Leader-Follower Formation for UAV Robot Swarm Based on Fuzzy Logic Theory

Wilson O. Quesada[1], Jonathan I. Rodriguez[1], Juan C. Murillo[1], Gustavo A. Cardona[2], David Yanguas-Rojas[2], Luis G. Jaimes[3], and Juan M. Calderón[1,4(✉)]

[1] Department of Electronic Engineering, Universidad Santo Tomás, Bogota, Colombia
{wilsonquesada,jonathan.rodriguez,juanmurillog, juancalderon}@usantotomas.edu.co
[2] Department of Electrical and Electronics Engineering, Universidad Nacional de Colombia, Bogota, Colombia
{gacardonac,dryanguasr}@unal.edu.co
[3] Florida Polytechnic University, Lakeland, FL, USA
ljaimes@floridapoly.edu
[4] Bethun-Cookman University, Daytona Beach, FL, USA
calderonj@cookman.edu

Abstract. This paper proposes an algorithm based on a fuzzy logic approach, capable to guide a robot swarm with the aim to keep a leader-follower formation without colliding with other swarm agents. The fuzzy system is programmed and evaluated originally in Matlab, where several experiments were performed. The results depicted a robot swarm showing some bio-inspired behaviors, such as swarm agents surrounding the leader when it is in a static position or when it is traveling from one place to another place. Finally, the proposed fuzzy system was implemented on a drone swarm using V-Rep. The drones simulation shows the swarm navigating together and keeping the leader in the center of the swarm when it is static and following the leader when it is moving. These results could be evaluated in a future work using drone robot swarm in real environments.

Keywords: Swarm robotics · Autonomous mobile robots
Fuzzy logic theory · Drone swarm

1 Introduction

The use of robots' teams for the realization of multiple tasks has become more extended in the recent years, since it is a more reliable system because, in the possibility that one or more robots get to damage the development of the task can take more time due that reduces the number of robots working but the

L. Rutkowski et al. (Eds.): ICAISC 2018, LNAI 10842, pp. 740–751, 2018.
https://doi.org/10.1007/978-3-319-91262-2_65

task will be satisfactory contrary to what happens in the systems of a single sophisticated and expensive robot as it was decades before. Obviously, the arrival on the market of cheaper and improved robotic platforms such as quadcopters opened the possibilities of the implementation of this type of systems at present how is shown in [1], thus improving the performance of the robots in the tasks assigned to them optimizing the time.

There are many applications where multi-robot systems become important because they can perform well in multiple tasks, for example in [2] they show how to make localization of the robots using relative observation between them, this is one of the main challenges to work with multi-robot systems, it is necessary to know all time about the self-localization and the distance with the neighbors. Another approach has been bioinspiration of animals such as ants or the case of swarm bees presented in [3,4], since it may be the case to have a swarm of robots working on the same goal, avoiding collisions and seeking for points of interest how is shown in [5]. Similarly in [6] explore the idea of the implementation of heterogeneous robot network for search and rescue operations. On the other hand, the use of robots' teams have been highly researched and implemented in the case of decentralized navigation and in exploration and mapping tasks, these ideas have been exploited in [7,8] respectively.

The formation control problem has been widely studied in the recent years as a useful tool for the control of multiple homogeneous and heterogeneous robots during the execution of multiple missions such as autonomous formation flight presented in [9]. They consider autonomous aircrafts that can present communication and transmission failure among them and the needed of redundancy to accommodate it. In the same way, other approach was developed in [10] where they used leader and follower formation control on under-actuated autonomous underwater vehicles, it is worth to mention that underwater vehicles have communication range problems taken in to account in this work.

In this work, we present a fuzzy logic based approach oriented to the leader-follower formation control of a swarm of homogeneous robots that allows the swarm to avoid collision between robots and keep the formation based on the leader movements. The other sections that complete this paper are organized as follows: Sect. 3 Swarm Navigation, Sect. 4 Experiments and Results, Sect. 5 Conclusions.

