Flat Tension Functions and Minimally Rigid Graphs for Tasks of Synchronization and Control of Multi-Agent Robotic Systems

Andrea Maybell Peña, Miguel Zea, *Member, IEEE*, and Luis Alberto Rivera, *Member, IEEE*
Department of Electronics, Mechatronics and Biomedical Engineering
Universidad Del Valle de Guatemala
Guatemala, Guatemala
{pen15127, mezea, larivera}@uvg.edu.gt

***Abstract*—This work focuses on the development of a synchronization and formation control method for multi-agent systems. The method is based on a modified consensus equation, the implementation of minimally rigid graphs and the use of flat tension functions. The goal is to bring a group of agents to a certain formation, and then lead them to a common goal, while avoiding obstacles. The use of minimally rigid graphs makes obstacle avoidance easier, compared to using totally rigid graphs, given the looser constraints for the swarms of robots. Using flat tension functions helps dealing with issues that minimally rigid graphs may present due to the lack of edges in the graphs. As a consequence, agents will keep the formation as closely as possible in the absence of obstacles, and they will adjust their positions and even switch formation if needed, when obstacles are present.**

***Index Terms*—Multi-agent systems, swarm robotics, formation control, consensus equation, flat tension functions**

I. Introduction

The study of multi-agent systems is a branch of robotics and computer systems that has gained a lot of attention over the past couple of decades due to its extensive applications [1]. This work focuses on a very specific application of swarm robotics: synchronization and formation control. The main goal is to develop a control method for multi-agent robotic systems which could be used in search missions. The method is based on a modified consensus equation, the implementation of minimally rigid graphs, and the use of flat tension functions to deal with problems that arise in the absence of edges between nodes in the graphs. The method consists of two parts: formation success without collisions, and obstacle avoidance through formation switch. A leader-follower technique is implemented to mobilize the swarm. The control applied is based on a centralized, event-triggered strategy. The efficiency of the method was verified in a controlled simulation considering physical dimensions and constrains of the agents.

The paper is organized as follows: Background is presented in Section II. Section III provides details of our proposed methods. In Section IV, the experiments and results are presented. Conclusions are given in Section V.

II. Background

A. Multi-Agent Control

One problem of extensive interest for multi-robotic systems is formation shape control. It is known that by the use of rigid graph theory it is possible to achieve a formation by controlling certain inter-agent distances. Sun *et al*. [2] discuss some generalized controllers found in the literature for rigid formation stabilization. Zhang *et al*. [3] present a three dimensional dynamic formation control using rigid graphs. They reduce the computational cost by not relying on a Global Positioning System, similar to what is done in [4]. Coppola and de Croon [5] present a way to optimize the behavior of a swarm in a pattern formation task using local behavior, expressed as a local state-action map. Cai and Queiroz [6] implement rigid graph theory to introduce a rigidity-based adaptive formation control law for multiple robotic vehicles. In [7], Olfati and Murray provide a graph theoretical framework to formally define formations of multiple vehicles. They propose creating larger rigid-by-construction graphs by combining smaller rigid sub-graphs.

Several techniques for implementing multi-agent control have been developed. These differ in terms of the sensors and actuators used by the agents, as well as the way the agents communicate with each other [8]. Four techniques can be distinguished for mobilization and communication: leader-follower, behavior based control, virtual structure techniques and control theory based techniques [9]. Control strategies can also be classified as centralized or decentralized [10]. A centralized strategy implies that there is a control entity supervising all robots [11]. A decentralized approach has no supervisor, only relative positions of each agent with respect to its neighbors are fed back [8]. Computational Intelligence methods have been proposed for multi-agent control systems, such as the Neuro-Evolutionary learning method proposed by Tuzel *et al*. [12], which allows to train leaders and influence swarm behaviours.

