

Composing Swarm Robot Formations Based on Their Distributions Using Mobile Agents

Ryotaro Oikawa¹(✉), Munehiro Takimoto¹, and Yasushi Kambayashi²

¹ Department of Information Sciences, Tokyo University of Science,
2641 Yamazaki, Noda 278-8510, Japan
{oikawa,mune}@cs.is.noda.tus.ac.jp

² Department of Computer and Information Engineering,
Nippon Institute of Technology, 4-1 Gakuendai, Miyashiro-machi,
Minamisaitama-gun 345-8501, Japan
yasushi@nit.ac.jp

Abstract. This paper presents a control algorithm for composing formations of swarm robots. The composed formations represent specific symbols. The algorithm is based on our mobile software agent based control model where robots acquire any control program as necessary. The robots start working with just generic control programs and they acquire control programs for specific tasks from outside as mobile software agents. Since the mobile software agents can migrate from one robot to another autonomously, they can migrate to the most conveniently located robot for a specific task. This control model eliminates unnecessary movements of physical robots, contributing to suppressing the total costs of any multi-robot systems. Our control algorithm for composing formations takes the advantages of this mobile agent model. We have implemented our algorithm in a simulator and conducted experiments to demonstrate the feasibility of our approach.

Keywords: Mobile agent · Formation control · Multiple robots · Swarm intelligence

1 Introduction

In the last decade, robot systems have made rapid progress not only in their behaviors but also in the way they are controlled. In particular, a control system based on multiple software agents can control robots efficiently [1]. Multi-agent systems introduced modularity, reconfigurability and extensibility to control systems. It has made the development of control systems easy on distributed environments such as multi-robot systems.

On the other hand, the excessive interactions among agents in the multi-agent system may cause problems in the multiple robot environments. In order to mitigate the problems of excessive communication, mobile agent methodologies

have been developed for distributed environments. In a mobile agent system, each agent can actively migrate from one site to another site. Since a mobile agent can bring the necessary functionalities with it and perform its tasks autonomously, it can reduce the necessity for interaction with other sites. In the minimal case, a mobile agent requires that the connection is established only when it performs migration [2].

The model of our system is a set of cooperative multiple mobile agents executing tasks by controlling a pool of multiple robots [3]. The property of inter-robot movements of the mobile agents contributes to the flexible and efficient use of the robot resources. A mobile agent can migrate to the robot that is the most conveniently located to a given task, e.g. the nearest robot to a physical object such as a soccer ball. Since the agent migration is much easier than the robot motion, the agent migration contributes to saving power consumption [1]. We took the advantage of the fact that any agents on a robot can disappear as soon as they finish their tasks. If the agent has a policy of choosing idle robots rather than busy ones, in addition to the power-saving effect, it would result in more efficient use of robot resources.

We proposed our model in a paper and have shown the effectiveness of saving power consumption [1] and the efficiency of our system for searching targets [4,5], transporting them to a designated collection area [6], clustering robots [7], and serialization of them [8,9]. Choosing idle robots is a new feature we have introduced in the current system. In our system, all the robots have no specific tasks; they are all general purpose. In such a system, the same robot can perform various tasks such as mentioned above and when the robot completes a task it can proceed another task immediately or become idle state waiting for the next task. Our mobile agent system can work efficiently in such a situation because mobile agents can select the suitable robot for the current task without central control and do not burden other robots working for different tasks.

In the previous paper, we focus our attention on the formation control of robots [10]. Formation control is one of the most important tasks in the multi-robot applications, especially when individual robot has limited abilities and a given task requires collective actions. For example, robots may aggregate for coordinated search and rescue, collectively transporting large objects, exploring and mapping unknown area, or maintaining formations for area defense or flocking [11].

We have utilized the concept of the ant colony optimization (ACO) for the formation control. ACO is a swarm intelligence-based method and a multi-agent system that exploits artificial stigmergy for the solution of combinatorial optimization problems. In fact, we have implemented both the ants and the pheromone as mobile software agents. They are Ant agents (AAs) to drive robots and Pheromone agents (PAs) to attract AAs. Each AA creates PA at the locations for its neighbor robots to occupy, which diffuses to attract neighbor AAs through migrations. Once a PA reaches the AA to attract, the AA migrates to the robot nearest to the target location, and then drives the robot to the location, following the guidance of the PA. The control manner achieved distributed formation without any leader, and the migration properties of the agents contribute to suppressing the total cost of the formation.

