

Article

Robot Swarm Navigation and Victim Detection Using Rendezvous Consensus in Search and Rescue Operations

Gustavo A. Cardona ¹ and Juan M. Calderon ^{2,3*}

¹ Department of Electrical and Electronics Engineering, Universidad Nacional de Colombia, Bogota 110131, Colombia; gacardonac@unal.edu.co

² Department of Electronic Engineering, Universidad Santo Tomás, Bogota 110131, Colombia

³ Department of Computer Science & Engineering, Bethune-Cookman University, Daytona Beach, FL 32114, USA

* Correspondence: juancalderon@usantotomas.edu.co or calderonj@cookman.edu

Received: 31 December 2018; Accepted: 21 March 2019; Published: 25 April 2019



Abstract: Cooperative behaviors in multi-robot systems emerge as an excellent alternative for collaboration in search and rescue tasks to accelerate the finding survivors process and avoid risking additional lives. Although there are still several challenges to be solved, such as communication between agents, power autonomy, navigation strategies, and detection and classification of survivors, among others. The research work presented by this paper focuses on the navigation of the robot swarm and the consensus of the agents applied to the victims detection. The navigation strategy is based on the application of particle swarm theory, where the robots are the agents of the swarm. The attraction and repulsion forces that are typical in swarm particle systems are used by the multi-robot system to avoid obstacles, keep group compact and navigate to a target location. The victims are detected by each agent separately, however, once the agents agree on the existence of a possible victim, these agents separate from the general swarm by creating a sub-swarm. The sub-swarm agents use a modified rendezvous consensus algorithm to perform a formation control around the possible victims and then carry out a consensus of the information acquired by the sensors with the aim to determine the victim existence. Several experiments were conducted to test navigation, obstacle avoidance, and search for victims. Additionally, different situations were simulated with the consensus algorithm. The results show how swarm theory allows the multi-robot system navigates avoiding obstacles, finding possible victims, and settling down their possible use in search and rescue operations.

Keywords: swarm-robotics; rendezvous consensus; robot navigation; victim-detection

1. Introduction

Natural disasters and wars are some of the worst events that humanity has had and will have to face, since in these kinds of situations it is almost impossible to evacuate people in the affected area, causing many more deaths and having a devastating impact. However, the catastrophe is not the only problem, considering that after a catastrophe, the area is still dangerous due to issues such as landslides, debris, damage to the electrical system, and gases, among others. This is a reason rescue teams should be cautious because their lives are still at risk. It causes the survivor search to take longer losing valuable time, which is the most important resource in order to keep survivors alive. It opens the door to some researchers that have taken action in the matter. They involve robotics to help paramedics and rescuers in some

tasks, improving their performance. For instance, in [1] they show some advantages, disadvantages, and difficulties that were observed in the use of different robotic platforms that performed SAR (Search And Rescue) tasks on the World Trade Center terrorist attack. This is a good benchmark for research related to this matter that is being developed in recent years and it is present in the robotics road-map.

Robotics is presented as a viable alternative thanks to the use of sensors and actuators which allow to minimize human failures. Taking this into account, some works [2–4] explore the idea of robotics platforms with a hybrid mechanics design. The proposed mechanism allows the robot to navigate on land and air avoiding get stuck in complex environments. Additionally, it is worth highlighting the amount of information that is required to be processed by a single robot in order to satisfying all tasks. Consequently, some researchers focus on the design of semi-autonomous robots as presented in [5,6]. On the other hand, multi-robot research has shown a different alternative to solve the aforementioned task, for instance, the advantages of using robotic swarms in exploration tasks is mentioned in [7,8]. Advantages such as coverage of a larger area in less time in contrast to the use of a single platform. In the same way, the robustness and flexibility of the robotic swarm is higher than a single robot, considering that if a robot from the swarm fails or get stuck, the operation and the success of a mission will not be compromised.

As was previously mentioned, robotics has become important in SAR operations. Taking into account that by using multi-robot systems some tasks such as exploration, mapping, search and location of victims and classification, among others can be optimized. Tasks related to the exploration of an unknown area become the first important step in wanting to save lives as well as mapping the area to locate victims. The work shown in [9,10] uses robot network to accomplish navigation and exploration tasks, making use of Voronoi tessellation, and graph theory. On the other hand, it is necessary to take into account the control law that allows a swarm to create attraction towards the victims and repulsion towards obstacles and robots in order to avoid collisions. This approach is achieved using attractive and repulsive potential functions as shown in [11], among other works that have been made to solve the different challenges that entail the exploration task.

Taking into account that it is difficult to have access to a real disaster scenario, it is worth mentioning that the present work is carried out by simulation. In robotics, simulation has become an essential tool for research, due to the fact that in recent years simulators have reached a high realistic level in comparison with the real environment as shown in [12]. On the other hand, simulators allow testing algorithms without putting people or equipment at risk. In addition, without losing performance if simulator keeps the necessary characteristics of the real world as shown in [13]. As previously mentioned, is difficult to be able of performing the algorithm in a real environment, in addition to this, having available a high number of drones with all the necessary sensors require a huge investment. Reasons why the present algorithm is carried out in V-Rep, which has a good performance in this kind of tasks that involve the search an rescue.

The main contributions of the present paper are threefold. First, the development of a navigation swarm system based on the artificial potential approach, which uses attraction and repulsion forces to keep swarm communication between agents and avoiding collisions, respectively. Additionally, an attraction force pointing to the location of a possible victim is added to the navigation system in the same way as proposed in our previous projects [14–16]. Second, the policy of sub-swarm generation when some robots detect a possible victim, through creating a weighted graph based on the distance in order to perform a k-nearest algorithm and then break the agents' links to the main swarm. Finally, with the generated sub-swarm is performed a control formation algorithm to put all of the agents close to the possible survivor and then apply a sensor network estimation consensus. The existence of a victim is determined by the sensor network consensus modeling each sensor with least-squares theory.

The other sections that complete this paper are organized as follows: Section 2, Related Work, in which the benchmark in robotics search and rescue is exposed. Section 3, Navigation Process, explains how the navigation algorithm is carried out using artificial potential functions and how the sub-swarm is generated.

Section 4, Victim Detection, shows how the formation control around the victim is implemented and then the consensus applied in the sensor network estimation. Section 5, Simulation and Results, shows how the theory was applied in a swarm of drones through V-rep simulator and finally, Section 6 corresponds to the document conclusions and some future work alternatives.

2. Related Work

Robotic swarms face a variety of challenges such as navigation, communication and collective decision making. These challenges generate in the swarm characteristics robustness and reliability to the system allowing to perform tasks such as exploration and location of targets. In navigation task, there is a great variety of algorithms such as presented in [17] which addresses the characteristics of different algorithms based on swarm intelligence found in nature, such as ant colony, swarm of bees, and wolf pack, among others. The characteristics of the different algorithms generate a great variety of possibilities to choose according to the needs of the challenge. In [18] they use two bio-inspired algorithms; The first algorithm is the particle swarm optimization algorithm, which has good results in exploration work, but sometimes the swarm stagnates in local minima. The second algorithm is the Darwinian particle swarm optimization algorithm, where the swarm is divided into small sub-swarms which share the best solution with the other sub-swarms avoiding being trapped in local minima.

As mentioned before, the communication in robotic swarm systems is a research topic that tries to solve problems such as noise due to interference in the environment, loss of information and limitations in the transmission distance and bandwidth of the system. In [19] a low resource consumption communication protocol called RRTLAN is presented which can be implemented in an 8-bit micro controller. An asynchronous communication protocol is shown in [20]; this protocol was tested on robotic platforms that search for a moving target using a multi-robot cooperation strategy. In [21] they use the electrical networks of a collapsed building as a communication channel for a cooperative robotic system that is performing exploration tasks. In [22] a communication protocol called Chopin is proposed and implemented, this protocol was implemented using Robotics Operating System (ROS) in real time. The communication process in a swarm system can be a problem depending on the algorithm implemented, considering that they are systems that manage a great amount of information.

Given that a robotic swarm system has a large amount of information coming from each agent of the swarm, a method is required to calculate a consensus value of the acquired information and then a decision is made collectively. In [23] they present a security system for the detection of the intruders which makes a consensus among the sensors located in common areas. In image sensors such as thermal cameras, wireless cameras and security cameras present a large amount of noise due to the environment and lighting, in [24] they use different data fusion techniques implemented in the pixels of the image captured from different cameras. This allows giving greater reliability of the information given by the image resulting from the fusion of the data. On the other hand, platforms designed for search and rescue work need portable energy sources such as batteries or renewable systems. Energy management is very important to allow the platform operates as long as possible without interruptions. In [25] an algorithm is proposed to find an optimal path between two points and using a cross-layer algorithm to improve energy performance. In [26] they propose a system where the transmission of information is not carried out without first reaching a consensus among the different agents. The preliminary consensus is made to eliminate noise faults or malfunction of some sensors and thus achieve energy savings since no redundant information is transmitted.

Finally, the victim's detection is the aim of this robotics application. The detection of survivors trapped under debris is a problem in which robotics can help search and rescue teams, unfortunately, there are scenarios that are not allowed to perform frequently tests to evaluate the performance of robotic

systems. In [27] they present a model of the respiratory signal resulting from an ultra-wideband radar, which allows the detection of this signal through obstacles. In [28] they perform mathematical methods to separate the signals given by the ultra-wideband radar, allowing the identification of a person's vital signs. In [29] they present an app which uses the mobile phones of the rescue teams, allowing to record the sounds and estimate the victim's location through the use of a central processing system. Some of the applications of robotic swarms and cooperative robotics in search and rescue are presented in [30]. Some of the mentioned applications stand out; the creation of communication networks as in [31] where a swarm of drones equipped with WIFI and GPS modules is used to create a network, allowing the rescue team to coordinate the search of survivors in case the local communication system is out of service. Another application is the recognition of fire zones without compromising the integrity of firefighters. An example of terrain recognition is exposed in [32] where a 3-dimensional map is created by exchanging information between the platforms that perform the exploration of the environment.

In consequence of the nature of disasters, it is difficult to test robotics platforms that provide support to search and rescue tasks. In [33] they use V-REP for the creation of simulation environments, in addition to this, sensors and actuators can be integrated into predefined or user-created platforms. In [7] present an analysis of programs that allow the simulation of multi-robot systems for the development of bio-inspired algorithms. After carrying out the tests in simulation programs, it is necessary to test the prototypes in an environment with characteristics similar to those found in a disaster. In [34], the Department of National Security and the National Institute of Standards and Technology are coordinated to design three tests that will allow the evaluation of requirements identified by rescuers. In [35] they give standards for the elaboration of a scale scenario and thus be able to analyze the performance of the platforms using sensors and performance matrices. In [36] they develop a test scenario that emulates a disaster zone in which they carry out performance tests of a navigation algorithm in multi-robot systems.

