Controlling Formations of Robots with Graph Theory

Jan Carlo Barca, A. Sekercioglu, and A. Ford

Department of Electrical and Computer Systems Engineering,
Monash University, Melbourne, Australia
research@jancarlobarca.com,
ahmet.sekercioglu@eng.monash.edu.au,
adam.ford.eng@gmail.com

Abstract. A number of techniques that allow autonomous multi-robot systems to be held within formation-like structures exist but they are limited by a high communication load, high energy usage and a lack of robustness. This research improves on state-of-the-art formation control schemes for leader-follower type multi-robot systems by employing mechanisms that enable groups of robots to move in two-dimensional formations without the need for inter robot communication. We also incorporate techniques that enable the robots to move back into formation in a precise manner when external interferences have caused the formation shape to deteriorate. The control system is derived through the use of graph theory and has been tested rigorously in a realistic simulator to prove its applicability to multi-robot control.

1 Introduction

A multi-robot (MR) system consists of a group of robots that act autonomously and is designed to solve problems that would be impossible for a single robot. In single robot systems, there is a huge responsibility put onto one robot which, in turn causes a single point of failure. MR systems address this issue by creating redundancies and expanding responsibilities, therefore providing a decreased system fail rate. As the field of robotics progresses, there is a need to focus research efforts more towards distributed-type robotic systems [1].

Formations in robotics can be defined as the inter-relational physical structure where the MR system is kept in tightly pre-defined restricted patterns. It allows the ability to move large objects [2] compared to just one robot, and decreases the overall time and effort required for large area exploration and mapping. The applications of formations within MR systems are extensive, they can range from military Unmanned Aerial Vehicle surveillance, mapping and surveying of environments and exploration of unpredictable territory to mobile sensory networks and satellite high-resolution imagery [3,2,4]. As the world becomes more dangerous and conflicting, there arises a need to produce systems, such as those that incorporate formations that enhance the safety of human beings.

One of the advantages of having a MR system maintaining formation is that all robots can depart from and arrive at a waypoint at the same time. There is also a psychological effect if used within a military application that if a huge group of objects is

S. Lee et al. (Eds.): Intelligent Autonomous Systems 12, AISC 194, pp. 563–574.
springerlink.com

moving towards a group of enemies, it can have a detrimental effect on morale. In our work we focus on triangular and diamond formations due to the benefits associated with these particular formations. A triangular formation allows the system of robots to be gradually exposed to an area, and offers formation leniency because no entity follows directly behind another. While diamond formations allow forward, side and back protection to the group when used within a military scenario and entering environments where being ambushed is a possibility [2,5].

There are some areas of formation control that can be improved on. High computational usage can, for example, prevent formation control algorithms from running to their full potential [6]. In our work, which builds upon the techniques presented in [2,4] we reduced this problem by using graph theory and minimal constraints to enable formation control without the need for inter robot communication, thus reducing energy consumption. Other variables that come into play when MR systems are deployed in dynamic real world environments are the disturbances caused by the interactions between different robots in the system and the interferences from obstacles in the environment [4,6], which can make it hard for the system to retain a precise formation structure. This problem can be observed in several well established formation control techniques for MR systems such as the morphogenesis based techniques presented by [7-9] and the potential fields based technique presented by [10]. None of these techniques can ensure that the exact formation shape is regained if the system is affected by external disturbances. To address this issue we incorporate a gradient decent function that enables the robots to move back to their exact positions in the formation after they have been forced away by external interference. A difference between other formation control techniques that use minimally rigid graphs and our work is that our technique does not rely on Global Positioning Systems (GPS) to correct the positions of robots in the formation. This is advantageous as it opens up for using the technique in areas where GPS is unavailable, such as indoors, underground, in the ocean or in space. Relevant definitions from graph theory are introduced in the following section.

