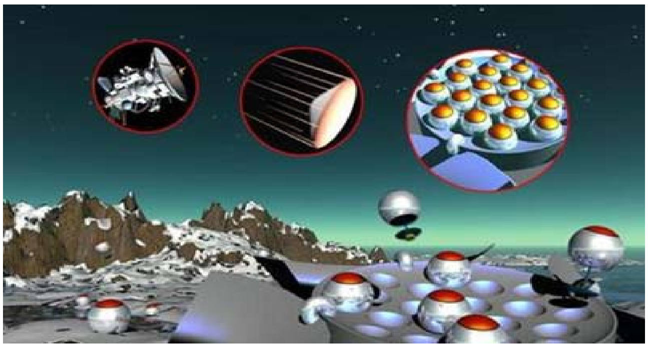


Fig. 1. An application of robots cooperation to explore the unknown planet.

For such operations, relying on a group of simple robots for delicate operations has various advantages over considering a single complex robot. For instance, first it is usually more cost-effective to manufacture and deploy a number of cheap robots rather than a single expensive one; then, higher number yields better potential for a

---

[1]Manuscript received March 15, 2009; revised April 15, 2009; accepted May 15, 2009

doi:10.4304/jcm.5.1.31-38

system resilient to individual robot failures; third, smaller robots have obviously better mobility in tight and confined spaces, and the last but the least, a group can survey a larger area than an individual robot, even if the latter is equipped with better sensors.

Flocking is a form of collective behavior of large number of interacting agents with a common group objective [36]. For many decades, scientists from rather diverse disciplines including animal behavior, physics and biophysics, social sciences, and computer science have been fascinated by the emergence of flocking/swarming/schooling in groups of agents via local communication. Examples of these agents include birds, fish, penguins, ants, bees, and crowds. In detail, robot flocking is the ability of a group of robots to move in formation and to preserve it while moving, like a flock of birds or insects (see Fig. 2). Concretely, simple robots are allowed to move, with basic rules governing their movement. The moving in formation has many applications, for instance, transporting large objects, exploring hazardous areas and surveillance.

Fig. 2. A group of birds moving together, staying close to each other and keeping some desired formation while moving.

Since many applications of cooperative robotics consider cheap robots dwelling in hazardous environments, fault tolerance is of primary concern. While, to the best of our knowledge, only a few work discussed fault tolerance of flocking of mobile robots in recent years, e.g., [38, 39].

Specially in flocking applications, all robots coordinate to generate a certain formation. If some robots are crashed, the other robots can not distinguish alive robots in waiting state from the crashed ones (The crashed robot will stop working and never recover.). Thus, the desired formation with the current robots can not maintain while moving. Therefore, it is necessary to find a method that can distinguish the correct robots in waiting state from the crashed ones. Yoshida et al. [13] proposed a fault-tolerant algorithm to select the active mobile robots from a group of mobile robots. Unfortunately, the authors only considered initial crash faults of robots, i.e., a faulty robot, which makes no motion from the beginning of execution of the algorithm.

II. FLOCKING PROBLEM

Informally, flocking is the formation and maintenance of a desired pattern 1 while moving, by a team of mobile robots. In most of research, a leader-followers approach [12] is considered. That is, at any time, there exists a robot leader in the system to lead the other robots, called followers. This leader is elected and known by the other robots in the system. In other words, the followers just need to follow the leader wherever it goes, and to keep the given formation while moving. More formal definition of flocking is proposed in [5].

Considering from the computational viewpoint, there are a number of fundamental questions that have inspired this work:

1. How do all robots work to satisfy the above three rules in ***distributed*** way?
2. How do robots in flocks perform obstacle avoidance?
3. How do robots form into a desired formation and maintain it during flocking?
4. What is the minimum assumption that needs to make robots flock?
5. What happens if some robots are crashed?

For question 1-4, there are some recent studies. However, for question 5, most of work did not consider this case. Following these questions, the other question may arise, such as how a robot knows other robots are correct or crashed, and how the remaining correct robots effectively cooperate each other. Take these questions in mind, in the following we analyze the existed robots flocking algorithm. Then we try to find the solution and potential direction of the topic.

