Journal of Theoretical and Applied Information Technology
15th August 2021. Vol.99. No 15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A DECENTRALIZED, LOW COMPUTATIONAL COST STRATEGY FOR COORDINATION AND SEARCH WITH A ROBOT FLOCK

FREDY H. MARTÍNEZ S.

Facultad Tecnológica, Universidad Distrital Francisco José de Caldas, Bogotá D.C, Colombia

E-mail: fhmartinezs@udistrital.edu.co

ABSTRACT

The coordination of a flock of robots is a high demand application in applications such as motion planning, navigation, herding (tracking and/or tracing), area coverage (exploration, search and rescue, etc.), object transportation (surrounding and moving together), and compound tasks, all of which are currently heavily researched in robotics. Many approaches have been proposed to solve this problem, but they largely compromise system characteristics such as fault tolerance, capacity, efficiency, and in particular cost, since real implementations require special hardware. This paper proposes a coordination strategy for a system composed of small robots of minimalist design under the condition of minimum processing and sensing capacity. The communication requirements have been limited to a local communication strategy sufficient to achieve the relative orientation of each swarm member. The usefulness of the scheme is evaluated by simulation in specialized search tasks in an unknown region. The results show the high capability of the scheme and the ease of implementation on real prototypes.

Keywords: *Collective movement, Flocking, Leader-follower, Swarm robotics, Self-organizing systems*

1. INTRODUCTION

Multi-agent systems in robotics are becoming a field of application with strong commercial implications at the industrial level. Many tasks are inefficient when performed by an autonomous robot, and the collective models observed in animals are very attractive for this type of task. One of the basic behaviors of a system composed of a group of robots is its movement in a flock [1, 2]. This behavior is characterized by the simultaneous movement of all the agents of the system as if the system were a single entity [3]. This functional characteristic seeks to mimic the dynamics of groups of animals such as birds or fish when they perform a complex task together [4]. In the biological model, the flocking behavior increases the sensing and response capacity of the system (one of the individuals can detect a problem, and make the whole system respond accordingly), a feature that is also very desirable in a swarm system composed of robots or artificial agents [5, 6]. To achieve this behavior requires that each agent in the system adjusts its speed and alignment (i.e., defines its movement strategy) in correspondence with the movement of the other robots, avoiding collisions while responding to the global dynamics of the system [7, 8]. This scheme allows proposing a multi-agent system architecture that can be controlled for the development of complex tasks.

Considering the implications of the use on real prototypes, this research proposes a global flocking behavior strategy for small robotic platforms [9]. This motion strategy turns out to be key in robotic tasks of motion planning, navigation, herding (tracking and/or tracing), area coverage (exploration, search and rescue, etc.), object transportation (surrounding and moving together), and compound tasks from the previous ones [10, 11, 12, 13]. In particular, we focus on the task of searching for elements of interest in an area, such as locating people in collapsed environments, or any other specific element in an environment, ignoring a priori quantity and characteristics of the environment [14, 15]. The algorithm is proposed and evaluated as a hardware platform-independent strategy.

One of the typical tasks of animal flocks is to forage in an environment. The paper presents a search algorithm based on flocking behavior and local interaction for a swarm of small robots [16, 17]. It is a navigation scheme that mimics the behavior of swarms or flocks that explore an environment in search of specific elements, in which the group of agents chases a leader along with a navigation environment, while each of the agents performs specific searches in the environment [18]. The identification of elements of interest in the environment should allow the formation of sub-groups in regions of the environment identified as possible solutions to the search problem [9, 19, 20]. The search parameters may correspond to the vital signs of a person in a collapsed environment, the detection of a metal in an area with anti-personnel mines, or the morphological characteristics of an endangered animal in its natural habitat.

Like any other complex system, navigation in flocks presents certain problems. Agents in the system can collide with each other, all individuals must navigate the environment while avoiding obstacles, and different elements in the environment can affect the navigation path according to the sensing capabilities of the agents [21, 22, 2]. Robots are designed to identify specific targets in the environment (people, metals, certain parameters in images, etc.) as well as possible obstacles that limit their movement and other agents [23]. Since the location within the system is different for each agent in the system, each individual will have its sensing capability relative to the region of the environment in which it is located [24, 25]. When a sub-group of agents detects a search parameter in a region, the algorithm indicates that this group of agents should remain in the region. Also, the more agents in a region, the more attractive the region is to other agents [26, 27]. This behavior mimics bacterial Quorum Sensing (QS), in the biological model, a bacterium recognizes the population intentionality through the concentration of chemicals it deposits in the environment when it detects a high intensity it understands that there is a large number of individuals, and initiates an attack (virulence), which in a robot is expressed as the convergence to a possible solution (stay in the region). These biological behavioral schemes can be integrated to form a simple but high-performance solution.