2 Theory – Definitions and Preliminary Remarks

Leader-follower type MR systems usually have one *main leader (ML)* that guides the movement of the followers by means of *sub-leaders (SL).* These *SL* are located at intersection points of branched graphs, which are rooted in the *ML,* and guide the movement of robots further down in the graph hierarchy. Followers can therefore have more than one leader guiding their movement even if only one *ML* exists.

To define the physical relations between the individual robots, the use of graph theory is seen to be greatly beneficial because of its algorithms and definitions that have been developed to find rigid and minimally-constrained structures. Therefore to gather a total understanding on the theories behind this research, a broad knowledge of graph theory is required. A series of relevant definitions and concepts from graph theory are therefore described in this section.

Directed edges are represented by the constraints that are defined within a graph. For example, if two vertices (***a***) and (***b***) exist and they are connected by an edge, then the edge is directed if ***a*** connects to ***b***, but ***b*** is not connected to ***a***. If ***a*** connects to ***b***, and ***b*** also connects to ***a***, then the edge is undirected. It is often desirable to use directed edges in MR systems as only one robot in a pair is responsible for maintaining a particular inter robot constraint. This is beneficial as the energy consumption associated with maintaining the constraint can be kept at a minimum. A graph with only directed edges is referred to as a digraph.

To devise a graph that is structured and rigid, it is required that all constraints are able to be followed and the distances between each vertex must remain constant at any time. For a two-dimensional graph to be defined as minimally rigid the graph must be in a state that if any edge was to be removed it would cause the graph to be no longer rigid [11]. One can check if a graph with (v) number of vertices is minimally rigid by testing if the number of edges (e) satisfies ‘$|e| = 2\ |v| - 3$’ [11]. It is beneficial to use minimally rigid graphs in MR systems as no unnecessary constraints are used to uphold desired formations, which in turn ensures that particular formation shapes can be maintained in an energy efficient manner.

For a rigid graph to become persistent, it must have its undirected edges replaced by directed edges [12]. This can be done by replacing each of the undirected edges with directed edges in a way that retains the rule that a digraph is persistent only if any two vertices in the graph can preserve their relative distance constraints during any motion [12]. Fig. 1 illustrates a persistent and a non-persistent graph. Next section describes how constraints that have been defined with graph theory can be maintained by an MR system.

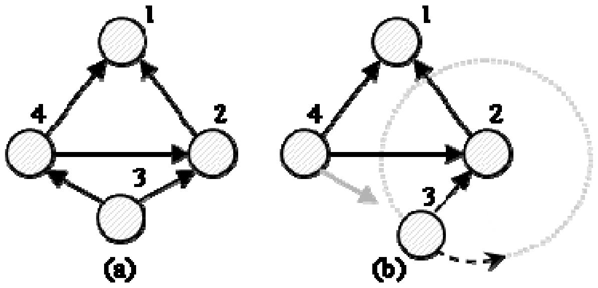

Fig. 1. Persistent and non-persisten graphs: (a) is persistent as all vertices can maintain their relative distance during any motion, and (b) is non-persistent as vertex 4 is unable to satisfy all its distance constraints when vertext 3 moves along the dotted circle

3 Dual Leader Averaging Follower Algorithm

An overview of how the followers operate is given in Fig. 2. All robots in the MR system make use of this control method with the exception of the *ML*, which is controlled by a human operator. (The *ML* can also alternatively follow pre-defined paths.) One can observe that the control method enables the robots to take relative distance and angle measurements to one or two leaders into account during the control process. How these measurements are captured with a distance measuring device and flow through the control system can also be observed. Any distance measuring device can be used to support the control process as long as the device makes it possible to determine relative distances and angles to other robots. Each component of the control method is discussed in the following subsections.