In this paper, we propose an improved algorithm for multi-robot formations. In forging the new algorithm, we have focused on the efficiency. The new algorithm calculates the locations suitable for efficiently composing a formation so that the robots can suppress the duration time and energy consumption. In this algorithm, we introduce two kinds of mobile agents; the guide agent and node agents. The guide agent migrates among the robots in order to collect the locations of the robots and calculates the optimal locations of the robots for the formation based on the conceptual barycenter of them. The node agents drive the robots to the locations that the guide agent calculates.

The structure of the balance of this paper is as follows. In the second section, we describe the background. The third section describes the mobile agent model and the algorithm for our formation control. In the fourth section, we report the results of the numerical experiments using a simulator based on our algorithm. Finally, we conclude our discussions in the fifth section.

2 Background

Kambayashi and Takimoto have employed mobile agent system for our framework [3]. The framework helps users to construct intelligent robot control software by migration of mobile agents. They have implemented a team of cooperative search robots in order to show the effectiveness of their framework [5]. At the same time they have demonstrated that their framework contributes to energy saving of multiple robots [1, 4]. They have achieved significant energy saving for search robot applications.

Formation control strategies can be classified into three strategies [12]; behavior-based, virtual-structure, and leader-following. Behavior-based strategy defines simple behaviors or motion primitives for each robot. Balch and Arkin [13] proposed the motor scheme control. In virtual-structure strategy, virtual structure considers the entire formation as a rigid body. The motion of each robot is translated from the motion of the virtual structure, which is determined by the definition of the dynamics of virtual structure. Lewis and Tan [14] proposed one of pioneering works. In leader-followed strategy, some robots are selected as leaders, and the other robots follow the leaders. Das et al. [15] proposed one of the most popular control techniques using a feedback linearization control method. Cheng [11] proposed not only a formation control algorithm but also a formation generation algorithm using Contained Gas Model in which robots act like a particle in a container.

We proposed distributed formation control algorithm using mobile agents that is inspired by ant colony optimization [10]. In the approach, the guide agent plays the role as the leader. Unlike leader-followed strategy, however, the guide agent does not drive robots to compose formations. Instead, the guide agent has the virtual-structure of the objective formation and calculates the optimal locations of the formations. After calculating the locations to which the swarm robots are supposed to move, the guide agent generates node agents and makes them drive the robots to the locations. Our approach assumes that each robot knows the shape of the formation.

3 The Mobile Agents

Our system model consists of robots and two kinds of mobile software agents. We assume that the robots have simple capabilities of locomotion such as driving wheels, measuring distance and angle through an optical camera and a sensor as well as some communication devices with which they can communicate each other through a communication network such as a wireless LAN.

In our algorithm, all the controls for the mobile robots are achieved through the mobile agents. They are a guide agent (GA) and node agents (NAs). There is one GA in the system, which calculates the location of the conceptual barycenter of the formation and all the locations for the robots to occupy, based on the conceptual barycenter. NAs physically drive the robots to the locations to compose the formation. Here, consider that several formations are continuously performed. In that case, the distribution of robots in the field would be different every time. In order to efficiently adapt the formation to each distribution, while suppressing the duration time for composing a formation and the total length of traces of robot movements, the GA determines the target locations of the robots based on the center coordinate, i.e. the conceptual barycenter, of each distribution.

