

Bacterial quorum sensing applied to the coordination of autonomous robot swarms

Fredy H. Martínez S., Fernando Martínez S., Holman Montiel A.
Universidad Distrital Francisco José de Caldas, Facultad Tecnológica, Colombia

Article Info

Article history:

Received Feb 20, 2019

Revised Mar 2, 2019

Accepted Aug 14, 2019

Keywords:

Bacteria
Path planning
Quorum sensing
Robots
Swarm

ABSTRACT

This paper proposes a strategy for the coordination of a swarm of robots in an unknown environment. The basic idea is to achieve the autonomous movement of the group from an initial region to a target region avoiding obstacles. We use a behavior model similar to bacterial Quorum Sensing (QS) as a technique for the coordination of robots. This behavior has been described as a key element in the interaction between bacteria, and we use it as a basic tool for local interaction, both between the robot and between the robot and the environment. The movement of the swarm of robots, or multi-agent robotic system, is shown as an emerging behavior resulting from the interaction of agents (in the context of artificial intelligence) from basic rules of behavior. The proposed strategy was successfully evaluated by simulation on a set of robots.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Fredy H. Martínez S.,
Universidad Distrital Francisco José de Caldas, Facultad Tecnológica,
Carrera 7 #40-53, Bogotá D.C., Colombia.
Email: fhmartinezs@udistrital.edu.co

1. INTRODUCTION

A group of robots, or robot swarm, working in a coordinated way, have some advantages for the development of tasks over a single robot [1-3]. As with uninformed search algorithms, such as genetic algorithms [4] and ant colony [5], a swarm of robots offers the possibility that each agent in the system (assuming that the group of robots behaves like a system) can observe specific characteristics of the task that other agents do not see, which can accelerate the time required for the development of the task, or even in some cases achieve it [6, 7].

The strength of a swarm lies in the ability of its individuals to share information [8, 9]. Each individual senses local environmental information, which would correspond to a state of the system [10]. A swarm of n individuals can simultaneously observe n different states of the problem. If each individual communicates and shares this information with other nearby individuals, then the swarm becomes a multi-agent system that processes information in parallel [11, 12].

There are many applications for this type of systems. A typical example is the exploration of unknown environments [13, 14]. In this kind of task each agent can explore a small part of the environment, and at the same time share with their colleagues, the information collected. The group of robots can map large environments very quickly and safely compared to the use of a single robot. The security we are talking about is related to the robustness of the swarm, that is, if one individual fails, others can still complete the task, this does not happen when using a single robot. These kinds of strategies seek to divide the problem into simpler and more limited sub-problems, which in fact can be solved with simpler and cheaper robots [15].

Another type of application of the swarm of robots is related to tasks that can only be carried out thanks to the joint and simultaneous work of all the agents in the system [16]. Another famous example of this type of application is the movement of the piano. A piano (or any other heavy element) turns out to be too heavy to be moved by a single robot. However, if a group of robots surround it and push in a coordinated way, then the robot system will be able to move the piano [17, 18].

The swarm of robots can also be heterogeneous [19]. While it is normal for all agents in the system to execute the same code, there is no reason why robots should be equal to each other. In fact, it is easier and cheaper to build them if they do not have to be identical, and differences can help solve the task [20]. It is possible that under specific conditions one robot will assemble with another, in a kind of self-assembly coordinated by environmental conditions [21]. This can be useful in tasks where it is required to identify a particular condition or combine two or more robots to activate specific code sequences.

The inspiring principle for the design of these swarms of robots lies in nature. In nature, there are countless examples of animals working together towards a common goal [22, 23]. This collaborative work is observed at all levels, in cells, birds, insects, schools of fish in the sea, and even in humans with the support of social behaviors. We have selected as a biological model the bacterial Quorum Sensing. Under this behavior, bacteria are able to express different kinds of behaviors (according to a single genomic code) in response to chemical stimuli in the environment. In particular, their aggressive behavior has been documented when from these readings they determine that there are a large number of individuals in the neighborhoods, enough to trigger an attack [24].

The following part of the paper is arranged in this way. Section 2 presents preliminary concepts and problem formulation. Section 3 illustrates the design profile and development methodology. Section 4 we present the preliminary results. And finally, in Section 5, we present our conclusions.