As shown previously, Assistance in disaster areas is a fairly large and necessary field of research. The robots for search and rescue of victims have emerged as a part of the solution to several challenges that must be solved. When we look at the large number of researches that have been done on the subject, it seems that everything is done. However, our approach shows that there are still details that can be improved such as, other works use the concept of swarm navigation but almost all of them focus on the connectivity maintenance. In this work we allow the system to disconnect agents from the swarm depending on possible victim detection. In this way, once a possible victim is detected, a group of robots disconnects from the main swarm that keeps navigating. This sub-swarm determines if there is or not a victim based on a consensus algorithm. Additionally, other authors consider the detection of a victim based only on the performance of a sophisticated type of sensor. In this work, after applying a formation control to sub-swarm, we proposed a classification between victim or false positive based on consensus approach over a modeled sensor network using minimum least squares theory.

Finally, as a method to test the proposed swarm navigation system, an environment with obstacles, victims and aerial robotics were simulated using V-Rep [33]. This is virtual robotics environment tool that allows to proximate to the real implementation approach.

3. Navigation Process

In nature, there are several species that display swarm behavior, such as bacteria, insects, birds, fish, lions, hyenas, and horses among others. As can be expected, these groups are composed of individuals, who possess specific distinct capabilities and behaviors, but when they all get together as one group and operate as such, they will behave differently in order to work as a group. One of the more notable examples of this group behavior is a swarm of honey bees choosing a place to build a new honeycomb, this phenomenon was presented by Seeley et. al. in [37], whose paper showed that this swarm behavior, which

takes place during the spring and summer, it is best exemplified when the colony outgrows its honeycomb and then proceeds to divide itself to find new grounds and then gets back together as a swarm once it finds the correct location to build a bigger hive. To be more specific, The new site selection begins with several hundred scout bees, who leave the colony in search of potential new sites, once the scouts find an appropriate candidate, they return to the hive in order to report with a wiggle dance where the potential new place is located. Once the scouts select the correct location from the pre-selected sites, they will steer the rest of the swarm to the new location by chemical stimulation. After the selection step is completed, the division process of the hive begins, and the old queen with nearly half of the colony leaves the hive to build the new one, while a younger queen stays with the other half of the colony in the old honeycomb.

It is also important to highlight that the ability of the swarm to navigate can be affected by a set of critical factors, such as collisions between themselves, large obstacles in the route, pheromone communication errors, erroneous scout indications and even poor sense of direction.

We can represent the swarm of bees using graph theory with the application of the concepts worked on by Mesbahi in [38]. One can describe the system by defining N agents representing each member of the bee swarm, or in the case of our application a robot, and a connection representing the communication and transmission of information between the agents, either bees or robots. As mentioned by Mesbahi, we first need to define the agents of the system (V, E) , where V is the set of nodes in the system and E stands for the topology of the system, we also need to define $N(i)$ as the neighborhood or adjacent nodes to node i . Regarding the links between the agents, we can describe them as a dynamic system.

It is advantageous to apply graph theory to control swarm behavior, because it allows us to apply concepts such as the Laplacian matrix, which represents the system and allows the consensus analysis, connectivity grade calculation, and applying a control law for the whole system. In addition to that, we assume that the system has moderated sensor noise, links are bidirectional given the capability for any robot to share and receive information with the agents within own communication range. The design of the swarm navigation algorithm focuses on the generation of simple navigation rules that are applied to each of the agents individually. However, the sum of the individual decisions of the agents generates a collective behavior typical of a swarm. The applied criteria to the simple navigation algorithm of the swarm agents are based on Reynolds rules [39]. These rules settle down the base conduct for the generation of swarm behavior, such as (1) avoid collisions with neighbors, (2) match speed and direction with neighbors, and (3) stay close to neighbors.

Before establishing the interaction policy to fit the Reynolds' rules, it is necessary to define the agent concept and the kinematic model of the robotics agent used in the current work. Each agent is described as a point in the space with position and velocity. We assumed that each agent can detect the position and velocity of the rest of the swarm agents. The swarm agents interaction is represented by graphs (V, E) where $V = \{1, 2, \dots, N\}$ is a set of nodes and $E = \{(v_i, v_j) : i, j \in V, i \neq j\}$ represents the communication and sensing topology between the i^{th} agent and each of the other j^{th} swarm agents as depicted in Appendix A.

The dynamic model employed for the robot is a simple integrator model. Without loss of generality, since as exposed in [40] it is possible to impose integrator dynamics for multiple robot structures employing kinematic control

$$\dot{x}_i = u_i, \quad (1)$$

where \dot{x}_i represents the linear speed of a robot and u is the control signal applied to the system in order to impose the desired behavior. The navigation approach in this work consists of the displacement of a robot swarm through an environment $S \in \mathbb{R}^2$, in which, on the one hand, we have areas clear to move composed by the set S_c and on the other hand, S_o the areas with presence of obstacles that forbid the robots to move in it (Note that $S = S_c \cup S_o$).

Interaction between swarm agents: The interaction of the swarm agents is defined by the attraction and repulsion strategy allowing the navigation into the environment. This strategy allows the agents to keep a comfortable distance over the other swarm agents. The attraction parameter tries to maintain in close proximity every element of the swarm and thus gives the group a mechanism to remain grouped together. The attraction parameter is defined by (2)

$$-k_a x_{ij} \quad (2)$$

where k_a is the attraction force and x_{ij} is the distance between the agent i to j , this mechanism can be local (with a restriction by the sensing range) or global (agents can move other agents from the group regardless of how far they are).

The repulsion parameter allows the agent to keep a comfortable distance from the other members of the swarm. This avoids collision between agents when the swarm is moving. The repulsion mechanism can respond in two different ways. First one, when a comfortable distance is reached and its parameter is expressed by (3)

$$[-k(\|x_{ij}\| - d)](x_{ij}) \quad (3)$$

where $\|x_{ij}\| = \sqrt{(x_i - x_j)^T(x_i - x_j)}$, $k > 0$ is the repulsion magnitude and d is the comfortable distance between i^{th} agent and the j^{th} agent. On the other hand, if two agents are too close to each other, the parameter is represented by (4).

$$k_r \exp\left(\frac{-\frac{1}{2}\|x_{ij}\|^2}{r_s^2}\right)(x_{ij}) \quad (4)$$

where $k_r > 0$, represents the repulsion magnitude and $r_s > 0$, is the repulsion range.

Interaction between agents and obstacles: The obstacles produce a repulsion force over the agents. This repulsion is modeled using (4) where $k_r > 0$, and r_s is related to the size of the obstacle.

Interaction between agents and victims: The victims exert an attraction force, where $k_r < 0$ and r_s is proportional to the victims density in the near area.

Sub-Swarm Generation: Every agent is exposed to different attraction and repulsion forces, which help to keep the swarm coherent and at the same time pushes the swarm towards the goal. The victims generate an attraction force over close agents and this force can disturb the normal behavior of the swarm and reduce the velocity of the agents in proximity to the victims. The k_r magnitude was put in place so that it can stop the closest agents to the victims. When agents are navigating close to the victims, at least one agent must stop near the victims. Once at least one agent has stopped close to the victim and generated the alarm about victim detection, there is a relevant variation in the process that has to be exposed, which is the generation of a sub-swarm due to the localization of possible victim. The creation of a sub-swarm is generated by K-nearest neighbors approach. The K-nearest algorithm behaves as a classifier by selecting as his name indicates the k closest robots j to robot i . When the neighborhood has been selected we create a graph $G_{ss} = (V_{ss}, E_{ss})$ where V_{ss} is composed by the neighbors of robot i and E_{ss} is constructed taking into account a weighted function $W_{ss} : E_{ss} \rightarrow \mathbb{R}$ in the following way $W_{ss} = \frac{1}{x_{ij}}$. Allowing in this way, that the closer robots have more relevance to the classification of robot i between take itself to the sub-swarm or remain in the principal.

As can be seen from Figure 1 the sub-swarm process is given when some robots in the network detect a possible victim. In the instant when a victim is detected the main inconvenience is to decide which robots will remain at the swarm and which of them will generate a new one. Consequently with the K-nearest algorithm in Figure 1 is possible to appreciate when a non-filled dot representing the robot that will be classified as a member of the sub-swarm blue dots or not green dots. It is possible to note that he is assigned to the sub-swarm becoming to a blue dot and breaking communication links with the rest of the main swarm.

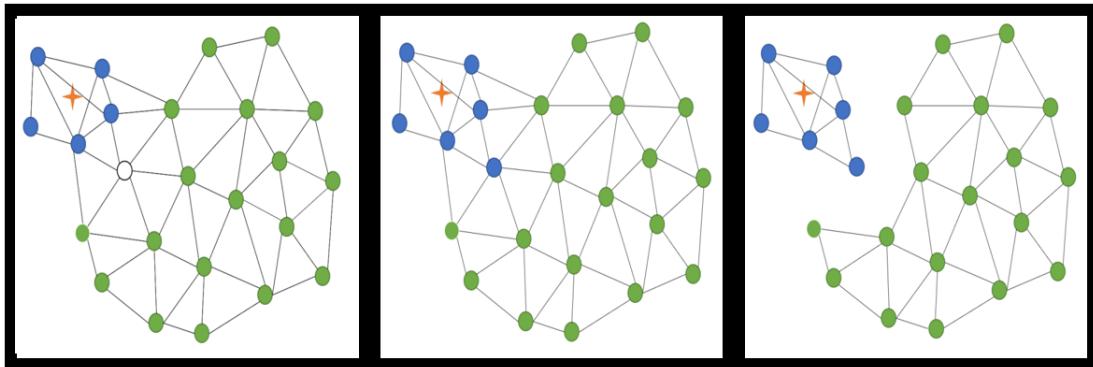


Figure 1. Sub-swarm generation.

4. Victim Detection

One of the most investigated approaches in robotics applied to the search and rescue missions is non-convex environment navigation including collision avoidance, as was shown in the previous section. Although navigation is really important in search and rescue operations, detection of a victim is the main task in the mission. This is the reason some techniques for the improvement of victims' detection using a mobile robot network have been addressed. Which are formation control and distributed estimation consensus, with the aim of improving the measurements reliability of victims' detection coming from the sensors, based on assigning a better sub-swarm location around the victim and classifying between victims and false positive carrying out a sensor network consensus, respectively. It is worth mentioning that each robot has a sensor capable of sensing if somewhere there is a possible victim, giving place to a sensor network system. In this way, it is not important for the purposes of this project to specify the type of sensor used in the model.