Most of the solutions proposed in the literature for bouncing flocks fall into the problem of the actual implementation. The schemes require complex robots capable of executing the algorithms in real-time, which are too complex for small robots with hardware limitations. In addition, many of the strategies still implement a central control unit, which reduces the robustness of the system and goes against the decentralized principles of these systems.

2. PROBLEM STATEMENT

The problem can be defined as follows. A swarm of agents, with local sensing and displacement capabilities, composed of n individuals, all identical in structure and capability, are deposited in a connected, compact, planar environment identified as W. The particular design of each agent is not critical to the algorithm; instead, the properties of the motion are fundamental. The movement of the agent responds to local readings of the environment as well as to the tracking of the leading agent.

This swarm of agents must be able to self-coordinate and define its position and movement strategy according to the location of the other agents in the system. The self-organization strategy must allow the joint navigation of the system and the development of basic tasks throughout the environment.

W can be decomposed into connected regions denoted by r, capable of containing a subset of n. The collection of all regions is denoted by R. Some of these regions contain target points to be located by the agents as the convergence of their behavioral algorithm. Using Linear Temporal Logic (LTL) to specify the task, if $\Pi = \{\pi_1, \pi_2, ..., \pi_n\}$ is a set of Boolean propositions such that π_i is true if and only if a subset of n is in $r_i \in R$, then the navigation and search task can be specified for each agent as:

Navegation and search for targets:

$$(\Diamond\pi_1 \wedge \Diamond\pi_2 \wedge \cdots \Diamond\pi_n)\ U\ \Diamond\pi_k \qquad (1)$$

Where the following temporal operators are used:

- ○: Next
- v: Disjunction
- ∧: Conjunction
- =>: Implication
- <=>: Equivalence
- ◊: Eventually
- []: Always
- *U*: Until

and π_k is a target region. Activation of the QS causes the robot to move from an initial explorer state o to a subsequent virulent state o', which is specified as $o\ U\ o'$.

The agents are not familiar with W, they do not know where the target points are, nor do they know how many of them there are. Their possible location

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

can only be determined by reading their sensors, which have a limited range (much smaller than the size of W). There are regions in W inaccessible to the agents called obstacles, the collection of all obstacles is called $\boldsymbol{O}$, and W - $\boldsymbol{O}$ = E, which corresponds to the free space along which the agents can navigate. The movement of the system is controlled by the readings of each agent, and in particular of the leading agent.

As a general case, agents have contact sensors (very short-range obstacle distance sensors) that allow them to identify obstacles, other agents, and ∂W. In such circumstances, the agents will change their navigation direction randomly relying on the dynamics of the system to cover the whole environment. They also possess specific sensors of limited range (but sufficient to detect targets) and are centered on the vertical axis that allow them to detect target regions. It is assumed that agents can identify target regions if they are in the range of the sensors and those other agents cannot occlude the agent's sensing capability. The specific type of sensors is not important, but their ability to identify specific elements in the environment at close range, including other agents. In principle, the information collected by an agent is not shared as such to the other members of the system, which allows for a much simpler and relaxed communication system.

The motion control policy of each agent i dictates that it must follow the movements of its target agent j unless it identifies a target region, which has a higher degree of attraction for the agent. If the two agents are close to each other (at a distance of less than ρ), then the two agents are considered as one as long as this condition holds. The relationship between an agent and its target agent cannot be reciprocal, i.e., these two agents cannot mutually establish the other as a target, since they would stop moving and separate from the system. One of the agents, the lead agent, will navigate the environment without following any agent. The system will always have at least one leading agent, and any agent in the system can be a leader. The assignment as lead agent is given if the agent does not detect other agents to follow. This agent navigates the free space avoiding obstacles until it finds an agent to follow. Agents separate from the system eventually join it, and those that are followed direct the navigation of the system and sub-systems that form along with the development of the task (Fig. 1).

Figure 1: Pseudo-Code Of Control Policy For Each Flock Member

3. METHODS

Most of the strategies for swarm-like systems have only been demonstrated on prototypes with a small number of agents, largely because the algorithms require expensive robots with high processing, communication, and sensing capabilities. Not only is it costly to implement a large swarm with these robots, but it is also costly to reduce the size of existing robots while maintaining these features. These features also go against the minimalist principles of agents in these types of systems. This research seeks to propose a coordinated navigation strategy of a multi-robot system that retains a simple structure to implement on low-cost robots, which implies hardware with limitations in processing, sensing, and communication. The movement strategy of our multi-robot system is flocking, which will allow the search for specific elements in the environment.

