# Group of Robots Inspired by Swarm Robotics Exploring Unknown Environments

Patrick Prieto Soares
Lucas Botoni de Souza
Márcio Mendonça
Electrical Engineering Department (DAELE)
Federal University of Technology - Paraná (UTFPR)
Cornélio Procópio, Brazil
{p.prietosoares; lucasbotoni}@hotmail.com
mmendonca@utfpr.edu.br

Rodrigo H. C. Palácios
Computational Engineering Department (DACOM)
Federal University of Technology - Paraná (UTFPR)
Cornélio Procópio, Brazil
rodrigopalacios@utfpr.edu.br

Joao Paulo Lima Silva de Almeida
Federal Institute of Paraná
Jacarezinho, Brazil
joao.almeida@ifpr.edu.br

***Abstract*—This paper's goal is to apply concepts of Swarm Robotics and a Fuzzy Logic Controller (FLC) on multiple autonomous mobile robots (multi-agents) in order to complete a foraging task in semi-unknown or unknown environments. In this work the limits of the searching area are known, but everything within it is unknown. The foraging task simulates a real life application of robots on rescue missions. Although it is a simulated environment, the autonomy of the robots for this kind of operation is tested in different simulated scenarios. A reactive (real-time) architecture is used to enhance the robots global robustness on dealing with unpredictable situations. It is expected to collaborate and serve as inspiration for future works on this research area.**

***Keywords—Swarm robotics, Foraging, Unknown environment, Autonomous mobile robot.***

## I. Introduction

In recent years, the application of robots in various fields is increasing. A few examples are the rovers used by NASA to explore the planet Mars, classifying its climate and geology, and the robot roomba980, an autonomous vacuum cleaner. These tasks can be better performed when a group of autonomous mobile robots is used, in order to achieve a complex global objective by means of simple local interactions.

As it can be noticed, exploring unknown environments can be a difficult task for robots. The main problems are environment modelling, perception, localization and decision-making control [1]. Those principles are important because an autonomous mobile robot must be capable of reacting to obstacles and unpredictable events [2].

In order to develop more flexible robot architectures, different intelligent computational techniques can be applied to deal with the infinite states a robot can achieve in an unknown environment. The work [3] proposes using a Fuzzy Logic Controller (FLC) on a swarm of robots while avoiding obstacles and keeping formation.

A multiple agent system (MAS) is well defined if its dynamics reflect some synergy, that is, global behaviors emerge from the individual capabilities to reach specific objectives [4]. This concept applied in robotics results in a multi-robot system, as well as a swarm of robotics.

In general, the robots present a few MAS aspects such as: mobility, autonomy, reactivity, benevolence (agents don't have conflicting goals and always try to achieve their goals), adaptability (robots ability to adapt to changes in the environment while completing the task) and collaboration (agents share information among themselves) [5].

Thus, robots supported and developed using an adequate computational intelligent tool and architecture can be able to interact and adapt on different scenarios [5]. According to Maes concept on the reference work [6], an autonomous agent is a computational system that inhabits a complex and dynamic environment, being able to sensor and to actuate autonomously in order to achieve its goals.

Swarm robotics is a new approach to coordination of multi-robotic systems, consisting of the number of physically simple robots, without a central coordination, among other characteristics (discussed in the development of this article) which suggest to be suitable for this research.

Swarm robotics differs from traditional multi-agent systems (MAS) due to their command structures and non-hierarchical, non-centralized control. Moreover, a distributed and on-line control strategy is important to perform a stigmergy behavior. In this case, there is not a central controller or a robot leader to compute the control actions of each robot of the fleet, such as in [4].

Swarm robotics is inspired by the concepts of Swarm Intelligence. In this way, Swarm Intelligence and bio-inspired computation have become increasingly popular in the last two decades. Bio-inspired algorithms such as ant colony algorithm, bat algorithm (BA) [7]. The term "swarm" is used to refer a large group of locally interacting individuals with common goals [8]. Thus as multi-agent systems, swarm robotics systems are fully distributed and self-organized [9].

This strategy concept is indicated when there are restrictions or delays in robots communication and a fast response is required (real-time) [10].

In other words, from the cooperation among robots emerges what is called collective behavior or intelligence. Usually, it resembles a natural behavior for completing tasks such as ants foraging [11], honey bees building a nest [12], a school of fishes swimming in formation [13], but this cooperation can also be used in optimization problems such as foraging [14]. Thus, it consists on using numerous autonomous robots cooperating among themselves in order to complete tasks.

978-1-5090-6020-7/18/$31.00 ©2018 IEEE

This work has some inspiration in the strategy called Brick and Mortar, strategy used to explore minimum time in emergency situations dangerous for humans [15]. The main goal of this paper is to develop and simulate an autonomous mobile robot architecture capable of foraging for victims and rescuing them in unknown environments. It must be able to interact with the environment accordingly while detecting unknown static obstacles. In this way, the main challenge of this work is to generate global behaviors from local rules [16][17].