To determine the center coordinate of the formation, we use the concept of the barycenter of all robots in the field. The GA migrates among robots in order to collect locations of the robots and calculate the barycenter of them as if they are connected into one object. Upon the completion of GA's calculation, NAs drive the robots to the locations that the GA determines based on the conceptual barycenter. We assume that the objective shape is represented as a set of point coordinates. We also assume that the mobile agents on the robots do not know the absolute coordinates of robots but they can measure the relative coordinate of neighbor robots using sensors or cameras. In our algorithm, all mobile agents uses relative coordinate from the base robot that is selected by the GA. Each relative coordinate is represented as a vector value. When an mobile agent migrates from the base robot R_1 to the robot R_n along the path (R_1, \dots, R_n) , the relative coordinate \mathbf{p}_n of robot R_n are calculated as follows:

$$\mathbf{p}_n = \sum_{i=1}^{n-1} \mathbf{v}_i \quad (1)$$

\mathbf{v}_i is the vector value from robot R_i to robot R_{i+1} . The \mathbf{v}_i can be measured by sensors or cameras.

3.1 Guide Agent

In this section, we describe the algorithm for the GA. The GA can migrate to robots through communication networks. First, the GA visits robots scattered in the field to find an idle robot. Once the GA finds an idle robot, the GA appoints that idle robot to be the base robot to calculate the relative coordinates. Then, the GA traverses all the reachable idle robots one by one in order to collect their

locations. When the GA visits all the reachable idle robots, the GA calculates the conceptual barycenter \mathbf{g} as follows:

$$\mathbf{g} = \frac{1}{n} \sum_{i=1}^n \mathbf{p}_i \quad (2)$$

n is the number of robots and \mathbf{p}_i is the relative coordinate of each robot from the base robot, which is calculated by (1). The information of the original formation F is represented as follows:

$$F = \{\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_i, \dots, \mathbf{f}_n\}$$

We assume that \mathbf{f}_i is the vector value from the conceptual barycenter \mathbf{f}_o of the original arrangement of nodes in formation F . Henceforth we call the conceptual barycenter as just the barycenter. The GA translates them to the vectors from the base robot. The GA uses the barycenter \mathbf{g} in the field instead of barycenter of the original formation. Thus, the vectors from \mathbf{g} are identical to the vectors from \mathbf{f}_o . To translate the vectors from \mathbf{g} to the vectors from the base robot, the GA adds \mathbf{g} to each \mathbf{f}_i as follows:

$$\mathbf{p}_{f_i} = \mathbf{f}_i + \mathbf{g}, \mathbf{f}_i \in F \quad (3)$$

Since each \mathbf{f}_i is identical to the vector value from \mathbf{g} , each \mathbf{p}_{f_i} becomes the target locations for robots to move. Upon completion of the calculation of all the vector values, the GA generates NAs that drive robots to \mathbf{p}_{f_i} .

Example 1 (Calculating locations). Consider that the objective shape is the square as shown in Fig. 1. The vector \mathbf{f}_i in this figure is a relative coordinate from the barycenter of the original formation that is represented as a small circle at the center in the figure. Figure 2 shows the initial state of the positions of robots and the pale color circle in the figure represents the base robot that becomes the base coordinate of the positions of robots. That is, the base robot has coordinate (0,0). The GA calculates the barycenter \mathbf{g} by (2), which is the vector (or the relative coordinate) from the base robot (Fig. 3). After that, the GA overlaps the barycenter of the original formation and the barycenter \mathbf{g} in the field as shown in Fig. 4, where \mathbf{f}_i is the relative coordinate from the barycenter of the original formation. To translate \mathbf{f}_i to the vector from the base robot (\mathbf{p}_{f_i} in Fig. 4), the GA add \mathbf{g} to \mathbf{f}_i followed by (3), so that the relative coordinate \mathbf{p}_{f_i} is calculated.

3.2 Node Agent

In this section, we describe the algorithm for NAs. The GA calculates the relative coordinates $(\mathbf{p}_{f_1}, \dots, \mathbf{p}_{f_n})$ from the base robot. n is the number of robots of the formation. For each robot, the GA generates the corresponding NA, and gives each NA the relative coordinate \mathbf{p}_{f_i} as the target location to which it

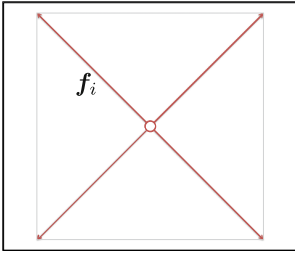


Fig. 1. Objective formation.

