

## Research Article

# Modelling Oil-Spill Detection with Swarm Drones

**F. Aznar,<sup>1</sup> M. Sempere,<sup>1</sup> M. Pujol,<sup>1</sup> R. Rizo,<sup>1</sup> and M. J. Pujol<sup>2</sup>**

<sup>1</sup> *Department of Computer Science and Artificial Intelligence, University of Alicante, 03090 Alicante, Spain*

<sup>2</sup> *Department of Applied Mathematics, University of Alicante, 03090 Alicante, Spain*

Correspondence should be addressed to F. Aznar; [fidel@dccia.ua.es](mailto:fidel@dccia.ua.es)

Received 30 October 2013; Accepted 17 January 2014; Published 9 March 2014

Academic Editor: Benito Chen-Charpentier

Copyright © 2014 F. Aznar et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Nowadays, swarm robotics research is having a great increase due to the benefits derived from its use, such as robustness, parallelism, and flexibility. Unlike distributed robotic systems, swarm robotics emphasizes a large number of robots, and promotes scalability. Among the multiple applications of such systems we could find are exploring unstructured environments, resource monitoring, or distributed sensing. Two of these applications, monitoring, and perimeter/area detection of a given resource, have several ecological uses. One of them is the detection and monitoring of pollutants to delimit their perimeter and area accurately. Maritime activity has been increasing gradually in recent years. Many ships carry products such as oil that can adversely affect the environment. Such products can produce high levels of pollution in case of being spilled into sea. In this paper we will present a distributed system which monitors, covers, and surrounds a resource by using a swarm of homogeneous low cost drones. These drones only use their local sensory information and do not require any direct communication between them. Taking into account the properties of this kind of oil spills we will present a microscopic model for a swarm of drones, capable of monitoring these spills properly. Furthermore, we will analyse the proper macroscopic operation of the swarm. The analytical and experimental results presented here show the proper evolution of our system.

## 1. Introduction

Maritime activity has been increasing gradually in recent years. For example, around 42000 vessels (excluding the fishing ones) pass throughout the North Sea, carrying products that can adversely affect the environment, such as oil, which can produce high levels of pollution in case of being spilled into sea. Moreover, many pollutants are accidentally spilled from ships during “normal” operations. These spills are probably small but become significant due to the large number of ships.

Dramatic incidences of marine pollution, such as the Prestige oil spill off the Spanish north coast [1–3], have highlighted the potential for human-caused environmental damage. In attempting to mitigate or avoid future damage to valuable natural resources caused by marine pollution, research has been undertaken by the scientific community to study the processes affecting the fate and distribution of marine pollution and especially to model and simulate these processes. Furthermore, active systems, able to detect and

track such kind of spills, are an invaluable tool to help to locate and clean the affected resources [4–6].

Moreover, 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 [7–10]. As it has been shown [10] it is different from other multirobotic systems, because, for example, the swarm robots are relatively simple.

However, the distributed nature of these systems makes it difficult to develop an architecture that correctly models a swarm and coordinates it 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 [11, 12]. These models will be tested to evaluate

the performance of the swarm before its deployment in real UAVs. This is mainly because this kind of systems has emergent properties that make it difficult to predict the overall operation of the swarm when only the local behaviour of the agents is being analysed.

In this paper, we will present a swarm behaviour to be used with robotic drones in order to cover the perimeter or the area of an oil spill. We will use GNOME [13], a very realistic environmental simulation system for marine pollutants. With the data provided by this simulation, using real maritime information and air currents predictions, we will create a virtual oil spill in the Spanish coast. We will show that our behaviour is able to both detect and track the spill for a given period of time. Moreover, we will also show, through the use of the macroscopic model derived by the local rules of the robots, that the swarm will track the oil pollutants. A Fokker-Plank equation will be derived from the microscopic behaviour in order to predict the probability that a drone stays in a given position at time  $t$ .

## 2. Pollutant Dispersion Model

Whether the trajectory model is accurate, adequate, or correct is often questioned. The accuracy and adequacy of a model are associated with the data used for the calculations and modelled physical processes. The modelling of an oil spill is not a simple task, mainly due to the number of factors influencing the trajectories of pollutants: sea currents, winds, and even the gravitational force or surface tension of water.

There are several applications that model and simulate pollutant discharges into the sea. These applications are based on the modelling of the most important forces that interact on an oil slick. In this paper, one of these applications has been used to obtain realistic data of an oil spill off Spanish coast. As it will be seen in subsequent sections, we use both ocean currents and wind data. These data will be used for testing a realistic oil spill using a swarm of flying robots (drones).

