

# A Swarm Behaviour for Jellyfish Bloom Detection

Fidel Aznar, Mar Pujol, Ramón Rizo

---

## Abstract

In this paper we will deal with the issue of swarm behaviour for jellyfish detection using UAVs (Unmanned Aerial Vehicles). Swarm behaviour is inspired by the functioning of biological swarms. They are characterized by being fully distributed, scalable and fault-tolerant. Initially, we will study the behaviour of jellyfish and their impact and interaction with industry. Motivated by the need to improve current detection systems, we will propose a swarm behaviour, that will be formalized with a microscopic model. We will discuss both the convergence and the scalability of the model. Finally, a macroscopic model will be provided to predict the probability that an individual is placed in a position at a given moment.

---

## 1. Introduction

In this section, the importance of jellyfish and their economic impact in today's society will be introduced. The need for improvement in current jellyfish detection systems will be motivated. We will propose a very appropriate technique to carry out this task: swarm robotics. The advantages of swarming systems over conventional systems will be discussed. Furthermore, the importance of modelling and simulating when developing swarm behaviour systems will be shown.

Jellyfish are a little studied organism of marine ecosystems. Their populations are naturally highly variable; many species respond to favorable conditions by rapid population increases but these rapid population changes themselves make them difficult to monitor [2, 12]. Moreover sudden outbursts known as "blooms" followed by population crashes are observed in these organisms [26, 22, 30, 2, 5, 15].

Considering the potentially large impacts of jellyfish on marine ecosystems [27] and on mariculture operations, there are surprisingly few data on

jellyfish distribution and abundance. This lack of data is partly a consequence of problems with sampling and monitoring these delicate organisms [27]. A further consideration is that blooms may arise both through the processes of rapid population growth and aggregation.

More attention has been paid to jellyfish in recent years because of their interference in human enterprises, their ecological importance, and a qualitative perception that their population is increasing [3]. These perception are suggested in the literature, underlying that jellyfish blooms may be increasing in frequency and size [30, 31] and could be linked to a wide variety of factors including global warming, climate cycles, coastal pollution, over-fishing and even marine litter. However, records of jellyfish abundance and distribution are sparse and it has been difficult to confirm whether blooms are really increasing.

The interference in human activities could be confirmed by a large number of reports on problems in the European salmon farming industry related to jellyfish blooms. When blooms develop around aquaculture sites they can lead to a variety of problems. Jellyfish directly interfere with many human activities (reviewed by [30, 31]), specifically, through stings (beach closures, tourism impacts, injuries, deaths), clogging intakes (coastal power and desalination plants, mining and military operations, shipping, aquaria), interference with fishing (clogged and split nets, spoiled catch, stung fishers, damaged gear, capsized boats), aquaculture (fish deaths, pens fouled by polyps), and marine biological surveys (interference with trawls and acoustic surveys). They also have ecosystem impacts with indirect effects on fisheries resources that are difficult to quantify, such as their roles as predators of zooplankton, fish eggs and ichthyoplankton, as vectors for parasites, as food for fish, and as refugia and food for some species of juvenile fish (interactions reviewed by [28]).

The economic impacts of jellyfish blooms to marine finfish aquaculture were brought into public awareness by a massive bloom of *Pelagia noctiluca* (the mauve stinger) that killed more than 100,000 fish in Northern Ireland's only salmon farm in 2007. However, in addition to such exceptional high profile events, mortalities caused by jellyfish are potentially an on-going and recurrent problem for the aquaculture industry. Some jellyfish also benefit humans (reviewed in [30]), notably as food [16], and potentially for their use in drugs (e.g., [36, 25]). Unfortunately, such benefits may be outweighed by the direct and indirect negative impacts of jellyfish blooms.

Remote sensing has proved successful in providing early warning of blooms.

A range of projects funded by The Crown Estate [23] and the EU are also developing and parametrising physical/biological mathematical models to predict the location, abundance and transport of blooms of harmful phytoplankton such as *Karenia mikimotoi*. Current methods, that combine aerial surveys with satellite data showed promise but there are also specific limitations. Researchers were unable to assure whether the features they had seen were due to jellyfish because of the problems obtaining adequate in situ or airborne data describing the spatial extent of particular blooms. Cloud cover was also an issue.

Nowadays, this is a clear requirement for improving data on jellyfish distribution and abundance in order to monitor trends and to identify whether distributions are random or if hot-spots occur [7]. There is a need to develop methods that allow reproducible standards at a reasonable cost [27]. As it has been mentioned in [23] the availability of such data would eventually allow the development and testing of predictive models which might provide warnings to the industry of impending blooms and inform site selection for new farms. Given sufficient warning, the industry can potentially take precautionary actions such as harvesting or moving fish or employing physical barriers such as mesh screens.

The benefits of early warning schemes are firstly, a better communication between individual fish farm sites, allowing thus the development of integrated bloom control or mitigation strategies when a bloom is predicted/experienced. Secondly, the fact that participating parties have access to clear “real-time” information describing the water quality throughout the study region, allows personnel to make informed decisions regarding fish husbandry. Moreover, email, fax or mobile phone alerts can be rapidly generated to participating parties as soon as certain water quality criteria are exceeded anywhere in the study region.

As it has been mentioned before, proposing and developing an automatic procedure to locate and identify the distribution and abundance of jellyfish, could be of great significance. This method could be used in conjunction with the features listed in [27] to improve our knowledge about jellyfish live. Although there are some studies on the detection of jellyfish remotely, they all require human intervention for conducting and checking data acquisition. We propose to use a swarm robotic system that will use UAVs to perform this task in an autonomous way.

Swarm robotics is an approach to solve problems inspired by the collective behaviour of social animals and it is focused on the interaction of multiple

robots. It is a different approach to classical artificial intelligence, where the main goal is usually to develop behaviours that mimic human brain function. Swarm Robotics is based on the metaphor of social insects, and emphasizes aspects like decentralized control, limited communication between agents, local information, emergence of a global behaviour and robustness [6].

In general, robots behave like little insects, i.e. they do not have many skills individually, but when combined in large groups, they provide amazing results. Their underlying modular technology allows parallel tasks and has many applications, including: searching and rescuing people in natural disasters, replacing human workers in dangerous environments and exploring unknown or dangerous environments, amongst others. Swarm robotics is also used to identify the minimum requirements of agents that interact biologically, thus allowing the understanding of certain biological phenomena, such as the collective behaviour of bacterial systems [33].

To complete a sophisticated task, a single robot must be designed with complicated structure and control modules resulting in high cost of design, construction and maintenance. Single robot is vulnerable especially when a small broken part of the robot may affect the whole system and it's difficult to predict what will happen. The swarm robotics can achieve the same ability through inter-group cooperation and takes the advantage of reusability of the simple agents and the low cost of construction and maintenance. The swarm robotics also takes the advantage of high parallelism and is especially suitable for large scale tasks [37].

The advantages of swarm robotics are [37]: parallelism (the population size of swarm robotics is usually quite large, and it can deal with multiple targets in one task), scalability (the interaction in the swarm is local, allowing the individuals to join or quit the task at any time without interrupting the whole swarm), stability (similar to scalability, the swarm robotics systems are not affected greatly even when part of the swarm quits due to the majeure factors), economical (the cost of swarm robotics is significantly low in designing, manufacturing and daily maintaining: the whole system is cheaper than a complex single robot even, if hundreds or thousands of robots exist in a swarm) and energy efficiency (since the individuals in the swarm are much smaller and simpler than a giant robot, the energy cost is far beyond the cost of a single robot compared with the battery size). This is the reason why swarm robotics has become an important research field in last decade.