2. PROBLEM FORMULATION

A collection of n robots (numbered 1 to n) is placed into a compact, connected planar workspace $W \subset \mathbb{R}^2$. Let ∂W denotes the boundary of W . The free space within this environment is denoted by E and is an open subset of W . Within W there are obstacles corresponding to areas of the two-dimensional plane not accessible to robots. These obstacles are finite in number and size and independent of each other. We will denote \mathcal{O} as the set of all obstacles. In this way, $W = E \cup \mathcal{O}$.

Let $\mathcal{r} = \{r_1, r_2, \dots, r_i\}$ be a set of regions defined by some geometrical strategy in E . Let \mathcal{R} denote the collection of all regions. Each of these regions has a geometrical centroid point p_i that represents it, where $\mathcal{p} = \{p_1, p_2, \dots, p_i\}$ is the set of all these points in E . We can associate to each one of these points a value of slope, which we will denote as m , which will correspond to some law of navigation defined on the plane of the environment. In coherence, we can define a function $G(x, y)$ on the plane \mathbb{R}^2 in such a way that the function assigns this value of slope to each point $p_i(x, y)$ of the environment. The set of all the points in the environment is called \mathcal{p} , therefore $p_i \subset \mathcal{p}$. The assignment of values is made in such a way that the smallest value corresponds to the centroid of the target region.

For the design of the regions, it is important to consider the size of the environment and the size of the robots. The aim is that the robots can be easily organized in a region, that the size of the region is considerably larger than the size of the robot, and that the region is small in relation to the whole environment. These restrictions guarantee the mobility of the robots and the design of the navigation path. With respect to the movement of the robot along the path, under the above considerations the robots are small, and therefore can be modeled as mobile points. Although each robot has a geometry and kinematics, this is not considered relevant in our strategy. The objective of the strategy is not to explicitly control the movement of the robots, not even to determine their state, but to propitiate their displacement from local information deposited in the environment. This is achieved through a restriction of movement under which the robots are forced to travel through the regions until they find a specific event to guide their movement.

This behavior of the robots can be compared to balls on a billiard table. The balls impact every boundary of the table and bounce back to go through another portion of the table. This type of behavior will be the one programmed by the robots but modified in order not to respect the law of mirrors for the angle of reflection. Instead, they will bounce off obstacles and boundary of the environment on a random angle. This type of movement ensures that the robot travels through the region in a finite time, and therefore finds in it the information it needs to navigate. For a single robot, we define a discrete transition system D_1 that

simulates the original hybrid system. Let the state space of the discrete system be \mathcal{R} . The transition system is defined as (1):

$$D_1 = (\mathcal{R}, r_0, \rightarrow_1) \tag{1}$$

In which r_0 is the region that initially contains the robot. The transition relation $r \rightarrow_1 r'$ is true if and only if r and r' are neighboring regions with a common border.

The environment with the regions corresponds to a simplification of the navigation problem. In such sense, this description can be represented by a labeled transition graph. In this representation, the vertices would correspond to the regions defined in the environment, and the edges would correspond to the transitions between the regions as shown in Figure 1.

D_1 is a representation of the system, which is modeled as a hybrid system. The dynamics of each robot in each region is continuous, but the transition from one region to another is triggered by a discrete event. We propose that with proper triggering events it is possible to guide the robots along a navigation route on D_1 . We use the control description of a hybrid system as a consequence of the description of the robots and their movement in the regions. The system is made up of n robots, therefore D_1 is generalized by making an n -fold Cartesian product of the transition graph. This results in a discrete transition system D_n that simulates the motions of all robots.

We develop an event-based system, each robot starts with an initial control mode. During execution, the control mode may change only when receiving a sensor observation event y . We define the set of all the possible observation events that a robot can detect and obtain some relevant information from the environment. This set is called Y , therefore $y \in Y$.

In the proposed strategy of movement, there is no state feedback. It is not possible to determine explicitly the position and orientation of any of the robots of the system, much less of the system as such. Instead, we propose a control strategy based on information feedback. Each robot collects local information through its sensors, and from this information builds an information space through $\phi: I \times Y \rightarrow I$ mapping. Certain events produce certain actions of the robot according to this mapping. This corresponds to the discrete events of the hybrid system. To coordinate the movement of the robot from the events, we define a set of $\pi: I \rightarrow U$ form control policies, which are explicitly defined for each task, and allow the robot to define its movement by means of the sensor observation history as shown in Figure 2.