The robot is structured inspired on the subsumption architecture proposed by Rodney Brooks [18]. In addition, there are applied multiple robots based on the same model, cooperating in order to get better results on rescuing the victims. Both cases results are compared and discussed.

Therefore, this paper intends to contribute with a strategy based on autonomous robotics, multiple agents system and specially swarm robotics concepts for low-cost and low-processing power. Moreover, these agents perform local and limited interactions because they do not have global environmental information.

Besides of the applications cited previously, which targets were static (victims or hostages for example). The strategy of this work can be adjusted to search dynamic targets and to operate in 3D environments. That way, it can be adapted for recent swarm robotics applications, such as marine platforms repairs, rescues, and even agriculture instances, for example, pulverizing the soil using a group of drones.

This work was organized as follows: Section II discusses the development, presenting the concepts of subsumption architecture, fuzzy systems, as well as swarm robotics. Section III presents the methodology used for the simulations made in the MatLab® software, using two environments in two foraging scenarios: one robot and four robots simultaneously. Section IV presents the results obtained for the proposed conditions, and Section V contains the conclusions and future works.

## II. Development

This section explains how the subsumption architecture was applied to the robot modelling process, mentioning the sub-behaviors goals and how they work and alternate. In the sequence it is explained how the Fuzzy System was implemented to generate pulses according to the obstacles detected by the robot sensors. Lastly, which concepts of Swarm Robotics were applied to enhance the desired aspects of the robot.

### *A. Subsumption Architecture*

The subsumption architecture, proposed by Rodney Brooks, is a tool on developing robots that has brought a new perspective to the research area. Before it, the robots were based on Artificial Intelligence (AI), which were modelled based on symbolic expressions of the environment.

Therefore, small changes in the environment could collapse the robots decision-making process causing failures, unexpected and undesirable behaviors, like the robot crashing with obstacles. In this work, this architecture was chosen due to its viability on implementing real projects.

The architecture proposes that robots should behave reactively to external stimulus instead of trying to comprehend it. The robot expected complex behavior is decomposed into sub-behaviors [19]. This is justified by Zadeh's incompatibility principle, which reveals that as a system complexity grows, the harder it will be to foresee how it will behave [20].

This principle is noticed in [21], in which Braitenberg starts with a simple vehicle with a sensor and its connections to the robots motors. Then he starts adding more and different sensors connected in different ways to the robot and tries to foresee how the robot will behave to different external stimulus. He also notes that certain behaviors resemble human ones, such as liking, loving or hating something.

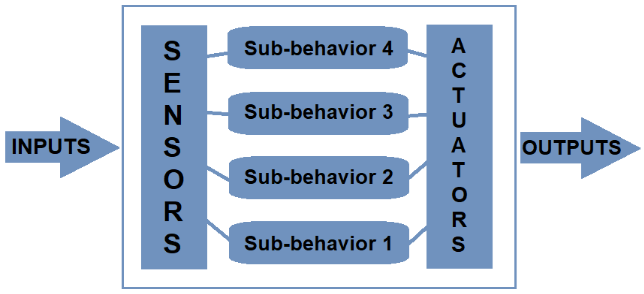


Fig. 1. Subsumption architecture.

As it can be seen on Fig. 1, every sub-behavior operates in parallel with others where they are inhibited or activated according to the sensors actual data, generating the desired output. Each sub-behavior has its own purpose and they shall be implemented one by one. After every addition, the operation and sub-behaviors alternating algorithm must be tested extensively.

In the end, this architecture allows the robot to deal with stimulus from the environment in real time, interacting with dynamic aspects of it and pursuing one or more goals. If those aspects can be noticed on the robots behavior, it suggests the robot is autonomous [19].

For this paper, the robots were responsible for three tasks: to locate all victims, to rescue them and to detect the most unknown obstacles as possible until the last victim is rescued. The complex behavior expected from the robot to operate in such circumstances and complete the goals was divided into four sub-behaviors.

The first sub-behavior implemented was free movement. It starts every iteration of the algorithm and generates an unitary pulse for the motors attached to the wheels of the robot. So, it moves forward at maximum speed if the sensors do not detect anything.

The second sub-behavior is meant to deal with obstacle avoidance. It is activated if there is any obstacle (static or dynamic) being detected by the robot. A Fuzzy Logic Controller (FLC) was used to generate the pulses to the motors.

The option for this intelligent computational tool was based on its capabilities of dealing with uncertain/imprecise information such as noise on the sensors data or dynamic obstacles (other robots operating on the same environment).