Multi-swarm robotic systems differ from other multi-robotic systems because [32]: (1) robots in a swarm are autonomous robots located in a certain

environment, (2) the swarm has a large number of robots, (3) the swarm is composed of small groups of homogeneous robots, (4) robots are relatively simple, and (5) the robots have local sensors and their communication skills are limited. These features ensure that the coordination between the robots is distributed and that the system is fault tolerant, since due to the redundancy of robots each of the agents in the system is not essential and can be replaced by another agent.

These kind of systems will be easily scalable, allowing more agents to be added or deleted according to task demands. This is essential for the task we want to develop. Given the variability in size of jellyfish blooms, being able to increase or decrease the number of UAVs operating for a specific period is a major requirement in order to develop the task correctly.

However, the distributed nature of these systems makes difficult to develop an architecture that correctly models a swarm and that coordinates the swarm to perform complex tasks. In order to design a swarm behaviour a mathematical model must be provided for both, the individual agents and the whole swarm. These models will be tested to evaluate the performance of the swarm before its deployment to the real UAVs. This is mainly because this kind of systems have emergent properties, that make difficult to predict the overall operation of the swarm when only the local behaviour of the agents are analysed.

As discussed above, and due to decentralized control, swarm systems are able to solve complex problems that a single agent cannot solve. Most times, the swarm is more than the sum of its agents, and the emergent properties of the system allow us to solve tasks that have not explicitly been defined in the microscopic behaviour. Designing this kind of behaviour is complex, particularly when they need to be highly scalable. Therefore, in most of these systems, defining formal models and simulations with a large amount of robots, is essential in order to understand the system evolution.

A sample of interest of this kind of robotics is shown in the development of swarms totally adapted to the aquatic medium [4]. These swarms are being used for detecting and monitoring the marine environment, such as the development of deployment behaviours for extracting the temperature of a certain area [9]. We can also find several papers related to the detection of jellyfish [19, 17, 18] but these systems are oriented to its extermination and require previous localization of the blooms so they have not shared the same purposes of this work.

In swarm robotics research field, several works for the covering and track-

ing tasks can be found, such as those presented in [6, 1, 10, 21, 8, 34, 35]. However, we have found that many of these works do not take into account the special characteristics of the UAVs and therefore are not energetically applicable [6, 10, 21, 8]. Others works, although were designed for this type of vehicle, are not applicable to the required localization and tracking tasks and are not adapted to the specific needs of non-structured environments, required in marine exploration and jellyfish blooms detection [34, 35].

In this article we will propose a swarm behaviour for UAV, that will be formalized with a microscopic model. This behaviour will be able to locate and track a jellyfish bloom that may be composed with several jellyfish groups. We will discuss both the convergence and the scalability of this model. In addition, a macroscopic model, that will predict the place or places where the agents will be aggregated, will be discussed.

In the following section we will specify the features of both, UAV's swarm and the environment to be used, given the localization and tracking task to be carried out.

## 2. Test Environment Design

We must not forget that our ultimate interest is to deploy this system using multiple UAVs in maritime surroundings. Therefore, it is crucial to properly design the parameters of both, the model and the simulations to be as close as possible to reality. In this section we will design our test environment considering the main features of the UAV robots (and adapting the sensors and the flight planning to the tasks of detecting and monitoring jellyfish blooms), taking into account the specific needs of these tasks. In addition, the swarm simulator, used for the experimentation section of the microscopic model will be presented.

There are therefore several studies that test remote detection of jellyfish blooms. For this purpose they use aircrafts, that trace the environment at a fixed altitude between 150m and 300m. Human experts inside the plane are able to visualize an area of almost  $8 \text{ km}^2$  every 5 minutes, thus obtaining, for an altitude of 150m, a vertical field of view (Vertical FOV) of 300m.

We propose to use a swarm of UAV robots. In this case we want to trace *Aurelia aurita* jellyfish, the most common in the Mediterranean Sea, in order to notify and analyse the blooms that may appear in the Mediterranean tourist coasts, especially on the beaches.

Due to the fact that the morphology of these jellyfish is more difficult to detect than, for example *Cyanea capillata* (the lion’s mane jellyfish) and *Chrysaora hysoscella* (compass jellyfish), we have decided to use low altitude flights, about 30m above sea level. Given that UAVs will be equipped with a 10mm lens and a sensor with focal length multiplier of 1.6, we get a vertical FOV of 45m per agent. Although this paper does not describe the vision algorithm for the jellyfish detection, it is important to highlight that low altitude flights facilitate so much the image segmentation algorithm. In order to simplify the simulation of the swarm, we assume that a single UAV has a detection area of 3x3 meters, regardless of its height, just below it.

The UAVs we want to simulate are multicopters, although the behaviour presented here would be easily adaptable to fixed-wing drones. The main features of this kind of UAVs is that they can remain stationary in a given point and can move with a velocity around 35km/h. As shown in several studies, the jellyfish blooms move at a max speed of 1km/h, so that the UAV swarm has enough velocity to detect and track the bloom. In the following simulations we will neglect the jellyfish displacement, given that the speed of the robotic swarm is much greater.

The displacement of this type of jellyfish is mostly horizontal. They are located from the sea surface (0 to 5m) to about 15m depth. Thereby, large jellyfish accumulations are perfectly visible from 30m height. The size of jellyfish blooms goes from a few  $m^2$  (the most common) to pests, such as the salmon farm one, which reached  $26km^2$ . This requires a high scalable system, that must be taken into account when developing swarm behaviours.

Furthermore, in order to develop our behaviour in a successful way, each UAV needs to use a local communication mechanism between nearby agents. Devices like XBEEs and networks like DigiMesh can be very useful when working with distributed systems [11]. In our simulator, we assume a local communication radius of 200 meters, much lower than the real one (with a range over 1km) to ensure system robustness. In addition, a mechanism for local positioning of an agent from their neighbors must be provided. This mechanism can be as easy as using GPS signal (where available) or to use alternative methods, such as to measure the signal strength of nearby neighbors for position triangulation. Since our purpose is to find blooms of jellyfish in areas close to the beach, we will assume that we have GPS positioning to conduct the tests in our simulator. However, we want to emphasize that our behaviour does not require global positioning whenever you provide any mechanism to locate an agent with respect to its nearest

neighbors.

Finally, we will assume that the drones will implement some obstacle avoidance system. As we work with multicopters, the easiest option is to force the vehicles to fly in different heights. Moreover, if using the neighbours signal strength to locate the UAVs in the swarm, it is relatively simple to take into account the signal strength to determine a security perimeter between agents.

We want to underline that both, the discussed UAVs features together with the simplicity of the behaviour to be executed (discussed in the next section) enable us to build the system with low cost drones, around 250\$ each unit. This is critical because a swarm behaviour must have a large enough number of agents for their successful implementation. When working with swarms, a low price per unit means more agents in the swarm, considering all implications.

For the simulation of the system, MASON simulator for multi-agent systems has been used [20]. Based on the MASON simulator, we have developed a continuous 2D environment, which contains discrete resource cells, and each cell having the same size as an agent (1x1m approx.). Unless specified, all agents will be positioned randomly in the environment before starting the simulation. For each of the simulations there is an option to limit the maximum size of the environment (by default, it is assumed that the environment is bounded). We will work with environments between 1 to 5  $km^2$ .

In the next section, we will design a swarm behaviour able to locate a bloom of jellyfish in the sea. Considering the features listed above, we will provide a microscopic model for a swarm system of UAVs.

### 3. Microscopic Model

Modelling is a method used in many research fields, to better understand the internals of the investigated system, that has several advantages for swarm-robotics. The existence of possible risks for the robots and the limited power of the robots require a human observer to follow the experiments and do some house keeping works periodically. The time spent on these experiments and possible risk of losing the robots, even if a human observer exists, become a bottleneck when several experiments are needed to validate the results of the studies. To eliminate these problems it is safer and easier to model the experiments and simulate them on computers. Another importance of modelling for swarm robotic studies appears when the scalability of



the experiments are tried to be tested. Most of the time, scalability requires to test the control algorithms on more than hundreds of robots *which is cost prohibitive* [1].