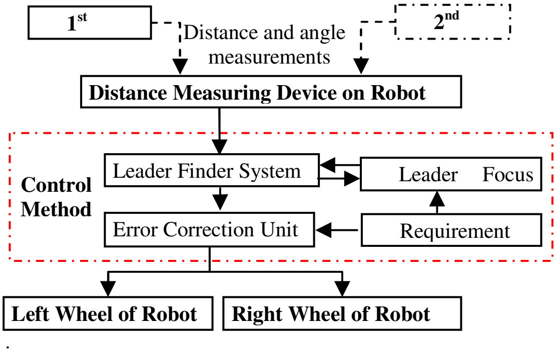

Fig. 2. Overview of how followers operate

3.1 Leader Focus Unit

The Leader Focus Unit (LFU) assists the Leader Finder System (LFS) with “locking” onto allocated leader/s. The LFU offers this support by transmitting the expected relative angle/s between the robot at hand and its associated leader/s to the LFS at startup. The LFU obtains this information from the Requirement Unit (RU) which contains all necessary constraints for maintaining a particular formation shape. After startup the LFU continues to assist the LFS by transmitting the previous relative angle/s between the robot and its leader/s. The purpose behind the LFU is to reduce the processing load associated with iterating through data from the distance measuring device when searching for leader/s.

3.2 Leader Finder System

The LFS checks if a robot is located at the relative angle/s provided by the LFU. If a robot is not detected at an expected angle, then the search arc is expanded horizontally with one resolution in left and right directions along the horizontal axis. This expansion of the search arc is repeated incrementally until a robot is detected (Fig. 3). Empirical results show that this approach works when the robots are distributed according to desired formation patterns from the outset. (Refer to [5] for a technique that can be used to generate desired formation patterns.) Once a robot is detected the actual relative angle and distance to the robot are passed over to the Error Correction Unit (ECU). If the LFU transmitted two angles to the LFS, then the search is repeated for the second angle to find a second leader. Feedback is also sent to the LFU if there is a discrepancy between the angle/s transmitted to the LFS and the actual angle/s to the leader/s.

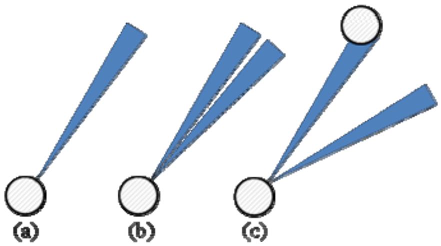

Fig. 3. Search for a leader: (a) the search is initiated in the angle provided by the LFU, b) the search arc is expanded in left and right directions along the horizontal axis if no robot is detected in the initial region, and c) the search arc is expanded incrementally until a robot is found

3.3 Error Correction Unit

The ECU enables robots to reduce the discrepancy between their desired and their actual relative position with respect to their leader/s so that relevant formations can be maintained. A mechanism that is employed when a robot has two leaders will be discussed first as most robots in our formations have two leaders.

3.4 Error Correction with Two Leaders

This mechanism compares the measurements provided by the LFS with desired distance and angle measurements obtained from the RU, in order to calculate an error vector that pushes the robot at hand towards a desired relative pose with respect to two leaders. The idea is illustrated in Fig. 4, where one can observe how follower (*F1*) moves according to the $e_{x,ave}$ and $e_{y,ave}$ vectors to reduce the discrepancy between the actual and the desired relative pose with respect to leaders (*L1*) and (*L2*).

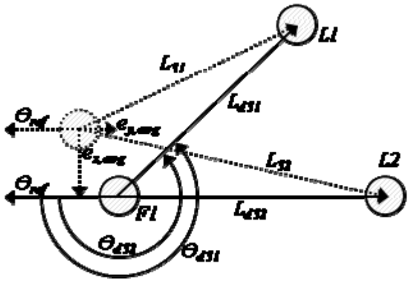

Fig. 4. Reducing discrepancy between desired and actual relative pose of follower *F1*

The mechanism initiates by calculating two separate sets of error vectors with respect to *L1* and *L2*. Corresponding vectors are then synthesized to produce average error vectors, which *F1* acts upon while attempting to mirror the two leader velocities. The preliminary error vectors are calculated according to (1) to (4) and synthesized using (5) and (6).