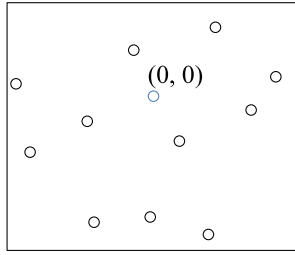


Fig. 2. Initial state. (Color figure online)

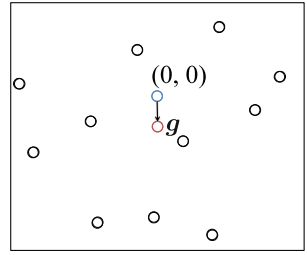


Fig. 3. Barycenter.

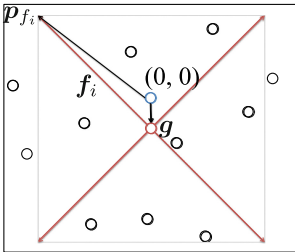


Fig. 4. Translation.

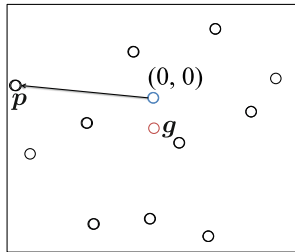


Fig. 5. Location of an NA.

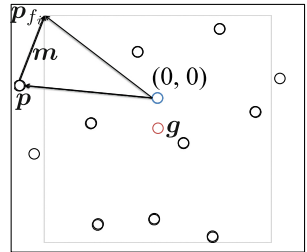


Fig. 6. The movement.

should drives the robot. The NA calculates the movement vector \mathbf{m} to the target location as follows:

$$\mathbf{m} = \mathbf{p}_{f_i} - \mathbf{p} \tag{4}$$

Coordinate \mathbf{p} is the current coordinate of the NA from the base robot as shown in Fig. 5, and movement vector \mathbf{m} is a relative coordinate from current robot to target location as shown in Fig. 6. If a NA finds some robots that are nearer to the target location than the current robot, the NA migrates to the nearer robot instead of driving the current robot in order to suppress the duration time and energy consumption for composing the formation. After each migration, the coordinate \mathbf{p} of NA is updated as follows:

$$\mathbf{p} \leftarrow \mathbf{p} + \mathbf{v}$$

Vector \mathbf{v} is the vector from the current robot to the nearest robot that is measured by sensors or cameras.

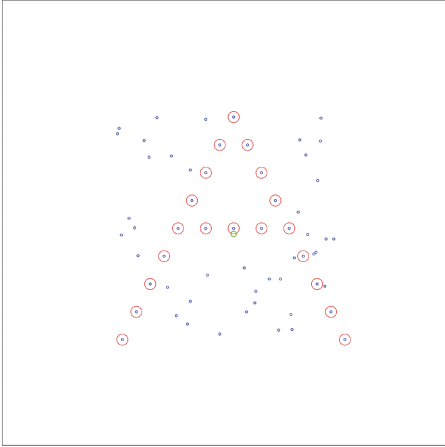


Fig. 7. Formation of letter A. (Color figure online)

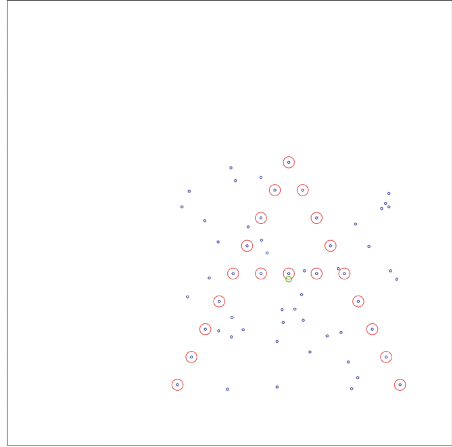


Fig. 8. Formation of letter A (2). (Color figure online)

4 Experimental Results

In order to demonstrate the correctness and effectiveness of our method, we have implemented our algorithm on a simulator, and conducted numerical experiments. In the experiments, we assume the following conditions.

1. Robots are scattered in an 800×800 square field in the simulator.
2. The view range of each robot is 150 units, and the range of wireless network is wider than the view range.
3. Each robot can move 2 units in each step in the simulator.
4. The initial locations of robots are randomly set without overlapping.
5. Each robot is represented as a circle on the grid field.