In addition, when designing swarm-control mechanisms, researchers and engineers are faced with the challenge to develop a set of rules at the individual (microscopic) level such that a desired behaviour at the group (macroscopic) level is achieved. This is a very difficult task since there is no general systematic way to devise individual behaviours that reliably achieve a desired group behaviour. Thus design choices can usually only be tested in experiments or simulations [38].

In this section we want to design a microscopic model, which is able to locate and track a jellyfish bloom. Although for the system deployment UAVs will fly at 30 high (and as we have seen this gives us a 45m vertical FOV), we will assume that for our simulations a particular agent can locate a bloom only if it is just below its position within a 3x3 meter box (regardless of their flight altitude).

This is done for two reasons: firstly, we can test the robustness of the behaviour when the detection is extremely limited, including in the model the possible difficulties of the visual system together with the environmental conditions (that could affect the detection). Secondly, it provides a model that does not depend on the resource to be searched (which determines, among other factors, the flight altitude). It is important to keep in mind that although the dynamics of the bloom and the meteorological conditions (e.g. wind) will not be modelled they will not affect significantly the presented model. On the one hand, if there are winds whose speed is close to the one of the individuals of the swarm (35km/h) these devices cannot be used. At lower speeds and taking into account the average movement of a bloom (1km/h) the swarm can perform its task independently of the surrounding winds and tides (which can also be very difficult to predict in certain situations). Due to its low speed we consider the dynamics of the blooms to be negligible, also achieving a more generic behaviour, independent of the species to be followed and much simpler (which is very important to allow the calculation of the macroscopic model).

Moreover, we will assume the most common size of bloom of a few  $m^2$  [29], because they are the most frequently found in the the coastal area where we want to deploy the system.

It seems reasonable to think that the behaviour of a given UAV will be different, depending on whether it wants to locate a bloom or to monitor

it. In this case, we suggest that an UAV could be in two different states (only one at a time), the state *wander* and the state *resource*. Therefore, our microscopic model will be very simple: all the agents will start within the *wander* state. If a jellyfish bloom is detected, agent will change to state *resource*. Finally, if the bloom is lost, the agent will return to *wander* state. In the following section, we will describe in detail the behaviour of the agents in each of these states.

### 3.1. Wander

At the beginning of the behaviour execution agents starts at *wander* state. At that time they have no idea where to find a jellyfish bloom. Therefore, all directions are equally valid and they will move randomly. Each agent  $i$  will select a displacement direction that will be slightly modified each few seconds:

$$\mathbf{v}_{wander_r}(t) = \mathbf{v}_{wander_r}(t-1) + \mathbf{rand}() \cdot \mu_1 \quad (1)$$

where  $\mathbf{rand}()$  is a gaussian random vector with zero mean and unit variance and  $\mu_1 \in \mathbb{R}$  is a scalar term that specifies the intensity used to modify current direction. However, it is possible that some agent has detected a nearby jellyfish bloom. In this case this agent will communicate its location to those UAV close to its position. Agents in *wander* state that receive this communication will no longer move entirely at random, but will move towards it depending on their distance to the bloom will move towards it. In this way, we define that for each agent  $i$  in state *wander*:

$$\mathbf{v}_{wander_g} = \sum_{n \in neigh(i)} \left( (\mathbf{r}_i - \mathbf{r}_n) \times \frac{1}{dist(\mathbf{r}_i, \mathbf{r}_n)^2} \right) \quad (2)$$

where  $r$  is the position of the agents (relative or absolute),  $neigh(a)$  is the function that returns the agents inside the detection range of  $a$  and  $dist$  the euclidean distance between two points. The meaning of this equation is simple: an agent must move to the average of the vectors that have reported their neighbors, depending on how close it is to them (the closer the higher intensity). In our case, we decided that the strength of attraction of such vectors is decreased with the square of the distance.

Agents in *wander* state must perform one more task. If an agent tells them that he found resources it must relay such information. Thus, an agent in *wander* state that receives a vector pointing to a resource location will

update it, depending on their status and will broadcast it to its neighbors. This allows us to exceed the limit imposed by local transmitting distance. However, we must take into account two factors: Firstly, in the proposed algorithm we do not require swarm cohesion and therefore some isolated groups of agents, that will not be notified about the position of the bloom, could exist. Secondly, the attraction intensity varies with the square of the distance, so the zones far away from the resource will not be influenced by it. This is consistent with the design of our swarm: only those agents close enough to the resource should track it, the other agents must be prepared to locate other different blooms.

Thus, the velocity of an agent in *wander* state is defined as

$$\mathbf{v}_{wander} = (1 - \mu_{goto}) \cdot \mathbf{v}_{wander_r} + \mu_{goto} \cdot \mathbf{v}_{wander_g} \quad (3)$$

where  $\mu_{goto} \in [0, 1]$  is the intensity used by an agent to follow the information of the resources detected by another agents.

### 3.2. In Resource

Once the resource has been identified by an specific UAV it should go towards it. The agent must do several things: firstly, it should move just above the resource and stay on it. Secondly, the possibility of collision with other agents must be avoided. Moreover, once it is placed on the resource it must inform to their neighbors about its location. Finally, the stochastic nature of their sensors and their actuators must be taken into account.

More specifically, the velocity of an UAV is defined given three terms:

$$\mathbf{v} = \alpha_{compass} \cdot \mathbf{v}_c + \alpha_{obstacle} \cdot \mathbf{v}_o + \alpha_{rand} \cdot \mathbf{v}_r \quad (4)$$

where  $\alpha_{compass} \in \mathbb{R}$ ,  $\alpha_{obstacle} \in \mathbb{R}$ ,  $\alpha_{rand} \in \mathbb{R}$  and  $\alpha_{compass} + \alpha_{obstacle} + \alpha_{rand} = 1$  are terms to adjust the behaviour.

$\mathbf{v}_c$  specifies the robot direction, which is determined by the area with more resource intensity. The larger the sensor detection area the more important  $\mathbf{v}_c$  will be, thus moving the agent faster to its goal.

$$\mathbf{v}_c = \left\| \sum_{s \in S} (\mathbf{pos}(s) - \mathbf{r}_i) \cdot s \right\| \quad (5)$$

where  $S$  is the set of readings obtained by the resource sensor in a specific time and  $\mathbf{pos}$  is the position of a given reading. We assume that the resource

intensity is between the range  $[0, 1]$ , where 0 means a complete lack of resource and 1 full resource detection.

$\mathbf{v}_o$  defines an obstacle avoid vector, obtained from all nearby robots detected at an specific time:

$$\mathbf{v}_o = \left\| \sum_{j=1}^{|R|} \mathbf{r}_j - \mathbf{r}_i \right\| \quad (6)$$

where  $R$  is the set of detected robots,  $\mathbf{r}_j$ , the position of the detected robot  $j$ , and  $r_i$  the position of the current robot.

Finally, we will take into account the accuracy of the transmitted locations: there are several factors that could make these locations not to be optimal. We will include therefore a random component to model this uncertainty in the movement of the robot:  $\mathbf{v}_r(t) = \mathbf{v}_r(t - 1) + \alpha_{rand} \cdot \mathbf{rand}()$ .

In Table 1 it is possible to observe an example of the parameters of the microscopic model used in the experimentation and a detailed description of its effects.