2 Related Work

In the recent years, there have been multiple approaches to the formation control problem from different perspectives. Some of the most representatives are the behavior-based, the artificial potential based and the leader-follower based among others. The behavior-based formation control strategies impose multiple high-level instructions or desired behaviors on the robot's team that allow them to fulfill the mission requirements. In particular, in the work presented in [11] there are presented behavior-based strategies that allow a robot team to navigate through environments with multiple obstacles preserving some basic geometric formations implementing the control signals based on 4 high-level behaviors: avoid static obstacles, avoid robots, maintain formation and move to goal. With those considerations, a robot team is capable of navigating through environments with obstacles without colliding restoring the formation once the obstacle has been sorted.

Alternatively, the artificial potential formation control based strategies employ certain functions called artificial or virtual potentials that in a similar way as the effect of an electromagnetic field or a spring, cause that the robots move according to the desired behaviors. In particular, a repulsive artificial potential may be employed for the robots to avoid collisions. For instance, in the work presented by [12] there is artificial potential based control laws that allow a robots team to maintain formations employing only local information. This distributed approach minimizes the individual artificial potential function of each robot and guarantees the connectivity maintenance and the rigidity of the formation.

On the other hand, regarding the leader-follower formation control based strategies, those employ two types of agents, the leaders who are in charge of determining the formation parameters (position, scale, and rotation) and the followers that maintaining certain distances to their neighbors and to the leaders achieve the desired behaviors. It is worth to mention that there are approaches that do not consider the leader agent as one of the actual robots, but a virtual agent that is tracked by all the roots of the team. For instance, in the work presented in [13] there is presented a leader-follower formation control algorithm where all the robots on the team define a virtual leader that is only a virtual artifact of the optimization process that allows each of the robots to move referenced to it achieving the desired behaviors.

Another example of the leader-follower approach is presented in [14] where it is introduced the idea of combining the Complex-Laplacian matrix with the mutator replicator dynamics. In this work, there are two agents that are the leaders and the rest of them are considered followers. The formation is rigidly defined by the position of the two robot leaders and the position in the graph of each one of the robots is decided through an agreement protocol that employs the mutator-replicator dynamics. Once the positions are decided, the Complex-Laplacian matrix is employed to achieve the appropriate position of each robot on the formation completing the task.

Although most of the formation control algorithms are based on traditional logic principles, there are also approaches that employ the fuzzy logic ideas in order to obtain flexible behaviors with uncertainty and/or with different desired operation regions. In [15] it is presented an exploration and monitoring scheme, where employing fuzzy operators with multiple inputs and outputs, it is possible to control a team of robots to maintain a formation while they are avoiding collisions. In this work, in particular, the robot's model has an important consideration that is that their sensing range does not cover the whole 360 degrees around the robot but only 240 degrees increasing the complexity of the analysis.

3 Swarm Navigation

The principal challenge of the present work is to propose an algorithm to guide a swarm of robots. The robot swarm has to follow a leading robot without colliding with the other robots, but not too far from the nearest robot. To accomplish this requirement, it is proposed a system based on fuzzy logic and this algorithm is simulated using MATLAB and V-Rep for drone robot simulations.

3.1 Initial Variables

Three input variables are taken into account: the number of robots without including the leading robot (Nr), optimal distance to the nearest robot (Lr) and optimal distance to the leader (Ll). The simulation tests were performed with different values for each of input variables.

3.2 Fuzzy System

A fuzzy system is proposed with the aim to accomplish the requirements of the challenge. The robots have to navigate following a leader and at the same time, it has to avoid colliding with other robots of the same swarm. The proposed system has two fuzzy subsystems using the Mamdani approach. The complete system has two outputs, one that determines the velocity towards the closest robot and the second output determines the velocity towards the leader. Also, each fuzzy system has two inputs: the first one is the distance from the robot to the leader and the second one is the distance from the robot to the nearest robot, so two fuzzy subsystems were defined for output 1, output 2 respectively. Figure 1 shows the schema of the complete system, including a vectorial adder of the outputs for determining the final velocity and direction of the reference robot.