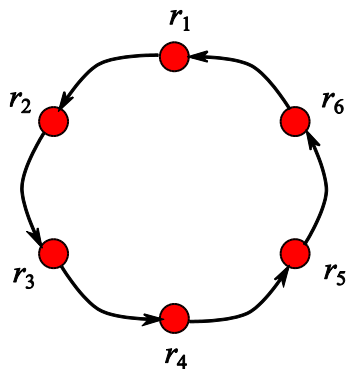


Figure 1. Transition graph for a circular path through the regions r_1, r_2, r_3, r_4, r_5 and r_6

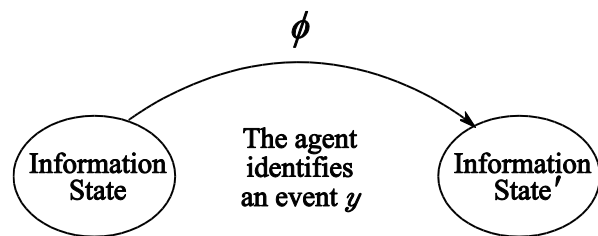


Figure 2. Information-state transition

All robots are programmed with the same code, which by similarity with the biological model we call the genome. This genome contains information related to the control policy defined for the task. That is, according to local readings, the robot will switch between behaviors according to the information detected in each region. Each behavior is a new state of the robot, therefore, the system moves from the observations of each agent.

In the task defined at this stage of the investigation, the agents must meet in a target region moving through the different regions of the environment (the initial location of the robots is random). The local readings will correspond to intensity values located as marks in the environment that can be detected and interpreted by the robots (landmarks). These can be color marks or coded digital values. Local

communication is restricted to the identification of neighboring robots within a short range. The detection of obstacles, environmental boundaries, and other robots is done by simple contact sensors.

3. METHODOLOGY

In general, there are two main control strategies for the coordination of a swarm of robots: the use of a central control unit or the use of a distributed control scheme on the robots. It is widely known that the centralized scheme, besides being highly complex, has low reliability, which is why many non-centralized strategies have been postulated. We propose the use of bacterial QS as a decentralized coordination strategy for a group of small autonomous robots with low processing capacity. QS corresponds in biology to the ability of a bacterium to detect population density levels, and respond by genetic regulation (internal code of the bacterium) through a specific action. This ability is supported by local readings that can be made by the bacteria in the medium, and by the information that the same bacteria chemically deposits in the medium (communication with other bacteria).

In our QS model, each robot plays the role of a bacterium, or agent if the classical artificial intelligence nomenclature is used. The agents navigate the environment reading local information through their sensors. The central idea is to make the agents sense this local information, and from there achieve collective coordination. The task proposed in this research is path planning to a target region, which is why parameters such as the inter-agent distance inherent in formation tasks are not relevant. The robots maintain a safe distance between them and with respect to the obstacles from the information of their sensors (contact sensors).

The navigation path can be coded in the environment by some traditional path planning strategy, this would correspond to the design of the navigation environment, and our strategy requires small modifications to the actual navigation environment. For example, a simple option is the use of potential fields. We can define a virtual potential field over the navigation environment in such a way that the values stable the path for the robots. This virtual field can be imagined as a sheet placed as a roof in the navigation environment, in which heavy metal spheres are placed along the desired path. The lowest values of the sheet will correspond to the gradient that the robots must detect and follow.

The proposed strategy requires the design of the navigation environment. This design refers to the definition of the regions, including the region target, and the installation of landmarks in each of the regions. These landmarks are special tags with coded values that are identifiable by the robots through their sensors. These landmarks encode values according to the intensity value corresponding to the location point. The intensity value is assigned to the entire navigation environment according to the design of the potential field or equivalent strategy selected so that the value is minimal in the target region, and increases as it moves away from this region. The robots freely navigate the free space looking for information in the environment. This is, in fact, a search process of the algorithm during which the robots identify the landmarks and react according to the readings. The difference between readings gives the robots the possibility to calculate the gradient of the readings and select a navigation direction that reduces the distance to the minimum, i.e., to the target region. The use of the potential field as heuristics does not result in the problem of local minima since this heuristics is used as an exploration strategy, which in fact ensures the convergence of the strategy.