The third sub-behavior was responsible for dealing with states that presents imminence of collision. It is activated when the three ultrasonic sensors located in the front of the robot detects an obstacle closer than its safety distance. That means there is an obstacle way too close to the robot and may harm its integrity. In the simulations presented in this paper, the safety distance was defined as 5 cm, but for real life applications, it would need to be readjusted in order to fit its goals.

Therefore, in order to avoid it, this sub-behavior generates a positive and a negative pulse forcing the robot to turn its direction quickly to evade the obstacle.

This sub-behavior is also used to quick avoid other robots, since they can be considered as dynamic obstacles and in certain situations might be in collision course. The direction of the turning movement is changed after a few times it is activated. This direction-alternating algorithm was added to prevent the robot from staying stuck on the same loop of states over time.

The fourth and last sub-behavior added is responsible for the victims rescue. After locating a victim and signalizing them (the victims change colors on the simulation after being located), this component of the complex behavior will be activated. It aligns the robot to the target and controls its speed through a multivalued algorithm.

This speed control is supposed to be a safety measure on behalf of the victim, so the robot can arrive to its location in a steady and safe speed for the rescue operation. When the robot is close enough to the victim, it is considered rescued.

This mode declassifies this robot model as purely reactive because it has some sort of trajectory planning. It also contains a memory algorithm in which after a certain amount of iterations, if the robot still has not rescued the victim, it will forget it and let it be rescued by another robot in a better state. This happens because when a robot locates a target he is instantly responsible for rescuing it as its main goal.

This sub-behavior will just be inhibited by the sub-behavior of collision imminence. Thus, allowing the robot to rescue victims close to obstacles without activating other sub-behaviors. Furthermore, when a robot is busy rescuing a victim, the remaining robots close to it are responsible for avoiding the collision.

The state machine of the sub-behaviors (desired aspects of the complex individual behavior) is shown on Fig. 2, while the events are explained on Table I. As it can be noticed, the iteration starts by default on sub-behavior 1 and depending on the environment the robot reacts activating another sub-behavior. For example, if a sensor detects any obstacle within its coverage area, the robot activates the fuzzy system to generate the pulses necessary to avoid it.

TABLE I. EVENTS DESCRIPTION

| Event | Description |
|---|---|
| A | Obstacle detected by any ultrasonic sensor |
| B | Obstacle detected by the three ultrasonic sensors within the robots safety zone |
| C | Victim found |
| D | Victim still targeted to be rescued |
| E | Robots memory algorithm activates |

On the other hand, if a victim is found triggering the event C, the robot will activate the sub-behavior responsible for rescuing the victim and it stays focused on this task until it completes the task, forgets it or faces an obstacle within its safety zone.

When the sub-behavior of collision imminence is activated, it will not go back to any other state, because the highest priority of the robot was designed to be the maintenance of its integrity. It is not expected from a rescue robot to get damaged while on duty. If he keeps his integrity, it will be able to save more victims on the long run. Therefore, it is better to avoid collisions and sometimes let other robots in a better situation to rescue the victim.

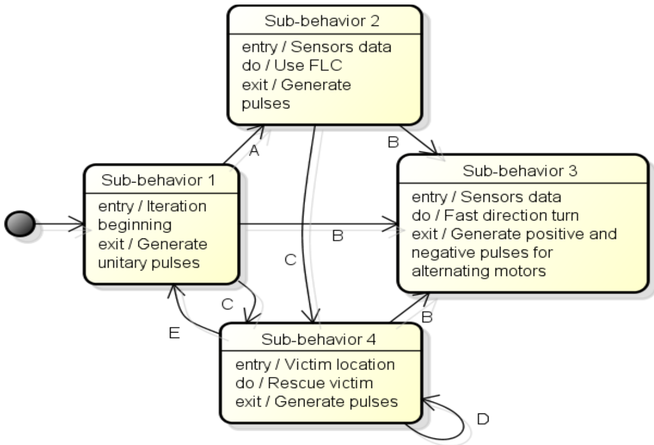


Fig. 2. State machine.

## B. Fuzzy System

A fuzzy system was developed to generate the pulses as outputs of the sub-behavior responsible for obstacle avoidance. The inputs are the actual data obtained by the robots three ultrasonic sensor: frontal sensor (FS), left sensor (LS), right sensor (RS).

Each input has a universe of discourse that goes from 0 to the maximum range of the sensor defined by the user on the beginning of the simulation (12 cm in the simulations presented in this paper). This universe of discourse is divided into five membership functions: two trapezoidal at the edges and three triangular at the center. The outputs follows the same pattern but its universe discourse is [0,1].

The system contains 125 rules based on the membership functions of the inputs and adjusted on empirical trial. An example of the surfaces generated by those rules can be seen on Fig. 3; *we* and *wd* are the angular velocities for the left and right wheels, respectively.