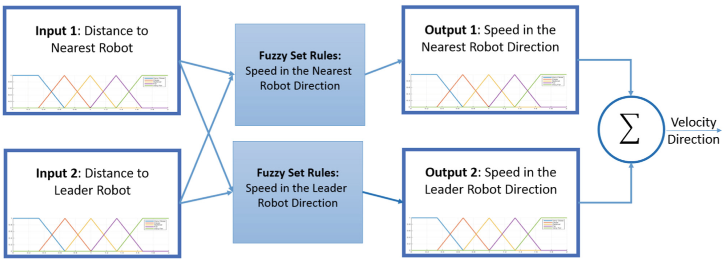

Fig. 1. Outputs of the fuzzy system

Inputs: As was explained previously, the fuzzy system has two inputs, first one depends on the distance to the nearest robot and the second one is the distance to the Leader. Both inputs are composed of five fuzzy set defined as Very Close, Close, Optimal, Far, and Very Far. These fuzzy sets are defined by triangular fuzzy function for the three central sets and two trapezoidal functions for Very Close and Very Far sets as shown in Fig. 2. The universe of discourse for inputs is defined by the optimal distance to the leader robot (Ll) and the nearest robot (Lr). For these, the optimal value is located in 1, the closest distance is zero, and farther distance is 2.

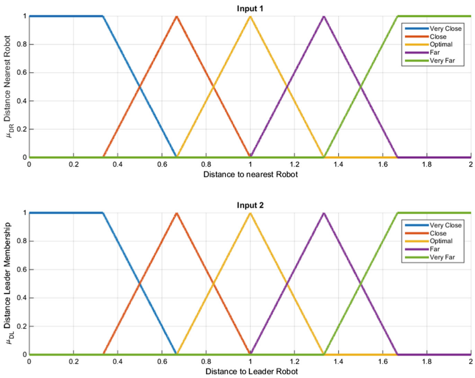

Fig. 2. Inputs of the fuzzy system

Outputs: The fuzzy system has two outputs to determine the displacement velocity toward the nearest robot and the leader respectively. Both outputs have five fuzzy set defined as Moving Far Faster (MFF), Moving Far (MF), Stop (S), Moving Close (MC), and Moving Close Faster (MCF). These set are represented by trapezoidal functions except the "Stop" fuzzy set who uses a triangular function as shown by Fig. 3.

Distance Calculation: The robot that is executing the action of movement is taken as reference robot. All distances between each swarm robot and the reference robot are calculated. The smallest distance is taken as distance to the nearest robot (Lr). This distance is the first input of the fuzzy system. The distance between the reference robot and the lead robot (Ll) is also calculated since this is the second input of the fuzzy system. The aforementioned distances calculation is performed for each of the robots that make up the swarm. The calculation of the distance is made using the formula 1 of distance in the Cartesian plane.

$$Ll, Lr = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

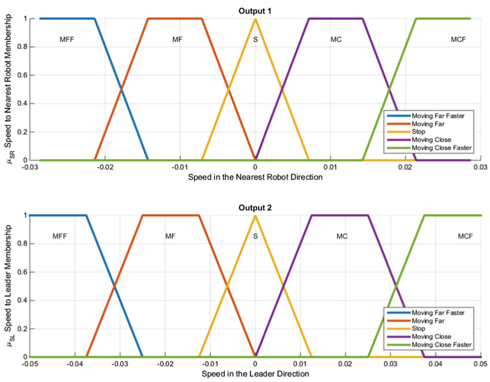

Fig. 3. Outputs of the fuzzy system (Color figure online)

Fuzzy system rules: Given the system has two outputs, there are two sets of decision rules. The rules of each set calculate the minimum product between the membership of two sets, each set belongs to a different system input. The output is calculated by the rule belongs to one of the sets that belong to the system output. The general form of the rule is shown in the Eq. 2

$$\textbf{If}(Input(1) \text{ is } mf(i)) \textbf{ and } (Input(2) \text{ is } mf(j)) \textbf{ then } Output \text{ is } mf(k) \quad (2)$$

where i and j are the subindex for the membership function of the input 1 and 2 respectively, and k is the subindex of the fuzzy set output that is being evaluated.