## 4. Experimentation

In this section, we will conduct several experiments using the previously presented microscopic model. We will analyse its operation using velocities, distances and blooms shapes as close as possible to the final implantation environments. First, the behaviour will be executed over a  $5km^2$  environment with 3 blooms analysing its evolution with 20 UAVs. Next, we will analyse the convergence for this map, developing 600 simulations on the same environment and showing how the behaviour is able to locate and maintain on the 3 different blooms. To determine if the behaviour is generalizable to any type of map we will generate several maps (300) with different number of blooms, on which the behaviour will be evaluated. Finally, the scalability will be analysed by varying the number of UAVs between 5 and 100, analysing the times required for bloom covering.

### 4.1. Localization

In order to develop our tests, several small and medium sized areas from previous aerial surveys have been extracted (eg. [24, 29, 23]). These zones have been introduced into our simulator as a testing bench to evaluate the microscopic behaviour of the system. We have worked with areas of  $1km^2$

and  $5km^2$  (25 times higher than the range of the simulated communication modules). In order to cover these areas we have assumed that we have a swarm of 20 UAV, which is arranged randomly in the environment. The jellyfish spots have mean sizes near  $23m^2$ . For all the experiments presented here we have used the values specified in Table 1.

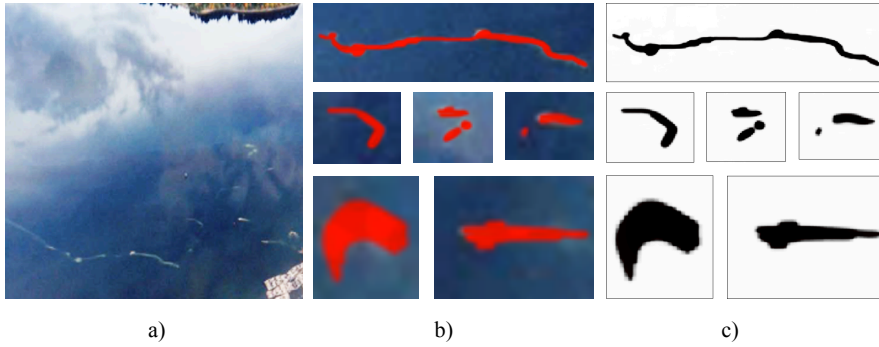


Figure 1: a) Example of aerial image with jellyfish areas. b) Extraction of jellyfish areas from previous image. c) Segmentation of the jellyfish groups as used in the simulator

Now we will present the results obtained by simulating the behaviour specified in our microscopic model. First, we will present a single execution of the algorithm where the evolution of the swarm towards the resource will be shown. Next, the convergence of the system will be commented, taking into account the 600 simulations developed with our model.

In Figure 2a an example of a test environment, with an area of  $5km^2$ , and three jellyfish blooms is shown. In Figure 2c the percentage of agents (of the total of the swarm) that are located above a bloom is shown. As can be seen, not only the agents find all blooms (which happens at  $t = 537s$ ), but the 100% of individuals are capable of standing on the bloom from minute 40 until the end of the simulation at  $t = 3000s$

Although illustrative, this example only indicates the system performance for a given map. That is why, for making more exhaustive tests, we have decided to generate maps with random jellyfish blooms. The generation is performed with a basis on real jellyfish bloom photographs, by rotating and placing them randomly. To avoid overlap, when generating more than one spot, their position is forced to be at least twice the distance of communication of the UAVs.

In Table 2 the localization results from the 100 simulations performed for each map size (1km and 5km) and for each number of jellyfish blooms (1 to

Table 1: Microscopic model parameters used in the experimentation

Parameter	Value	Description
$\mu_1$	0.3	Weight of the random speed. The higher $\mu_1$ the more random and less uniform movement. Depends directly on the used actuator
$\mu_{goto}$	0.3	Determines how the speed of <i>Wander</i> state is calculated. Establish the relationship between wandering and following the recommended resource. The range varies between $\mu_{goto} = 0$ , where it wanders and $\mu_{goto} = 1$ , where the resource is followed directly, with no wandering
$\alpha_{compass}$	0.3	Weight within the computation of state <i>In resource</i> speed, related to resource location
$\alpha_{obstacle}$	0.25	Weight within the computation of state <i>In resource</i> speed, related to obstacle avoidance
$\alpha_{rand}$	0.45	Weight within the computation of state <i>In resource</i> speed, related to random movement

Table 2: Average time in seconds taken by the swarm to locate the spots (jellyfish blooms) specified in the maps A, B and C. Only the simulations that locate all the spots will be taken into account (100% for one spot, 96% for two and 76% for three). Maps are randomly generated with one, two or three spots (A, B and C respectively), with a spot area from 20 to  $26m^2$ .

	A	B	C
1 km	181.56	596.46	645.37
5 km	301.58	926	801.11

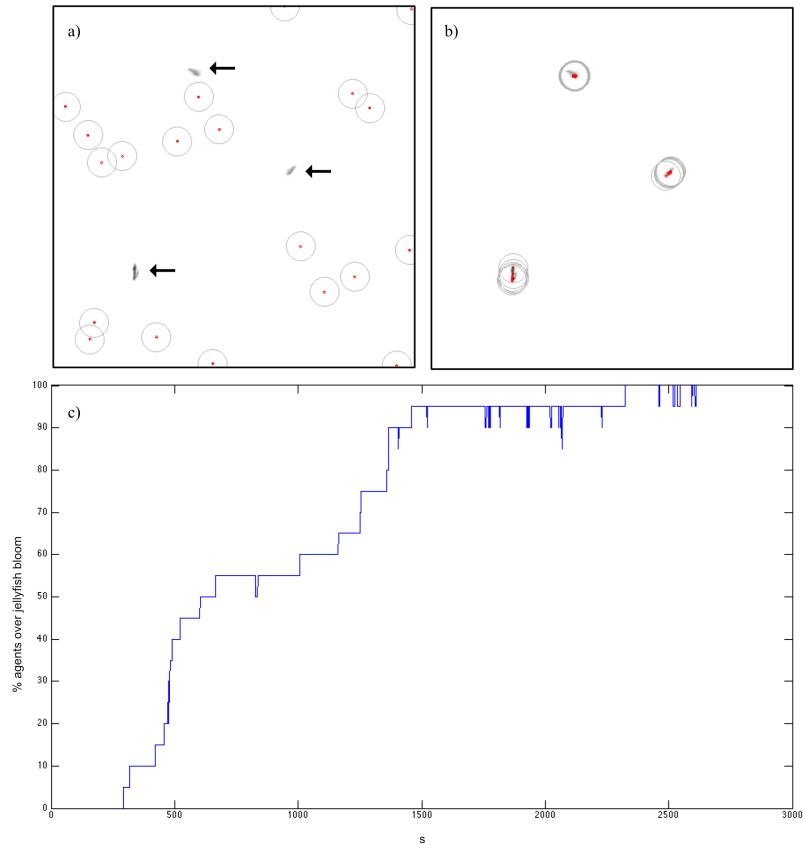


Figure 2: a) An environment of  $5km^2$  with three jellyfish blooms (marked with arrows) and the initial arrangement of the individuals of the swarm are shown. Robots are represented with a red dot and a circle concentric to their position that marks their area of communication. Initially, the agents are distributed randomly in the environment. b) Position of the agents after 3000s of simulation. As can be seen, all of them are over one of the three blooms. c) This graph shows the percentage of agents (of the total of the swarm) that are located above a bloom. Not only the agents have found all the blooms (which happens at  $t = 537s$ ), but the 100% of individuals are capable of standing on the bloom from  $t = 2335s$  until the end of the simulation  $t = 3000s$

3, with an area between 20 and  $26m^2$ ) are presented. In this table the time needed by the swarm to locate all the resources (with a minimum of two agents per resource) is shown.