Flocking is characterized by the coordinated movement of agents according to a target. The goal of our system is to explore the entire free space E in search of a region of interest π_k. This region contains a special element that can be detected and identified at close range by any of the agents, so the exploration turns out to be more efficient with a larger number of individuals. However, given the limited processing capacity of the agents, the decision-making is restricted to following simple control policies. This design principle goes hand in hand with biological models (fish, birds, bees, etc.) in which system dynamics emerge as a consequence of simple local interaction. These control policies must succeed in defining for each individual its relative position concerning the other agents and its movement strategy (direction and speed).

The strategy we propose is based on the ability of each robot to identify obstacles at close range around it and to be able to discern if it is another agent in the system. This is a common strategy in swarm systems consistent with biological systems. Robots cannot estimate their odometry, nor is it required to define their motion. Instead, it is proposed to identify only their location relative to their neighbors and obstacles in the environment. There is also no communication between agents beyond this ability to detect each other, and relies on the self-organization of the system to produce higher-order dynamics to solve the task. Unlike other schemes, we do not use anchor nodes in our proposal, all agents have an identical design (homogeneous agents). This feature increases the robustness and failure resilience of our scheme (damage of any of the agents does not affect the system performance).

A serious functional problem with swarm prototypes is that the location of individuals in the system depends on their communication, and if the communication is slow compared to the speed at which the agents move, it is impossible to estimate positions accurately. We solve the problem by simplifying the communication to a scheme in which each agent is informed from its local readings, which includes the identification of nearby agents. This eliminates communication delays, as well as protocol issues and the amount of data to be transmitted. In this way, the noise produced by the movement of the robots during transmission is eliminated.

Each agent in the system is capable of measuring the distance to neighboring robots within its range. Each agent can be assimilated to a node or vertex of a graph with edges (undirected graph) that connects the agents aware of the presence of its neighbors (connects the agents it detects with its sensors, Fig. 2). All agents in the system are identical, or with statistically negligible variations, so they all have the same sensing range, which causes the detection to be mutual or bidirectional. If an agent detects at least three other agents around it, from these distances it can establish its relative position by 2D trilateration, and consequently adjust its direction of movement and velocity (follow the closest one, increase its velocity if it is too far away, and reduce its velocity or move away if it is too close to another agent). If there are less than three agents around it, it simply follows the closest one, and if it detects none, it scans the environment looking for agents, target region, and dodging obstacles.

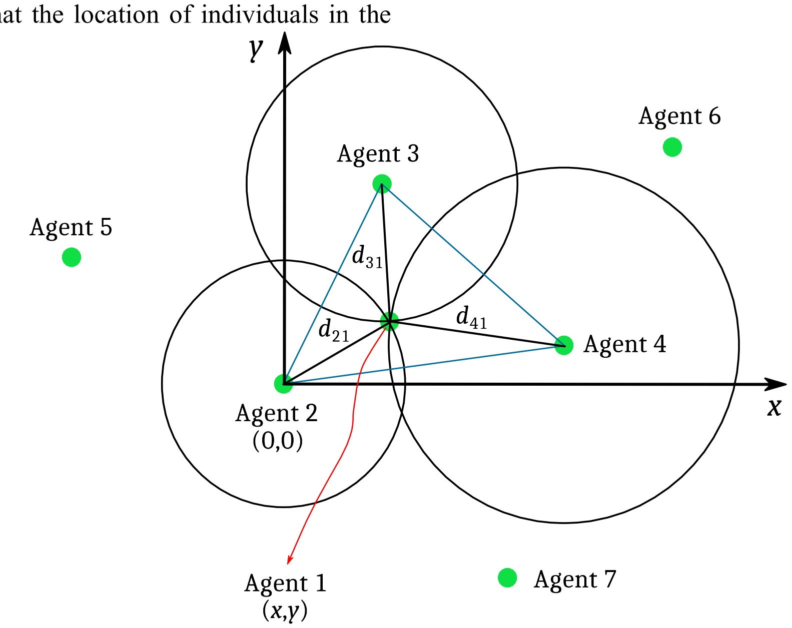

Figure 2: The Relative Positioning Of Agents In The System. The Nodes In Green Correspond To Agents. Agent 1 Seeks To Establish Its Relative Location

From the perspective of Agent 1, using its sensors this agent detects the distance to the three closest agents: Agent 2 (distance d_{21}), Agent 3 (distance d_{31}), and Agent 4 (distance d_{41}). These three distances correspond to the radii of three circles centered on Agents 2, 3, and 4, which intersect at the position of Agent 1 (Fig. 2). If one takes as the origin of the two-dimensional plane the position of one of the detected agents, for example, the closest one (assuming it is Agent 2, with $d_{21} < d_{31}$, and $d_{21} < d_{41}$), then it is possible to calculate the relative position of Agent 1 concerning the closest agent and to track it over time by simple geometric relations (the agents correspond to the vertices of triangles for which Agent 1 measures the sides and angles, note the blue triangle in Fig. 2, and the triangles formed with the distances that can be solved with the cosine theorem, determining the relative coordinates of the other agents).