Energy consumption is an important aspect to take into consideration for any kind of robotic system. It is desired to have a low energy consumption while still guaranteeing

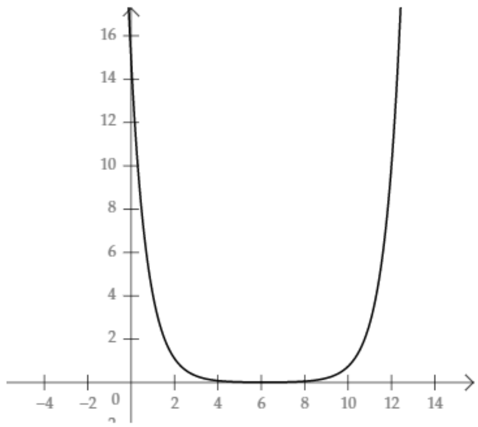


Fig. 1: Hyperbolic cosine function

proper control performance. One approach that has been proposed is the event-triggered control strategy. Its controller actuation happens only at some specific instants determined by predefined event conditions [13].

B. Flat Functions

A flat function [14]–[16] is a smooth function $f : \mathbb{R} \to \mathbb{R}$ whose derivatives are all zero at a given point x_0:

$$f^{(k)}(x_0) = 0, \quad \forall k \geq 0 \quad \text{but} \quad f \neq 0 \tag{1}$$

Among flat functions, the most well-known might be the one shown in Equation (2) [16]:

$$f(0) = 0 \quad \text{and} \quad f(x) = e^{-\frac{1}{x}} \quad \text{for} \quad x > 0 \tag{2}$$

This function appears, for example, in the Arrhenius Equation, which establishes the connection between the rate of a reaction constant k and the absolute temperature T of a particular chemical reaction [14].

The hyperbolic cosine is another example of a flat function. Figure 1 shows the graph of the function described in Equation (3).

$$f(x) = 0.01 \cosh(1.3x - 8) \tag{3}$$

C. Minimally Rigid Graphs

For a two-dimensional graph to be defined as minimally rigid, the graph must have a state such that if any edge is removed, the graph is no longer rigid. If a graph meets the condition $e = 2v - 3$ where, e is the number of edges and v is the number of vertices, it is said that the graph is minimally rigid [17]. If a graph has a smaller number of edges, it is no longer considered rigid, if it has more, the graph remains rigid but is no longer minimally rigid. A complete graph is a totally rigid graph.

III. Methods

We first wanted to ensure that the agents, starting at random positions, would reach a consensus. That is, they should be able to come close to each other, and do so without colliding. To do that, a collision avoidance control is applied (Equation (4)), where $N(i)$ is the set of adjacent agents, or neighbors, of the i^{th} agent in the network, $\dot{x_i}$ is the velocity of the agent, and r is the radius of the agents (assumed circular in this work).

$$\dot{x_i} = -\sum_{j \in N(i)} \frac{(\|x_i - x_j\| - 2r)(x_i - x_j)}{(\|x_i - x_j\| - r)^2} \tag{4}$$

Once all agents are close enough without colliding they change to a model that includes formation control (Equation (5)), where the weight ω_{ij} is the derivative of a tension function between agent i and agent j.

$$\dot{x_i} = -\sum_{j \in N(i)} \omega_{ij} \left(\|x_i - x_j\|\right)(x_i - x_j) \tag{5}$$

To find the tension, rational functions based on those proposed in [18] were tested. These functions take into account agents' dimensions and desired distances between agents, which depend on the formation graph chosen. To determine the best model, we considered the Mean Squared Error (Equation (6)) and the energy required by the system (Equation (7)) as performance metrics. n is the number of agents in the system, x_{ij} is the distance between agents i and j, and d_{ij} is the desired distance between those agents.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} (x_{ij} - d_{ij})^2 \tag{6}$$

$$E = \sum_{i=1}^{n} \int_{t_i}^{t_f} \|\dot{x_i}\|^2 \, dt \tag{7}$$

In our experiments, the best results were obtained using the weights ω_{ij} shown in Equation (8). Note that the corresponding tension considers both collision avoidance and formation control.

$$\omega_{ij} = \frac{4(\|x_i - x_j\| - d_{ij})(\|x_i - x_j\| - r) - 2(\|x_i - x_j\| - d_{ij})^2}{(\|x_i - x_j\| - r)^2(\|x_i - x_j\|)} \tag{8}$$

The use of totally rigid graphs imposes excessive restrictions, making it difficult for swarms of robots to avoid obstacles as they move towards a goal. That is why we use minimally rigid graphs. However, some complications may arise [19]. Having a zero at any a_{ij} position of the adjacency matrix of a minimally rigid graph could mean two different things. Either the distance between agents i and j is zero, or there is no edge connecting those agents. Given that we consider agents with $r > 0$, the first meaning applies only to positions a_{ii}. In the absence of an edge between agents i and j, there is no information about the distance between the agents. Typical consensus functions such as the ones described above have a single, well-defined minimum, and agents seek to reach that minimum. When applying the formation control using the adjacency matrix to define the distances between the agents, the zeros corresponding to edge absences have the effect of reducing the distance between unconnected agents, even though the collision avoidance control tries to keep the distance.