III. EXISTED ROBOTS FLOCKING ALGORITHMS AND DISCUSSION

The flocking problem has been long studied in various perspectives. Studies can be found in different disciplines, from artificial intelligence to engineering [2, 5, 9, 12].

In 1986, Reynolds introduced three heuristic rules to lead to creation of the first computer animation of flocking [30]. Here are three quotes from [30] that describe these rules:

- *Flock Centering*: attempt to stay close to nearby flockmates
- *Obstacle Avoidance*: avoid collisions with nearby flockmates
- *Velocity Matching*: attempt to match velocity with nearby flockmates

Let us mention that these rules are also known as cohesion, separation, and alignment rules in the literature.

In natural systems, flocking algorithms were designed based on behavioral patterns that can be found in bees, flies, frogs, birds, fish, ants, for instance [30]. His work demonstrated that flocking is an example of emergent behavior arising from simple rules.

Since his contribution, many flocking strategies have reported in the field of swarm robotics. These strategies can be classified into global and local perspective strategies. As a global perspective strategy, Carpin and Parker [21] introduced a cooperative leader-follower approach that could handle a heterogeneous team with different types of sensors using communications. As an extension of Carpin and Parker [21], Parker et al. [29] introduced a tightly-coupled navigation assistance approach by a leader with rich sensing capability as the central figure of a robot team. Such strategies make the leader more costly and the team becomes less robust to the failure of the leader. More recently efforts have been made to reproduce this type of behavior in artificial systems, where solutions are designed using heuristics [35].

Hayes and Dormiani-Tabatabaei [8] studied the flocking problem using the principles of swarm intelligence. In particular, they provided a leader-less flocking algorithm that is more conductive to implementation in embodied agents. The key point of their algorithm is that it uses the time derivative of the perceived center of the flock to align the robots without explicit knowledge of robot heading.

Another approach for the flocking problem is the control theory approach [11]. With this approach, each robot has to follow certain control laws to converge to a stable state. For instance, Saber and Murray [9] provided a dynamic graph theoretic framework that enables the modeling flocking agents in the presence of obstacles. Zhang et al. [11] also considered fault tolerant formation control of Unmanned Air Vehicles (UAV). In particular, they provided a fault tolerant control scheme to deal with global position system (GPS) sensor failure and wireless communication packet losses. Moreover, they studied the effectiveness of their solution by conducting simulations in a wireless network environment for real-time formation flights. In [36], the author presented a theoretical framework for design and analysis of distributed flocking algorithms. Two cases of flocking in free-space and presence of multiple obstacles are considered.

Recently, there are a few works which address flocking issue from a computational viewpoint. The first such study was by Eervai and Prencipe [5]. In their work, the authors first provided a formal definition for the flocking problem based on a leader-followers approach. The authors proposed a flocking algorithm that applies to formations that are symmetric with respect to the movement of the leader without an agreement on a common coordinate system (except the unit distance). Coble and Cook [4] considered the case when robots can communicate between each other by exchanging messages. In particular, they applied a symbolic machine learning approach to deal with uncertainties in communication among autonomous robots. In other words, they provided a set of attributes that a robot needs to consider when determining whether its neighbors have been destroyed or it is temporarily obstructed from communication. In the leader-follower method, a robot is selected as the moving reference point. Gervasi and Prencipe [5] proposed a computational solution based on CORDA [16] with weak assumptions such as asynchrony, anonymity, no memory and a simple behavior cycle. In their study, all followers generate a geometric pattern symmetrically with respect to the pre-selected leader. Balch and Arkin [28] studied a paradigm of reactive behaviors for four formation patterns, where the robots were assigned their role such as the leader or the follower with a unique ID.

On the other hand, local perspective strategies can be divided into the leaderless method, where a specific robot is not assigned to conduct its robot swarm, and the leader-follower method, according to implementation method. The leaderless methods are based on interactions between individual robots mostly inspired by evidence from biological systems or natural phenomena. These ideas were mostly borrowed from the physical phenomena or organism of animals and insects in nature [22-23], or behavior-based approaches [24]. Folino and Spezzano [25] proposed a parallel spatial clustering algorithm combining a smart exploratory method with a density-based cluster algorithm to discover clusters of arbitrary shape and size in spatial data. Shimizu et al. [26] introduced emergent behaviors for two dimensional modular robots reconfigurating their geometric shape based on coupling between a connectivity control algorithm and nonlinear oscillators. Another leaderless approach based on local interactions is presented in [10, 27].