This strategy forms the search algorithm for the target agent. To complete the movement strategy, a dynamic is programmed that makes the agent be in continuous movement, at the same time that it dodges obstacles and performs local readings of the environment. When these readings produce a search value above a certain region threshold, the agent stops following the target agent and stays in the region. When the population in this region exceeds a certain population threshold, a Quorum Sensing (QS) mechanism is triggered that causes the agent's behavior to change from explorer to virulent. This change indicates that the agents have located the π_k target region within the W environment.

4. RESULTS

The strategy was implemented on a swarm consisting of the TurtleBot 3 Burger by Robotis (Fig. 3). This platform is 13.8 cm x 17.8 cm x 19.2 cm in size, but its great advantage is that it has a 360-degree LiDAR sensor (Laser Distance Sensor LDS-01) capable of easily detecting distances to obstacles around the robot (detection distance 12 cm to 3.5 m, 1-degree angular resolution, and 300±10 rpm scan rate). Similar robots were assembled to evaluate the performance of the strategy with similar but not identical agents. In any case, the LiDAR configurations and other features were similar to allow a homogeneous system. This is an understudied feature on real platforms that provides real operational capabilities of these systems.

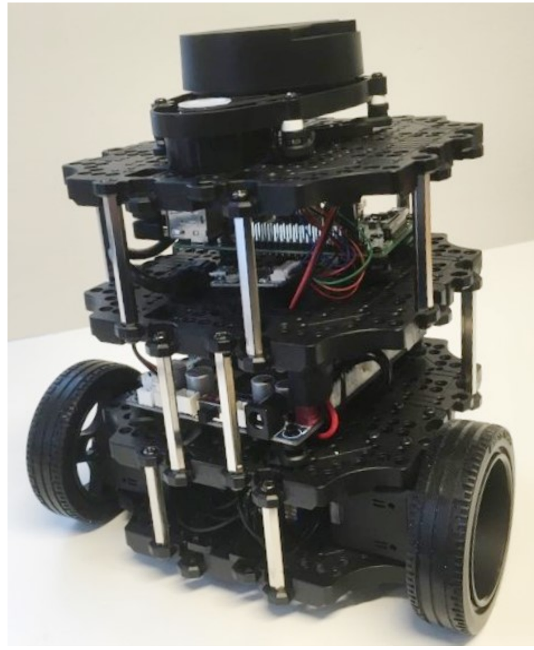

Figure 3: TurtleBot 3 Burger by Robotis

The simulations of this system were performed in a proprietary application developed in Python 3.7.10 with support for Numpy 1.19.5, Scipy 1.4.1, and Matplotlib 3.2.2. The simulations assume the idealized characteristics of the TurtleBot 3 Burger robot, i.e., circular robots with a radius of 0.105 m, the maximum forward speed of 0.22 m/s, with perfect sensing, and no explicit communication with other robots. The algorithm was implemented under the principle of collective movement along with the environment with a minimum of collisions between robots. The first test consisted of allowing free navigation of the swarm without designing any target region. Under these conditions, the swarm must navigate together along with the free space E.

Fig. 4 shows the results of one of these tests. Fifty agents were used in the swarm, each with a sensing capability of up to 3.5 m, and a minimum collision dodging range of 0.5 m. The test environment was designed with a size of 10 m x 10 m, and three rectangular obstacles were placed at global coordinates (2, 4), (9, 9), and (6, 1). The robots were randomly placed around the initial point (1, 8) (the dispersion around this point considered the size of the robot), and the behavior was simulated for the equivalent of 120 s (multi-agent system time).

The second part of the tests consisted of including a target region and observing how it is identified by the flock of robots. This region was defined within the global coordinates (7, 3), (8, 3),

Journal of Theoretical and Applied Information Technology
15th August 2021. Vol.99. No 15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

(8, 2), and (7, 2). It is a square area of 1 square meter in which a sub-group of the swarm should gather. The other conditions of the environment, robots, and system dynamics were kept the same. Fig. 5 shows the behavior of one of the tests performed.

The different simulations demonstrate the capability of the system, and therefore of the proposed algorithm, to scout in a finite time the navigation environment as a flock of robots, while dodging obstacles and searching the environment for a region of interest. This research is currently continuing with the implementation of the algorithm on real robotic platforms.