To prevent this problem, we propose using a flat tension function for unconnected agents. Thus, the agents will not be forced to come together, and the formation control will decide where the agents should position themselves. We used a hyperbolic cosine as our flat tension function (Fig. 1). This function has a wide "minimum" region, where there is little difference in the cost values. Thus, even if the formation control places the agents with different distances between each other, the cost will remain practically the same.

Using the function defined in Equation (3), the resulting weights are as shown in Equation (9). Note that the weights of the consensus equation in Equation (5) remain as in Equation (8) for connected agents. The parameters α and β can be adjusted as needed.

$$\omega_{ij} = \begin{cases} \alpha \times \frac{\sinh(100\alpha\|x_i - x_j\| - \beta)}{\|x_i - x_j\|}, & d_{ij} = 0 \\ \frac{4(\|x_i-x_j\|-d_{ij})(\|x_i-x_j\|-r)-2(\|x_i-x_j\|-d_{ij})^2}{(\|x_i-x_j\|-r)^2(\|x_i-x_j\|)}, & d_{ij} \neq 0 \end{cases} \tag{9}$$

Our proposed method also considers obstacle avoidance, for which an independent control is needed. We can add the term shown in Equation (10) to the control law given by Equation (5), so that the velocity of each agent is affected by the distance to the obstacles as well. d_{ik} is the distance between agent i and obstacle k, and γ is a parameter that can be adjusted.

$$\omega_{ik} = \frac{-\gamma}{(d_{ik} - 1)^2} \tag{10}$$

The last element of our method is formation switching. The agents are able switch formation, if needed, to make it easier for the swarm to avoid obstacles and reach the goal. For that, the error between the current position of the agents and the desired position in all the possible configurations must be measured. The one with the smallest error will be the graph chosen, and the agents will reconfigure themselves [18].

IV. Experiments and Results

Figure 2 shows the two formation graphs chosen for our experiments. We define success in reaching the desired formation as having a MSE smaller than 0.07 for the final formation. This threshold was chosen since it was the MSE with which the formation was still visually successful, that is, it was similar enough to the test graph. Furthermore, we consider the complete method to be successful if the swarm is able to reach the goal. This implies that the agents are able to reach the formation and avoid the obstacles in the environment to get to the destination.

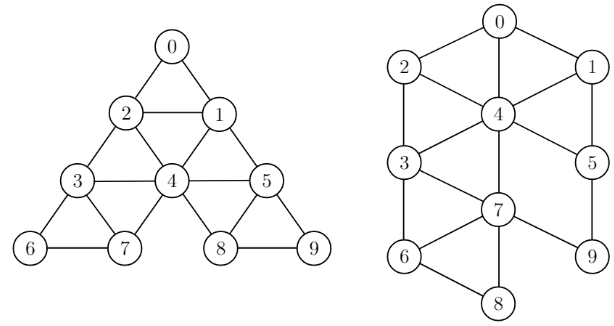


Fig. 2: Minimally rigid formation graphs used in our experiments.

We first ran simulations in Matlab to test our method and to determine appropriate parameters. The percentages of success in formation by varying α and β, are shown in Figures 3 and 4. The range of values of these parameters were chosen so that the resulting flat function had a wide flat region at the desired range. We found that the parameters that worked best for Equation (9) were $\alpha = 0.013$ and $\beta = 8.0$, and the best parameter for Equation (10) was $\gamma = 0.01$.

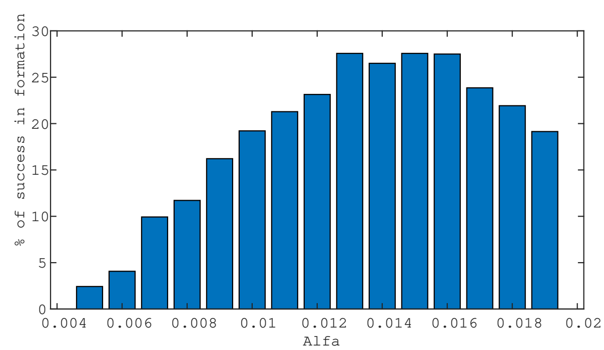


Fig. 3: Results of varying α.