4.1. Formation Control

Taking into account, that sub-swarms were generated in the navigation process due to some robots from the robot network detect a possible victim through their sensors and decide to leave the main swarm to create a new sub-swarm. Consequently, an algorithm capable of relocating all the robots in the sub-swarm around to the possible victim location is carried out in order to improve measurements from the sensor network on the detection of a victim. Giving to the sensor network the ability to increase their probability of well-classifying and avoid that a far robot introduces noise in the consensus. The algorithm has the capability to bring robots closer to the victims using the rendezvous algorithm and then take agents to the perimeter of a circle making certain modifications to the rendezvous consensus algorithm exposed in [38]. In which the Laplacian matrix and the state vector are used, giving rise to the consensus equation that models the dynamics of the system, expressed in (5), and explained with more detail in Appendix A. Where the state vector contains the dynamical information of the agents.

$$\dot{x} = -Lx \quad (5)$$

According to [38] the consensus equation shown above will have a space of agreement $A = \{X \in R^n \mid x_i = x_j \forall i, j\}$ in which, when $\dot{x}_i = 0$ there will be a consensus. On the other hand, in the time domain, we have $x(t) = e^{-Lt}x_0$ in which, when the limit when t tends to infinity is evaluated and performing an expansion in Taylor Series is possible to note that except for the first term all the others are zero. The first term turns out to be $x(t) = e^{-\lambda_1 t}((u_1)^T x_0)u_1$ letting that the first eigenvalue is zero because the whole exponential part tends to 1 which causes the following expression $((u_1)^T x_0)u_1$. Extending it to

all the agents in the graph, we have that the consensus of a non-directed graph is the average of the initial conditions of the system, this means that the consensus can be written as follows

$$\text{consensus} = \sum \frac{x_{0i}}{n}. \quad (6)$$

Taking into account that this rendezvous consensus allows all agents to reach the same position in the space, the next step is to guarantee the connection between the agents while the signal control is modified to reach the goal position for each agent based on the approach shown in [11] which in order to solve the problem of maintenance communication, artificial potential functions was performed used to guarantee connectivity and keeping a distance among agents with the aim to avoid collisions. The artificial potential functions are of the form,

$$\psi_{ij} = \frac{1}{\rho_2^2 - ||x_{ij}||^2} - \frac{1}{||x_{ij}||^2 - \rho_1^2}, \quad (7)$$

where ψ_{ij} is the artificial potential function between the agents i and j , ρ_2 corresponds to the radius of connectivity of the agents and x_{ij} corresponds to the distance between the agents i and j . From these functions, the dynamics of the agents is imposed by $\dot{x}_i = -\sum_{j \in N_i^\sigma} \nabla_{x_i} \psi_{ij}$, note that this function tends to infinite when the distance between the agents tends to ρ_2 . In fact, the more the distance between agents increases, the more the intensity of the attraction among the robots increments to avoid communication breaks.

The artificial potential functions approach demonstrates connectivity maintenance by evaluating the total energy of the graph connections given by

$$\psi = \frac{1}{2} \sum_{i=1}^n \psi_i = \frac{1}{2} \sum_{i=1}^n \sum_{j \in N_i^\sigma} \psi_{ij}, \quad (8)$$

where ψ is the total energy of all the links and ψ_i corresponds to the energy of all the links concerning the agent i . From this total energy, it is ascertained whether it is growing, maintaining or reducing with time from its derivative with respect to time according to

$$\dot{\psi} = \frac{1}{2} \sum_{i=1}^n \sum_{j \in N_i^\sigma} \dot{x}_{ij}^T \nabla_{x_i} \psi_{ij}, \quad (9)$$

demonstrating that the total energy of the links tends to decrease over time and since the energy of the links grows due to the loss of connectivity, it is also demonstrated that the links are maintained. As expected the cooperative behavior of agents in a sub swarm converges to the desired location and then they spread out around the target in a circle formation as shown as follow in Figure 2.

Consequently, as can be seen in Figure 11 agents that have been chosen from the main swarm to generate a sub-swarm do not have an order in space and some of them can be far from the possible victim. The fact that some robots could be far from the victim can cause misdetection due to a possible error in measurements from the sensors produced by noise. As a result of possible errors coming from long distances between robots and the victim, a formation control is proposed to resolve this inconvenient by performing a position consensus in the sub-swarm, setting the possible victim as a target point. Once consensus has been reached, avoiding collisions among robots through a repulsion effect between each robot to the others, the next step is to perform a consensus in a circle shape around the victim, allowing in this way that each robot has approximately the same probability to well classifying. Taking into account that each robot from the sub-swarm has his own sensor to determine if there is or not detecting a victim, the next step is concerning to a distributed consensus in the sensor network that will be shown in the next subsection.

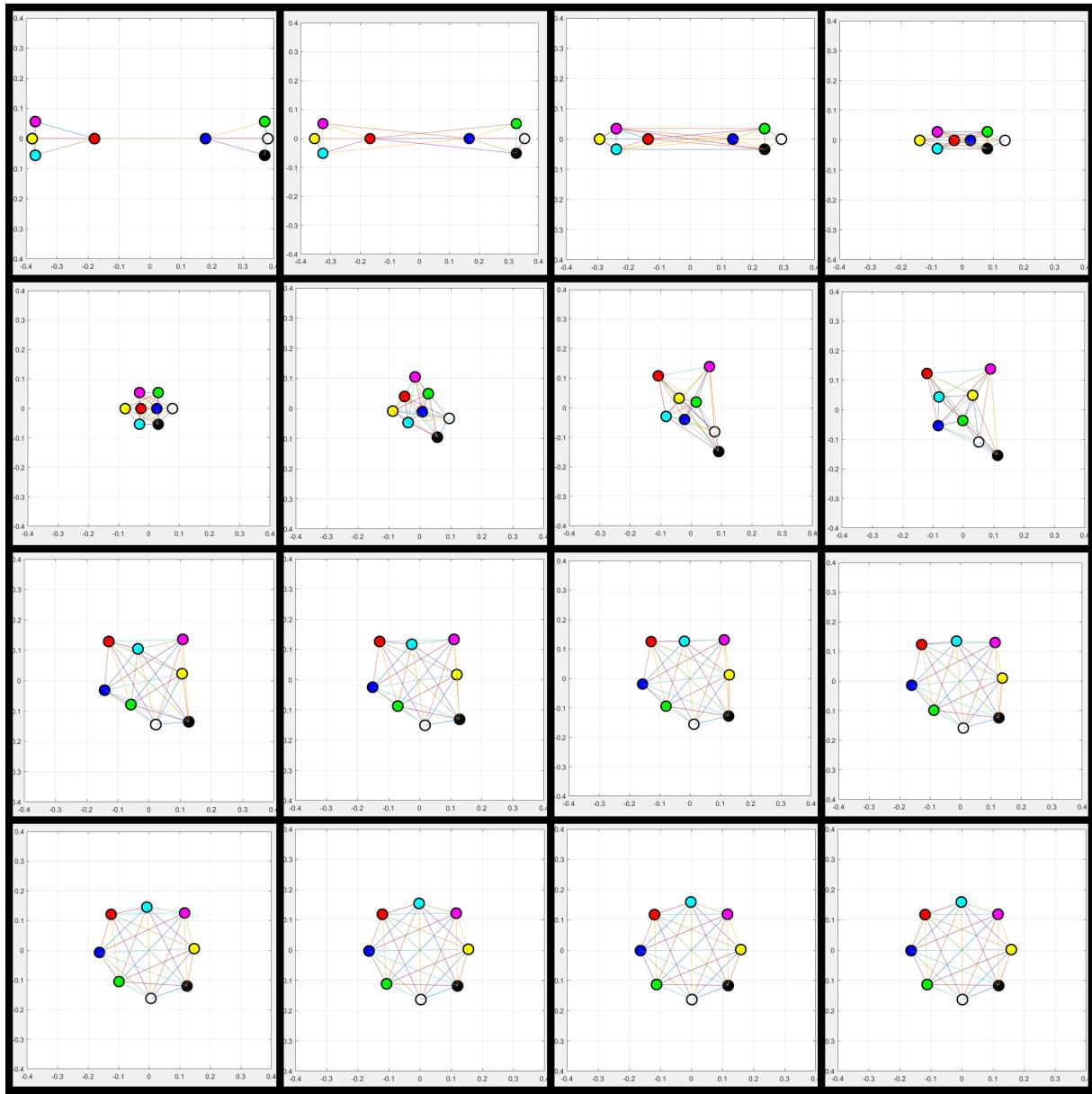


Figure 2. Consensus around a circle.

4.2. Distributed Estimation Consensus

Sensors are one of the most relevant parts for successful a victim detection process, due to that they are in charge of capturing all information from the environment. It is relevant to note that real sensors also capture noise in the measurements, which can come from electromagnetic noise, poor conditioning or even external factors depending on the type of sensors. This is the reason that in the present work, sensors are modeled using distributed linear least square with the presence of additive zero-mean Gaussian noise as shown in Figure 3 the sensing of a Gaussian signal without noise and in the presence of noise, respectively, based on the work presented in [38]. The underlying model involves the estimation of a linear function of a variable $\beta \in \mathbb{R}^q$ which is additively affected by noise ε_i in each of the n sensors on the network, how was previously mentioned,

$$\sigma_i = H_i \beta + \varepsilon_i, \quad (10)$$

where $\sigma_i, \varepsilon_i \in \mathbb{R}^{p_i \times 1}$ and $H_i \in \mathbb{R}^{p_i \times q}$ in which each value of the vector σ_i is a measurement channel of *i*th sensor, on the other hand, H is assumed to be of rank q which assures that the measurements are not entirely redundant.

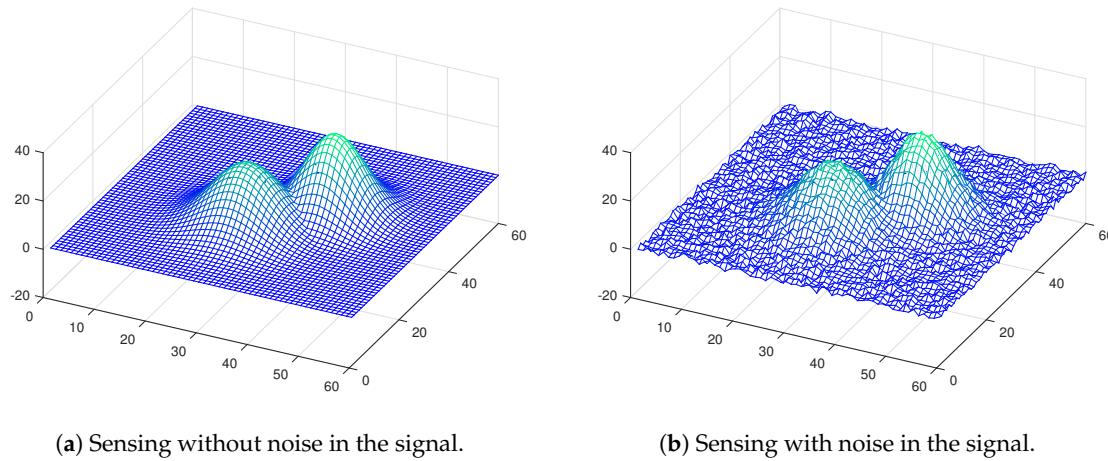


Figure 3. Additive Gaussian noise applied to a measurements in sensors.