Gervasi and Prencipe [5] have proposed a flocking algorithm for robots based on a leader-followers model, but introduce additional assumptions on the speed of the robots. In particular, they proposed a flocking algorithm for formations that are symmetric with respect to the leader's movement, without agreement on a common coordinate system (except for the unit distance). However, their algorithm requires that the leader is distinguished from the robot followers.

Canepa and Potop-Butucaru [12] proposed a flocking algorithm in an asynchronous system with oblivious robots. First, the robots elect a leader using a probabilistic algorithm. After that, the robots position themselves according to a specific formation. Finally, the formation moves ahead. Their algorithm only lets the formation move straight forward. Although the leader is determined dynamically, once elected it can no longer change. In the absence of faulty robots, this is a reasonable limitation in their model.

IV. FAULT TOLERANT FLOCKING

In this section, we introduced the famous system models and summarized the existed fault tolerant flocking algorithms. Currently, there are two famous system model, called asynchronous model (CORDA model) and semi-synchronous model (Suzuki-Yamashita model). Based on the different system model, two kinds of fault tolerant algorithms were proposed.

A. System Model

(1) CORDA model

First, we introduce the CORDA model proposed by Prencipe [16]. The system consists of a set of autonomous mobile robots $R = \{r_1, \cdots, r_n\}$. A robot is modeled as a unit having computational capabilities, and which can move freely in the two-dimensional plane. Robots are seen as points on the plane.

In addition, they are equipped with sensor capabilities to observe the positions of the other robots, and form a local view of the world.

The local view of each robot includes a unit of length, an origin, and the directions and orientations of the two x and y coordinate axes. In particular, we assume that robots have a partial agreement on the local coordinate system. Specifically, they agree on the orientation and direction of one axis, say y. Also, they agree on the clockwise/counterclockwise direction. The robots are completely *autonomous*. Moreover, they are *anonymous*, in the sense that they are a priori indistinguishable by appearance. Furthermore, there is no direct means of communication among them.

In the CORDA model, robots are totally asynchronous. The cycle of a robot consists of a sequence of events: Wait-Look-Compute-Move.

- **Wait**. A robot is idle. A robot cannot stay permanently idle. At the beginning all robots are in Wait state.
- **Look**. Here, a robot observes the world by activating its sensors, which will return a snapshot of the positions of the robots in the system.
- **Compute** In this event, a robot *performs* a local computation according to its deterministic algorithm. The algorithm is the same for all robots, and the result of the compute state is a destination point.
- **Move**. The robot *moves* toward its computed destination. The robot moves toward its computed destination, but the distance it moves is unmeasured; neither infinite, nor infinitesimally small. Hence, the robot can only go towards its goal, but the move can end anywhere before the destination.

In the model, there are two limiting assumptions related to the cycle of a robot.

Assumption 1: *It is assumed that the distance travelled by a robot r in a move is not infinite. Furthermore, it is not infinitesimally small: there exists a constant* $\Delta_r > 0$, *such that, if the target point is closer than* Δ_r, *r will reach it; otherwise, r will move toward it by at least* Δ_r.

Assumption 2: *The amount of time required by a robot r to complete a cycle (wait-look-compute-move) is not infinite. Furthermore, it is not infinitesimally small; there exists a constant* $\tau_r > 0$, *such that the cycle will require at least* τ_r *time.*

(2) Suzuki-Yamashita model

In this model, each robot is able to sense its surroundings, perform computations on the sensed data, and move toward the computed destination. The behavior of a robot constitutes its cycle of looking, computing, moving and being active wait. We define an activation of a robot as follows:

Activation. The sequence "*Look- Compute- Move-Wait*" is called the cycle of a robot. If one such cycle of a robot is executed, we call this one activation of a robot. In SYm model, time is represented as an infinite sequence of discrete time instants t_0, t_1, t_2,…, during which each robot can be either active or inactive. In particular, the robots execute their activities of observation, computation and movement in instantaneous fashion, and thus, a robot observes other robots only when a cycle begins (i.e., when they are stationary).