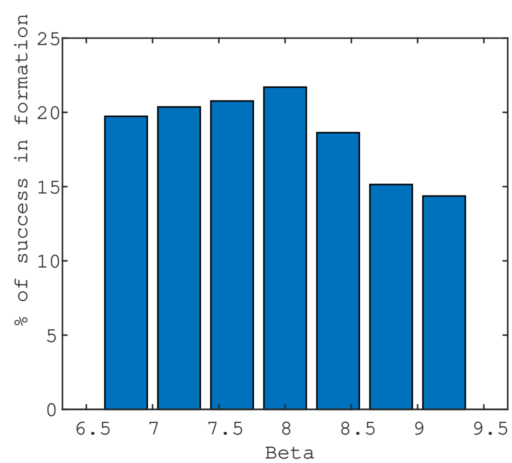


Fig. 4: Results of varying β.

After choosing the appropriate parameters, we ran hundreds of simulations in the Webots simulation environment [20]. Webots allows modelling the physical parameters of the agents. We chose E-Puck robots for our experiments. We tested two different obstacle configurations, and the two formation configurations shown in Figure 2. We ran experiments using totally rigid graphs, and our proposed method with minimally rigid graphs and flat tension functions.

Table I shows that in most cases in which totally rigid graphs were used, the agents achieved the initial formation, but they were not successful in reaching the goal. Having a formation control with too many restrictions did not allow the swarm to deform and therefore could not be reconfigured. On the other hand, when using minimally rigid graphs, most cases were successful. In 82% of the experiments, the agents converged to the formation and reached their goal. Only in 5% of the tests, the agents were unable to reach the goal, even though they converged to the formation. These results show that allowing slight deformations in the configuration to admit new formations greatly improves the chances of avoiding obstacles. Figures 5 and 6 illustrate the difference between the two rigidity conditions.

TABLE I: Successful arrivals at the goal

Obstacle Config.	Rigidity	Formation Failure	Goal	No Goal
1	Total	0	58	42
1	Minimal	14	86	0
2	Total	0	50	40
2	Minimal	12	78	10

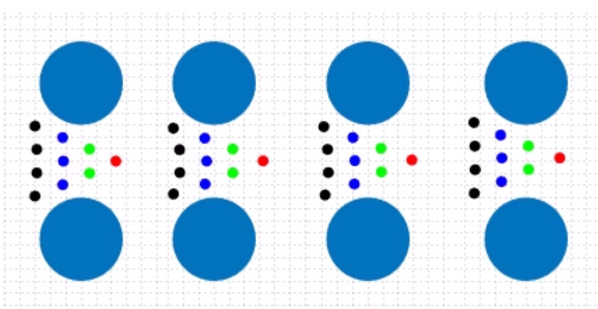


Fig. 5: Using totally rigid graphs.

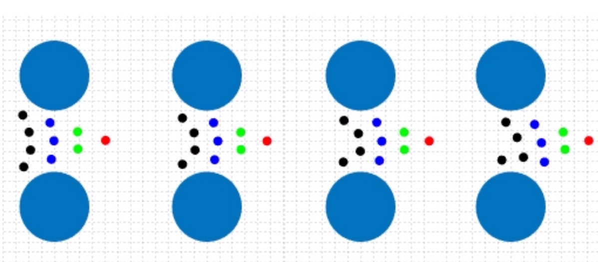


Fig. 6: Using minimally rigid graphs.

Examples of the trajectories followed by the agents in successful runs, both in Matlab and in Webots simulations, are shown in Figures 7 - 11.