Now, we introduce an additional element to the algorithm, the QS. This also corresponds to a local event identifiable by the robots, but not coded in the environment, but a consequence of the dynamics of the robots. It corresponds to the reading of the medium that informs about the population density in the region, that is, the number of robots in the same region. A larger population makes the region more attractive, so the effect of QS on the algorithm is to reduce the convergence time. From the mathematical point of view, the gradient in the navigation environment can be represented by two ordinary differential equations of the first order (2). In this representation, the slope at each point corresponds to the intensity.

$$\begin{aligned} \frac{dx}{dt} &= \mathbf{G}_1(x, y) \\ \frac{dy}{dt} &= \mathbf{G}_2(x, y) \end{aligned} \tag{2}$$

In (2) the derivatives on the left represent the speed of change in each of the axes of the environment. As a visual representation, the origin of this system would correspond to the centroid of the region target. These functions try to push the robots towards the origin. A possible design for these functions is shown in Figure 3.

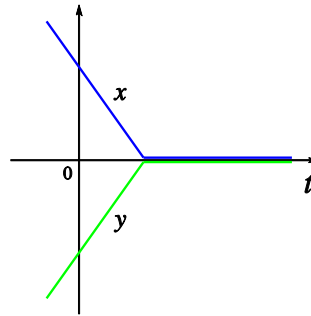


Figure 3. Linear approximation of intensity designed for a target region

As can be seen, these curves do not allow a path between them, that is, in fact, their design characteristic. The curves are symmetrical with respect to the x-axis, with slopes that push towards the origin. A possible structure of the curves is (3):

$$\begin{aligned} x &= a \times e^{\frac{-y}{\tau}} - a \\ y &= a - a \times e^{\frac{-y}{\tau}} \end{aligned} \tag{3}$$

In this equation a is a reference value and τ is the time constant. This constant adjusts the slope, and could eventually be defined as a function of population density. With differential writing the equations become (4):

$$\begin{aligned} \frac{dx}{dt} &= -\frac{a}{\tau \times e^{\frac{-y}{\tau}}} \\ \frac{dy}{dt} &= \frac{a}{\tau \times e^{\frac{-y}{\tau}}} \end{aligned} \tag{4}$$

Exponential representation requires growth control, which is why one variable is subtracted from the other. For a time constant of (5):

$$5\tau = 5 / \tau = 1 \tag{5}$$

We show the behavior of the slope in Figure 4, and of each of the axes in Figure 5.

In this way, a $P(x,y)|_{p_0 \rightarrow p_1}$ function defines the path of the robots from their region of origin to the target region as shown in Figure 6. Let $v(x,y)$ be the potential field on E which guides the movement of robots. In order that the robots are oriented along $P(x,y)|_{p_0 \rightarrow p_1}$ the intensity of the field should be minimal for the points belonging to $P(x,y)|_{p_0 \rightarrow p_1}$. In addition, the field intensity should increase as the points away from the route. This increase can be calculated along a normal line to the path traced. For any point $p(x,y)$ which does not belong to the path $P(x,y)|_{p_0 \rightarrow p_1}$, the value of the intensity of the field should be proportional to the perpendicular distance from the point to the path.