In SYm model, the robots are autonomous, in the sense that they cannot be distinguished by their appearance, and they do not have any kind of identifiers that can be used during the computation. Also, they cannot communicate with each other explicitly, the only way to communicate is by vision. The local view of each robot includes a unit of length, an origin and the directions and orientations of the two x and y coordinate axes. In particular, we assume that robots have a partial agreement on the local coordinate system. Specifically, they agree on the orientation and direction of one axis, say the y axis. Also, they agree on the orientation clockwise/counterclockwise. The robots are not oblivious, that means, they has memory to remember their past information. Also, in our model, a robot can see all the other robots in the environment since the local view of robots make robot network disconnected easily.

B. Fault Tolerant Flocking Algorithms

As we know, there exist two fault tolerant flocking algorithms [39, 41]. In [39], the authors proposed a flocking algorithm based on CORDA model, and in [41] a semi-synchronous flocking algorithm is proposed. In these two algorithms, they only addressed ***crash failures***. That is, a robot may fail by crashing, after which it executes no actions (no movement). A crash is permanent in the sense that a faulty robot never recovers. However, it is still physically present in the system, and it is seen by the other non-faulty robots. A robot that is not faulty is called a ***correct*** robot.

(1) Flocking algorithm in CORDA model

The variant of the problem in [38] requires that the robots form and move while maintaining an approximation of a regular polygon, in spite of the possible presence of faulty robots---robots may fail by crashing and a crash is permanent. Although they did consider the presence of a leader robot to lead the group, the role of leader is assigned *dynamically* and any of the robots can potentially become a leader. In particular, after the crash of a leader, a new leader must eventually take over that role.

To the best of our knowledge, this work is the first to consider flocking of asynchronous (k-bounded) robots in the presence of faulty robots. Here, “k-bounded” means a scheduler with which between two consecutive full activation cycles of the same robot, another robot can execute at most k full activation cycle. With k-bounded scheduler, the paper [38] proposed a fault-tolerant flocking algorithm that allows a group of asynchronous robots to self organize dynamically to form an approximation of a regular polygon, and maintain this formation while moving. The algorithm relies on the assumption that robots' activations follow a k-bounded asynchronous scheduler, and that robots are non-oblivious (i.e., have a limited memory of the past).

This flocking algorithm allows correct robots to move in any direction, while maintaining an approximation of the polygon. Unlike previous works (e.g., [5], [12]), the algorithm is fault-tolerant, and tolerates permanent crash faults of robots.

The shortcoming of this algorithm is: the effectiveness of the algorithm relies on the precision of sensor and movement strongly. If there is any sensor or movement error, the algorithm will not work. Furthermore, if memory corruption has occurred in some robots, the whole robot cooperation will become failure. Therefore, it still has long way to go in the real robot application.

Another point that we mentioned is that in [38], the authors designed a perfect failure detector, which with very strong assumption that no robots can visit its and other robots past positions during “k(k+1)” activations. Furthermore, as we know from [40], there are eight kinds of failure detectors, among which the perfect failure detector is a reliable but very difficult to achieve.

(2) Flocking algorithm in Suzuki-yamashita model [41]

Initially, all robots are located on a regular polygon. The goal (requirements) of our algorithm is to maintain an approximation of a regular polygon, and to reform a new regular polygon with the correct robots in the absence of crash of robots during flocking. More interesting point of the algorithm is that it can make robots rotate freely.

The main idea of this algorithm is as follows:

- Assign a unique persistent rank for each robot by rank assignment module;
- Select the correct robots by failure detector module;
- Based on the rank of robots, select a unique leader from the set of correct robots;
- Based on positions of the leader and the other correct robots, a robot computes the target position and moves to satisfy the flocking requirements.

In [41], the flocking algorithm lifts the limitation of formation rotation in [39], yet in semi-synchronous model.

In all, these two flocking algorithms addressed dynamic agreement problem and at the same time they tolerated the crash failure of mobile robots. All robots use the same algorithms but they can effectively coordinate each other. That would be very useful in the practical applications, like moving heavy objects in the places that is dangerous for human beings, or rescue people after earthquake.