Taking into account that the aim of the Least Squares is to minimize the error, in the centralized case, we have by isolating it $\varepsilon = \sigma - H\beta$ and then applying the square error $\varepsilon^2 = \varepsilon\varepsilon' = (\sigma - H\beta)'(\sigma - H\beta)$ the result of which is $\varepsilon'\varepsilon = \sigma'\sigma - 2\sigma'H\beta + \beta'H'H\beta$ which is a function $f(\beta)$ that depends only on β . In order to find the local or global minimums it is necessary to find the error gradient and the Hessian matrix, which in this case are $\nabla f(\beta) = \frac{df}{d\beta} = -2H'\sigma + 2H'H\beta$, by isolating β it is found a minimum when,

$$\hat{\beta} = (H'H)^{-1}H'\sigma, \quad (11)$$

additionally to verify that is a minimum function, it must satisfy $\frac{d^2f}{d\beta^2} = 2H'H > 0$ taking as u a non-null vector by multiplying $u'H'Hu > 0 \rightarrow (Hu)'(Hu) \rightarrow \|Hu\|^2 > 0$. It is possible to conclude that due to the Hessian matrix it is always positive because the function is convex, and the critical point is not just a minimum is the optimum.

Coming back to the distributed sensor network, it can be written as $\hat{\beta} = (\sum_{i=1}^n H_i'H_i)^{-1}(\sum_{i=1}^n H_i'\sigma_i)$ by solving the follow distributed optimization function $\min_{s.t. \beta \in \mathbb{R}^q} \sum_{i=1}^n f_i(\beta)$, as was shown previously each $f_i : \mathbb{R}^q \rightarrow \mathbb{R}$ are convex function as a result the target function is given by the average of the gradient functions of each node as $f^* = \frac{1}{n} \sum_{j=1}^n (f_j)$, taking this into account, it can be evidenced that Equation (6) has the same form of the following

$$\hat{\beta} = \frac{1}{n} \sum_{j=1}^n \sigma_i, \quad (12)$$

concluding in this way that a consensus among the sensors can be achieved if some condition are satisfied, considering the iteration of a sensor as $\hat{\beta}_i(k+1) = \hat{\beta}_i(k) + \Delta \sum_{j \in N_i} w_{ij}(\hat{\beta}_j(k) - \hat{\beta}_i(k))$ in which $\hat{\beta}_i(k)$ illustrate the estimation of sensor *i* of variable β at time *k*,

$$\lim_{k \rightarrow \infty} \hat{\beta}_i(k) = \left(\frac{1}{n} \sum_{i=1}^n \sigma_i \right) \mathbf{1}, \quad (13)$$

if and only if the sensor network is connected and $\rho(L_w(G)) < \frac{2}{\Delta}$ where $\rho(L_w(G))$ correspond to the maximum eigenvalue of the graph in absolute values, then the agents on the network will converge to the average of its initial conditions as shown below in Figure 4.

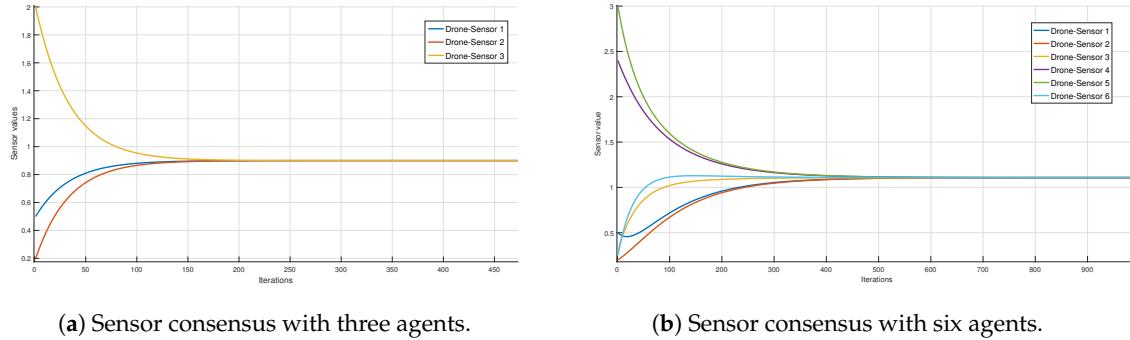


Figure 4. Sensor network converging due to the consensus applied.

5. Simulation Results

In order to perform experiments for swarm navigation, sub-swarm generation, and victim detection, the multi-agent model previously explained was implemented using Matlab and VRep. The Matlab implementation was performed with the aim to evaluate the proposed mathematical model in different scenarios and behavioral cases as previously named. VRep was used to perform simulations of the proposed swarm behavior model over 3D environments with a virtual model of real robots as Drones. The principal objective of the VRep simulations is to validate the proposed system using a more realistic environment as shown in Figure 5.



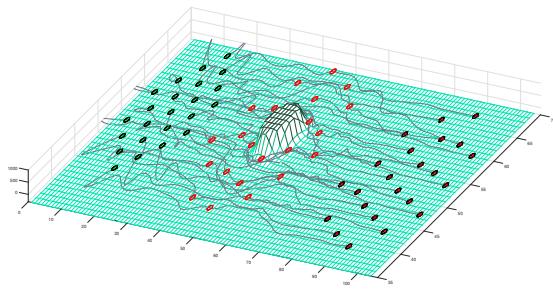
Figure 5. Simulation in a realistic environment and robot platforms using V-Rep.

Six different types of experiments were developed to verify the proposed model in several situations of navigation, sub-swarm generation, and victim detection. Every experiment is shown in both simulation environments as named previously (Matlab and VRep). Both types of simulations were performed using 23 agents or Drones respectively. The trace left by the agent is shown as a solid line behind every agent or

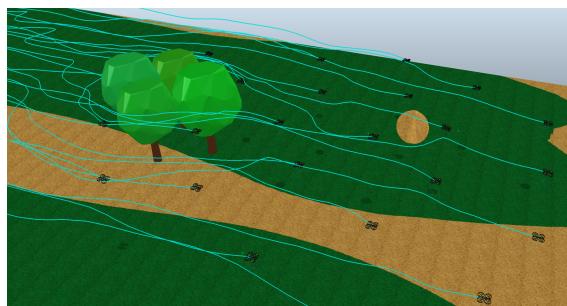
Drone according to the case. The idea of those lines is to show the path followed by every swarm's robot and depict the collective behavior generated by the swarm computational model. In Matlab simulations, the green circles represent the initial locations of the swarm's agents, the red filled circle is the current position of the agent, and the red empty circles are a sample location in the agent path through the simulation time. Next, every experiment is explained in detail and the results are analyzed.

5.1. Obstacle Avoidance

The first version of this experiment is performed using 23 agents and two cases of a single obstacle avoidance are performed. The first case is a small obstacle as shown in Figure 6. This part of the experiment depicts how the swarm goes around the obstacle in order to avoid it. In VRep simulation, the small obstacle is represented by a bunch of trees. This simulation shows how the obstacle is avoided by the swarm's drones flying around the trees. This is possible because the obstacle is relatively small and the agents are able to tolerate the obstacle between the attraction forces. The second version uses a big obstacle as depicted in Figure 7. In the second case, the agents avoid the obstacle by taking a side path. The big obstacle is represented by a building in VRep-simulation, where the swarm of drones navigates describing a side path to the building. This avoiding obstacle strategy occurs because there is not enough space between the attraction forces to allow broad obstacles to stay in between the agents.

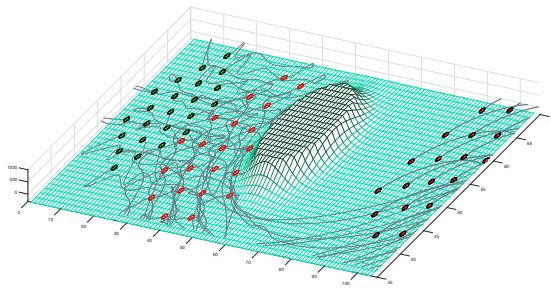


(a) Small-obstacle avoidance in Matlab.

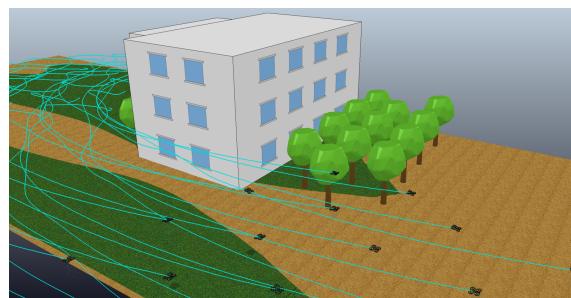


(b) Small-obstacle avoidance in a virtual environment.

Figure 6. Simulation of robots avoiding small obstacle.



(a) Large-obstacle avoidance in Matlab.



(b) Large-obstacle avoidance in a virtual environment.

Figure 7. Simulation of robots avoiding large obstacle.

5.2. Multiple Obstacles

This part of the experiment depicts how the swarm goes around several obstacles in order to avoid them. This is possible because the obstacles are relatively small and the agents are able to tolerate obstacles between the attraction forces, as depicted by Figure 8. This case uses nine obstacles distributed throughout the area between the start point and the goal point. For the virtual environment, obstacles are represented by nine trees distributed along the search area. It forces the swarm to navigate through obstacles while avoiding them and moving towards the goal point. Figure 8 shows the agents navigating and exploring the zone. At the same time, the path described by every agent is drawn and it depicts the explored area. VRep Video simulation of the multiple obstacle avoidance in Ref. [41].