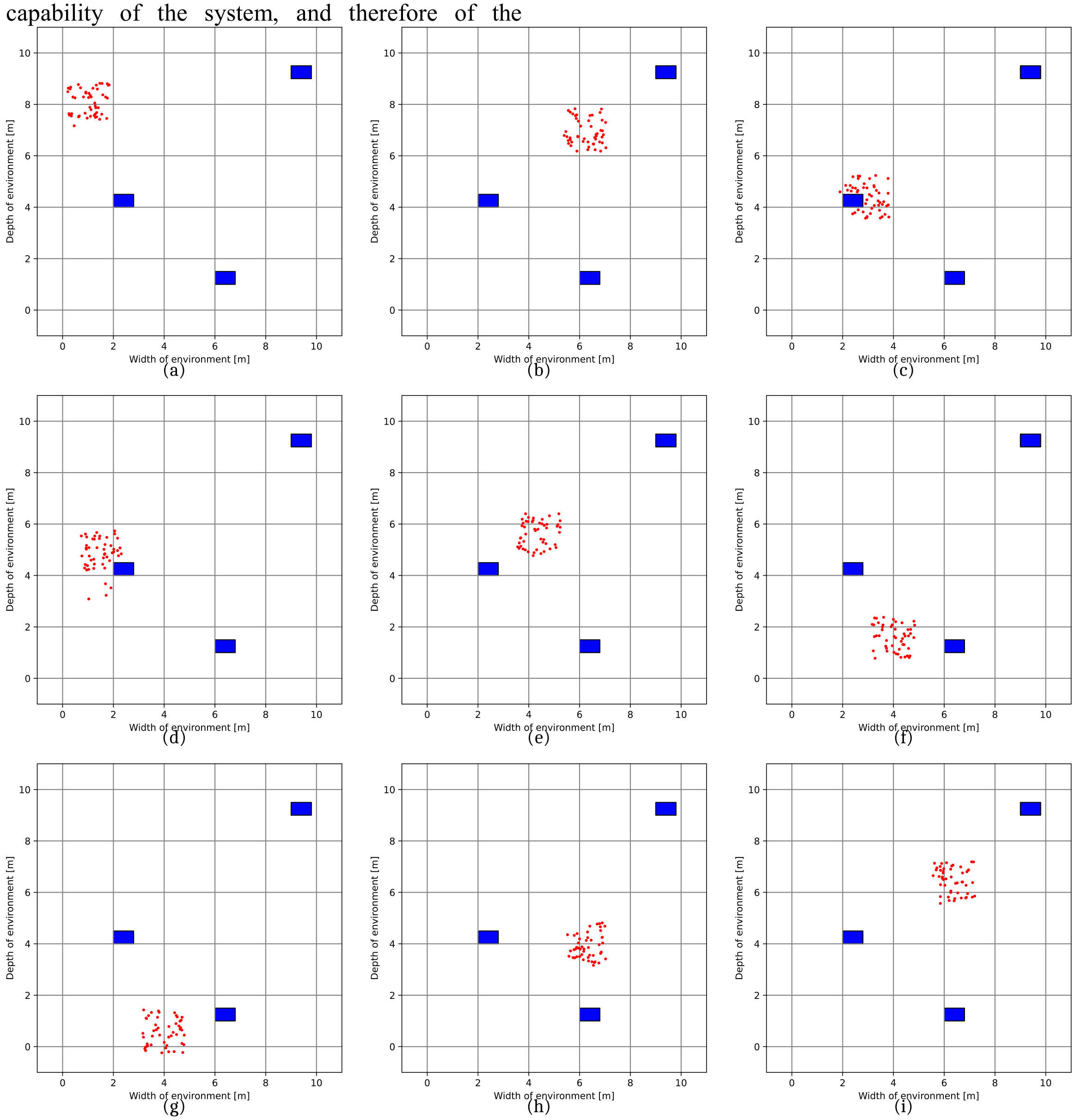

Figure 4: Flocking Simulation Starting From The Global Position (1, 8). The Captures Correspond To The Following Simulation Times: (a) 0 s, (b) 25 s, (c) 34 s, (d) 44 s, (e) 55 s, (f) 80 s, (g) 90 s, (h) 100 s, and (i) 110 s