V. POTENTIAL RESEARCH DIRECTIONS

From the analysis of the above fault tolerant flocking algorithm, we think there are following directions or requirements that can help to solve fault tolerance in robot flocking.

(1)*To avoid collision*

There are two kinds of collisions that need to be considered, one is collision between robots, the other one is the collision between robots and obstacles if exist in the environment. Only some of work considers two kinds of collision together, such as [39].

(2) *To keep robot formation*

That is robots need to keep the neighboring graph (also sensor or communication graph) connected during the entire execution of the algorithm. In some applications, a group of robots need to generate a desired formation during flocking by organizing by themselves. Also, during moving, all robots need to maintain such desired formation to finish the specific tasks.

(3) *To find the weakest Failure detector*

There are a lot of failure detection schemes [17] [18] that are explored in traditional distributed systems. Based on different model and assumptions, the implementation of failure detection schemes is different but the goal of them is to detect the other robots status (alive or crash).

The failure type of robots:

- *Initially dead robot*: A robot is called initially dead if it does not execute a single step of its local algorithm.
- *Crash permanent:* a robot is said to crash if it executes its local algorithm correctly up to some moment, and does not execute any step thereafter.
- *Transient failure (crash recovery)*: in this case, a robot executes its local algorithm correctly, but there is a transient moment it can not work correctly, after that it recovers to correctly work.
- *Byzantine failure:* a robot is said to be Byzantine, if it executes arbitrary steps that are not in accordance with its local algorithm.

In most of work, they considered the crash permanent failure, in which once a robot crash, it will not move any more and stay its place for ever. So, the other interesting research topics would be to consider transient (crash-recovery) model or byzantine model. Also, there are some other failure models like: a robot can communicate with other robots but it can not move, or a robot’s communication ability is failed but still can move. Like that, different failure models will bring in different problems and result in different fault tolerant algorithms. It is also interesting to discuss the possibility of designing flocking algorithm under the above failure models. If possible, it is useful to design different failure detectors, such as eventually weak failure detector [17, 40], and to apply to robot flocking.

(4) *To investigate the "weakest" system model*

When a robot can communicate with the other robots by wireless communication or global position system, and if each robot has its identification, the new coordination algorithm will be designed based on such robot ability. However, the communication between robots may bring in the delay and communication is not reliable due to limited bandwidth, range and interferences, especially in harsh environments, it will be a challenge to design an efficient flocking algorithm.

(5) *Possibility of shape rotation in other models*

A shape rotation flocking presented [41, 42] consider a semi-synchronous model called Suzuki-yamashita model with crash failure of robots, in which robots can crash but never recover. We know Suzuki-yamashita model is a semi-synchronous model, in which robots execute their activations in atomic way. An interesting question is whether it is possible to find shape rotation solution in more complex models: CORDA model where robots are totally asynchronous, systems where robots can crash and recover, or systems with Byzantine robots. The CORDA model is weaker than SYm, that means, if one flocking algorithm can in CORDA model, it implies that it also can work in SYm model. That makes the algorithm more general and has wider applications.

(6) *Flocking with other failure detectors[42]*

To design a fault tolerant algorithm, one important (core) question is to find a failure detect to distinguish the crashed processes and the correct ones. The fault tolerant robot research is no exception. In the presented fault tolerant flocking algorithms, a perfect failure detector is used by managing the moving of robots strictly.

As we know, the perfect failure detector is very strongest among eight failure detectors. We could call perfect failure detector be reliable failure detector, since it always provide the correct results. One interesting questions is: what if using the unreliable failure detector, like S failure detector or <>S failure detector [40].

Also, in the existed work, the robot needs very strict movement restriction to make sure the perfect failure detector work well. To loose such movement restriction and at the same time to make sure the effective coordination of robots, an adaptive failure detector would be a good choice and could bring more effective coordination.

VI. CONCLUSION

In this paper, we first summarized the existed flocking approaches and analyzed their characteristics. Then we discussed the existed fault tolerant flocking algorithm and then explored the potential research directions about robots flocking for a group of mobile robots. Our work opened a lot of interesting research direction and be useful for the researchers to address this interesting flocking topic.