First, we conducted experiments for formation of letter A in different initial arrangements of robots as shown in Figs. 7 and 8. In the figures, where small black circles represent robots, large circles represent robots with a NA, and the small pale color circle represents the barycenter, both formations are composed around the center of the area where robots scatter. The results show that GA calculates the barycenter and the NAs compose the formation around it correctly.

4.1 Comparison to the Other Algorithm

We also conducted experiments to compare this novel approach with our previous approach [10] in terms of efficiency we gain and the total cost we reduce. We investigated the efficiency by the duration time, and the total cost by the total length of traces of robots for composing formations. In the algorithm, when all the NAs arrive at the target locations means that the formation task is

completed. On the other hand, in our previous work, robots gradually compose a formation, and then continue fine adjustments after the formation composing is mostly completed. Thus, it was hard to decide when the task ends in our previous work. Therefore, we have to decide allowable deviation, and check whether the current locations of robots are under it in every time steps.

In order to show the preciseness in our new algorithm, we measured gaps between the current locations and the objective ideal positions, and calculated the average of them. First, we calculated the barycenter of points for both actual and ideal points, and then calculated all the coordinate of robots relative to the barycenter. After that, we measured the distances between the relative coordinates of actual robots and the ideal relative coordinates, and calculated the averages D of them for various shapes with the difference numbers of robots as follows:

$$D = \frac{1}{n} \sum_{i=1}^n \|I_i - A_i\| \quad (5)$$

In the equation, I_i is the ideal relative coordinate of a robot from the barycenter, A_i is the actual relative coordinate of the robot from the barycenter and n is the number of NAs composing the shape. Figure 9 shows the average distances of this algorithm and the previous work for convergence time, where the horizontal axis is time and the vertical axis is the average distance calculated by (5). In the experiment, the objective formation is a circle with radius 150 units, the number of nodes of the formation is 20, and the number of robots that are scattered in the field is 40. We assume that NAs that compose a formation are generated after the GA calculates the barycenter and the target locations based on it, and hence, the line chart for the newly proposed approach, which represents the change of average distances, starts from 100 time steps after calculation of the barycenter. As shown in the figure, the experimental result shows our newly proposed approach is remarkably more efficient than the previous approach even if it includes the time for calculating the barycenter and the target locations. This is because each NA knows its own target location to move, so that NAs can go straight to the locations.

On the other hand, the previous approach determines the target locations dynamically. It continuously adjusts the current locations once they arrive around the targets. Indeed, in most cases, all NAs completed their task by the 140 steps, while the previous approach took approximately 600 time steps to settle down. Also, for the same reason, the total length of traces of the newly proposed approach becomes much shorter than the previous approach. The trace of the newly proposed approach is 30 times as short as the previous approach. Previous approach takes 25000 units and newly proposed approach 800 units.

4.2 Migration

Migrations to other robots can contribute to suppressing the duration time and energy consumption for composing formations. To show this effect, we conducted

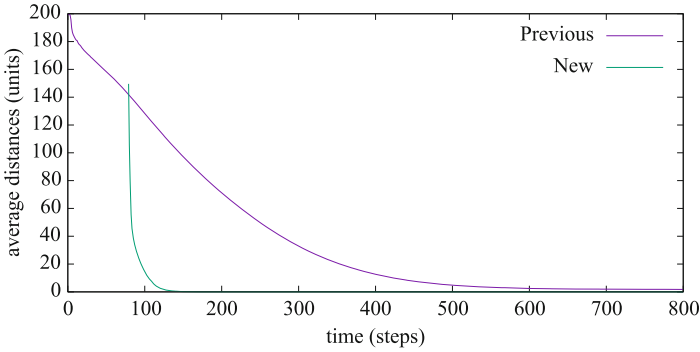


Fig. 9. Comparison to our previous approach.