The fuzzy set rules for output 1 and 2 are depicted in Tables 1 and 2 respectively.

Table 1. Set of Rules: Speed in the nearest robot direction

		Distance to nearest robot				
		Very Close	Close	Optimal	Far	Very Far
Distance to Leader	Very Close	MFF	MF	S	MC	MCF
	Close	MFF	S	S	S	MCF
	Optimal	MFF	S	S	S	MCF
	Far	MFF	S	S	S	MCF
	Very Far	MFF	MF	S	MC	MCF

Finally, to generate the output data, the defuzzification process is required. This method calculates a single scalar quantity from each fuzzy output set. This work used the bisector method for defuzzification process.

Table 2. Set of Rules: Speed in the leader direction

		Distance to nearest robot				
		Very Close	Close	Optimal	Far	Very Far
Distance to Leader	Very Close	MFF	MFF	MFF	MFF	MFF
	Close	MF	S	S	MF	MF
	Optimal	S	S	S	S	S
	Far	MC	MC	MC	MC	MC
	Very Far	MCF	MCF	MCF	MCF	MCF

Final calculation of the speed: The fuzzy system output was described in the previous section. It represents the magnitude of the movement towards the nearest robot and towards the leader since they are two fuzzy systems for two outputs. If the magnitude is negative, it means that the robot must move away from either the nearest robot or the leader respectively. The two outputs are vectorial added, taking into account the angle with respect to the leader and the nearby robot, to generate the vector of movement that points to the direction in which the robot must move. Finally, the new position of the robots is updated according to the vectorial addition of the fuzzy system outputs.

4 Experiments and Results

Several experiments were performed with the aim to test the proposed navigation system based on fuzzy logic approach.

4.1 Initial Robot Position

Since the robots are not initially organized in a specific formation. The robotic agents were initially located randomly within the navigation space. The algorithm does not depend on the initial position of the robots and instead try to organize robots through the navigation process.

4.2 Experiments

Five experiments were performed, where the three first show the behavior of the simulated swarm using MATLAB. These three experiments try to show how the robots are grouped around the leader and how the robots follow the leader when it moves to a different place. In these experiments, the leader robot is a red star and swarm robots are depicted by blue circles. The last two experiments were performed using VRep (Robotics simulation software). These two experiments depicted the self-organization of drones around the leader using the proposed algorithm, and how the drone swarm navigates following the leader when it is moving.

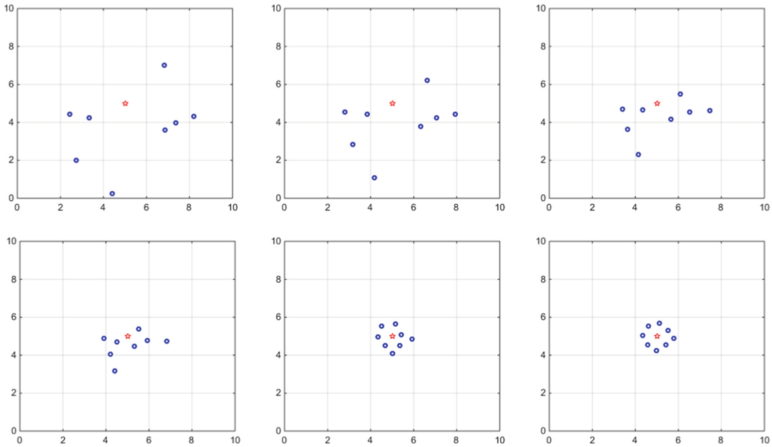

Fig. 4. Experiment 1 (Color figure online)

Experiment 1: This experiment was performed using 8 robots. These robots where localized randomly in a universe of 10X10 units of area. Figure 1 shows the sequence about how robots navigate from the original random position to the nearest point close to the leader. At the end of Fig. 1 is shown how the robot swarm is organized surrounding the leader.