Simulations of more than 120 minutes have been discarded (and counted as incorrect) if all the resources have not been detected. The detection ratio for one jellyfish bloom is 100%, for two 96% and for three decreases until 76%.

When working with two or three spots we found that sometimes the agents cannot find a spot. Our behaviour has mechanisms to prevent this from happening: firstly, the attraction of remote spot location, obtained by the messages been transmitted by other agents decay with the square of the distance. In addition we could use  $\mu_{goto}$  to graduate their attraction intensity. Secondly,  $\alpha_{obstacle}$  may be increased so that when more agents come to a more intense area will be the repulsion vector of the agents, leading them outside the spot. However, in this section the same model parameters, specified in Table 1, will be used for all the test so that the comparison between experiments is simplified.

Therefore, we consider that the presented behaviour develops the expected results for jellyfish bloom localization and tracking. Moreover, we have demonstrated experimentally that this behaviour is able to localize more than one bloom, even if the distance between spots is much larger than the UAVs communication range.

#### 4.2. Scalability

In the previous section we have tested the performance of the swarm in a simulator, defining its settings based on the real environment where the swarm will be deployed. We have evaluated the convergence of the swarm for several jellyfish blooms with a limited number of agents. Nevertheless, all the tests have been developed with the same number of UAVs. In this section we will test the scalability of the system by evaluating its performance regarding the number of used agents.

We define the scalability of the system as the ability of the swarm to be functional and to develop the behaviour independently of the number of individuals that form it. It should be noted, however, that the presented microscopic model does not allow the online request of new individuals by design (since we cannot assure the availability of communication coverage with the base and assume that all available drones will be launched at the first stage of the behaviour). However, it is relatively easy to know the



number of drones necessary to surround a bloom if its size is known (since given the UAV area of perception and its flying height we can know how many agents would be required). Otherwise, a first flight of the swarm will reveal the size of the bloom to be covered in subsequent flights.

We will rely on a fixed 5km environment with a jellyfish area of  $25m^2$ , where several convergence tests will be conducted.

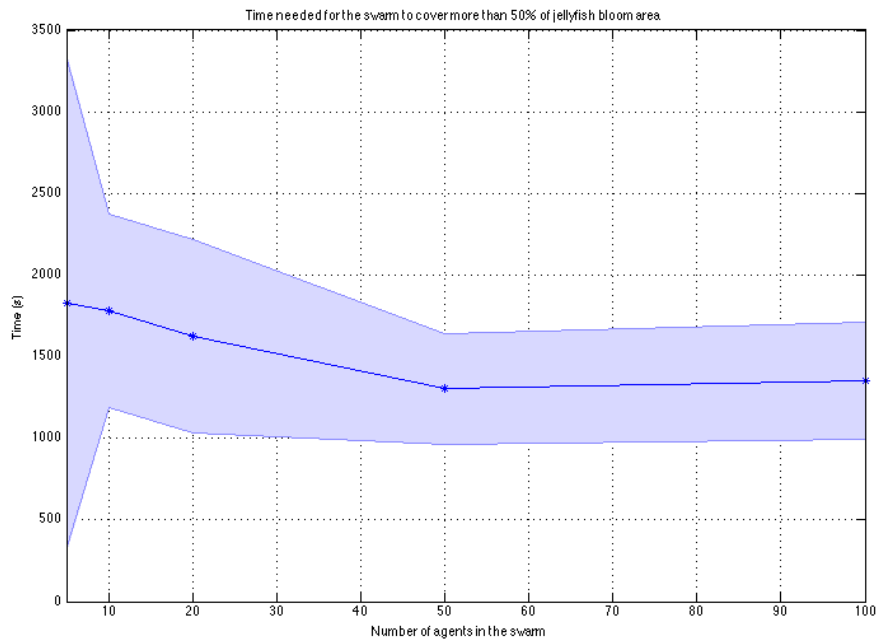


Figure 3: Time required by the swarm for being over a jellyfish bloom (more than the 50% of its agents). The mean and the standard deviation of the convergence time is shown for 20 executions for each swarm size: 5,10,20,50 and 100

In Figure 3 the time required for more than 50% of agents to be placed above a resource is shown. We have conducted 20 different executions for each different number of agents (5,10,20,50 and 100). For example, as it can be seen in the figure, for a swarm with 10 agents the 50% of the swarm locates the bloom in less than 30 minutes. On the basis of these results we could summarize that this behaviour is highly scalable. The optimal number of agents to be used depends on the features of the environment and the maximum number of available agents.

When we have tested a small size swarm we find that some initial time, until one of the robots locate the resource, is required. That time is determined directly by the area to be explored, the size of the bloom and the number of swarm agents. With current environment size a swarm with 10 agents goes away from nearly random search behaviours, as shown with the 5 UAVs swarm, reducing the execution variability. Moreover, from 50 agents the difficulty of spot covering (more than the 50% of swarm individuals must cover a spot) is incremented. This is due to two factors: the number of agents is increasing (more agents have to cover the resource) and not all the agents can be located over the bloom (because of the collision avoidance behaviour).

Thus, after analysing experimentally the scalability of this behaviour, we believe that it meets the requirements to be deployed in a real system. As discussed above, both the size of the bloom and the exploration area can significantly alter the number of agents required to perform this task successfully.

## 5. Macroscopic Model

Previously, we have tested our microscopic model for jellyfish bloom localization and tracking. The behaviour defined in this model is executed locally by each of the swarm agents. We have verified experimentally both, the convergence and the scalability of the system. In this section we will present a macroscopic model that will describe the functioning of the system in a global way. Therefore, this model will show the swarm evolution and could be used to analyse and predict their evolution. The fundamental advantage of this type of modelling is that it does not require a multitude of microscopic simulations to visualize the global behaviour and the tendency of the swarm, and can therefore be used iteratively in the development of global modelling and behavioural adaptation. Finally, we will test your microscopic model, comparing its prediction with the results obtained through our microscopic model.

Swarm robotics systems can be modeled at two different levels: the individual level, or microscopic level, that models the characteristics of the single individuals and the interactions among them; the collective level, or macroscopic level, that models the characteristics of the entire swarm. The development of models for analysing swarm robotics systems at both levels of abstraction is still a subject of study and research. In fact, modelling

both the microscopic and the macroscopic level and their interaction is very difficult due to the nature of self-organized systems [1].

Ideally, a unified approach to modelling a robot swarm will derive parameters for a microscopic description from experiments, derive macroscopic equations from the microscopic model and perform microscopic simulations to validate the macroscopic description. Several factors seem to make such an approach challenging. First, microscopic simulations, while more accessible than physical experiments, generally require substantial computational resources if they involve a large number of robots. Second, physical experiments are expensive, time-consuming and can usually only be conducted under sanitized laboratory conditions. Third, deriving macroscopic descriptions from probabilistic microscopic ones is usually hard, in particular if spatial aspects need to be taken into account [38].

There are several techniques for analysing microscopic behaviours, such as the use of recurrence equations (obtained by a microscopic behaviour if it is defined as a finite state machine), the use of differential equations, probabilistic approximations. . . However most of these methods only allow to analyse the evolution of the agent state transitions globally.

In this paper, based on the proposed framework in [14], the probability distribution of the swarm position for a time  $t$ ) will be obtained. This will enable us to predict in a very detailed way the overall system behaviour. As commented in [14], once defined the microscopic behaviour, we can calculate the overall operation of the system using the Fokker-Planck equation:

$$\frac{\partial \rho(\mathbf{r}, t)}{\partial t} = -\nabla \cdot (\mathbf{A}(\mathbf{r}, t)\rho(\mathbf{r}, t)) + \frac{1}{2}Q\nabla^2 (B^2(\mathbf{r}, t)\rho(\mathbf{r}, t)) \quad (7)$$

where  $Q$  is the displacement by a collision.  $\nabla \cdot (\mathbf{r}, t) dr_x dr_y$  is the probability of encountering a robot at position  $\mathbf{r}$  within the rectangle defined by  $dr_x$  and  $dr_y$  at time  $t$ .  $\rho$  is the density of moving robots at time  $t$ .