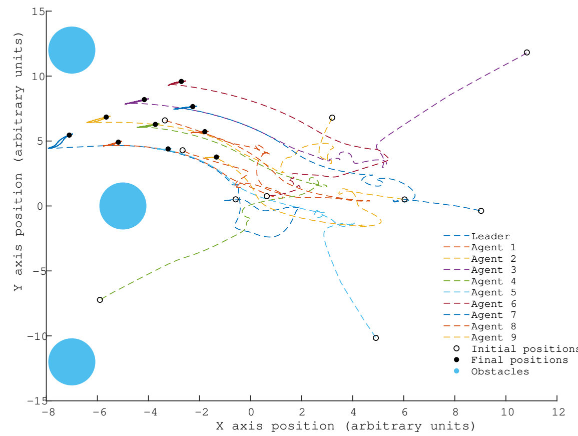


Fig. 7: Formation control using a totally rigid graph. Matlab simulation.

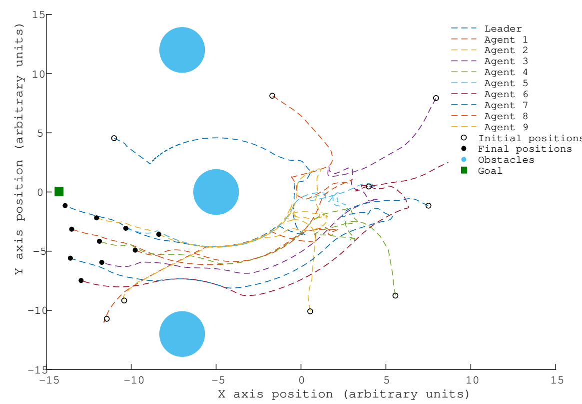


Fig. 8: First obstacle configuration. Matlab simulation.

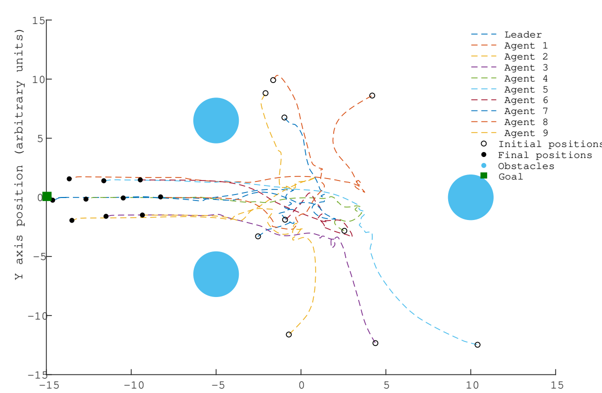


Fig. 9: Second obstacle configuration. Matlab simulation.

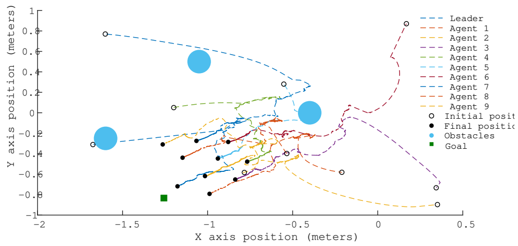


Fig. 10: Successful initial formation that undergoes deformation, with 3 formation switches. Webots simulation.

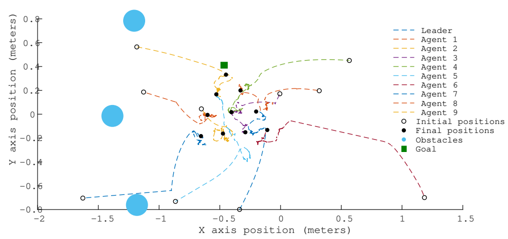
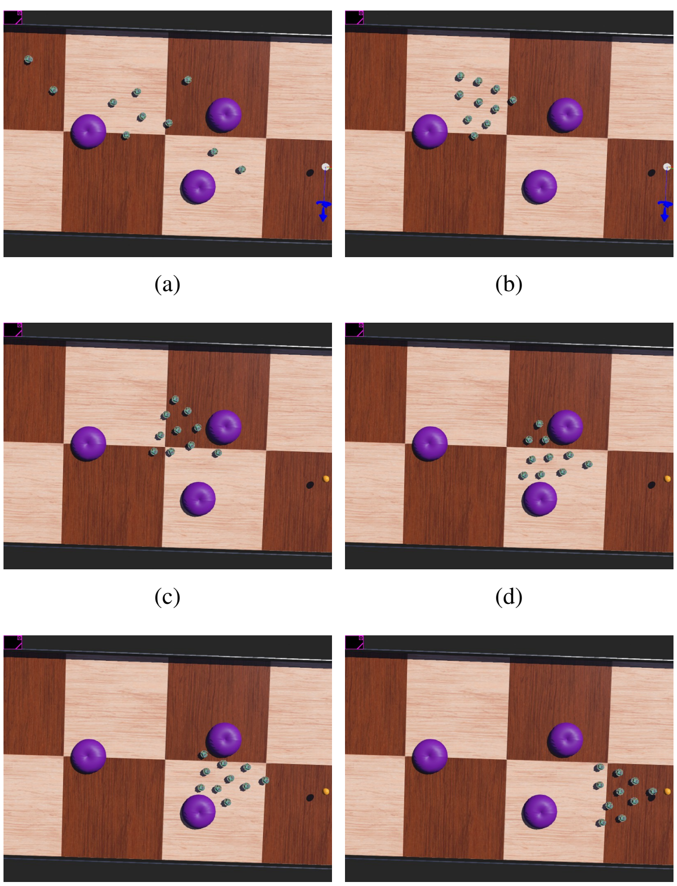