Experiment 2: This experiment uses 40 robots initially localized randomly. The complete sequence shows how the swarm navigates through the field approaching the leader. Once robots have arrived at the nearest point to the leader, it shows how the swarm is organized in layers around the leader and keeping the distance among the robot swarm partners.

Experiment 3: This experiment shows the swarm following the leader robot. The experiment was performed using 15 agents and they were localized randomly at the beginning of the simulation. The sequence shows how the robots navigate close the leader and when the leader starts to displace to another position all the robots navigate following him and keeping the formation established initially. The trace of every robot is shown in green lines in Fig. 3. As was explained initially the leader robot id red and the swarm robots are blue.

Experiment 4: This experiment tests the proposed algorithm using VRep Education Edition. This software allows the simulation of real robots as Drones, humanoids, and several mobile robots. Figure 4 shows the drone swarm navigating towards the drone leader. The initial position of every drone was assigned randomly. At the end of the sequence is shown the final position of the drone swarm, it is showing how the drones are surrounding the leader and keeping the distance to the leader and the distance among other drones.

Experiment 5: This is the last experiment and it uses 15 drones to evaluate the navigation of the drone swarm following the leader. The initial position of every drone was defined randomly as previous experiments. Figure 5 shows the navigation sequence where the drones approach the leader and follow him

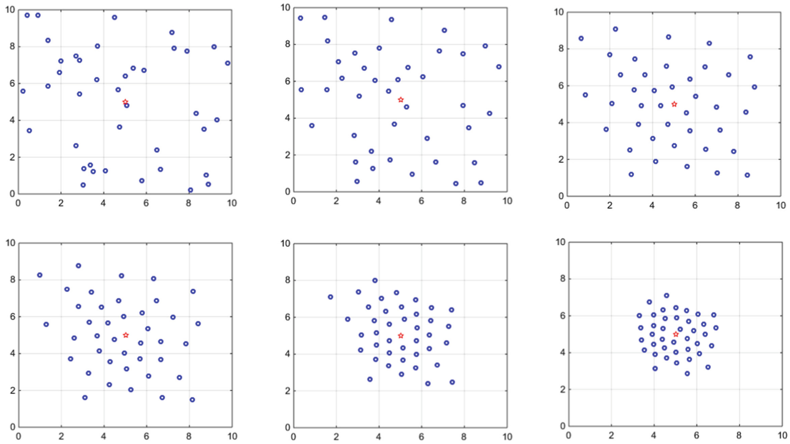

Fig. 5. Experiment 2

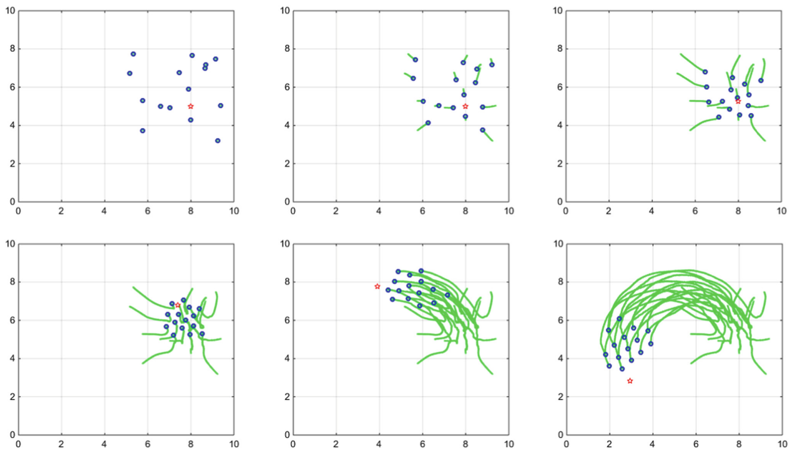

Fig. 6. Experiment 3

through the free navigation, just following him. The simulation shows the swarm keeping the formation and the distance among all the swarm robots and avoiding collide with other members of the swarm while they are navigating following the leader (Fig. 6).