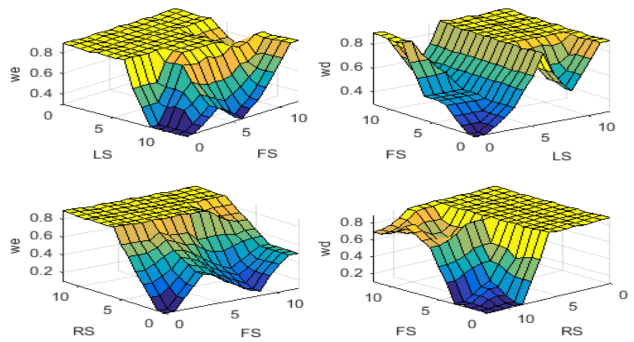


Fig. 3. Fuzzy surfaces for *wd* and *we.*

## C. Swarm Robotics

When using multiple robots, it is important to define how the interaction between the robots will be, for example, in this project there is not a command central, that way each robot is responsible for making decisions and exchange information solely with other agents close to them. In this project, concepts of Swarm Robotics were applied in order to guarantee better results and higher global robustness to real-time problems. A few concepts of this research area were used while programming the simulation algorithm. They can be seen on Table II.

It is important to mention that the local communication allows the robots to simulate data transmission among them just if the robots sensors, thus not having any central responsible for controlling the robots, are detecting them.

TABLE II. SWARM ROBOTICS CONCEPTS

| Concept | Description |
|---|---|
| Simplicity | Robots architectures presents low computational complexity and simple physical structure |
| Local Communication | Robots interact solely with the environment and other agents within theirs sensors coverage area |
| Decentralized Control | Each robot takes its own decisions, independently of other agents operating around |
| Parallelism | Each robot is in charge of multiple tasks (search and rescue victims) |
| Scalability | If a robot presents failures, the remaining is able to maintain the swarm operating accordingly |

When activating the communication algorithm any kind of information can be shared for example, how many victims the robots have already rescued at the moment, which sub-behavior is currently active, the robots position and direction, how many unknown obstacles have already been detected, among others.

Due to those aspects, the robots swarm presents certain flexibility against unpredictable events, for example sensors noise, or in cases which a robot presents failures and the remaining assume its function. Applying them and summing with the decentralized and reactive approach of the subsumption architecture, the robot developed has enough autonomy to operate in unknown environments.

## III. SIMULATION

Simulated environments with 2-D animation were developed to test and validate the robots architecture on the MatLab® software. According to Russel and Norvig [2], in order to attest the robots autonomy, it needs to successfully complete its tasks on three different scenarios. Therefore, there were developed two different rescue environments for the robots operation in two different scenarios, where it is supposed to avoid collisions with static and/or dynamic obstacles (other robots). The environments can be seen on Figs. 4 and 5.

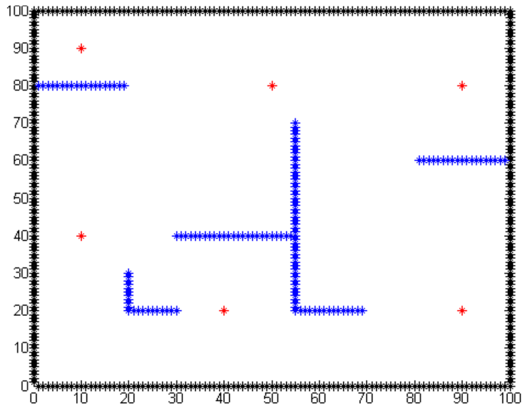

Fig. 4. Environment I.

After completing each one of the simulation iterations, a frame was taken in order to generate a simulation movie for better analysis of the complex situations the robots might get into. In those movies there were applied 30 frames per second for a more continuous image, that way the processing time of the simulation, iterations and the video time will be different.

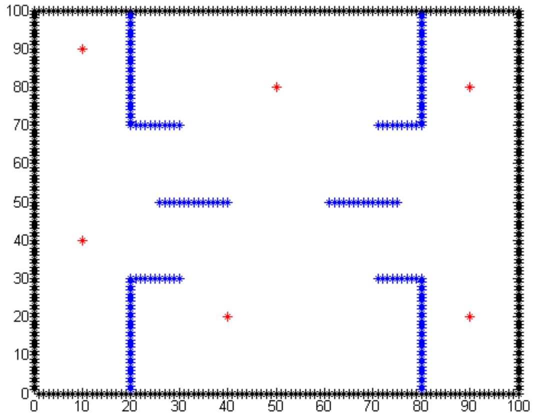

Fig. 5. Environment II.