This equation provides a method to statistically model a swarm of robots based on multi-particle modelling techniques from the field of quantum physics. Starting with a Langevin equation, which represents the behaviour of a single particle, the Fokker-Planck equation for the entire swarm is derived. As commented in [13], the Fokker-Planck equation implements the necessary abstraction of microscopic details as described above and treats rapidly-changing parameters as noise. The equation is still exact if this noise is generated by a Gaussian process, that is, if it is fully determined by the first two moments. It gives the temporal evolution of the probability density

describing the positions of the agents.

As it can be observed in this equation, all the swarm behaviour must be modelled using only two functions,  $\mathbf{A}$  and  $B$ . Function  $\mathbf{A}$  is a direction and describes the deterministic motion based on information provided by the environment and the information indirectly provided by other robots via the environment. Function  $B$  describes the random component of the motion.  $\mathbf{A}$  and  $B$  are characterized by the underlying control algorithm.  $B$  typically incorporates influences by the other robots that disturb the robot, for example, by the need of collision avoidance.  $\mathbf{A}$  might incorporate an external influence such as a light gradient.

Thus, function specified by  $\mathbf{A}$  depends on the direction used by robots. This direction is mainly determined by the messages sent by other agents to communicate any resource discovery event. We can therefore assume that a potential virtual field exists  $P(\mathbf{r}_i, t)$ , affecting each UAV  $\mathbf{r}_i$  at time  $t$ . It seems reasonable that the goal of this potential field will be found at the jellyfish areas.

In order to model our swarm we propose the following definition of  $\mathbf{A}$ :

$$\mathbf{A} = \left( \sqrt{1 - n(\rho)} \right) \frac{\nabla P}{\|\nabla P\|} \mathbf{v} \quad (8)$$

where  $n(\rho) = \frac{\min(\rho, \gamma_r)}{\gamma_r}$ , given that  $\gamma_r$  is the maximum expected density of robots. In this way  $\mathbf{A}$  will be defined as the normalized potential field gradient (since we are interested only in its direction) and multiplied by the nominal speed of the system. In an environment without collisions this equation would be already defined. However, the more robot density found in a point the higher probability of collision and thus, less probability of a robot to develop successful movements. In this way, term  $1 - n(\rho)$  penalizes the zones with high density of robots.

Function  $B$  defines the nondeterministic movement of the system. This movement is defined by the random component of *wander* state or by robot collisions (forced trajectory modifications). Thereby,  $B$  is defined as:

$$B = \gamma_w + \gamma_{c1} n(\rho)^{\gamma_{c2}} \quad (9)$$

where  $\gamma_w$  is a factor determined by the random movement of the agents in *wander* state and  $\gamma_{c1}$ ,  $\gamma_{c2}$  determine the influence on the number of agents in terms of the probability of suffering a collision in conjunction with the physical errors that could be found in the agents sensors and actuators.

Table 3: Macroscopic model parameters used in the experiments

parameter	value	Description
$\gamma_r$	0.0036	Maximum expected density of robots
$\gamma_w$	0.7	Determined by the random movement of robots
$\gamma_b$	0.85	Normalization term of the robots density
$\gamma_{c1}$	2	Determined by collisions and physical errors of UAV
$\gamma_{c2}$	0.1	Determined by collisions and physical errors of UAV

Finally, we define the virtual potential field  $P$ , that it is generated by the swarm communication as a partial differential equation (PDE):

$$\frac{\partial}{\partial t}P(\mathbf{r}, t) = m_{bloom}(\mathbf{r}, t) \cdot \gamma_b \rho(\mathbf{r}, t) \quad (10)$$

where  $m_{bloom}(\mathbf{r}, t) = \min(dist(\mathbf{r}, \mathbf{b})^2)$ , given  $\mathbf{b} \in Target$  any point of the jellyfish area and  $\gamma_b$  a normalization term of the robot density. Thus, the potential field is defined as the squared distance to the nearest jellyfish resource, multiplied by the amount of existing robots at that point (that would help to transmit this information).

### 5.1. Experimentation

In this section we will test the proposed macroscopic model. We will use a map with a medium size resource ( $140m^2$ ), to easily observe the robot density predicted by the model. The size of the map will be  $1km^2$ .

Our macroscopic model requires a number of parameters to be fitted. In this case, the adjustment has been developed experimentally. As discussed in [14] it is not a complex task and can be done automatically using learning techniques. More specifically, we have used the parameters presented in Table 3.

In addition, in order to solve the previous PDE equations we require that the agents share a certain starting position. For this test we have assumed that all the agents will start at position (100,100).

In Figure 4 the performance of the proposed model is shown. As it can be observed the swarm reaches the bloom at quite the same time as the microscopic model. At 40 minutes the swarm has fully located the jellyfish bloom.

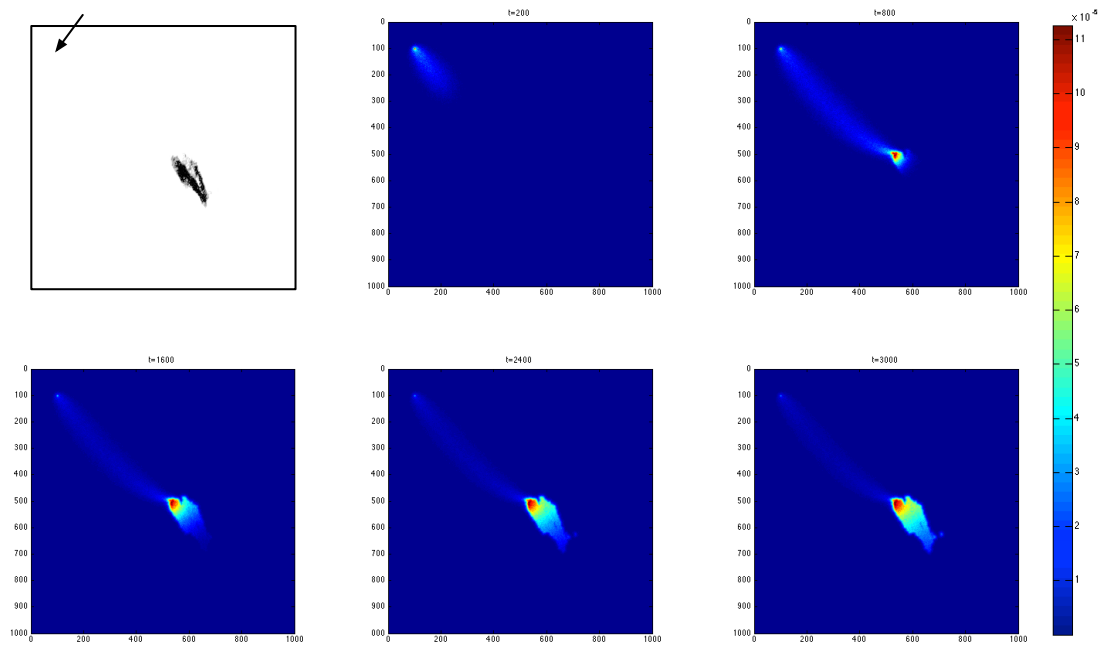


Figure 4: Picture on the left display the jellyfish bloom and the origin of the swarm. Remaining pictures show the evolution of  $\frac{\partial \rho(\mathbf{x}, t)}{\partial t}$  for each value of  $t$  in  $\{200s, 400s, 800s, 1600s, 3000s\}$ . Colour indicates the probability of an agent to be in a given position at time  $t$ .