5 Conclusions

The present work shows the use of fuzzy logic theory for the generation of leader-follower behavior in a robot swarm (Fig. 7).

The used fuzzy system is based on the Mamdani model. This system is in charge of calculate the velocity and movement direction of every robot in the

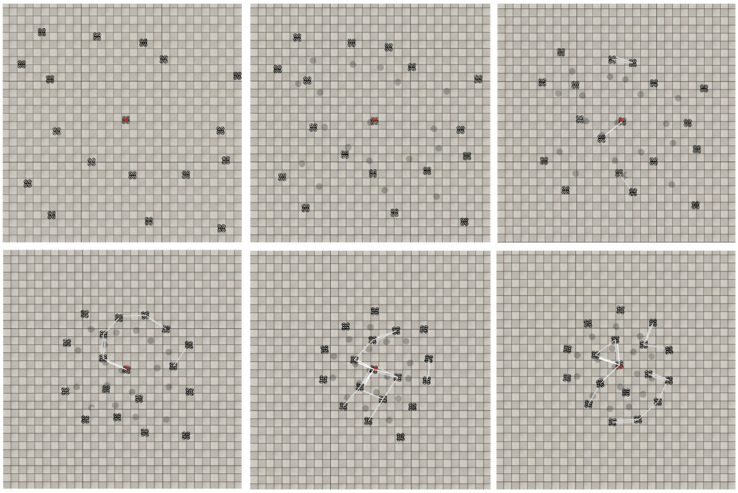

Fig. 7. Experiment 4

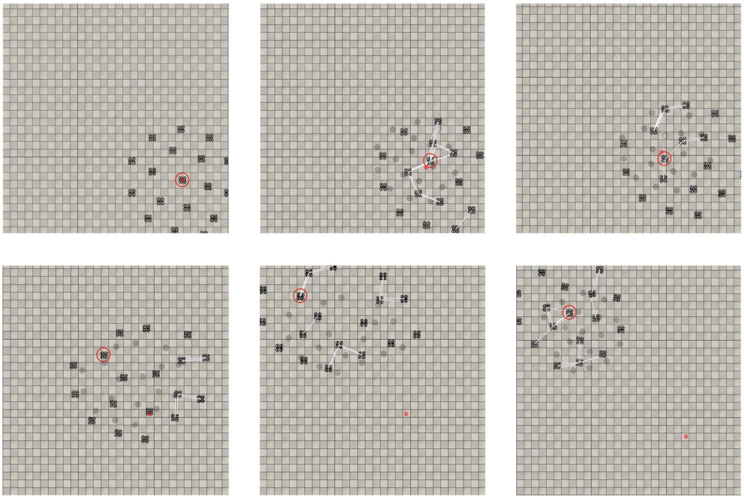

Fig. 8. Experiment 5

swarm. The fuzzy system uses as input two different distance approaches. The first one is the distance between evaluated agent and each other swarm agent.; the second one is the distance between evaluated agent and the leader. Using these two distance approaches, the fuzzy system determines the navigation direction of the evaluated agent (Fig. 8).

The robot swarm presents two very interesting behaviors. The first behavior appears when the leader is static, the agents are grouped around the leader, leaving the leader in the center of the swarm. The second behavior occurs when the leader moves from one place to another, in this behavior the swarm follows the leader towards the new location. Once the leader arrives at the new location, all the swarm agents localize themselves around the leader.