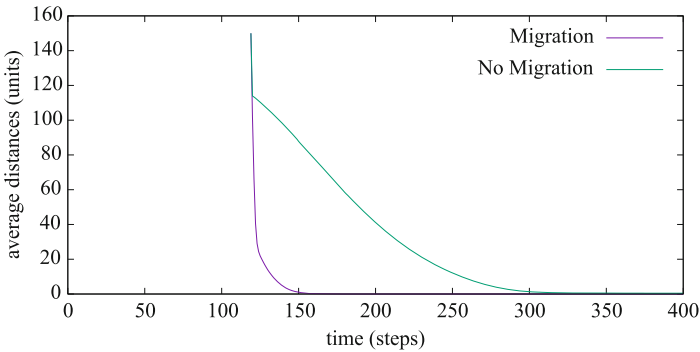


Fig. 10. Effect of migration.

an experiment under the condition where NAs continue to stay on initial robots without migrating to other robots. We compare the result with an experiment in which NAs use the migration to make robots move to their target locations, where the objective shape is a circle whose radius is 150 units. The number of nodes of the shape is 20, and the number of robots is 60. Therefore only 20 out of 60 robots are used to compose the shape. Figure 10 shows the average distance of the experiments. In the case with migration, the whole task was completed at 150 time steps, where the GA completed its task at 120 time steps and then NAs moved to their target in 30 time steps.

On the other hand, in the case without migration, it took 310 time steps to complete the whole task, where the total time of the task of NAs was 190 time steps that was 6 times as much as the case with migration. Also, as shown by the bar named “60” in Fig. 11, the total length of traces of robots for composing the formation was 600 for the case with migration, and 2700 for the case without migration. This result shows the traces with migration are 4.5 times as short as ones without migration in average. These results show the superiority of the migration.

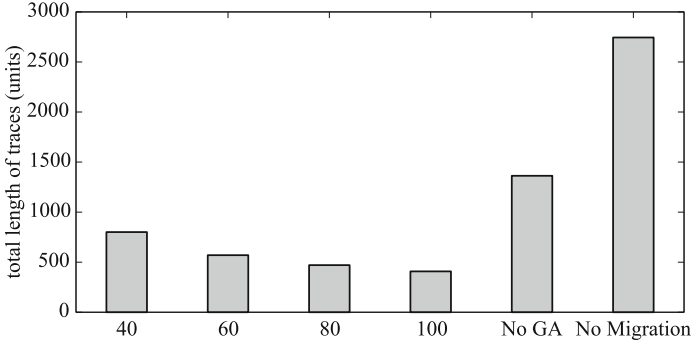


Fig. 11. Total length of traces for different condition.

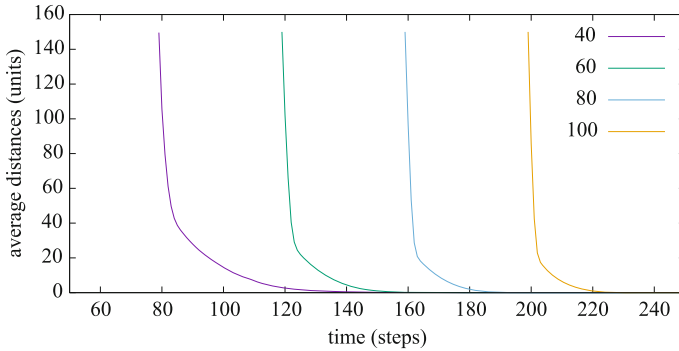


Fig. 12. Effect of the number of robots.

Finally, in order to check the influences of the number of robots to the duration time and energy consumption, we conducted experiments with the cases of the number of robots: 40, 60, 80 and 100. Like the other experiments, the objective shape is a circle whose radius is 150 units, and the number of nodes of the shape is 20. Figure 12 shows the average distances of these experiments. As shown in the figure, the GAs completed their tasks at time steps 80, 120, 160 and 200, respectively. This means that the cost of GAs is in proportion to the number of robots. On the other hand, NAs complete their tasks at 60, 30, 25 and 20 time steps, respectively. Also, as shown in Fig. 11, the total lengths of traces are 800, 570, 470 and 410, which is in inverse proportion to the number of robots. The result shows that the migration is suppressing energy consumption more effectively as the number of robots increases. In our simulator, mobile agents are able to migrate to other robots in one time step and robots move 2 units in one time step. Hence, the faster network speed is, the more important the number of robots becomes.