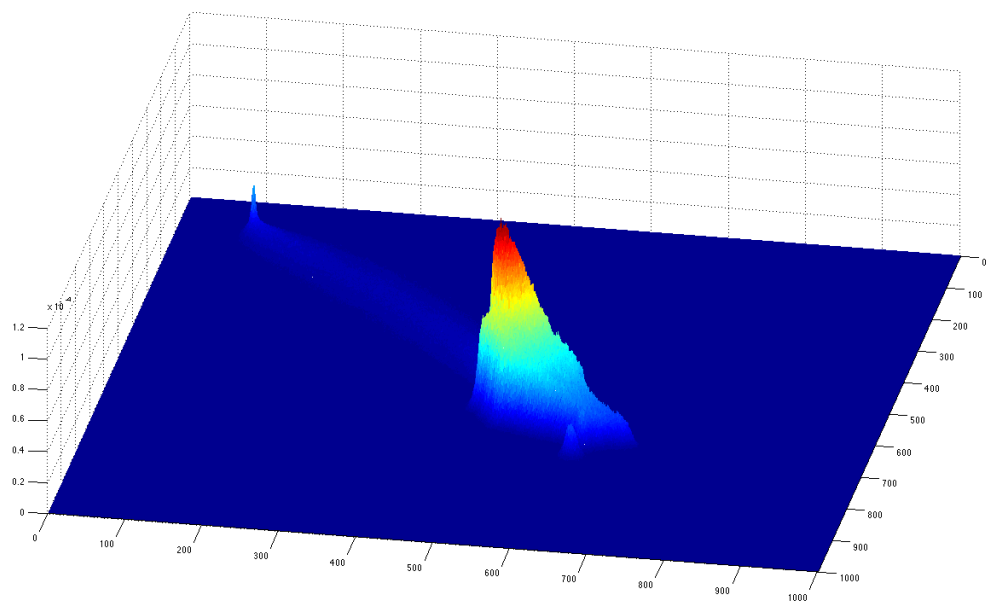


Figure 5: Three-dimensional representation of  $\frac{\partial \rho(\mathbf{r}, t)}{\partial t}$  for  $t = 3000$ .

As it has been shown in Figure 5, the model predicts that the swarm will cover the entire bloom. The jellyfish bloom areas with more robots will be those closer to the starting point, because the attraction produced by the local communication messages will attract the swarm to this area.

Although not discussed in this article, it is easy to verify how the predictions made by the macroscopic model are qualitatively consistent with the microscopic simulations relating to the covered area. However, some differences should be noted.

With a relatively small number of agents, like those that we have used in the macroscopic section (around 100 agents), we do not observe the remaining agents in the starting point, predicted by the macroscopic model, that can be observed in Figure 5. This can be caused because either the Fokker-Plank equation assumes that robots are particles (with no mass), causing the density of the environment to grow in an unrealistic way in some areas or because no tests have been performed with enough number of agents. Furthermore, due to other factors, the probability distribution does not match quantitatively with the predicted by the microscopic model, although it can be used qualitatively (eg. the same jellyfish bloom areas are detected). Nevertheless, a better fit between the two models is possible by applying learning techniques that adapt the macroscopic model parameters optimally as discussed in [13].

In this section we have presented a macroscopic model that predicts the evolution of a swarm for a given resource map. This model provides the probability distribution of an agent being at a specific position in a given time. The model can be used to check and adjust the behaviour of the microscopic model and to test the performance of the proposed behaviour in a much more quickly and concise way, without requiring any robotic simulation.

## 6. Conclusions

There is a need to acquire real data of jellyfish distribution and abundance in order to monitor trends and to identify whether distributions are random or if hot-spots occur. In this article, a swarm behaviour has been presented oriented to the detection and monitoring of jellyfish blooms using swarm robotics. The current jellyfish blooms acquisition and monitoring techniques require indirect detection systems or overflights in certain areas. They cannot be developed on a continuous basis due to their high costs.



Swarm robotics is based on swarm intelligence and takes its inspiration from biological swarm systems such as ants or bees. A swarm robotic behaviour is fully distributed, scalable and highly fault tolerable. These are easily scalable systems, which is fundamental for the task of detection and monitoring that we want to develop. It should be noted that, among other factors, the emergent properties and the decentralization of these systems make their design difficult and therefore require an analysis and modelling of both the behaviour and the individuals of the swarm.

In this line, we have contributed with a swarm behaviour modelled from the microscopic and macroscopic perspective. We have provided a microscopic model, verifying its convergence and its scalability. Its operation has been verified by performing hundreds of simulations on environments between 1 and  $5km^2$ . In these environments several blooms, obtained from previous aerial studies, have been randomly included. It has been proved that the swarm converges quickly to these blooms even in scenarios where the relationship between the jellyfish bloom and the environment size is relatively small. It has been proved that our model is able to locate several blooms spread in the environment. Furthermore, the scalability of behaviour has been tested for swarms of between 5 and 100 individuals.

We have also provided a macroscopical model, based on Fokker-Planck equations, which predicts the overall evolution of the swarm. This mathematical model calculates the likelihood of an agent to be placed in a position at a given time. This continuous model allows to determine the operation of the swarm without the need to perform expensive simulations of the microscopic behaviour. The macroscopic model confirms the microscopic model predictions in a qualitative way, becoming an essential tool for evaluating the global swarm behaviour.

As for future work, we are working with the preliminary implementation of this behaviour in a real platform. We will use low-cost, custom-developed multicopters to test this behaviour. The low computational needs required by this behaviour make possible to use cheap Arduino control boards to control the UAVs. Along this line, we will use XBEE 2.4GHz modules for local agent communication, providing a range of more than 1km. Our preliminary tests with this platform are being successful with swarms with a small number of individuals (about 5). We are currently adjusting the behaviour parameters for its real implementation in physical robots. We are also designing the visual sensor for jellyfish detection.

## Acknowledgement

This work has been carried out by the project "Intelligent Swarm Systems of Unmanned Aerial Vehicles for Security and Surveillance" TIN2013-40982-R. Ministerio de Economía y Competitividad (Spain). Project co-financed with FEDER funds.

- [1] Levent Bayindir and Erol Şahin. A review of studies in swarm robotics. *Turkish Journal of Electrical Engineering & Computer Sciences*, 15(2):115–147, 2007.
- [2] F Boero, J Bouillon, C Gravili, Maria Pia Miglietta, T Parsons, S Piraino, et al. Gelatinous plankton: irregularities rule the world (sometimes). *MARINE ECOLOGY-PROGRESS SERIES*-, 356:299, 2008.
- [3] Lucas Brotz, William WL Cheung, Kristin Kleisner, Evgeny Pakhomov, and Daniel Pauly. Increasing jellyfish populations: trends in large marine ecosystems. *Hydrobiologia*, 690(1):3–20, 2012.
- [4] Vasco Costa, Miguel Duarte, Tiago Rodrigues, Sancho Moura Oliveira, and Anders Lyhne Christensen. Design and development of an inexpensive aquatic swarm robotics system. In *OCEANS 2016-Shanghai*, pages 1–7. IEEE, 2016.
- [5] Michael N Dawson and William M Hamner. A character-based analysis of the evolution of jellyfish blooms: adaptation and exaptation. In *Jellyfish Blooms: Causes, Consequences, and Recent Advances*, pages 193–215. Springer, 2009.
- [6] Marco Dorigo, Elio Tuci, Roderich Groß, Vito Trianni, Thomas Halva Labella, Thomas Halva, Shervin Nouyan, Christos Ampatzis, Jean louis Deneubourg, Gianluca Baldassarre, Francesco, Stefano Nolfi, Francesco Mondada, Dario Floreano, and Luca Maria Gambardella. The swarmbot project. In *Kunstliche Intelligenz*, pages 31–44. Springer Verlag, 2005.
- [7] Thomas K Doyle, Jonathan DR Houghton, Sarah M Buckley, Graeme C Hays, and John Davenport. The broad-scale distribution of five jellyfish species across a temperate coastal environment. *Hydrobiologia*, 579(1):29–39, 2007.