References

[1] G. O. Young, "Synthetic structure of industrial plastics (Book style with paper title and editor)," in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.

[2] M. J. Daigle, X. D. Koutsoukos, and Gautam Biswas, "Distributed diagnosis in formations of mobile robots," IEEE Transactions on Robotics, vol. 23, no. 2, pp. 353-369, April 2007.

[3] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita, "A distributed memoryless point convergence algorithm for mobile robots with limited visibility," IEEE Transactions on Robotics and Automation, vol.15, no.5, pp. 818-828, October 1999.

[4] J. Coble, D. Cook, "Fault tolerant coordination of robot teams," available at: citeseer.ist.psu.edu/coble98fault.html.

[5] V. Gervasi and G. Prencipe, ``Coordination without communication: the case of the flocking problem", Discrete Applied Mathematics, vol. 143, no. 1-3, pp. 203-223, September 2004.

[6] P. Renaud, E. Cervera, and P. Martinet, "Towards a reliable vision-based mobile robot formation control", in Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3176-3181, Sendai, Japan, September 28 - October 2, 2004.

[7] G. Tanner Herbert, Ali Jadbabaie and George J. Pappas, "Stable flocking of mobile agents Part II: dynamic topology", available at citeseer.ist.psu.edu/article/tanner03stable.html.

[8] A. T. Hayes, P. Dormiani-Tabatabaei , "Self-organized flocking with agent failure: Off-line optimization and demonstration with real robots", in Proceedings of IEEE International Conference on Robotics and Automation, vol.4, pp. 3900- 3905, Washington, D. C, USA, May 11-15, 2002.

[9] M. Lindhe and P. Ogren, K. H. Johansson, "Flocking with Obstacle Avoidance: Cooperations with Limited Communication in Mobile Networks," in Proceedings of the 42th IEEE Conference on Decision and Control, pp. 2022 - 2028, Maul, Hawaii, USA, December 2003.

[10] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithm and theory," IEEE Transactions on automatic control, vol. 51, no.3, March 2006.

[11] X. Zhang, R. Xu, C. Kwan, L. Haynes, Y. Yang, M. M. Polycarpou, "Fault tolerant formation flight control of UAVs", International Journal of Vehicle Autonomous Systems, vol. 2, no.3-4 pp. 217 -235, 2004.

[12] D. Canepa and M. G. Potop-Butucaru, "Stabilizing flocking via leader election in robot networks", in Proceeding of 9th International Symposium on

Stabilization, Safety, and Security of Distributed Systems, pp.52-66, Paris, France, November 14-16, 2007.

[13] D. Yoshida, T. Masuzawa and H. Fujiwara, “Fault-tolerant distributed algorithms for autonomous mobile robots with crash faults," Systems and Computers in Japan, Vol. 28, No. 2, 1997.

[14] X. Defago, M. Gradinariu, S. Messika, P. Raipin-Parvedy, “Fault-tolerant and self-organizing mobile robots gathering", in Proceedings of 20th International Symposium on Distributed Computing (DISC 2006), volume 4167 of LNCS, pp. 46¨C60, Stockholm, Sweden, September 2006.

[15] I. Suzuki and M. Yamashita, ``Distributed anonymous mobile robot: Formation of geometric patterns," SIAM Journal of Computing, vol. 28, no.4, pp. 1347-1363, 1999.

[16] G. Prencipe, “CORDA: Distributed Coordination of a Set of Autonomous Mobile Robots," in Proceding of European Research Seminar on Advances in Distributed Systems, pp. 185-190, Bertinoro, Italy, May 2001.

[17] N. Xiong, Y. Yang, J. Chen, Y. He, “On the Quality of Service of Failure Detectors Based on Control Theory", in Proceedings of The IEEE 20th International Conference on Advanced Information Networking and Applications (AINA (1)), pp. 75-80, Vienna, Austria, April 18 - April 20, 2006.

[18] N. Xiong, Y. Yang, X. Defago, “Comparative Analysis of QoS and Memory Usage of Adaptive Failure Detectors,” The 13th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC'07), Melbourne, Victoria, Australia, December 17-19, 2007.