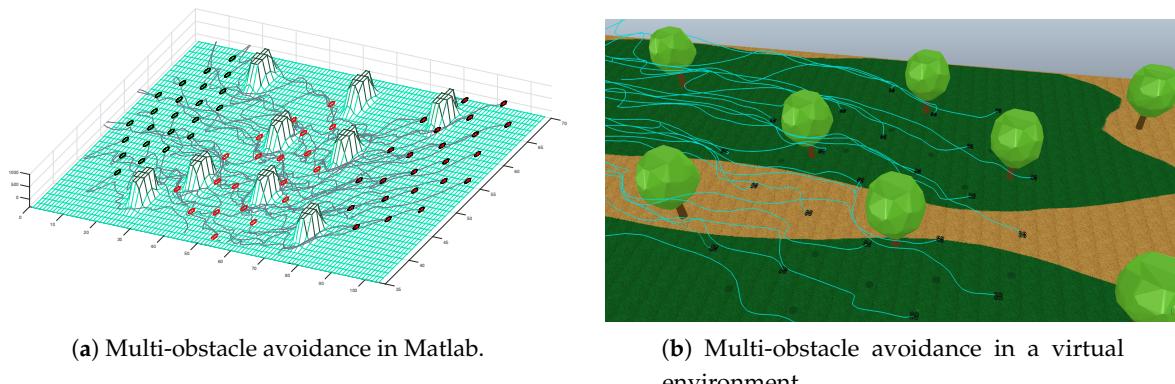


Figure 8. Simulation of robots avoiding multi-obstacle scenario.

5.3. Victim Localization

This test uses a flat terrain to show how the swarm localizes victims. This process is accomplished with the use of the sub-swarm generation process. Once an agent has localized a victim, it stops near the potential victim. At the same time, the swarm stops, because of the attraction force between the agents. At that moment the process called “Sub-Swarm Generation” detaches the agents that found victims. The detached agents create a new swarm surrounding the victims. The original swarm restarts its movement towards the target point and leaves behind the swarm agents that are in charge of the victims. Figure 9 shows both simulation styles, where the agents stop near the victims and surround them, while the swarm leaves them behind.

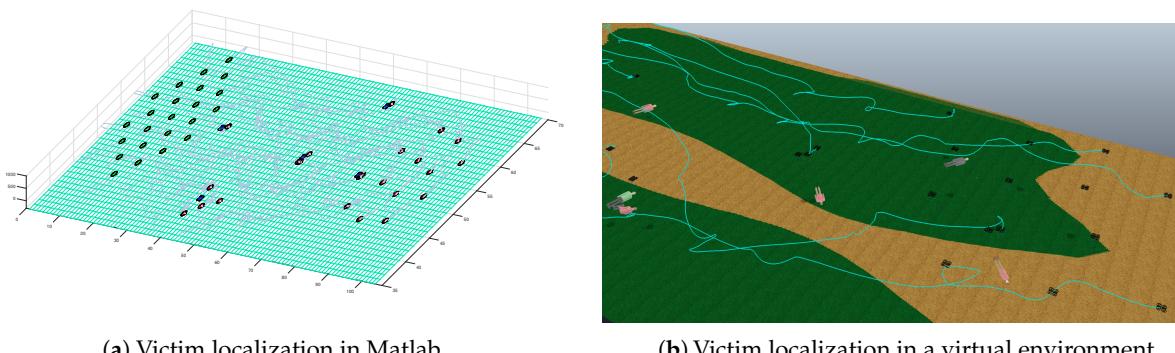


Figure 9. Victim localization in convex space.

5.4. Navigation and Victim Localization

This is a complete case where the swarm navigates through the area full of obstacles and some places with potential victims. The experiment uses 23 agents and 6 victims distributed in 5 victim places, given there are 2 victims in the same place as shown by Figure 10 this case depict how the swarm covered the area navigating through obstacles and localizing potential victims at the same time. The simulation shows how the proposed model generates sub-swarms with more drones in places where there are more victims or where the probability of finding victims is greater. This case is represented in the virtual environment simulation, where there is a place with two victims and the model assigns a sub-swarm with 6 drones. This victims place has more assigned drones than the other places. VRep Video simulation of Navigation and Victim Localization in Ref. [42].

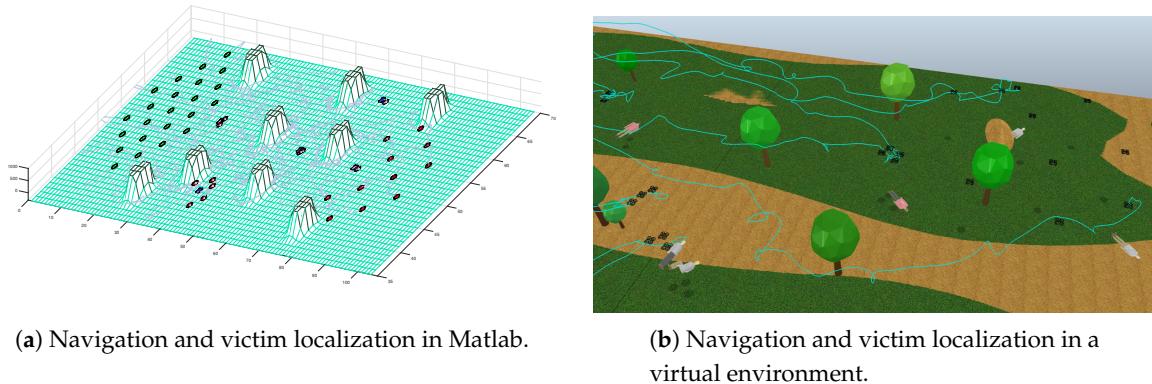


Figure 10. Navigation and victims localization in environment with presence of obstacles.

5.5. Sub-Swarm Formation for Distributed Estimation Consensus

This experiment shows two cases of possible victims with different detection probability. The first case presents two victims in an open field with a high probability to be detected by the drone sensors as shown in Figure 11. Given the victim detection is easy for this case, the swarm is made up of 6 agents located in the vicinity of the victims. After sub-swarm creation, the consensus and control formation algorithm is performed as shown in Figure 11 as explained previously in this paper the consensus looks for to approach every drone to the victim, reducing the uncertainty of sensing factor over the possible victim. Additionally, the formation control redistributes the drones in a circle around the victim with the aim to provide better assistance to the victims and balance the sensor measurements. The second case presents a victim with the body partially covered and a sub-swarm of three agents. The number of agents for this case is lower than the first case given the difficulty of the victim detection. Once the victim is detected, the sub-swarm executes the processes of consensus and control formation as shown by Figure 12.

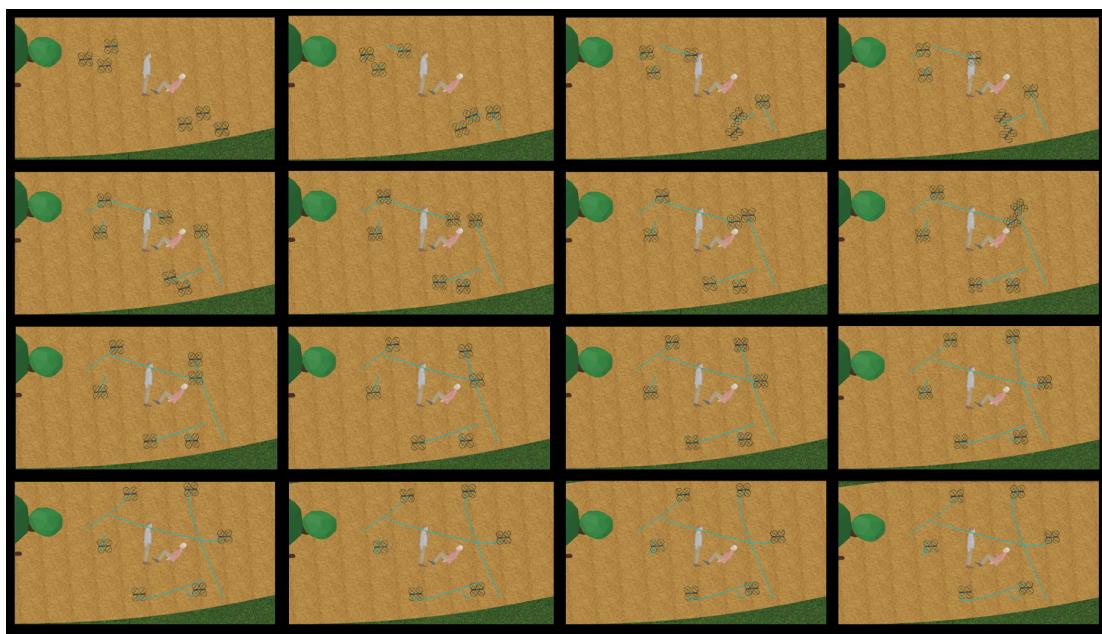


Figure 11. Formation control applied to 6 agents sub-swarm generated by localization of a victim.