- [8] Miguel Duarte, Jorge Gomes, Vasco Costa, Sancho Moura Oliveira, and Anders Lyhne Christensen. Hybrid control for a real swarm robotics system in an intruder detection task. In *European Conference on the Applications of Evolutionary Computation*, pages 213–230. Springer, 2016.
- [9] Miguel Duarte, Jorge Gomes, Vasco Costa, Tiago Rodrigues, Fernando Silva, Víctor Lobo, Mario Monteiro Marques, Sancho Moura Oliveira, and Anders Lyhne Christensen. Application of swarm robotics systems to marine environmental monitoring. In *OCEANS 2016-Shanghai*, pages 1–8. IEEE, 2016.
- [10] Frederick Ducatelle, Gianni A Di Caro, Alexander Förster, Michael Bonani, Marco Dorigo, Stéphane Magnenat, Francesco Mondada, Rehan O’Grady, Carlo Pinciroli, Philippe Rétornaz, et al. Cooperative navigation in robotic swarms. *Swarm Intelligence*, 8(1):1–33, 2014.
- [11] Robert Faludi. *Building wireless sensor networks: with ZigBee, XBee, arduino, and processing*. O’reilly, 2010.
- [12] Mark J Gibbons and Anthony J Richardson. Patterns of jellyfish abundance in the north atlantic. *Hydrobiologia*, 616(1):51–65, 2009.
- [13] Heiko Hamann. *Space-Time Continuous Models of Swarm Robotic Systems: Supporting Global-to-Local Programming*, volume 9. Springer, 2010.
- [14] Heiko Hamann and Heinz Wörn. A framework of space–time continuous models for algorithm design in swarm robotics. *Swarm Intelligence*, 2(2-4):209–239, 2008.
- [15] William M Hamner and Michael N Dawson. A review and synthesis on the systematics and evolution of jellyfish blooms: advantageous aggregations and adaptive assemblages. *Hydrobiologia*, 616(1):161–191, 2009.
- [16] YH Peggy Hsieh, Fui-Ming Leong, and Jack Rudloe. Jellyfish as food. *Hydrobiologia*, 451(1-3):11–17, 2001.
- [17] Sangrok Jin, Seokwoo Lee, Jihoon Kim, Jongwon Kim, and TaeWon Seo. Design of hovering underwater robotic platform to capture harmful marine organisms. In *Control, Automation and Systems (ICCAS), 2012 12th International Conference on*, pages 1747–1751. IEEE, 2012.

- [18] D Kim, JU Shin, H Kim, D Lee, SM Lee, and H Myung. Jeros: Jellyfish removal robot system. In *8th International Conference on Humanized System (ICHS), Daejeon, Korea, 2012*.
- [19] Donghoon Kim, Jae-Uk Shin, Hyungjin Kim, Hanguen Kim, Donghwa Lee, Seung-Mok Lee, and Hyun Myung. Development and experimental testing of an autonomous jellyfish detection and removal robot system. *International Journal of Control, Automation and Systems*, 14(1):312–322, 2016.
- [20] Sean Luke, Claudio Cioffi-Revilla, Liviu Panait, and Keith Sullivan. Mason: A new multi-agent simulation toolkit. In *Proceedings of the 2004 SwarmFest Workshop*, volume 8, 2004.
- [21] Ryuma Maeda, Takahiro Endo, and Fumitoshi Matsuno. Decentralized navigation for heterogeneous swarm robots with limited field of view. *IEEE Robotics and Automation Letters*, 2017.
- [22] Claudia E. Mills. *Jellyfish blooms: are populations increasing globally in response to changing ocean conditions?*, pages 55–68. Springer Netherlands, Dordrecht, 2001.
- [23] Thom Nickell, Keith Davidson, Clive J Fox, Peter Miller, and Graeme C Hays. Developing the capacity to monitor the spatial and temporal distributions of jellyfish in western scottish waters. *The Crown Estate*, 2010.
- [24] Department of Ecology of the State of Washington. Surface conditions reports.
- [25] Naoshi Ohta, Masato Sato, Kiminori Ushida, Mami Kokubo, Takayuki Baba, Kayoko Taniguchi, Makoto Urai, Koji Kihira, and Joji Mochida. Jellyfish mucin may have potential disease-modifying effects on osteoarthritis. *BMC biotechnology*, 9(1):98, 2009.
- [26] Jennifer E Purcell. Climate effects on formation of jellyfish and ctenophore blooms: a review. *JMBA-Journal of the Marine Biological Association of the United Kingdom*, 85(3):461–476, 2005.

- [27] Jennifer E Purcell. Extension of methods for jellyfish and ctenophore trophic ecology to large-scale research. In *Jellyfish Blooms: Causes, Consequences, and Recent Advances*, pages 23–50. Springer, 2009.
- [28] Jennifer E Purcell and Mary N Arai. Interactions of pelagic cnidarians and ctenophores with fish: a review. *Hydrobiologia*, 451(1-3):27–44, 2001.
- [29] Jennifer E Purcell, Evelyn D Brown, Kevin DE Stokesbury, Lewis H Haldorson, and Thomas C Shirley. Aggregations of the jellyfish aurelia labiata: abundance, distribution, association with age-0 walleye pollock, and behaviors promoting aggregation in prince william sound, alaska, usa. *Marine Ecology Progress Series*, 195:145–158, 2000.
- [30] Jennifer E Purcell, Shin-ichi Uye, Wen-Tseng Lo, et al. Anthropogenic causes of jellyfish blooms and their direct consequences for humans: a review. *MARINE ECOLOGY-PROGRESS SERIES-*, 350:153, 2007.
- [31] Anthony J Richardson, Andrew Bakun, Graeme C Hays, and Mark J Gibbons. The jellyfish joyride: causes, consequences and management responses to a more gelatinous future. *Trends in ecology & evolution*, 24(6):312–322, 2009.
- [32] Erol Şahin. Swarm robotics: From sources of inspiration to domains of application, 2005.
- [33] Adi Shklarsh, Gil Ariel, Elad Schneidman, and Eshel Ben-Jacob. Smart swarms of bacteria-inspired agents with performance adaptable interactions. *PLoS Comput Biol*, 7(9):e1002177, 09 2011.
- [34] Timothy Stirling and Dario Floreano. Energy efficient swarm deployment for search in unknown environments. In *International Conference on Swarm Intelligence*, pages 562–563. Springer, 2010.
- [35] Timothy Stirling and Dario Floreano. Energy-time efficiency in aerial swarm deployment. In *Distributed Autonomous Robotic Systems*, pages 5–18. Springer, 2013.
- [36] Takuya Sugahara, Masashi Ueno, Yoko Goto, Koichi Akiyama, Satoshi Yamauchi, Ryusuke Shiraishi, and Mikiharu Doi. Immunostimulation

- effect of the jellyfish collagen. In *Animal Cell Technology: Basic & Applied Aspects*, pages 293–299. Springer, 2009.
- [37] Ying Tan and Zhong-yang Zheng. Research advance in swarm robotics. *Defence Technology*, 9(1):18–39, 2013.
- [38] Matthias Vigelius, Bernd Meyer, and Geoffrey Pascoe. Multiscale modelling and analysis of collective decision making in swarm robotics. *PloS one*, 9(11):e111542, 2014.