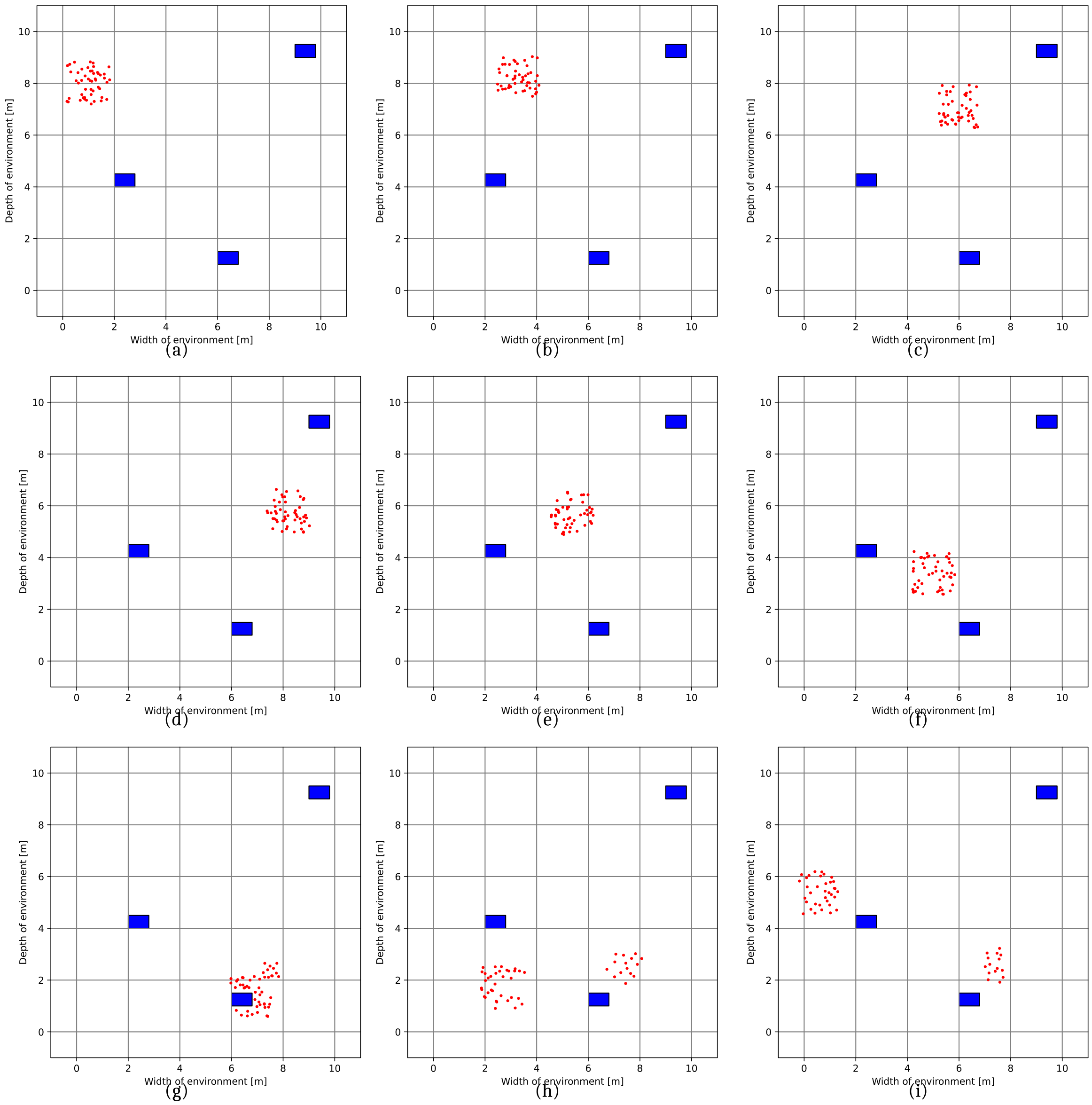

Figure 5: Flocking Simulation With Target Region Recognition Based On The Global Position (1, 8). The Captures Correspond To The Following Simulation Times: (a) 0 s, (b) 10 s, (c) 22 s, (d) 32 s, (e) 42 s, (f) 62 s, (g) 72 s, (h) 92 s, and (i) 112 s

6. CONCLUSION

This paper proposes a low computational cost algorithm for the autonomous coordination of a flock of robots whose task is the exploration of an environment and the identification of a region in which it meets certain specific characteristics. We start from the need to implement a robust and homogeneous scheme (without anchor agent) capable of running on real hardware with low processing, communication, and sensing capacity, which allows the self-organization of its agents while performing the exploration of the environment. To this end, we propose a dynamic that allows the flock to explore the entire environment in finite time, and a relative localization strategy that allows each agent to define in a simplified way its movement strategy (direction and speed). Along with these algorithms, a reactive control strategy is implemented in each agent based on local readings that allow each agent to avoid obstacles and other agents as it moves, and to identify specific regions in the environment. The design was formulated for the TurtleBot 3 Burger robotic platform, replicating its functional characteristics in a proprietary simulator developed in Python. Throughout multiple simulations, it was possible to establish the success of each of the blocks that make up the strategy, as well as its ability to find the solution to the search task. This research continues with implementations on real prototypes to identify limitations and possible improvements to develop a real-world application.