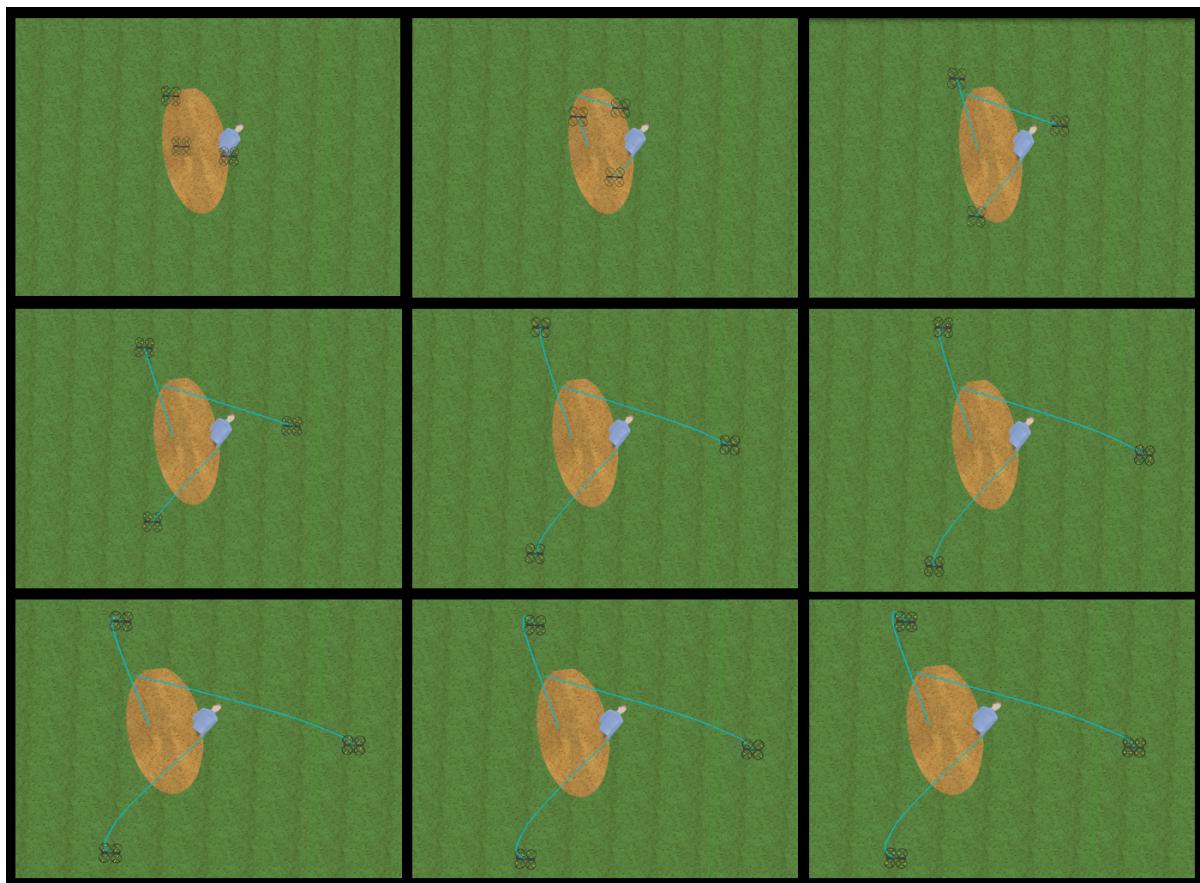
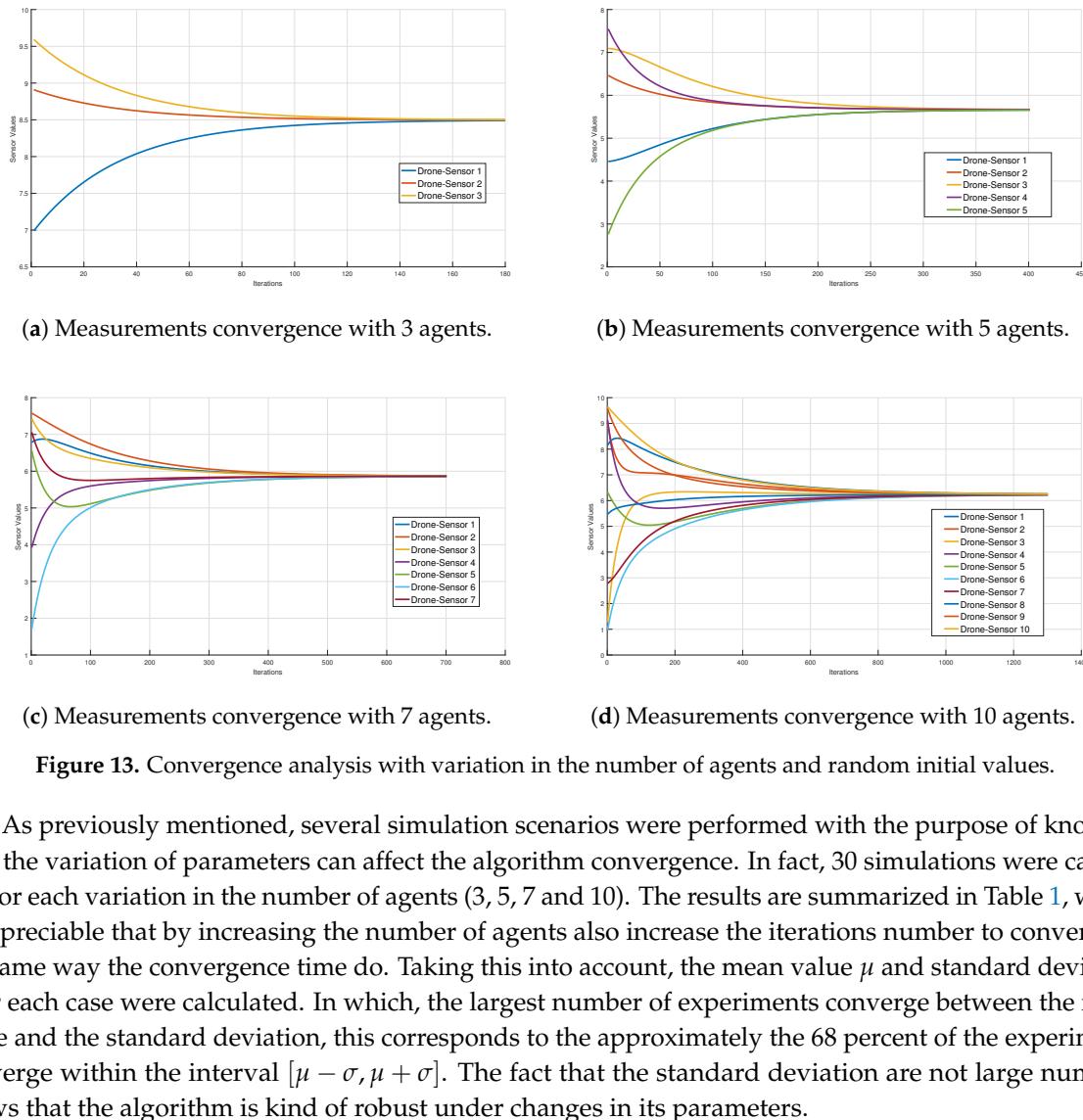


Figure 12. Formation control applied to the 3 agents sub-swarm generated by localization of a victim.

5.6. Convergence Analysis of the Distributed Estimation Consensus

In order to perform a convergence analysis, it is worth mentioning that the estimation algorithm was carried out in a computer with the following specs: 7-th generation core-i7 microprocessor, 32 GB installed memory (RAM) and NVIDIA GEFORCE 940MX. Several simulations were performed with the aim to know the algorithm performance under different parameter values. The principal parameter taken into account is the number of agents able to take a measurement, which in this case vary from 3 to 10. Additionally, in the interest to prove the robustness of the estimation algorithm, the measurement values are settled in a random way. In this way, the convergence of the algorithm depending on the number of agents is shown in Figure 13. Where it is appreciable how the convergence is affected by the number of agents. The greater number of agents giving measurements the more time it takes the algorithm to converge.



As previously mentioned, several simulation scenarios were performed with the purpose of knowing how the variation of parameters can affect the algorithm convergence. In fact, 30 simulations were carried out for each variation in the number of agents (3, 5, 7 and 10). The results are summarized in Table 1, where is appreciable that by increasing the number of agents also increase the iterations number to converge in the same way the convergence time do. Taking this into account, the mean value μ and standard deviation σ for each case were calculated. In which, the largest number of experiments converge between the mean value and the standard deviation, this corresponds to the approximately the 68 percent of the experiments converge within the interval $[\mu - \sigma, \mu + \sigma]$. The fact that the standard deviation are not large numbers shows that the algorithm is kind of robust under changes in its parameters.

Table 1. Convergence data depending on the number of agents.

Agents Number	Average Convergence Iteration	Standard Deviation	Average Convergence Time
3	176.6	8.3	90.6 (mS)
5	404.9	22.4	103.9 (mS)
7	602.6	39.4	144.5 (mS)
10	1162.4	102.3	226.6 (mS)

5.7. Effective Coverage Area

An important parameter that should be taken into account is the effective coverage area. This parameter is extremely important due to developed tasks that are related to the search and localization of human beings in a disaster zone. The idea of swarm drone exploration is supported by the aim to inspect every small space into the disaster zone at least once by at least one drone. The most insignificant failure in the exploration can be translated into the loss of human life, so it is preferable that the same area within the disaster zone be inspected by more than one drone. Generating in this way more robustness to the task of search and localization, this search and rescue approach gives great importance to the use of a drone swarm, which seeks to increase the probability of finding victims. The effective coverage area is defined as a corridor through which the swarm travels. This corridor is as long as the area to explore and the width depends on the number of drones since the width of the area depends on the number of drones that are located across the width of the swarm. The width of the coverage area is directly related to factors such as the number of drones, comfortable distance between drones (d) previously defined in (3), and the radius of sensors coverage (r) used to perform the inspection and search tasks. In order to analyze the exploration degree of the covered zone, several experiments were carried out in which the parameter r was varied in relation to d as shown in Table 2.

The experiment aim is the percentage calculation of the unexplored and explored area by at least one, two, three, or more drones. Figure 14 shows the simulation of the experiment and the representation of different degrees of exploration for the areas covered by the drone in its trajectory towards the endpoint.

The Figure 15 shows different experimental cases where the covered area by the swarm is depicted by several gray color levels through the path to the endpoint. Figure 15 shows how explored area changes according to the relationship between r and d . In Figure 15 from (a) to (d) is showed the covered area using a swarm of 23 agents. The covered area increases according to r value approaches to d . Figure 15e,f shows how the explored area is affected by the swarm agent number. The covered area percentage in relation to the values of r and d is depicted by Table 2, which describes seven experiments where the rate between r and d is studied from $r = d/15$ to $r = d$. From Table 2 it is concluded that explored area increase significantly through the increase of the r factor. The equilibrium point is reached when $r = d/4$ because is in this point, where 100% and 80% of the entire area is explored by at least two drones.

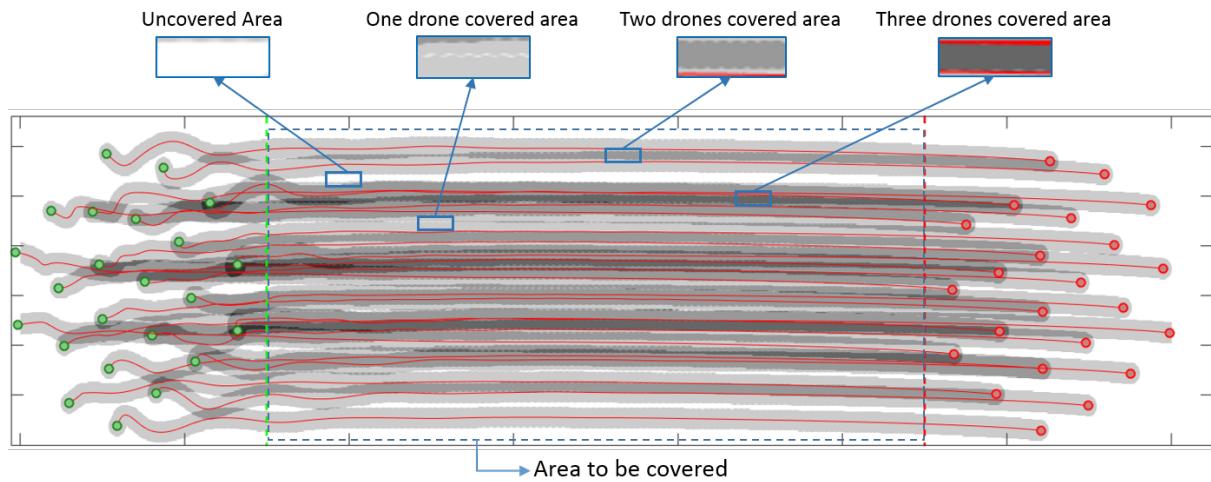


Figure 14. Formation control applied to the 3 agents sub-swarm generated by localization of a victim.