Conversely, when network speed is slow, visiting robots for collecting their locations might become a bottleneck of the task in terms of total time. The bar

named “No GA” in Fig. 11 shows the experiment where the GA does not collect any locations of robots. In this case, where the number of robots is 60 and the barycenter is not calculated, the length of traces becomes twice as long as the case with GA. These experiments show that the newly proposed approach can suppress energy consumption in proportion to network speed and the number of robots without sacrificing the total time.

4.3 Accuracy of Movement and Sensors

We have conducted experiments in the simulator assuming the ideal conditions. In such a condition, robots can move correctly with no error and sensors or cameras can measure precise distances and directions to other robots. In the real world, there are some movement errors and sensor errors caused by imperfect hardware. We simulate sensor errors, movement errors and angle errors.

Figure 13 shows the influences of errors for the newly proposed approach and the previous approach in the same error condition, where the objective shape is a circle whose radius is 150 units. The number of nodes of the shape is 20, and the number of robots is 60. Sensor errors and movement errors are 20 percent of sensed distance and movement length respectively. Angle errors is added within $[-10, 10]$ degrees. In the newly proposed approach, the duration time of the task was not influenced by the errors. Since robots could not measure the precise locations, however, the eventual formation was distorted. The average distance of robots from actual locations to the ideal locations was 20 units.

On the other hand, although the previous approach was also affected by errors, the eventual formation is less distorted than the newly proposed approach. This arises from the difference of the distance between attractor and attractee. In the newly proposed approach, NAs that are attractors tend to move to the target locations that are attractees in a long way repeating migrations, so that errors are accumulated. On the other hand, in the previous approach, each agent just attracts other agents to neighbor locations. Thus, the newly proposed approach

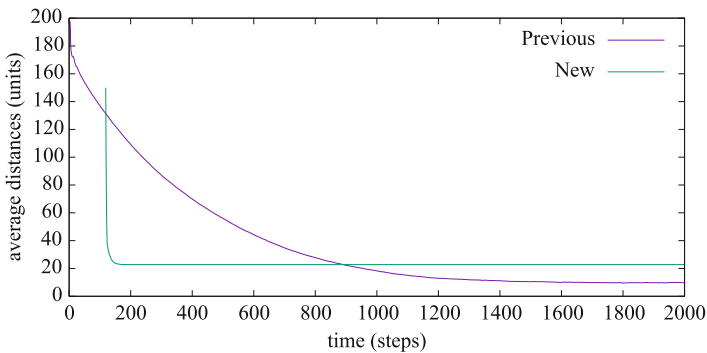


Fig. 13. Influence of error.

would be suitable for composing formations with not so many nodes for not so big shape.

5 Conclusions and Future Works

We have proposed a control algorithm for composing formations of swarm robots based on their distributions using many mobile software agents. In this approach, we introduced two kinds of mobile software agents, a guide agent (GA) and node agents (NAs). The GA traverses all the robots and calculates the conceptual barycenter of them, and then, calculates the suitable locations of the formation using the barycenter. The GA generates NAs that drive robots to the calculated locations. Each NA migrates to the robot that is nearest to the target location, and drives that robot to the location.

In order to show the effectiveness of our algorithm, we have implemented it on a simulator. And then we have conducted numerical experiments on the simulator. We have shown that our algorithm suppresses the duration time for a given formation task and energy consumption compared with our previous approach, and composes the formation at a suitable location based on the distribution of robots. Furthermore, we have shown that increasing the number of robots is effective to suppress energy consumption. We have simulated the errors caused in the real world, and shown that the proposed approach in this paper is not affected by the errors in term of the duration time of the formation task, while it tends to accumulate errors in the migration process compared with the previous approach.

As a future direction, we plan to conduct a control experiment. We will implement another multi-robot system where message-based approach being employed, and compare the result with that of our current agent-base system to show superiority of our agent-based approach. Also we plan to invent a new algorithm that adjusts the each location at certain intervals at low network cost in order to mitigate the problem of error accumulation.