7. LIMITATIONS AND FUTURE RESEARCH

A large number of simulations performed confirmed the algorithm's ability to control the swarm of robots without delays in the system's response. Throughout the tests, it was possible to confirm that in 100% of the cases the system was able to traverse the entire free space of the environment and correctly avoid the obstacles in it. Furthermore, it was able to always find the target region, even when the signal in this region was poor or there was more than one target region (part of the population remained in each of the possible solutions). However, at this point it was impossible to explicitly determine the dependence of the time required for the sweep of the environment or the identification of the target region as a function of the system parameters, i.e., size of the environment, characteristics of the obstacles, population size, functional characteristics of the robot, and tuning parameters of the algorithm (quorum threshold, tracking signal strength threshold, and obstacle avoidance distance). It was possible to demonstrate that the time required for the development of the task is finite, but as the future development of this research, it is necessary to develop a statistical analysis to determine the dependence of these variables on the parameters of the system.

Secondly, it is necessary to implement the algorithm and evaluate it on the TurtleBot 3 Burger robots to identify functional differences with the simulation model and adjust the model accordingly. Besides, it is necessary to identify possible functional problems of the system in the real world that prevent or limit its use, and the true scaling capabilities of the system. These tests should also include studies of the time required to develop the task for different parameters of the system to adjust the behavior of the simulation model so that it can be used as a validation and design tool before the use of real prototypes. Testing with prototypes should also include the study of effects on the system caused by significant functional variations between robots (as in real robots, what happens to the performance of the algorithm when not all robots work identically).

8. ACKNOWLEDGMENT

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 group ARMOS for the evaluation carried out on prototypes of ideas and strategies.

REFRENCES

[1] B. Balázs, G. Vásárhelyi, and T. Vicsek. Adaptive leadership overcomes persistence-responsivity trade-off in flocking: Adaptive leadership overcomes persistence-responsivity trade-off in flocking. Journal of the Royal Society Interface, 17(167), 2020. doi: 10.1098/rsif.2019.0853rsif20190853.

[2] A.R. Shirazi and Y. Jin. A Strategy for Self-Organized Coordinated Motion of a Swarm of Minimalist Robots. IEEE Transactions on Emerging Topics in Computational Intelligence, 1(5):326–338, 2017. doi: 10.1109/TETCI.2017.2741505.

[3] L. Bayindir. A review of swarm robotics tasks. Neurocomputing, 172:292–321, 2016. doi: 10.1016/j.neucom.2015.05.116.

[4] H. Zheng, J. Panerati, G. Beltrame, and A. Prorok. An adversarial approach to private flocking in mobile robot teams. IEEE Robotics and Automation Letters, 5(2):1009–1016, 2020. doi: 10.1109/LRA.2020.2967331.

[5] F. Martínez, F. Martínez, and H. Montiel. Bacterial quorum sensing applied to the coordination of autonomous robot swarms. Bulletin of Electrical Engineering and Informatics, 9(1):67–74, February 2020. ISSN 2302-9285. doi: 10.11591/eei.v9i1.1538.

[6] J. Lwowski, A. Majumdar, P. Benavidez, J.J. Prevost, and M. Jamshidi. Bird Flocking Inspired Formation Control for Unmanned Aerial Vehicles Using Stereo Camera. IEEE Systems Journal, 13(3):3580–3589, 2019. doi: 10.1109/JSYST.2018.2884051.

[7] S. Chen, H. Pei, Q. Lai, and H. Yan. Multitarget Tracking Control for Coupled Heterogeneous Inertial Agents Systems Based on Flocking Behavior. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 49(12):2605–2611, 2019. doi: 10.1109/TSMC.2017.2789335.

[8] M. Chamanbaz, D. Mateo, B.M. Zoss, G. Tokí, E. Wilhelm, R. Bouffanais, and D.K.P. Yue. Swarm-enabling technology for multi-robot systems. Frontiers Robotics AI, 4(APR), 2017. doi: 10.3389/frobt.2017.00012.

[9] E. Ordaz-Rivas, A. Rodriguez-Liñan, M. Aguilera-Ruíz, and L. Torres-Treviño. Collective Tasks for a Flock of Robots Using Influence Factor. Journal of Intelligent and Robotic Systems: Theory and Applications, 94(2):439–453, 2019. doi: 10.1007/s10846-018-0941-2.

[10] A.A. Ali, A.T. Rashid, M. Frasca, and L. Fortuna. An algorithm for multi-robot collision-free navigation based on shortest distance. Robotics and Autonomous Systems, 75:119–128, 2016. doi: 10.1016/j.robot.2015.10.010.