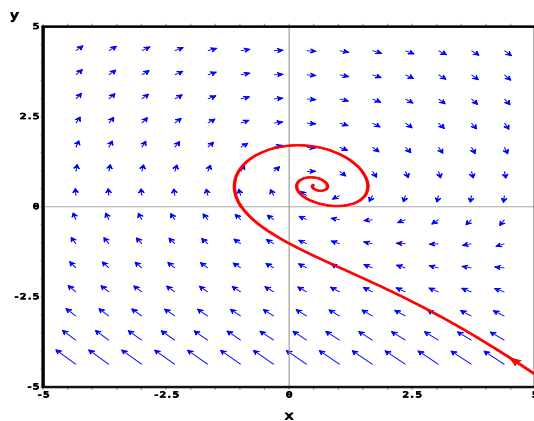


Figure 4. Designed slope field for a target region

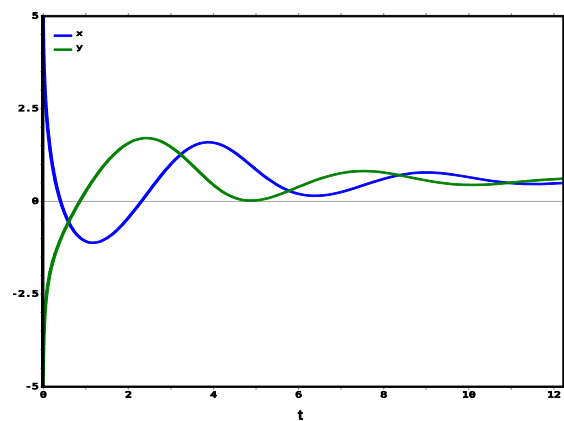


Figure 5. Exponential approximation of intensity behaviour in a target region

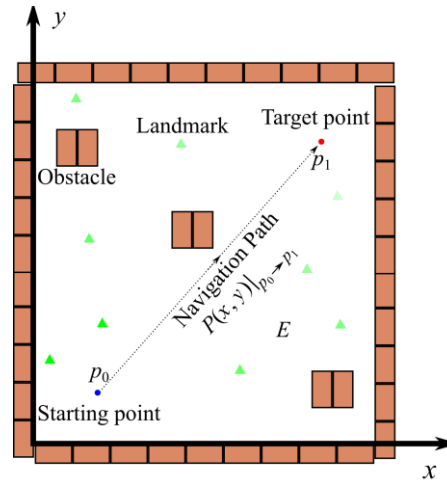


Figure 6. Path planning from p_0 to p_1 in two-dimensional environment with two obstacles. The design of the environment corresponds to the characteristics of the laboratory environment

The design of the intensity field should also consider the direction of advance, therefore, its value should decrease towards the target region. This can be shown more easily in graphical form. For example, if we want a robot to reach position (5, 4) (target region) from a random starting position, for example, (1, 1), we can think of an intensity field as shown in Figure 7. This design is simple, assigns the lowest value to the target region, and coherent values for the rest of the environment.

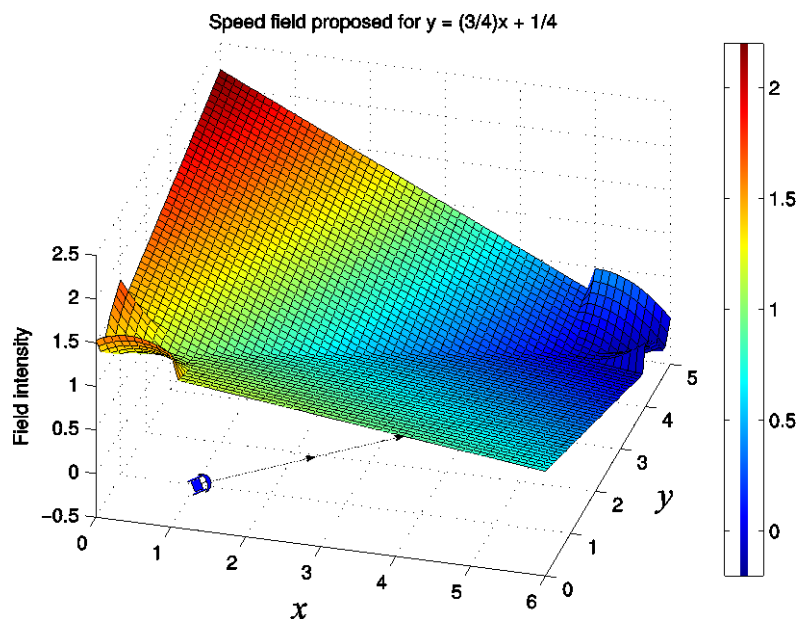


Figure 7. Example of an intensity field design. Path $y = \frac{3}{4}x + \frac{1}{4}$ between points (1,1) and (5,4)

4. RESULTS AND ANALYSIS

We simulate the strategy for the conditions of our laboratory and our test robots [25]. The working environment is a rectangle of 5 m x 6 m. The potential field was designed for this environment according to the conditions described above. This field works as an attraction field for robots. Regardless of the initial position of each robot in E , each robot will eventually navigate guided through the field to the target region. In the simulation shown in Figure 8, we arbitrarily select the position of a robot in (1, 3). Under the potential field design is shown in Figure 7, and assuming that the robot can read the value of the field at each point of

E , we program the robot's behavior to reduce the gradient of the field as it advances. However, to make the robot's behavior more realistic, we did not place landmarks throughout E . We only placed a small number of randomly distributed landmarks in E , so that the robot would make mistakes when calculating the gradient and direction of forward. In the end, however, the robot managed to find the target region. Since the initial position of the robot and the angle of rotation when an obstacle is found are random, to verify the behavior of the group we performed this simulation several times. In each case, the robot manages to reach the target region.