$$e_{x,31} = (L_{31}\cos(\theta_{31}) - L_{d31}\cos(\theta_{d31})) \quad (1)$$

$$e_{y,31} = (L_{31}\sin(\theta_{31}) - L_{d31}\sin(\theta_{d31})) \quad (2)$$

$$e_{x,32} = (L_{32}\cos(\theta_{32}) - L_{d32}\cos(\theta_{d32})) \quad (3)$$

$$e_{y,32} = (L_{32}\sin(\theta_{32}) - L_{d32}\sin(\theta_{d32})) \quad (4)$$

where:

$e_{x,31}$: x-coordinate for error displacement vector between *F1* and *L1*

$e_{y,31}$: y-coordinate for error displacement vector between *F1* and *L1*

L_{31} : Actual distance between *F1* and *L1*

L_{d31} : Required distance between *F1* and *L1*

θ_{31} : Actual angle between *F1* and *L1* with respect to θ_{ref}

θ_{d31} : Required angle between *F1* and *L1* with respect to θ_{ref}

$$x = e_{x,ave} = (e_{x,32} + e_{x,31})/2 \quad (5)$$

$$y = e_{y,ave} = (e_{y,32} + e_{y,31})/2 \quad (6)$$

where:

$e_{x,ave}$: x-coordinate average error displacement vector

$e_{y,ave}$: y-coordinate average error displacement vector

To control the forward velocity of *F1* a proportional derivative controller acts upon changes in the y variable with respect to *F1*'s required position as in (8). The velocity is only computed for the y axis as forward velocities are specified in a local coordinate system where the positive y axis always point directly in front of the robot.

$$\Delta y = \frac{y(t) - y(t-T)}{T} \quad (7)$$

$$V_{control} = \begin{cases} K_v.\Delta y + K_{dist}.\log(y(t)+1) - K_{dist}.1.05.\log(y(t-T)+1) + V_1, & y(t) \geq 0 \\ K_v.\Delta y - K_{dist}.\log(-y(t)-1) + K_{dist}.1.05.\log(-y(t-T)-1) + V_1, & y(t) < 0 \end{cases} \quad (8)$$

where:

Δy : Change in $e_{y,ave}$ over two sequential samples

$y(t)$: y-coordinate average error as a function of time

T : Sample rate of distance measuring device

V_1 : Current velocity of *F1* in millimeters per second

V_{contro} : Required velocity of *F1* in millimeters per second

K_{dist} : Distance constant for velocity control

K_v : Velocity constant for velocity control

While the y-coordinate changes a velocity control mechanism takes into account the rate at which this value changes and compensates to mirror the formation speed while still maintaining the displacement required. The angular velocity of *F1* is controlled using a proportional transparent dead-zone controller. The average angle of the error vector is calculated first (9). Then the magnitude of the average error vector is calculated (10) for use within the angular speed's dead-zone proportional controller (12). A secondary heading is also calculated (11) from the original readings and constraints for use within the controller. The heading result from (11) begins to influence the controller when *F1* enters a dead zone (magnitude of the error vector is less than d).

$$\begin{aligned} e_{\theta,ave} &= \tan^{-1}\frac{x}{y};\, x < 0 \\ & \tan^{-1}\frac{x}{y} + \pi;\, x \geq 0, y > 0 \\ & \tan^{-1}\frac{x}{y} - \pi;\, x \geq 0, y < 0 \end{aligned} \tag{9}$$

$$| e_{average} | = \sqrt{{e_{x,ave}}^2 + {e_{y,ave}}^2} \tag{10}$$

$$e_{direction} = \frac{(\theta_{31} - \theta_{d31}) + (\theta_{32} - \theta_{d32})}{2} \tag{11}$$

$$\omega_{control} = \begin{cases} K_{ang}.e_{\theta,ave}, | e_{average} | > d \\ K_{ang}.(e_{direction} - (e_{direction} - e_{\theta,ave})\frac{| e_{average} |}{d}), | e_{average} | \leq d \end{cases} \tag{12}$$

where:

$e_{\theta,ave}$	: Average angle of error vector
$\omega_{control}$	: Correction angle for *F1* in degrees per second
K_{ang}	: Constant for angular control system
d	: Dead-zone constant

3.5 Error Correction with One Leader

If *F1* only has one leader then distance and angle constraints are retained using (13) to (16). Equation (13) is used to calculate the distance error between the desired and the actual position of *F1*, whilst (15) is used to determine the heading error. The idea is illustrated in Fig. 5 where one can observe how *F1* moves according to the $e_{x,31}$ and $e_{y,31}$ vectors to reduce the discrepancy between the desired and actual relative pose with respect to *L1*.