[11] J. Cheng and B. Wang. Flocking Control of Mobile Robots with Obstacle Avoidance Based on Simulated Annealing Algorithm. Mathematical Problems in Engineering, 2020, 2020. doi: 10.1155/2020/7357464.

[12] T. Ibuki, S. Wilson, J. Yamauchi, M. Fujita, and M. Egerstedt. Optimization-based distributed flocking control for multiple rigid bodies. IEEE Robotics and Automation Letters, 5(2):1891–1898, 2020. doi: 10.1109/LRA.2020.2969950.

[13] M. Khaledyan, T. Liu, V. Fernandez-Kim, and M. De Queiroz. Flocking and Target Interception Control for Formations of Nonholonomic Kinematic Agents. IEEE Transactions on Control Systems Technology, 28(4):1603–1610, 2020. doi: 10.1109/TCST.2019.2914994.

[14] T. Andre and C. Bettstetter. Collaboration in Multi-Robot Exploration: To Meet or not to Meet? Journal of Intelligent and Robotic Systems: Theory and Applications, 82(2):325–337, 2016. doi: 10.1007/s10846-015-0277-0.

[15] F. Martínez, E. Jacinto, and F. Martínez. Obstacle detection for autonomous systems using stereoscopic images and bacterial behaviour. International Journal of Electrical & Computer Engineering, 10(2):2194–2172, 2020. ISSN 2088-8708. doi: 10.11591/ijece.v10i2.pp2164-2172.

[16] F. Martínez, E. Jacinto, and F. Martínez. Using bacterial interaction and stereoscopic images for the location of obstacles on autonomous robots. Bulletin of Electrical Engineering and Informatics, 9(3):906–913, 2020. ISSN 2302-9285. doi: 10.11591/eei.v9i3.2012.

[17] J. Motes, R. Sandstrom, W. Adams, T. Ogunyale, S. Thomas, and N.M. Amato. Interaction Templates for Multi-Robot Systems. IEEE Robotics and Automation Letters, 4(3):2926–2933, 2019. doi: 10.1109/LRA.2019.2923386.

[18] S. Nurmaini, Saparudin, B. Tutuko, and P. Aditya Putra. Pattern recognition approach for swarm robots reactive control with fuzzy-kohonen networks and particle swarm optimization algorithm. Journal of Telecommunication, Electronic and Computer Engineering, 8(3):155–160, 2016.

[19] R. Parasuraman, J. Kim, S. Luo, and B.-C. Min. Multipoint Rendezvous in Multirobot Systems. IEEE Transactions on Cybernetics, 50(1):310–323, 2020. doi: 10.1109/TCYB.2018.2868870.

[20] R.P. Ramos, S.M. Oliveira, S.M. Vieira, and A.L. Christensen. Evolving flocking in embodied agents based on local and global application of Reynolds' rules. PLoS ONE, 14(10), 2019. doi: 10.1371/journal.pone.0224376.

[21] Y.A. Romanova, E.E. Matveeva, A.N. Alexeev, and R.Y. Askhabov. Current status and problems of commercialization of innovations in robotic technology. International Journal of Applied Business and Economic Research, 14(9):5865–5874, 2016.

[22] D. Saito, N. Maruyama, Y. Hashimoto, and T. Ikegami. Visualization of dynamic structure in flocking behavior. Artificial Life and Robotics, 25(4):544–551, 2020. doi: 10.1007/s10015-020-00660-0.

[23] S. Thabit and A. Mohades. Multi-Robot Path Planning Based on Multi-Objective Particle Swarm Optimization. IEEE Access, 7:2138–2147, 2019. doi: 10.1109/ACCESS.2018.2886245.

[24] Y. Toshiyuki, S. Nakatani, A. Adachi, M. Kadota, and K. Ohkura. Adaptive role assignment for self-organized flocking of a real robotic swarm. Artificial Life and Robotics, 21(4):405–410, 2016. doi: 10.1007/s10015-016-0331-4.

[25] G. Vásárhelyi, C. Virágh, G. Somorjai, T. Nepusz, A.E. Eiben, and T. Vicsek. Optimized flocking of autonomous drones in confined environments. Science Robotics, 3(20), 2018. doi: 10.1126/scirobotics.aat3536.

[26] C. Yan, X. Xiang, and C. Wang. Fixed-Wing UAVs flocking in continuous spaces: A deep reinforcement learning approach. Robotics and Autonomous Systems, 131, 2020. doi: 10.1016/j.robot.2020.103594.

[27] Y. Yang, H. Yang, F. Liu, and L. Liu. Optimal control of distributed multiagent systems with finite-time group flocking. International Journal of Intelligent Systems, 35(9):1416–1432, 2020. doi: 10.1002/int.22264.