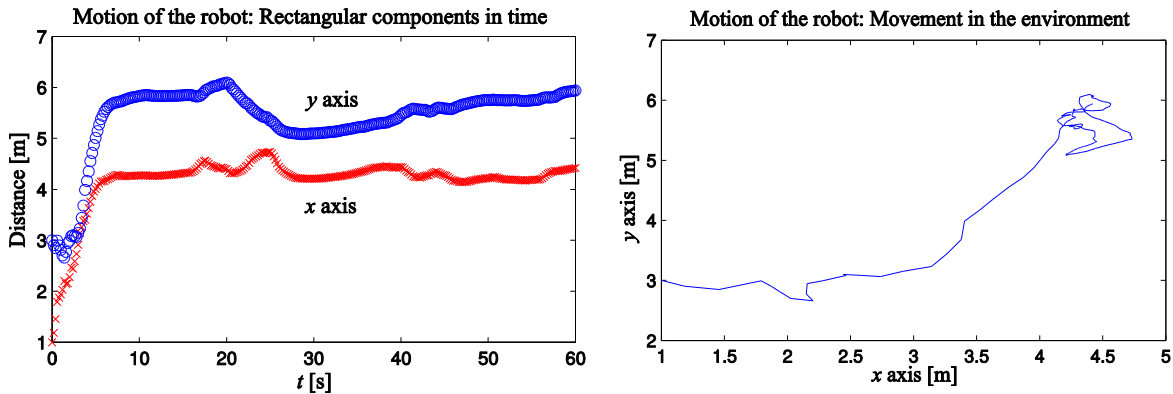


Figure 8. Motion of a robot in the environment

The landmarks always maintain their location in the environment, and codify the intensity value according to the design of the field at the point where they are placed. We have implemented these landmarks in different ways, in some experiments we use color marks on the ground with different color intensities. We have also used small tags with the values stored in their memory. In either case the robots were equipped with sensors capable of reading the coded information, and interpreted according to the control policy defined for the task. This together with the historical values read by the robot allowed the robot to calculate a gradient and define an action for the readings. If it also detects the presence of other robots, the value of the region increases in proportion to the number of robots detected, this is the QS added to the algorithm, and has the effect of accelerating convergence in favour of populated regions.

5. CONCLUSION

In this paper, we propose a strategy for motion planning of a swarm of robots based on simple rules for each robot, local readings, and bacterial QS to accelerate navigation. The strategy makes use of a heuristic to design a potential field in the environment to establish the value of landmarks to be installed in the environment. These landmarks will encode values of punctual intensities in the environment that the robots can read, and from them establish a gradient and a navigation direction. This direction of navigation is influenced however by the number of individuals present in the surroundings. If a robot detects other robots nearby, the area becomes more desirable for the robot in direct proportion to the number of robots detected, this is a feature that mimics the bacterial QS. From simulations, it was possible to verify the success of the strategy and proposes the use of experiments to validate the results in the real world.

ACKNOWLEDGEMENT

This work was supported by the Universidad Distrital Francisco José de Caldas, in part through CIDC, and partly by the Facultad Tecnológica. The views expressed in this paper are not necessarily endorsed by Universidad Distrital. The authors thank the research groups ARMOS and DIGITI for the evaluation carried out on prototypes of ideas and strategies.