Fig. 11: Successful initial formation, no formation switches. Webots simulation.

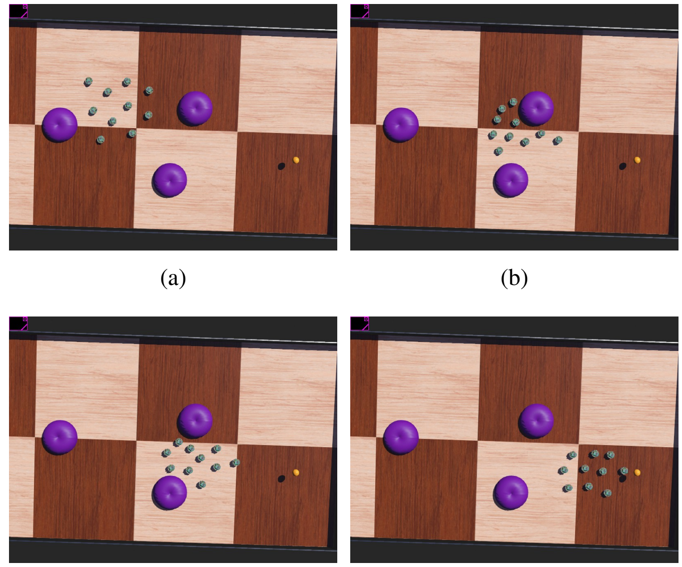


(a) (b)

(c) (d)

(e) (f)

Fig. 12: Example Webots run. The E-puck robots start at random positions (a), come to an initial formation (b), and move towards the goal (c) - (f).



(a) (b)

(c) (d)

Fig. 13: Example Webots run showing a formation switch.

Figures 12 and 13 show screenshots of the E-Puck robots in the Webots environment. Figure 12 (a) shows the E-Puck robots at the beginning of the simulations, scattered over the environment. In (b), the robots have come to the initial formation. In (c), the swarm is moving towards the goal and comes across an obstacle. In (d) and (e), the formation is deformed, which allows the swarm to avoid the obstacles. Finally, in (f), the swarm has reached the goal, and it is back in the triangular formation. Figure 13 shows a case in which there was a formation switch. As explained before, switching formations can make it easier for the swarm to avoid obstacles and reach the goal. The switch is triggered when the overall position error for the current formation becomes larger than the error for another formation.

V. Conclusions

This work contributes to research in the field of swarm robotics, specifically in the topics of synchronization and formation control. A combination of formation control and collision avoidance control as a single rational function allows agents to reach formations with a minimum mean square error using totally rigid graphs. However, the strict constraints in those graphs may reduce the success of the system in reaching a goal, mainly because of obstacle avoidance failures. By using minimally rigid graphs we can achieve better results, given the looser constraints for the swarms of robots. Nevertheless, problems may arise due to the lack of edges in the graphs. Our main contribution is the use of flat tension functions to overcome those problems. Instead of simply coming together, agents not connected in the graph can be adjusted to keep the desired formation as closely as possible. Our method also allows switching formations, which further helps the swarm in avoiding obstacles. The simulations ran in the Webots environment, considering physical constraints of the robots, show that the method could be viable in a real-world scenario.