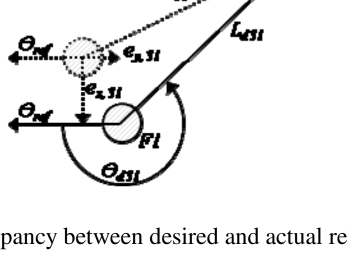

Fig. 5. Reducing discrepancy between desired and actual relative pose of follower *F1*

$$e_{distance} = L_{31} - L_{d31} \tag{13}$$

$$V_{control} = K_{dist}.\log(e_{distance}) \tag{14}$$

$$e_{direction} = (\theta_{31} - \theta_{d31}) \tag{15}$$

$$\omega_{control} = K_{ang}.e_{direction} \tag{16}$$

4 Simulation Design and Results

Two types of simulations that have been carried out to test the robustness of the proposed control method towards interferences that causes formation shapes to deteriorate are presented in this section. The interferences are introduced in the form of various accelerations and changes in the heading direction of the *ML*. Both simulation types were carried out with the Microsoft Robotics Developer Studio simulator suite and varying numbers of simulated robots equipped with a differential drive and a Laser Range Finder (LRF). The LRF has a viewing angle of 360°, a resolution of 1° and an update rate of 4 Hertz. The parameters that were used for implementation specific variables throughout the simulations are presented in Table 1.

Table 1. Parameters for implementation specific variables

T	0.25 seconds	K_v	0.168 millimeters
K_{dist}	60.057 millimeters	d	500 millimeters
K_{ang}	0.75 degrees	-	-

An overview of the constraints that were used in the triangular and diamond shaped formations that were employed throughout the simulations is provided in Fig. 6. One can observe that four different types of constraints were employed. The exact parameters associated with each constraint type are provided in Table 2.

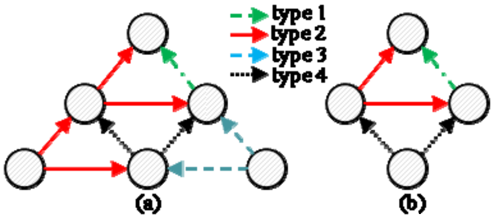

Fig. 6. (a) triangular and (b) diamond formations that were employed throughout the simulations

Table 2. Parameters for each constraint type

Type	Distance in millimeters	Angle 1 in degrees	Angle 2 in degrees
1	2000	210	-
2	2000	90	150
3	2000	210	270
4	2000	150	210

4.1 Robustness toward Increasing Leader Robot Acceleration

Simulations that investigated how robust the control method is towards accelerations of the *ML* are described in this section. It is important to investigate this issue as the *ML* will vary its velocity frequently in our future research as we intend to use the control method to operate MR systems in unpredictable and cluttered environments. Two sets of simulations were carried out to investigate the issue.

In the first set of simulations the velocity of the *ML* was increased from zero to 320 millimeters per second (mm/sec) before the average time it took for the robots to regain the formation was recorded. The outcome of this process is presented in Fig. 7. One can observe that it takes about two seconds for the followers to mirror the velocity of the *ML*. After this initial period the followers start to converge into their required positions. After 18.5 seconds the robots in the diamond formation converge. The robots then begin to oscillate in and out of the desired positions. Empirical results show that this behavior is a result of the relatively slow update rate of our LRF. We therefore expect that this behavior can be reduced by employing a distance measuring device with a faster update rate. Robots in the triangular formation converge and start to oscillate 7.1 seconds later, which shows that the control method enables both formations to converge to their desired shape.