REFERENCES

- [1] A. Kolling, P. Walker, N. Chakraborty, K. Sycara, and M. Lewis, "Human interaction with robot swarms: A survey," in *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 1, pp. 9-26, Feb 2016.
- [2] H. Zhao, H. Liu, Y. Leung and X. Chu, "Self-adaptive collective motion of swarm robots," in *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 4, pp. 1533-1545, Oct 2018.
- [3] S. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, "A survey on aerial swarm robotics," in *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 837-855, Aug 2018.
- [4] C. López. "Evaluación de desempeño de dos técnicas de optimización bio-inspiradas: Algoritmos genéticos y enjambre de partículas," *Tekhnê*, vol. 11, no. 1, pp. 49-58, 2014.
- [5] P. Burade. "Congestion management incorporation of facts devices using ant colony optimization," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 1, no. 2, pp. 139-150, 2012.
- [6] C. Pinciroli and G. Beltrame, "Buzz: a programming language for robot swarms," in *IEEE Software*, vol. 33, no. 4, pp. 97-100, July-Aug 2016.
- [7] H. Huang, "A Taguchi-based heterogeneous parallel metaheuristic ACO-PSO and Its FPGA realization to optimal polar-space locomotion control of four-wheeled redundant mobile robots," in *IEEE Transactions on Industrial Informatics*, vol. 11, no. 4, pp. 915-922, Aug 2015.
- [8] C. S. Karagöz, H. I. Bozma, and D. E. Koditschek, "coordinated navigation of multiple independent disk-shaped robots," in *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1289-1304, Dec 2014.
- [9] S. Campello Oliveira, G. Andrade Duarte, and H. Beltrao Cunha, "Unified coordination-communication strategy to swarm controlled mobile wireless sensor network," in *IEEE Latin America Transactions*, vol. 12, no. 5, pp. 951-956, Aug 2014.
- [10] S. Bandyopadhyay, S. Chung, and F. Y. Hadaegh, "Probabilistic and distributed control of a large-scale swarm of autonomous agents," in *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1103-1123, Oct 2017.
- [11] J. Wen, L. He, and F. Zhu, "Swarm robotics control and communications: imminent challenges for next generation smart logistics," in *IEEE Communications Magazine*, vol. 56, no. 7, pp. 102-107, July 2018.
- [12] I. Matazi, R. Messoussi, S. Bellmalle, I. Oumaira, A. Bennane, and R. Touahni. "Development of intelligent multi-agents system for collaborative e-learning support," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 7, no. 2, pp. 294-305, 2018.
- [13] R. Ramaithitima, M. Whitzer, S. Bhattacharya, and V. Kumar, "Automated creation of topological maps in unknown environments using a swarm of resource-constrained robots," in *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 746-753, July 2016.
- [14] C. Luo, S. X. Yang, X. Li, and M. Q. - Meng, "Neural-dynamics-driven complete area coverage navigation through cooperation of multiple mobile robots," in *IEEE Transactions on Industrial Electronics*, vol. 64, no. 1, pp. 750-760, Jan 2017.
- [15] A. R. Shirazi and Y. Jin, "A strategy for self-organized coordinated motion of a swarm of minimalist robots," in *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 1, no. 5, pp. 326-338, Oct 2017.
- [16] C. Juang, M. Lai, and W. Zeng, "Evolutionary fuzzy control and navigation for two wheeled robots cooperatively carrying an object in unknown environments," in *IEEE Transactions on Cybernetics*, vol. 45, no. 9, pp. 1731-1743, Sept 2015.
- [17] J. Chen, M. Gauci, W. Li, A. Kolling, and R. Groß, "Occlusion-based cooperative transport with a swarm of miniature mobile robots," in *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 307-321, April 2015.
- [18] D. Ruiz, B. Bacca, and E. Caicedo, "Control strategy based on swarms algorithms to cooperative payload transport using a non-holonomic mobile robots group," in *IEEE Latin America Transactions*, vol. 14, no. 2, pp. 445-456, Feb 2016.
- [19] R. Maeda, T. Endo, and F. Matsuno, "Decentralized navigation for heterogeneous swarm robots with limited field of view," in *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 904-911, April 2017.
- [20] A. Prorok, M. A. Hsieh, and V. Kumar, "The impact of diversity on optimal control policies for heterogeneous robot swarms," in *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 346-358, April 2017.
- [21] K. Szwaykowska, L. M. Romero, and I. B. Schwartz, "Collective motions of heterogeneous swarms," in *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 810-818, July 2015.
- [22] X. Yi, A. Zhu, S. X. Yang, and C. Luo, "A bio-inspired approach to task assignment of swarm robots in 3-D dynamic environments," in *IEEE Transactions on Cybernetics*, vol. 47, no. 4, pp. 974-983, April 2017.
- [23] Y. Mulgaonkar, A. Makineni, L. Guerrero-Bonilla, and V. Kumar, "Robust aerial robot swarms without collision avoidance," in *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 596-603, Jan 2018.
- [24] Q. Zhao, M. Li, Z. Wang, J. Li, and J. Luo, "A quorum sensing algorithm to control nanorobot population and drug concentration in cancer area," *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Zhuhai, 2015, pp. 42-47.
- [25] F. Martínez, H. Montiel, and E. Jacinto, "Hybrid fuzzy-sliding grasp control for underactuated robotic hand," *TELKOMNIKA (Telecommunication, Computing, Electronics and Control)*, vol. 17, no. 4, pp. 2070-2075, 2019.