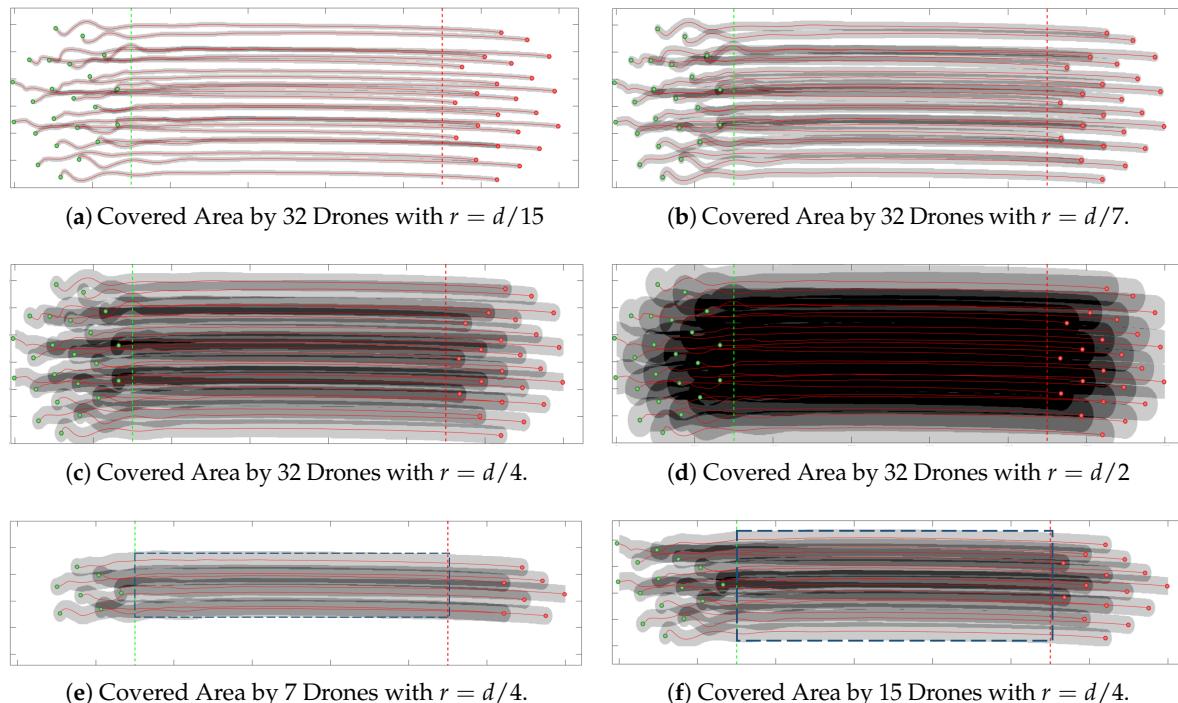


Figure 15. The covered area by a different number of drones with several relation values between r and d .

Table 2. Percentage of the covered area according to different rate values between r and d .

r	Uncovered	Covered	+2 Drones	+3 Drones	+4 Drones	+5 Drones
d/15	43.10%	56.89%	9.55%	0.0014%	0.0%	0.0%
d/8	13.51%	86.48%	42.01%	4.35%	0.0069%	0.0%
d/5	1.73%	98.26%	73.28%	24.57%	3.06%	0.0%
d/4	0.0%	100%	87.77%	54.55%	20.53%	0.20%
d/3	0.0%	100%	94.34%	80.14%	52.72%	14.56%
d/2	0.0%	100%	98.17%	94.04%	85.42%	71.26%
d	0.0%	100%	100%	100%	97.57%	96.11%

Table 3 shows how the areas explored by more than one drone increases as the number of drones increases. This result is quite logical, taking into account that increasing the number of drones also increases the number of rows in the swarm. In this case, more drones will follow roads similar to the path taken by the drones of the first rows and they will explore similar disaster zones. The relation between r and d was selected as $r = d/4$ with the aim to ensure the covered area close is to 100% and focuses the results on the percentage of areas explored by more than one drone.

Table 3. Percentage of the covered area according to different swarm size and $r = d/4$.

Drone Number	Uncovered	Covered	+2 Drones	+3 Drones	+4 Drones	+5 Drones
7	0.0%	100%	93.14%	16.63%	0.01%	0.0%
12	1.02%	98.87%	83.88%	29.8%	5.03%	0.0%
17	0.0%	100%	89.10%	48.48%	11.87%	0.0%
23	0.0%	100%	87.77%	54.55%	20.53%	0.20%

6. Conclusions and Future Work

An algorithm capable of performing some of the main tasks in search and rescue missions using aerial robotics has been proposed, based on swarm concepts. The algorithm uses the attraction and repulsion forces approach in order to keep the swarm as compact as possible or to be attracted by victims and at the same time avoid collisions or keep a minimum distance between agents, respectively. The algorithm uses the graph theory as the main tool to model and work with the robots' swarm. On this approach, the nodes represent the robots or sensor estimations and links are the distance between them or the weighted distance assigned to their location respect to the victim. Once that behaviors of navigation are achieved in the swarm, the next step is to create the policy that allows the swarm to generate a sub-swarm and disconnect the agents in the sub-swarm from the main one. As explained before, the generation of sub-swarm allows this algorithm to perform two tasks at the same time, which are the classification and assistance to a victim with the sub-swarm and continue the navigation seeking for more victims using the rest of the swarm.

As expected, the navigation process in a non-convex environment using artificial potential functions was successful as well as the implementation of sub-swarm generation in order to classify victims. The use of the k-nearest approach to a weighted graph has a good performance in the generation of sub-swarms close to a target location which could be a possible victim. The sub-swarm allows the system to perform a consensus algorithm at the same time that the other agents from the main swarm can still be navigating looking for more victims. This consensus in order to get closer to the victim and then through formation control surround it, improving in this way the probability of well victim detection. Once the robots get better location around the victim, they accomplish an estimator distributed consensus modeling the sensor by least square approach allowing sub-swarm to evaluate if they are sensing a victim or not.

Several experiments were carried out by using a robot simulator. This system allows using virtual models of real robots and the creation of environments with obstacles and people as victims whit-in. The simulation exposes how the algorithm generates a swarm behavior by allocating each robot behavior. The robots' swarm were capable of navigating without colliding with obstacles in the environment or with themselves and generates sub-swarms when they find a victim. Additionally, when a sub-swarm found a victim is showed how the formation control allows the robots to locate around the victims which is crucial in order to estimate if there is or not a victim in the place.

As future work, we intend to give more attention to the time-varying dynamical graphs in order to model extra behaviors concerning to the sub-swarm generation and the inclusion of heterogeneous agents. On the other hand, it is important to develop an algorithm that allows the system to get information from the detection of a victim, for instance, use of visual information to determine if the robot is in front of a

victim. Taking this into account, the next step is to apply this information coming from the sensors in each robot close to the victim in the distributed estimation consensus algorithm exposed in this work. Finally, create an algorithm in charge of the global victim localization that allows the system to inform rescuers the exact localization of a victim.

Author Contributions: J.M.C. leads the research project. G.A.C. and J.M.C. developed the technical/theoretical approaches by equal parts.

Funding: This work has been funded by “Treceava Convocatoria Interna de Proyectos de Investigacion FODEIN 2019 #1936006” at Universidad Santo Tomas Colombia, entitled “Generación de algoritmos de navegación en enjambres de robots mediante el uso de aprendizaje por refuerzo para el desarrollo de tareas de búsqueda y rescate de víctimas humanas en zonas de desastres”. and Department of Computer Science and Engineering of Bethune-Cookman University.

Acknowledgments: The authors express their thanks to all who participated in this research for their cooperation, especially thanks to Jose Leon for his help at the beginning of this research. The authors would like to give great thank to the hard work by the peer reviewers and editor.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Basic Concepts of Graph Theory

The graph theory is a tool that allows working in distributed systems. This means it can help to model the multi-agent system dynamics, the reason is important to clarify some concepts used through this work. First of all, let us explain what a graph $G = (V, E)$ could be, It is a representation of agents that is composed of nodes and links, which represent the agents $V = \{v_1, v_2 \dots v_i\}$ and the connection or capability of communication among agents $E = \{v_1v_1, v_1v_2 \dots v_1v_j, v_2v_j \dots v_iv_j\}$ respectively. In the same way, it is worth to explain concepts as Laplacian of a graph, adjacency and degree matrix. The Laplacian is composed by the operation between the adjacency matrix and the degree matrix how is shown in Equation (A1)

$$L(G) = D(G) - A(G), \quad (\text{A1})$$

where the adjacency matrix, as its name indicates describes which nodes have a way of connection. for this case bidirectional, in the case of an undirected graph the way to fill this matrix is shown in Equation (A2) where V_iV_j Represents a link among nodes V_i and V_j respectively. On the other hand, the degree matrix expresses in its main diagonal the number of connections that each node has, how is shown in Equation (A3).

$$[A(G)]_{ij} = \begin{cases} 1 & \text{if } V_iV_j \in E \\ 0 & \text{otherwise} \end{cases} \quad (\text{A2})$$

$$D(G) = \begin{bmatrix} d_{in}(1) & & \\ & \ddots & \\ & & d_{in}k \end{bmatrix} \quad (\text{A3})$$

Finally another way to represent the laplacian matrix of a directed weighted graph $L_w(G) = d_{lk}(G)W(G)(d_{lk}(G))'$ is using the concept of incidence matrix defined as Equation (A4) an the weighted matrix $W(G) = \text{diag}([w_1, \dots, w_n]')$ where $w_i > 0$ is the weight on the i th edge from the graph, indexed consistently with the column ordering in the corresponding incidence matrix.

$$d_{lk}(G) := \begin{cases} +1 & \text{if } k \in_i^+ \\ -1 & \text{if } k \in_i^- \\ 0 & \text{otherwise} \end{cases} \quad (\text{A4})$$