### C. Simulated Environments

The environments contain three major characteristics: victims, statics obstacles and rescue area. The rescue area (black dots) delimitates the region where the robot is supposed to operate. In a real operation it would be the searching area. This limit covers a 100 cm x 100 cm area and is considered as a known obstacle to the robot. However, these dimensions are at simulation level and can be changed, e.g. to m, in a search for victims in a real block.

The static obstacles (blue dots) have different formations and quantity in the environments. Throughout the simulation the obstacles are going to be detected by the robot, changing its color to black (known obstacles).

All the rescue environments contain six victims (red dots when unknown or green when signalized to be rescued by a robot) with the same location. Therefore, the difficulty of each stage is related exclusively to the obstacles placement.

### D. Results Parameters

In the beginning of the simulation, a few variables were defined such as the initial position of the robots, its directions, the wheel radius and base length, necessary for the kinematic model used to convert the pulses into movement. The kinematic model and the FLC used is similar to the work about autonomous robots [22], and so, because of its low cost and low computational complexity, it can be embedded on prototypes exploring real scenarios using this study.

In addition, were defined the robots sensors arrangement, its range and beam angle, and the rescue distance (it means how distant the robots center of mass got to be from the victim for it to be considered rescued). The user can also pick the colors that differentiate each robot.

First, it is used one single robot to perform the rescue operation, and then another test is run with three more robots. The simulation processing time is used as a computational comparison, while the amount of iterations is used for a real-time one. The processing time of the tests were obtained using the software functions *tic* and *toc*.

## IV. Results and Discussion

Each subsection presents its own results and discussion. In the first environment, the pulses (*wd* for the motor attached to the right wheel and *we* to the left wheel) referent to each robot were also plotted, they can be understood following the robots track (every 10 iterations a dot is plotted) and the Table III. The pulses are in fact the control actions generated in the end of the iteration to be used on the following one, thus the robot presenting the behavior expected in its current state.

TABLE III. PULSES LOGIC

| Pulses | Description |
|---|---|
| Both positive | Robot moves forward |
| *we* > *wd* | Robot turns right |
| *wd* > *we* | Robot turns left |
| One positive and one negative | Robot turns at the same location |

### *E. Environment I*

First, there was used one robot initially located at [40, 10] and direction 0º.

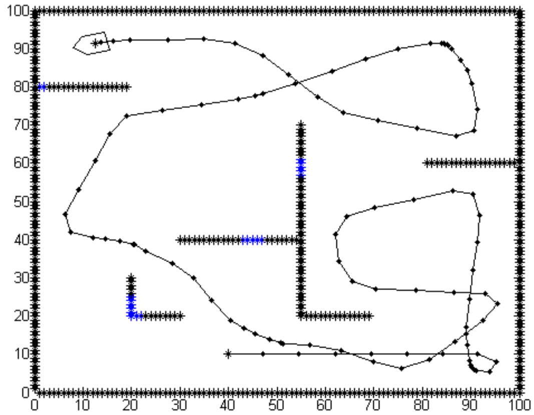

Fig. 6. Environment I with 1 robot.

It rescued all victims by itself after 976 iterations and approximately 1 minute and 31 seconds of simulation processing. The simulation can be seen on the link: https://youtu.be/IOIFOFHOBd0. The last frame is shown on Fig. 6, where it can also be noticed that almost all the obstacles were detected. The pulses can be observed on Fig. 7.

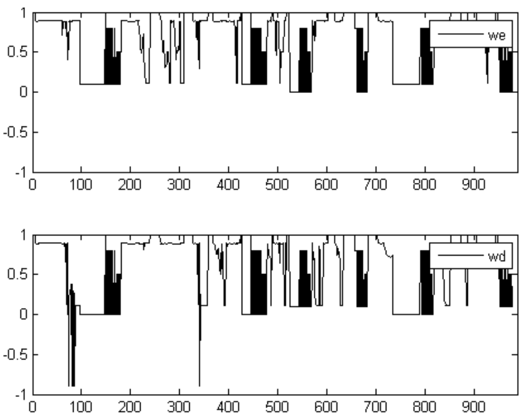


Fig. 7. Pulses for environment I with 1 robot.

Then, three more robots were added initially located at the positions, [60, 10], [90, 50], [90, 70] and directions 180º, 90º and 270º respectively. That way, the robots 1 (black track) and 2 (blue track) were facing one each other, just like robots 3 (cyan track) and 4 (magenta track).

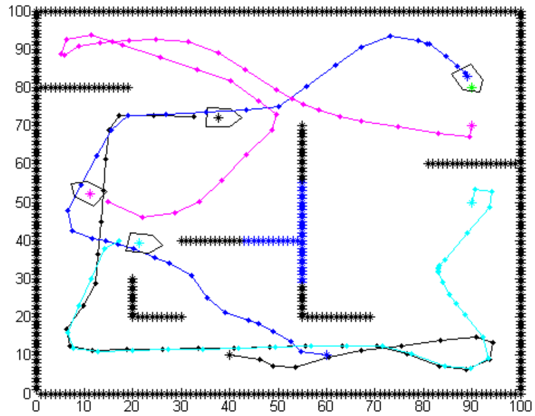

Fig. 8. Environment I with 4 robots.