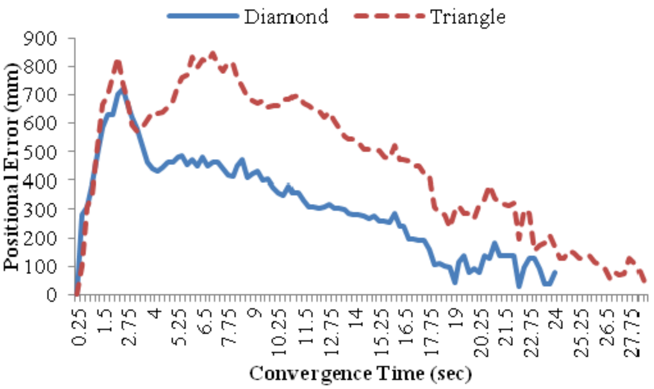

Fig. 7. Difference between desired and actual positions for followers as velocity of *ML* increased

In the second set of simulations the velocity of the *ML* was first increased from zero to 25.36 mm/sec before the average convergence time was recorded. This process was then repeated 30 times while incrementally increasing the velocity of the *ML* with 25.36 mm/sec per run. The results from this process are presented in Fig. 8. One can observe that the convergence time is close to zero for both formations until the velocity of the *ML* reaches 150 mm/sec. However, one can also observe that the convergence time increases exponentially when the velocity of the *ML* is increased beyond this point. By studying the two graphs in Fig. 8 it furthermore becomes clear that the triangular formation has a tendency to convergence later than the diamond shaped formation. Empirical results show that this is due to the relatively larger size of the triangular formation.

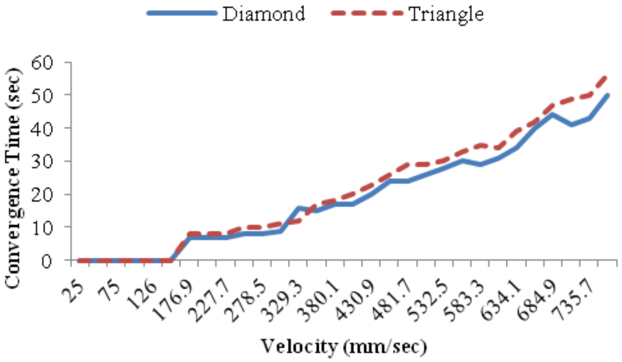

Fig. 8. Average convergence time of followers when target velocity of *ML* is increased

4.2 Robustness toward Increasing Curvature in Travel Path

This section describes simulations that were carried out to investigate how robust the control method is towards changes in the heading direction of the *ML*. It is important to investigate this issue as we intend to deploy our MR system into environments with obstacles in the future, and it is therefore expected that the *ML* will have to change heading direction frequently to avoid collisions. To investigate the issue we initially set the velocity of both wheels on the *ML* to 328 mm/sec and measured the average convergence time. The speed of the left wheel was then increased in intervals of 6.1 mm/sec until the control method failed to uphold the desired formation. Results from these simulations are depicted in Fig. 9. One can observe that the convergence time increases linearly as the turning curvature increases, and that the control method fails to uphold the formations when the left wheel has a speed of 540 and 556 mm/sec. Empirical analysis show that the control method fails at these speeds because the formations become so deteriorated at certain parts of the travelling path that the followers lock onto wrong leaders.