[19] X. Defago and S. Souissi, “Self-stabilizing Algorithm for Circle Formation by Disoriented Oblivious Mobile Robots", Japan Advanced Institute of Science and Technology, Research Report, IS-RR-2007-003, Ishikawa, Japan, February 2007.

[20] P. Flocchini, G. Prencipe, N. Santoro, P. Widmayer, “Hard Tasks for Weak Robots: The Role of Common Knowledge in Pattern Formation by Autonomous Mobile Robots", in Proceedings of 10th International Symposium on Algorithms and Computation (formerly {SIGAL} International Symposium on Algorithms), 1999.

[21] S. Carpin, L. E. Parker, “Cooperative leader following in a distributed multi-robot system”, Proc. IEEE International Conference on Robotics and Automation, 2002.

[22] D. Zarzhitsky, D. F. Spears, W. M. Spears, “Distributed robotics approach to chemical plume tracing”, Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1-6, 2005.

[23] H. Hanada, G. Lee, N. Y. Chong, “Adaptive flocking of a swarm of robots based on local interactions”, Proc. IEEE Swarm Intelligence Symposium, pp. 340-347, 2007.

[24] J. M. Esposito, T. W. Dunbar, “Maintaining wireless connectivity constraints for swarms in the presence of obstacles”, Proc. IEEE Conference on Robotics and Automation, pp. 946-951, 2006.

[25] G. Folino, G. Spezzano, “An adaptive flocking algorithm for spatial clustering”, in Proc. 7th International Conference on Parallel Problem Solving from Nature: J. J. M. Guervos (eds): Lecture Notes in Computer Science, vol. 2439, pp. 924-933, 2002.

[26] M. Shimizu, T. Mori, A. Ishiguro, “A development of a modular robot that enables adaptive reconfiguration”, Proc IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006.

[27] G. Lee, Y. Hanada, N. Y. Chong, C. Kim, “Adaptive flocking algorithm for robot swarms: lessons from a school of fish”, Proc. 13th International Conference on Advanced Robotics, pp. 1232-1237, 2007.

[28] T. Balch and R. C. Arkin, “Behavior-based formation control for multi-robot teams,” IEEE Transactions on Robotics and Automation, vol. 14, no. 6, pp. 926-939, 1998.

[29] L. E. Parker, B. Kannan, F. Tang, M. Bailey, “Tightly-coupled navigation assistance in heterogeneous multi-robot teams”, Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004.

[30] C. W. Reynolds, “Flocks, Herds, and Schools: A Distributed Behavioral Model”, Journal of Computer Graphics, vol. 21(1), 79-98, 1987.

[31] K. Schreiner, “NASA's JPL Nanorover Outposts Project Develops Colony of Solar-powered Nanorovers”, In IEEE DS Online, 3(2), 2001.

[32] K. Konolige, C. Ortiz, R. Vincent, A. Agno, M. Eriksen, B. Limketkai, M. Lewis, L. Briesemeister, E. Ruspini, D. Fox, J. Ko, B. Stewart, and L. Guibas, “CENTIBOTS: Large-Scale Robot Teams”, in Journal of Multi-Robot Systems: From Swarms to Intelligent Autonoma, 2003.

[33] B. R. Bellur, M. G. Lewis, and F. L. Templin, “An Ad-hoc Network for Teams of Autonomous Vehicles”, in Proc. of IEEE Symposium on Autonomous Intelligence Networks and Systems, 2002.

[34] J.S. Jennings, G. Whelan, and W.F. Evans, “Cooperative Search and Rescue with a Team of Mobile Robots”, in Proc. 8th International Conference on Advanced Robotics, 193--200, 1997.

[35] D. C. Brogan and J. K. Hodgins, “Group Behaviors for Systems with Significant Dynamics,” In Autonomous Robots Journal, vol. 4, 137--153, 1997.

[36] Reza Olfati-Saber, “Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory,” IEEE Transactions on Autonomic Control, Vol, 51, No. 3, March 2006.

[37] H. Yamaguchi and G.Beni, ``Distributed Autonomous Formation Control of Mobile Robot Groups by Swarm-based Pattern Generation", Proc. of the 2nd Int. Symp. on Distributed Autonomous Robotic Systems (DARS 96), pp. 141-155, 1996.