This test lasted 346 iterations (for all four robots), until every target was rescued, during 2 minutes and 57 seconds. The execution time of the experiments may change due to different processors, so being used as a parameter of comparison between the topologies of one and four robots. The obstacles on the center of the environment were not detected because no robot got close to them. The results can be seen on Figs. 8, 9, 10, 11 and 12. The plotting colors are related to the corresponding robots track color and the simulation can be accessed through the link: https://youtu.be/nbTAUni8jso. It can be noticed negative pulses at the moment the imminence of collision sub-behavior is activated.

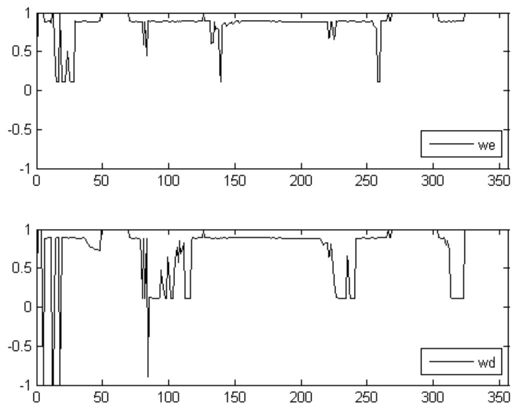


Fig. 9. Pulses of robot 1 for environment I.

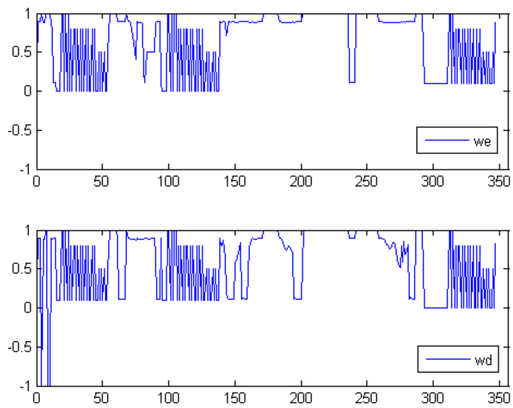


Fig. 10. Pulses of robot 2 for environment I.

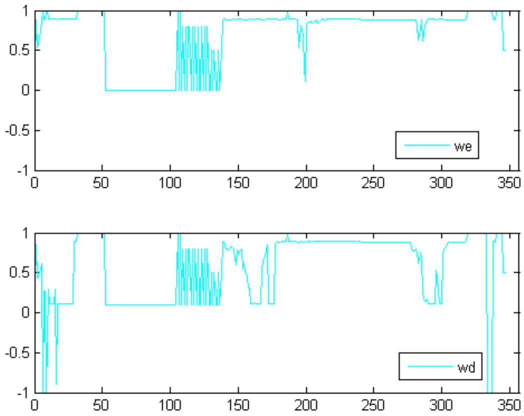


Fig. 11. Pulses of robot 3 for environment I.

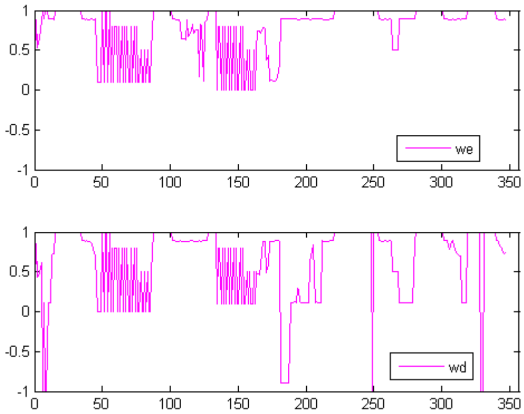


Fig. 12. Pulses of robot 4 for environment I.

## *F. Environment II*

Finally, the solo robot rescued every victim after 11 minutes and 34 seconds, lasting 5536 iterations. It was able to detect every obstacle by itself. The result can be seen on Fig. 13, or through the link: https://youtu.be/4_Vn9qcSnJk. It can be noticed that the robot got stuck for a long period on the left side of the environment because of the narrow corridor formed by the obstacles.