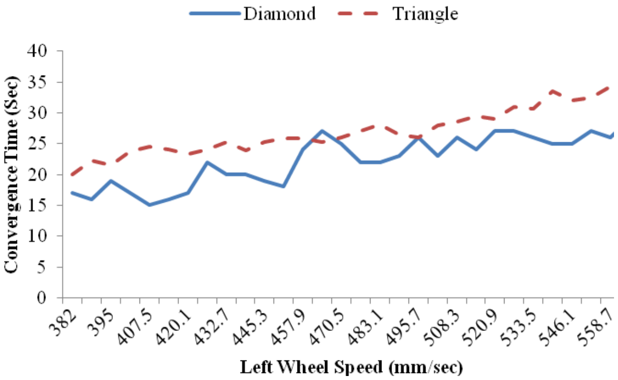

Fig. 9. Convergence time as turning curvature of *ML* is increased

5 Conclusions

A control method that enable groups of robots to move in formations without the need for inter robot communication has been presented. Simulations carried out to investigate how robust the control method is towards changes to the velocity and the heading direction of the *ML* was also described. Results from these simulations show that both triangular and diamond formations converge to their desired shapes when the control method is employed. However, results also show that the robots have a tendency to oscillate around their ideal positions after they have converged due to the relatively slow update rate of our LRF. In addition, it was found that the control method can fail to preserve formation shapes when the turning curvature of the *ML* is large as our robots easily lock onto wrong leaders in these situations. Our current research investigates how: i) robots can converge efficiently to their ideal positions with higher sensor update rates, ii) to preserve formations when follower robots lose sight of their leaders due to external disturbances such as obstacles, and iii) to make the control method more scalable. We also perform tests on real robots.

Acknowledgments. This research was supported in part by Lise and Arnfinn Hejes Grant for Education and Research. We would like to thank Adv. Harald Røer for the way he administered the grant. We would also like to thank Dr Dale Ford and Ashleigh Garling for their comments and insightful suggestions for improving the paper.

References

1. Parker, L.E.: Current Research in multirobot systems. Artif Life Robotics 7, 1–5 (2003)
2. Chen DS, J., Yang, J., Chen, H.: Leader-Follower Formation Control of Multiple Non-Holonomic Mobile Robots Incorporating a Receding-Horizon Scheme. The International Journal of Robotics Research 29(6), 727–747 (2010)
3. Florian Durfler, B.F.: Geometric Analysis of the Formation Problem for Autonomous Robots. IEEE Transactions on Automatic Control 55(10), 2379–2384 (2010)

4. Castro, R., Alvarez, J., Martinez, J.: Robot Formation Control using Backstepping and Sliding Mode Techniques. Paper presented at the Electrical Engineering, Computing Science and Automatic Control, 6th International Conference, Toluca (January 2009)
5. Barca, C., Sekercioglu, A.: Generating Formations with a Template based Multi-Robot System. In: Australasian Conference on Robotics and Automation, Melbourne, pp. 1–9 (2011)
6. Yuqing He, J.H.: Multiple Robots Formation Control Based on Receding Horizon Optimization. Paper presented at the IEEE International Conference on Information and Automation, June 22-25 (2009)
7. Guo, H., Meng, Y., Jin, Y.: Analysis of local communication oad in shape formation of a distributed morphogenetic swarm robotic system. In: IEEE Congress on Evolutionary Computation, Barcelona, pp. 1–8 (2010)
8. Mamei, M., Vasirani, M., Zambonelli, F.: Experiments of Morphogenesis in Swarms of Simple Mobile Robots. Applied Artificial Intelligence 18(9-10), 903–919 (2004)
9. Jin, Y., Meng, Y., Guo, H.: A morphogenetic self-organization algorithm for swarm robotic systems using relative position information. Paper Presented at the UK Workshop on Computational Intelligence, Colchester (2010)
10. Barnes, L., Valavanis, M.K.: Swarm Formation Control Utilizing Elliptical Surfaces and Limiting Functions. IEEE TRansactions on Systems, Man and Cybernetics 39(6), 1434–1445 (2009)
11. Anderson, B., Yu, C., Fidan, B., Hendrickx, J.: Rigid Graph Control Architectures for Autonomous Formations. IEEE Control Systems Magazine (2008)
12. Luo, X.-Y., Li, S.-B., Guan, X.-P.: Automatic generation of min-weighted persistent formations. Chinese Physics B 18(8), 3104–3114 (2009)