[38] Samia Souissi, Yan Yang and Xavier D¡äefago, "Fault-tolerant flocking in a k-bounded asynchronous system", Technical report, IS-RR-2008-004, Japan Advanced Institute of Science and Technology, September 2008.

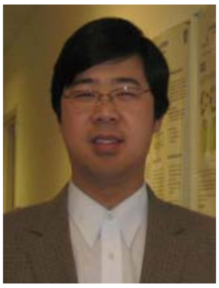
[39] Yan Yang, Naixue Xiong, Nak Young Chong, Xavier Defago, "A Decentralized and Adaptive Flocking Algorithm for Autonomous Mobile Robots," in the Proceeding of International Symposium on Advances in Grid and Pervasive Systems (GPC 2008), pp. 262-268, Kunming, China, May 25-28, 2008.

[40] Tushar Deepak Chandra and Sam Toueg, "Unreliable Failure Detectors for Reliable Distributed Systems," Journal of the ACM, 43:2, March 1996, 225-267.

[41] Yan Yang, Samia Souissi, Xavier Defago and Makoto Takizawa, "Fault-tolerant Flocking of Mobile Robots with whole Formation Rotation" , The IEEE 23rd International Conference on Advanced Information Networking and Applications (AINA-09), University of Bradford, Bradford, UK, May 26-29, 2009.

[42] Yan Yang, "Decentralized Fault-tolerant Flocking Algorithms for a Group of Autonomous Mobile Robots", Dissertation, Japan Adavanced Institute of Science and Technology, March 2009.

Biography



Naixue Xiong received his both PhD degrees in Wuhan University, and Japan Advanced Institute of Science and Technology, respectively. Both are on computer science. Now he is a research scientist in the Department of Computer Science, Georgia State University, Atlanta, USA. His research interests include Communication Protocols, Network Architecture and Design, and Optimization Theory. Until now, he published over 100 research articles. Some of his works were published or submitted in IEEE JSAC, IEEE or ACM transactions, and IEEE INFOCOM. He has received the Best Paper Awards in the 10th IEEE International Conference on High Performance Computing and Communications (HPCC-08) and the 28th North American Fuzzy Information Processing Society Annual Conference (NAFIPS2009). He is a member of IEEE, TCPP, and IET. Email: nxiong@cs.gsu.edu.

Jing He is a PhD student in Department of Computer Science, Georgia State University, Atlanta, GA, USA. She has published lots of papers. She is interested in distributed system, dependable computing, and communication network. Email: jhe9@cs.gsu.edu

Yan Yang got her Ph.D degree in March 2009 from Japan Advanced Institute of Science and Technology. Since 2008, she worked as a post doctor in Center for Asian and Pacific Studies in Seikei University, Tokyo, Japan. Now she works in Department of Computer Science, Western Illinois University, IL, USA. She is interested in distributed system, dependable computing and communication network.

Yanxiang He received his Ph.D. degrees in Computer School, Wuhan University, China. Now he is the Dean of Computer School Wuhan University, Wuhan, Hubei, China. Until now, he published over 150 research articles and over 15 books. His research interests include Distributed parallel processing, Data mining, Mobile computing, Software engineering, Information systems and E-Business. Email: yxhe@whu.edu.cn

Tai-Hoon Kim received his M.S. degrees and Ph.D. in Electrics and Electronics & Computer Engineering from the Sungkyunkwan University, Korea. Now, he is currently a professor of Hannam University. He wrote 16 books about the software development, OS such as Linux and Windows 2000, and computer hacking & security. And he published about 100 papers by 2006. He was a chair and program committee of international conferences and workshops. He was a guest editor of AJIT and FGCS Journal, and now he is an editor-in-chief of JSE and IJSIA Journal. He researched security engineering, the evaluation of information security products or systems with Common Criteria and the process improvement for security enhancement. In these days, he researches also some approaches and methods making IT systems more secure.

Lin Chuan received his Ph.D degree from school of computer science of Wuhan University, China. Now, he is an associate professor of Department of information and computation science, Wuhan University. His research interests include wireless sensor networks and DSP.