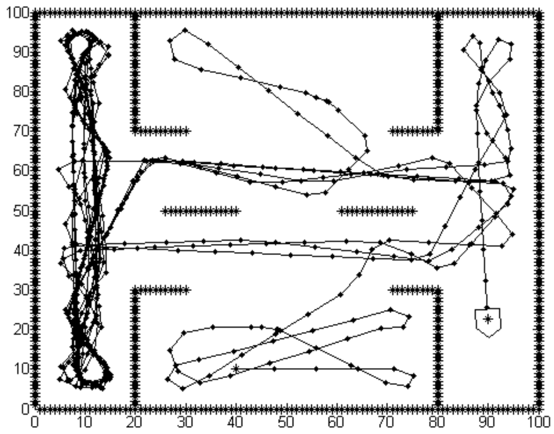

Fig. 13. Environment II with 1 robot.

Then, for the second phase using the complete set of robots, the simulation lasted 599 iterations during 5 minutes and 1 second, and every obstacle was detected. It can be seen how the robot operated in this scenario on Fig. 14 or accessing the link: https://youtu.be/tNwmAfvhn8U.

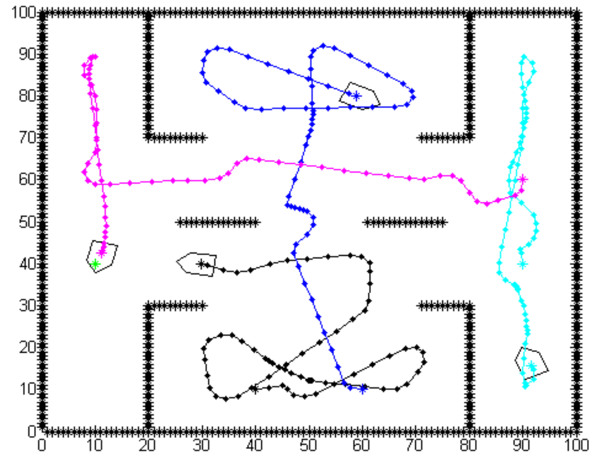

Fig. 14. Environment II with 4 robots.

## *G. General Results*

The robots were successful on every environment, rescuing all the victims and detecting as many unknown obstacles as possible even if it lasted longer on some of the tests. However, what happens during tests is unpredictable just like in real life operations so it is plausible some simulations take longer to be completed.

Some situations can be seen on the videos, such as on environment II with multiple robots in which the robots sometimes get stuck alternating between avoiding the obstacle wall and the robot passing close to it (it can also be noticed on the pulse figures). In addition, on the same environment with just one robot, it can be seen the victim located on [x = 10, y = 90] passing through the frontal and right sensor blind spot. Therefore, all the results were compiled and shown on Table IV.

TABLE IV. SIMULATION RESULTS

| **Environment** | **1 Robot** | | **4 Robots** | |
|---|---|---|---|---|
| | ***Processing time*** | ***Iterations*** | ***Processing Time*** | ***Iterations*** |
| I | 01'31" | 976 | 02'57" | 346 |
| II | 11'34" | 5536 | 05'01" | 599 |

At first, determining the search time by the number of robots in each scenario is not a trivial calculation; however, it was expected that the processing time using four robots would be faster than using one, due to the decrease of the search area of each robot. Another aspect to be considered are the maneuvers to avoid collisions with the scenario and the other robots.

It does not happen just because there are more robots completing the same tasks, but also because the interaction among the robots create new situations, regularly forcing the robots through different paths, which leads the robots to find the victims faster than what one single robot would take.

It was verified that the tests using multiple robots had considerable better results, even though they are physically and algorithmically simple. This better performance is expected in this research area and in real life rescues could be the difference between life and death for injured victims.

# V. CONCLUSION

Initial simulated results showed autonomy. There were no kind of collisions and all victims were rescued without any interference of the user. The robots were responsible for taking their own decisions and completing the tasks.

Thus, the hierarchy and logic behind each of the sub-behaviors, and the alternating system implemented on the subsumption architecture worked as desired, guarantying higher flexibility to unpredictable situations.

In short, these results showed five robots (one working solo on the first scenario and later four agents working together on the second scenario) with autonomous aspects, avoiding obstacles e rescuing targets, besides presenting desired aspects such as reactivity, parallelism, scalability, benevolence, among others.

Those aspects grant flexibility and autonomy to the robot model applied in this work, with a good probability of the goals being achieved faster, furthermore enhancing its performance.

It is expected for future works to apply different intelligent computational techniques (e.g. Fuzzy Cognitive Maps) to generate the outputs of the sub-behaviors, consuming less processing time. Implement larger environments where a bigger swarm can be applied to test different techniques such as Group Explosion. In addition, one base can be created for the robots to return every time they rescue a victim or to recharge the battery when it gets low. Also, different strategies can be investigated for other applications such as agriculture.

## REFERENCES


[1] E. D. S. Costa and M. M. Gouvea Jr., "Autonomous Navigation in Dynamic Environments with Reinforcement Learning and Heuristic," 2010 Ninth International Conference on Machine Learning and Applications, Washington, DC, 2010, pp. 37-42.