More specifically, we have used GNOME, an interactive environmental simulation system designed for the rapid modelling of pollutant trajectories in the marine environment. GNOME simulates oil movement due to winds, currents, tides, and spreading. It was developed by the Hazardous Materials Response Division (HAZMAT) of the National Oceanic and Atmospheric Administration Office of Response and Restoration (NOAA OR&R). HAZMAT uses GNOME during spill events to calculate the spill trajectory and the associated uncertainty in that trajectory. As it has been mentioned in [14], GNOME can be used for predicting how winds, currents, and other processes might move and spread oil on the water, for studying how predicted oil trajectories are affected by inexactness (uncertainty) in current and wind observations and forecasts, and for predicting the chemical and physical changes of the oil spill during the time that it remains on the water surface.

One of the benefits of this tool is the possibility of using external files containing information on tides, currents, and so forth. If this information is added, then GNOME uses it to make the trajectory prediction of the spill (even using

uncertainty during simulation) in the specified region. The output from the model consists of graphics, movies, and data files for postprocessing in a geographic information system (GIS) or NOAA Emergency Response Division's (ERD1) in-house model GNOME Analyst.

The model used by this tool is general and applies to trajectory problems. It is an Eulerian/Lagrangian model that is two-dimensional (2-D) in space, but vertically isolated layered systems can be modelled. Shoreline maps are inputs to the model, so any area can be modelled. The model automatically handles either hemisphere or east or west longitude.

More specifically, random spreading (diffusion) is calculated by a simple random walk with a square unit probability. The random walk is based on the diffusion value,  $D$ , that represents the horizontal turbulent diffusion of the spill in the water. During a spill, the value is calibrated as based on overflight data. In this way diffusion and spreading are treated as stochastic processes. Gravitational and surface tension effects are ignored, as these are only important during the first moments of a spill. Complex representation of subgrid diffusion and spreading effects are ignored.

The main diffusion equation used in GNOME is presented in [13], where  $D_x$  and  $D_y$  are the scalar diffusion coefficients in  $x$  and  $y$  directions and  $C$  is the pollutant concentration:

$$\frac{\partial C}{\partial t} = D_x \cdot \frac{\partial^2 C}{\partial x^2} + D_y \cdot \frac{\partial^2 C}{\partial y^2}. \quad (1)$$

GNOME simulates this diffusion with a random walk with any distribution, with the resulting diffusion coefficient being one-half the variance of the distribution of each step divided by the time step:

$$D_x = \frac{1}{2} \cdot \frac{\sigma^2}{\Delta t}, \quad (2)$$

where  $\sigma^2$  is the variance of the distribution of diffused points and  $\Delta t$  is the time elapsed between time steps.

Evaporation in GNOME is modelled by a simplistic 3-phase evaporation algorithm where the pollutant is treated as a three-component substance with independent half-lives [15]:

$$\begin{aligned} X_{\text{prob}} = & \left( P_1 \cdot \left( 2^{-t_i/H_1} - 2^{(t_{i-1}-2t_i)/H_1} \right) \right. \\ & + P_2 \cdot \left( 2^{-t_i/H_2} - 2^{(t_{i-1}-2t_i)/H_2} \right) \\ & \left. + P_3 \cdot \left( 2^{-t_i/H_3} - 2^{(t_{i-1}-2t_i)/H_3} \right) \right) \\ & \times \left( P_1 \cdot 2^{-t_i/H_1} + P_2 \cdot 2^{-t_i/H_2} + P_3 \cdot 2^{-t_i/H_3} \right)^{-1}, \end{aligned} \quad (3)$$

where  $t_i$  and  $t_{i-1}$  are the time elapsed (age; in hours) at time step  $i$  and the previous time step  $t - 1$ , respectively, since the LEs release.  $H_1$ ,  $H_2$ , and  $H_3$  are the half-lives of each constituent (in hours) for the pollutant, and  $P_1$ ,  $P_2$ , and  $P_3$  are the percentages of each constituent (as decimals) for the pollutant.

Spilled substances are modelled as point masses (up to  $10,000^4$ ) called LEs (Lagrangian elements) or “spots” (from “spill dots”). Spills can be initialized as one-time or continuous releases and as point or line sources or evenly spaced in a grid on the map for diagnostic purposes.

Once GNOME executes a simulation, the solution is produced in the form of a trajectory. GNOME provides two solutions to an oil spill scenario, the best estimate trajectory and the uncertainty trajectory. The best estimate solution shows the model result with all of the input data assumed to be correct. The uncertainty solution allows the model to predict other possible trajectories that are less likely to occur, but which may have higher associated risks. In this paper we will use the uncertainty solution of pollutant particles (represented by its LEs) for generating a continuous pollutant map. More details of this mathematical model could be obtained in [13].