References

1. Casper, J.; Murphy, R.R. Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **2003**, *33*, 367–385. [[CrossRef](#)] [[PubMed](#)]
2. Ventura, R.; Lima, P.U. Search and rescue robots: The civil protection teams of the future. In Proceedings of the 2012 Third International Conference on Emerging Security Technologies, Lisbon, Portugal, 5–7 September 2012; pp. 12–19.
3. Latscha, S.; Kofron, M.; Stroffolino, A.; Davis, L.; Merritt, G.; Piccoli, M.; Yim, M. Design of a Hybrid Exploration Robot for Air and Land Deployment (HERALD) for urban search and rescue applications. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), Chicago, IL, USA, 14–18 September 2014; pp. 1868–1873.
4. Seljanko, F. Low-cost electronic equipment architecture proposal for urban search and rescue robot. In Proceedings of the 2013 IEEE International Conference on Mechatronics and Automation (ICMA), Takamatsu, Japan, 4–7 August 2013; pp. 1245–1250.
5. Doroodgar, B.; Ficocelli, M.; Mobedi, B.; Nejat, G. The search for survivors: Cooperative human-robot interaction in search and rescue environments using semi-autonomous robots. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA), Anchorage, AK, USA, 3–8 May 2010; pp. 2858–2863.
6. Pfotzer, L.; Ruehl, S.; Heppner, G.; Rönnau, A.; Dillmann, R. KAIRO 3: A modular reconfigurable robot for search and rescue field missions. In Proceedings of the 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO), Bali, Indonesia, 5–10 December 2014; pp. 205–210.
7. Tan, Y.; Zheng, Z.Y. Research advance in swarm robotics. *Def. Technol.* **2013**, *9*, 18–39. [[CrossRef](#)]
8. Husni, N.L.; Handayani, A.S.; Nurmaini, S.; Yani, I. Cooperative searching strategy for swarm robot. In Proceedings of the 2017 International Conference on Electrical Engineering and Computer Science (ICECOS), Palembang, Indonesia, 22–23 August 2017; pp. 92–97.
9. Yanguas-Rojas, D.; Cardona, G.A.; Ramirez-Rugeles, J.; Mojica-Nava, E. Victims search, identification, and evacuation with heterogeneous robot networks for search and rescue. In Proceedings of the 2017 IEEE 3rd Colombian Conference on Automatic Control (CCAC), Cartagena, Colombia, 18–20 October 2017; pp. 1–6.
10. Cardona, G.A.; Yanguas-Rojas, D.; Arevalo-Castiblanco, M.F.; Mojica-Nava, E. Ant-Based Multi-Robot Exploration in Non-Convex Space without Global-Connectivity Constraints. In Proceedings of the 2019 European Control Conference (ECC), Naples, Italy, 25–28 June 2019.
11. Zavlanos, M.M.; Egerstedt, M.B.; Pappas, G.J. Graph-theoretic connectivity control of mobile robot networks. *Proc. IEEE* **2011**, *99*, 1525–1540. [[CrossRef](#)]
12. Torres-Torriti, M.; Arredondo, T.; Castillo-Pizarro, P. Survey and comparative study of free simulation software for mobile robots. *Robotica* **2016**, *34*, 791–822. [[CrossRef](#)]
13. Zhang, F.; Leitner, J.; Milford, M.; Corke, P. Sim-to-real transfer of visuo-motor policies for reaching in clutter: Domain randomization and adaptation with modular networks. *World* **2017**, *7*, 8.
14. León, J.; Cardona, G.A.; Botello, A.; Calderon, J.M. Robot swarms theory applicable to seek and rescue operation. In Proceedings of the International Conference on Intelligent Systems Design and Applications, Porto, Portugal, 16–18 December 2016; Springer: Cham, Switzerland, 2016; pp. 1061–1070.
15. León, J.; Cardona, G.A.; Jaimes, L.G.; Calderon, J.M.; Rodriguez, P.O. Rendezvous Consensus Algorithm Applied to the Location of Possible Victims in Disaster Zones. In Proceedings of the International Conference on Artificial Intelligence and Soft Computing, Zakopane, Poland, 3–7 June 2018; Springer: Cham, Switzerland, 2018; pp. 700–710.
16. Quesada, W.O.; Rodriguez, J.I.; Murillo, J.C.; Cardona, G.A.; Yanguas-Rojas, D.; Jaimes, L.G.; Calderon, J.M. Leader-Follower Formation for UAV Robot Swarm Based on Fuzzy Logic Theory. In Proceedings of the International Conference on Artificial Intelligence and Soft Computing, Zakopane, Poland, 3–7 June 2018; Springer: Cham, Switzerland, 2018; pp. 740–751.

17. de Sousa Paula, P.; de Castro, M.F.; Paillard GA, L.; Sarmento, W.W. A swarm solution for a cooperative and self-organized team of UAVs to search targets. In Proceedings of the 2016 8th Euro American Conference on Telematics and Information Systems (EATIS), Cartagena, Colombia, 28–29 April 2016; pp. 1–8.
18. Couceiro, M.S.; Rocha, R.P.; Ferreira, N.M. A novel multi-robot exploration approach based on particle swarm optimization algorithms. In Proceedings of the 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Kyoto, Japan, 1–5 November 2011; pp. 327–332.
19. Edlinger, R.; Zauner, M.; Rokitansky, W. RRTLAN-A real-time robot communication protocol stack with multi threading option. In Proceedings of the 2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Linkoping, Sweden, 21–26 October 2013; pp. 1–5.
20. Wiltsche, C.; Lygeros, J.; Ramponi, F.A. Synthesis of an asynchronous communication protocol for search and rescue robots. In Proceedings of the 2013 European Control Conference (ECC), Zurich, Switzerland, 17–19 July 2013; pp. 1256–1261.
21. Pan, Q.W.; Lowe, D. Search and rescue robot team RF communication via power cable transmission line—A proposal. In Proceedings of the International Symposium on Signals, Systems and Electronics, ISSSE’07, Montreal, QC, Canada, 30 July–2 August 2007; pp. 287–290.
22. Araujo, F.; Santos, J.; Rocha, R.P. Implementation of a routing protocol for Ad Hoc networks in search and rescue robotics. In Proceedings of the 2014 IFIP Wireless Days (WD), Rio de Janeiro, Brazil, 12–14 November 2014; pp. 1–7.
23. Nurellari, E.; McLernon, D.C.; Ghogho, M. Distributed two-step quantized fusion rules via consensus algorithm for distributed detection in wireless sensor networks. *IEEE Trans. Signal Inf. Process. Netw.* **2016**, *2*. [[CrossRef](#)]
24. Ghassemian, H. A review of remote sensing image fusion methods. *Inf. Fusion* **2016**, *32*, 75–89. [[CrossRef](#)]
25. Misra, S.; Vasilakos, A.V.; Obaidat, M.S.; Krishna, P.V.; Agarwal, H.; Saritha, V. A fault-tolerant routing protocol for dynamic autonomous unmanned vehicular networks. In Proceedings of the 2013 IEEE International Conference on Communications (ICC), Budapest, Hungary, 9–13 June 2013; pp. 3525–3529.
26. Chelbi, S.; Duvallet, C.; Abdouli, M.; Bouaziz, R. Event-driven wireless sensor networks based on consensus. In Proceedings of the 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), Agadir, Morocco, 29 November–2 December 2016; pp. 1–6.
27. Li, X.; Qiao, D.; Li, Y.; Dai, H. A novel through-wall respiration detection algorithm using uwb radar. In Proceedings of the 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Osaka, Japan, 3–7 July 2013; pp. 1013–1016.
28. Li, J.; Liu, L.; Zeng, Z.; Liu, F. Advanced signal processing for vital sign extraction with applications in UWB radar detection of trapped victims in complex environments. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 783–791.
29. Friedman, M.; Haddad, Y.; Blekhman, A. ACOUFIND: Acoustic ad-hoc network system for trapped person detection. In Proceedings of the 2015 IEEE International Conference on Microwaves, Communications, Antennas and Electronic Systems (COMCAS), Tel-Aviv, Israel, 2–4 November 2015; pp. 1–4.
30. Couceiro, M.S.; Portugal, D.; Rocha, R.P. A collective robotic architecture in search and rescue scenarios. In Proceedings of the 28th Annual ACM Symposium on Applied Computing, Coimbra, Portugal, 18–22 March 2013; pp. 64–69.
31. Alvissalim, M.S.; Zaman, B.; Hafizh, Z.A.; Ma’sum, M.A.; Jati, G.; Jatmiko, W.; Mursanto, P. Swarm quadrotor robots for telecommunication network coverage area expansion in disaster area. In Proceedings of the 2012 SICE Annual Conference (SICE), Akita, Japan, 20–23 August 2012; pp. 2256–2261.
32. Rocha, R.; Dias, J.; Carvalho, A. Cooperative multi-robot systems: A study of vision-based 3-d mapping using information theory. *Robot. Auton. Syst.* **2005**, *53*, 282–311. [[CrossRef](#)]
33. Rohmer, E.; Singh, S.P.; Freese, M. V-REP: A versatile and scalable robot simulation framework. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 3–8 November 2013; pp. 1321–1326.
34. Messina, E.R.; Jacoff, A.S. Measuring the performance of urban search and rescue robots. In Proceedings of the 2007 IEEE Conference on Technologies for Homeland Security, Woburn, MA, USA, 16–17 May 2007; pp. 28–33.

35. Chiou, A.; Wynn, C. Urban search and rescue robots in test arenas: Scaled modeling of disasters to test intelligent robot prototyping. In Proceedings of the UIC-ATC'09—Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing, Brisbane, Australia, 7–9 July 2009; pp. 200–205.
36. Saeedi, P.; Sorensen, S.A.; Hailes, S. Performance-aware exploration algorithm for search and rescue robots. In Proceedings of the 2009 IEEE International Workshop on Safety, Security & Rescue Robotics (SSRR), Denver, CO, USA, 3–6 November 2009; pp. 1–6.
37. Seeley, T.D.; Buhrman, S.C. Group decision making in swarms of honey bees. *Behav. Ecol. Sociobiol.* **1999**, *45*, 19–31. [[CrossRef](#)]
38. Mesbahi, M.; Egerstedt, M. *Graph Theoretic Methods in Multiagent Networks*; Princeton University Press: Princeton, NJ, USA, 2010.
39. Reynolds, C.W. Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH Comput. Gr.* **1987**, *21*, 25–34. [[CrossRef](#)]
40. Bullo, F.; Cortes, J.; Martinez, S. *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms*; Princeton University Press: Princeton, NJ, USA, 2009; Volume 27.
41. Calderon, J. Mobile Robotics & Intelligent Systems. Robot Swarm Navigation-Obstacles. 2018. Available online: <https://youtu.be/6B5TVmT8knI> (accessed On 21 March 2019).
42. Calderon, J. Mobile Robotics & Intelligent Systems. Robot Swarm Navigation. 2018. Available online: <https://youtu.be/F2tsg9jzIoY> (accessed on 21 March 2019).



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).