References

[1] B. Khaldi and F. Cherif, "An overview of swarm robotics: Swarm intelligence applied to multi-robotics," *International Journal of Computer Applications*, vol. 126, no. 2, 2015.

[2] Z. Sun, Q. Liu, C. Yu, and B. D. Anderson, "Generalized controllers for rigid formation stabilization with application to event-based controller design," in *2015 European Control Conference (ECC)*. IEEE, 2015, pp. 217–222.

[3] P. Zhang, M. de Queiroz, and X. Cai, "Three-dimensional dynamic formation control of multi-agent systems using rigid graphs," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 11, p. 111006, 2015.

[4] J. C. Barca, A. Sekercioglu, and A. Ford, "Controlling formations of robots with graph theory," in *Intelligent Autonomous Systems 12*. Springer, 2013, pp. 563–574.

[5] M. Coppola and G. C. H. E. de Croon, "Optimization of swarm behavior assisted by an automatic local proof for a pattern formation task," in *Swarm Intelligence*, M. Dorigo, M. Birattari, C. Blum, A. L. Christensen, A. Reina, and V. Trianni, Eds. Cham: Springer International Publishing, 2018, pp. 123–134.

[6] X. Cai and M. de Queiroz, "Rigidity-based stabilization of multi-agent formations," *Journal of Dynamic Systems, Measurement, and Control*, vol. 136, no. 1, p. 014502, 2014.

[7] R. Olfati-Saber and R. M. Murray, "Graph rigidity and distributed formation stabilization of multi-vehicle systems," in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, vol. 3, 2002, pp. 2965–2971 vol.3.

[8] F. Roberti, J. M. Toibero, R. F. Vassallo, and R. Carelli, "Control estable de formación basado en visión omnidireccional para robots móviles no holonómicos," *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 8, no. 1, pp. 29–37, jan 2011.

[9] L. Dong, Y. Chen, and X. Qu, "Formation control strategy for nonholonomic intelligent vehicles based on virtual structure and consensus approach," *Procedia Engineering*, vol. 137, pp. 415–424, 2016.

[10] D. Manolis, T. Pappa, C. Diakaki, I. Papamichail, and M. Papageorgiou, "Simulation study of centralized vs decentralized approaches to the signal control of large-scale urban networks," 01 2017.

[11] M. Trabattoni, G. Valentini, and M. Dorigo, "Hybrid control of swarms for resource selection," in *Swarm Intelligence*, M. Dorigo, M. Birattari, C. Blum, A. L. Christensen, A. Reina, and V. Trianni, Eds. Cham: Springer International Publishing, 2018, pp. 57–70.

[12] O. Tuzel, G. Marcon dos Santos, C. Fleming, and J. A. Adams, "Learning based leadership in swarm navigation," in *Swarm Intelligence*, M. Dorigo, M. Birattari, C. Blum, A. L. Christensen, A. Reina, and V. Trianni, Eds. Cham: Springer International Publishing, 2018, pp. 385–394.

[13] L. P. Jie Cao, Zhihai Wu, "Distributed event-triggered average consensus of multi-agent systems," *International Journal of Computer Science (IIJCS)*, vol. 3, no. 10, Oct. 2015.

[14] P. Glaister, "A "flat" function with some interesting properties and an application," *The Mathematical Gazette*, vol. 75, no. 474, p. 438, dec 1991.

[15] Y. Pan and M. Wang, "When is a function not flat?" *Journal of Mathematical Analysis and Applications*, vol. 340, no. 1, pp. 536–542, 2008.

[16] H. Kodama, K. Masuda, and Y. Mitsumatsu, "Derivatives of flat functions," May 2018.

[17] L. Krick, "Application to graph rigidity in formation control of multi-robot networks," Master's thesis, University of Toronto, 2007.

[18] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks (Princeton Series in Applied Mathematics)*. Princeton University Press, jul 2010. [Online]. Available: https://www.xarg.org/ref/a/0691140618/

[19] Z. Sun and B. D. O. Anderson, "Rigid formation control with prescribed orientation," in *2015 IEEE International Symposium on Intelligent Control (ISIC)*, Sep. 2015, pp. 639–645.

[20] W. Open Source Robot Simulator, "https://cyberbotics.com/." [Online]. Available: https://cyberbotics.com/