In both behaviors, the swarm agents move according to the leader's location and at the same time, they avoid colliding with the other agents of the swarm. These behaviors were observed through several simulations carried out in Matlab. In addition, the same fuzzy system was programmed in a swarm of 17 drones using the robot simulation software called VRep. The simulations of the drone swarm depicted same behaviors that were observed in the Matlab simulations. In conclusion, this work shows how it is possible to generate navigation systems for swarm robots based on fuzzy logic and at the same time get behaviors from systems inspired by animal swarms such as bees.

Future work is based on the implementation and evaluation of the proposed system on real robot platforms and studies of some navigation features such as generated trajectories, traveled area, and efficiency in the evasion of obstacles.

References

1. Floreano, D., Wood, R.J.: Science, technology and the future of small autonomous drones. Nature **521**(7553), 460–466 (2015)
2. Martinelli, A., Pont, F., Siegwart, R.: Multi-robot localization using relative observations. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation 2005, ICRA 2005, pp. 2797–2802. IEEE, April 2005
3. Quijano, N., Passino, K.M.: Honey bee social foraging algorithms for resource allocation, part I: algorithm and theory. In: American Control Conference 2007, ACC 2007, pp. 3383–3388. IEEE, July 2007
4. Quijano, N., Passino, K.M.: Honey bee social foraging algorithms for resource allocation, part II: application. In: American Control Conference 2007, ACC 2007, pp. 3389–3394. IEEE, July 2007
5. León, J., Cardona, G.A., Botello, A., Calderón, J.M.: Robot swarms theory applicable to seek and rescue operation. In: Madureira, A.M., Abraham, A., Gamboa, D., Novais, P. (eds.) ISDA 2016. AISC, vol. 557, pp. 1061–1070. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-53480-0_104
6. Yanguas-Rojas, D., Cardona, G.A., Ramirez-Rugeles, J., Mojica-Nava, E.: Victims search, identification, and evacuation with heterogeneous robot networks for search and rescue. In: 2017 IEEE 3rd Colombian Conference on Automatic Control (CCAC), Cartagena, pp. 1–6 (2017). https://doi.org/10.1109/CCAC.2017.8276486
7. Tanner, H.G., Kumar, A.: Towards decentralization of multi-robot navigation functions. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation 2005, ICRA 2005, pp. 4132–4137. IEEE, April 2005
8. Simmons, R., Apfelbaum, D., Burgard, W., Fox, D., Moors, M., Thrun, S., Younes, H.: Coordination for multi-robot exploration and mapping. In: AAAI/IAAI, pp. 852–858, July 2000
9. Giulietti, F., Pollini, L., Innocenti, M.: Autonomous formation flight. IEEE Control Syst. **20**(6), 34–44 (2000)
10. Cui, R., Ge, S.S., How, B.V.E., Choo, Y.S.: Leaderfollower formation control of underactuated autonomous underwater vehicles. Ocean Eng. **37**(17), 1491–1502 (2010)

11. Balch, T., Arkin, R.C.: Behavior-based formation control for multirobot teams. IEEE Trans. Robot. Autom. **14**(6), 926–939 (1998)
12. Drfler, F., Francis, B.: Formation control of autonomous robots based on cooperative behavior. In: 2009 European Control Conference (ECC), pp. 2432–2437. IEEE, August 2009
13. Droge, G.: Distributed virtual leader moving formation control using behavior-based MPC. In: American Control Conference (ACC), pp. 2323–2328. IEEE, July 2015
14. Guzman-Hernandez, S., Mojica-Nava, E.: Formation control using replicatormutator dynamics for multiple mobile autonomous agents. Neurocomputing **172**, 337–344 (2016)
15. Yasuda, Y., Kubota, N.: Formation behavior of multi-robot for exploration and monitoring. In: 2012 Joint 6th International Conference on Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS), pp. 1232–1237. IEEE, November 2012