Acknowledgments. This work is supported in part by Japan Society for Promotion of Science (JSPS), with the basic research program (C) (No. 26350456), Grant-in-Aid for Scientific Research.

References

1. Takimoto, M., Mizuno, M., Kurio, M., Kambayashi, Y.: Saving energy consumption of multi-robots using higher-order mobile agents. In: Nguyen, N.T., Grzech, A., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2007. LNCS (LNAI), vol. 4496, pp. 549–558. Springer, Heidelberg (2007)
2. Binder, W., Hulaas, J.G., Villazon, A.: Portable resource control in the J-SEAL2 mobile agent system. In: Proceedings of the Fifth International Conference on Autonomous Agents (AGENTS 2001), pp. 222–223. ACM (2001)
3. Kambayashi, Y., Takimoto, M.: Higher-order mobile agents for controlling intelligent robots. *Int. J. Intell. Inf. Technol. (IJIIT)* **1**(2), 28–42 (2005)

4. Nagata, T., Takimoto, M., Kambayashi, Y.: Suppressing the total costs of executing tasks using mobile agents. In: 42nd Hawaii International Conference on System Sciences, 2009 (HICSS 2009), pp. 1–10, January 2009
5. Abe, T., Takimoto, M., Kambayashi, Y.: Searching targets using mobile agents in a large scale multi-robot environment. In: O’Shea, J., Nguyen, N.T., Crockett, K., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2011. LNCS, vol. 6682, pp. 211–220. Springer, Heidelberg (2011)
6. Shibuya, R., Takimoto, M., Kambayashi, Y.: Suppressing energy consumption of transportation robots using mobile agents. In: Proceedings of the 5th International Conference on Agents and Artificial Intelligence (ICAART 2013), pp. 219–224. SciTePress (2013)
7. Mizutani, M., Takimoto, M., Kambayashi, Y.: Ant colony clustering using mobile agents as ants and pheromone. In: Nguyen, N.T., Le, M.T., Świątek, J. (eds.) ACIIDS 2010. LNCS, vol. 5990, pp. 435–444. Springer, Heidelberg (2010)
8. Shintani, M., Lee, S., Takimoto, M., Kambayashi, Y.: Synthesizing pheromone agents for serialization in the distributed ant colony clustering. In: ECTA and FCTA 2011 - Proceedings of the International Conference on Evolutionary Computation Theory and Applications and the Proceedings of the International Conference on Fuzzy Computation Theory and Applications [Parts of the International Joint Conference on Computational Intelligence (IJCCI 2011)], pp. 220–226. SciTePress (2011)
9. Shintani, M., Lee, S., Takimoto, M., Kambayashi, Y.: A serialization algorithm for mobile robots using mobile agents with distributed ant colony clustering. In: König, A., Dengel, A., Hinkelmann, K., Kise, K., Howlett, R.J., Jain, L.C. (eds.) KES 2011, Part I. LNCS, vol. 6881, pp. 260–270. Springer, Heidelberg (2011)
10. Oikawa, R., Takimoto, M., Kambayashi, Y.: Distributed formation control for swarm robots using mobile agents. In: Proceedings of the Tenth Jubilee IEEE International Symposium on Applied Computational Intelligence and Informatics, pp. 111–116 (2015)
11. Cheng, J., Cheng, W., Nagpal, R.: Robust and self-repairing formation control for swarms of mobile agents. In: Proceedings of the 20th National Conference on Artificial Intelligence AAAI 2005, vol. 1, pp. 59–64. AAAI Press (2005)
12. Kanjanawanishkul, K.: Formation control of mobile robots: survey. *eng.ubu.ac.th* 50–64 (2005)
13. Balch, T., Arkin, R.: Behavior-based formation control for multirobot teams. *IEEE Trans. Robot. Autom.* **14**(6), 926–939 (1998)
14. Lewis, M., Tan, K.H.: High precision formation control of mobile robots using virtual structures. *Auton. Robots* **4**(4), 387–403 (1997)
15. Das, A., Fierro, R., Kumar, V., Ostrowski, J., Spletzer, J., Taylor, C.: A vision-based formation control framework. *IEEE Trans. Robot. Autom.* **18**(5), 813–825 (2002)