[2] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 3rd ed. Upper Saddle River, USA: Prentice Hall, 2009, 1132 p.

[3] A. S. Handayani, N. L. Husni, S. Nurmaini and I. Yani, "Formation control design for real swarm robot using fuzzy logic," 2017 International Conference on Electrical Engineering and Computer Science (ICECOS), Palembang, 2017, pp. 77-82.

[4] R. Calvo, J. R de Oliveira, M. Figueiredo, R Romero, “A bioinspired coordination strategy for controlling of multiple robots in surveillance tasks,” International Journal on Advances in Software, Vol. 5, 2012, pp. 146-165.

[5] M. Mendonça, I. R. Chrun, F. Neves, L.V. R. Arruda, “A Cooperative Architecture for Swarm Robotic based on Dynamic Fuzzy Cognitive Maps”, Engineering Applications of Artificial Intelligence, Volume 59, pp. 122-132, ISSN 0952-1976, 2017.

[6] B. İ. Kumova and S. B. Heye, "A survey of robotic agent architectures," 2017 International Artificial Intelligence and Data Processing Symposium (IDAP), Malatya, 2017, pp. 1-8.

[7] X.-S. Yang, Z. Cui, R. Xiao, A. H. Gandomi, and M. Karamanoglu, Eds., Swarm Intelligence and Bio-Inspired Computation Theory and Applications, 1st ed. Amsterdam, Netherlands: Elsevier, 2013, 450 p.

[8] J. C. Barca and Y. A. Sekercioglu, “Swarm robotics reviewed,” Robotica, Vol. 31, No. 3. pp. 345 – 359, 2013.

[9] E. Bonabeau, M. Dorigo, and G. Theraulaz, Swarm Intelligence: From Natural to Artificial Systems, 1st ed. New York, USA: Oxford University Press, 1999, 320 p.

[10] Á. Kisdi and A. R. L. Tatnall, “Future robotic exploration using honeybee search strategy: Example search for caves on Mars,” Acta Astronautica, Vol. 68, Issues 11–12, 2011, pp. 1790-1799.

[11] M. Rosalie et al., "Area exploration with a swarm of UAVs combining deterministic chaotic ant colony mobility with position MPC," 2017 International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, USA, 2017, pp. 1392-1397.

[12] J. H. Lee, C. W. Ahn and J. An, "A honey bee swarm-inspired cooperation algorithm for foraging swarm robots: An empirical analysis," 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Wollongong, NSW, 2013, pp. 489-493.

[13] S. Tolba, R. Ammar and S. Rajasekaran, "Taking swarms to the field: A framework for underwater mission planning," 2015 IEEE Symposium on Computers and Communication (ISCC), Larnaca, 2015, pp. 1007-1013.

[14] O. Zedadra, H. Seridi, N. Jouandeau and G. Fortino “A Cooperative Switching Algorithm for Multi-Agent Foraging. Engineering Applications of Artificial Intelligence, Vol. 50, pp. 302-319, April 2016.

[15] H. Hexmoor, Essential Principles for Autonomous Robotics (Synthesis Lectures on Artificial Intelligence and Machine Learning), 1st ed. San Rafael, CA, USA: Morgan & Claypool, 2013, 156 p.

[16] L. Wang, A. D. Ames and M. Egerstedt, "Safety Barrier Certificates for Collisions-Free Multirobot Systems," in IEEE Transactions on Robotics, vol. 33, no. 3, pp. 661-674, June 2017.

[17] J. Kubica, A. Casal and T. Hogg, “Complex behaviors from local rules in modular self-reconfigurable robots”, 2001, IEEE International Conference on Robotics and Automation. 1. 360 - 367 vol.1. 10.1109/ROBOT.2001.932578.

[18] R. A. Brooks, “A Robust Layered Control System for a Mobile Robot,” IEEE Journal of Robotics and Automation, vol. 2, no. 1, pp. 14-23, Mar. 1986.

[19] R. A. Brooks, “Intelligence without representation,” Artificial Intelligence, vol. 47, n. 1–3, pp. 139–159, 1991.

[20] L. A. Zadeh, “A Fuzzy Algorithmic Approach to the Definition of Complex or Imprecise Concepts,” Journal os Man-Machine Studies, Vol. 8:249-291, 1976.

[21] V. Braitenberg, Vehicles: Experiments in Synthetic Psychology., 1st ed. Cambridge, USA: MIT Press, 1986, 186 p.

[22] M. Mendonça, E. S. da Silva, I. R. Chrun and L. V. R. Arruda, "Hybrid Dynamic Fuzzy Cognitive Maps and Hierarchical Fuzzy logic controllers for Autonomous Mobile Navigation," 2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Vancouver, BC, pp. 2516-2521, 2016.