### 3. Swarm Design

In this section we will analyse the features that must have a robotic swarm to detect and follow an oil spill in effective way. We will also propose a specific microscopic behaviour for this task. As mentioned above, the modelling of this kind of pollutants is a complex task because of the interaction of many factors. However, it is easy to extract a set of characteristics that a system needs to locate this kind of spills.

On the one hand, the swarm must be able to detect and follow pollutants that can change and move in time mainly by the action of advection and diffusion. Depending on the application, two situations can be found: the origin of the spill is known or the swarm initially makes the detection process (in this work it is assumed that detection is one of the tasks to be performed). On the other hand, the appearance of several polluting slicks is very probable due, among other factors, to the transport and deposit of sediments on the coast while the oil slick disperses and evaporates.

The behaviour of the swarm must be highly robust and tolerant to failures and should be totally distributed. Therefore, all agents must have the same importance when performing the behaviour, without the existence of any agent more prominent than another. Finally, the behaviour should be highly scalable for two reasons: robust issues and performance of the behaviour, since as a first step it may be beneficial to use a reduced number of agents until they find any evidence of a particular discharge.

Although in this paper behaviours will be analysed in a simulated way, the features of agents are directly based on flying robots (our intention is to use these algorithms in drones). These drones will have a video camera that will use a vision algorithm to determine the existence of any oil slick in its visual field (as presented in [4, 16]). For security reasons, we assume that drones will fly at different heights, so that collisions in a position  $(x, y)$  are avoided. We also assume that due to flight altitude (about 500 m above sea level) the differences in the visual field caused by different flying altitudes are negligible. All tests, unless otherwise specified,

will be carried out by a swarm of 200 drones. It is a medium-sized swarm, appropriate for simulation.

Our main goal, once the behaviour is designed, is to determine the ability of the swarm to locate, converge, and follow an oil spill. Therefore, a macroscopic model to predict the global behaviour of the swarm and to verify its performance will be specified. More specifically, we propose a homogeneous behaviour, executed by all agents, consisting of three states. Initially, drones look for any trace of the spill in the environment. Once the spill is detected, the drone will head to it. Finally, the drone will try to stay inside it (to cover the slick) or in its perimeter (to mark it).

Figure 1 shows the finite state machine (FSM) that governs the behaviour of each robot. The initial state is *Wander*, since we initially assume that the position of the spill is unknown. The transition from *Wander* state is performed when the agent’s visual sensor detects an oil slick. In this case, the new state will be *Resource*. This state (*Resource*) has two possible transitions: the spill is not detected (transition *d*) so the system returns to *Wander* state or the amount of oil detected is  $>80\%$  of the image (transition *b*), so, the system switches to *InResource* state. The agent remains in this state if the detected oil percentage is  $>80\%$ ; otherwise, the system returns to *Resource* state.

Next, each behaviour specified in the FSM will be described in more detail.

At the beginning of the execution agents start at *Wander* state. At that time they have no idea where to find an oil spill. Therefore, all directions are equally valid and they will move randomly. The velocity of the drone at time  $t$  is defined as

$$\mathbf{v}_w(t) = \mathbf{v}_w(t-1) + \mathbf{rand} \cdot \mu_1, \quad (4)$$

where  $\mathbf{rand}$  is a random uniform vector defined within the interval of the maximum and minimum velocity of the drone, and  $\mu_1$  is the coefficient of variability on the current velocity. With values close to 1, the robot will move in a totally random way.

If a drone detects the resource, it heads to the resource to position itself at the perimeter or over it, depending on the desired behaviour. The velocity of the agent is defined by three factors:

$$\mathbf{v} = \alpha_1 \times \mathbf{v}_c + \alpha_2 \times \mathbf{v}_o + \alpha_3 \times \mathbf{v}_r, \quad (5)$$

where  $\alpha_1 + \alpha_2 + \alpha_3 = 1$  and these values define the intensity of each factor.

More specifically,  $\mathbf{v}_c$  specifies the direction of the robot. This direction is determined by the area with the larger intensity resource average:

$$\mathbf{v}_c = \gamma(S) \times \left\| \sum_{s \in S} ((\mathbf{pos}(s) - \mathbf{pos}(\text{rob})) \cdot s) \right\|, \quad (6)$$

where  $S$  is sensor readings that detect the resource at a given time,  $\mathbf{pos}(s)$  is the position vector of a reading, and  $\mathbf{pos}(\text{rob})$  is the position of the robot. We assume that the intensity of the resource is in the range  $[0, 1]$ , where 0 is the complete lack of resource and 1 is the unambiguous detection of it.

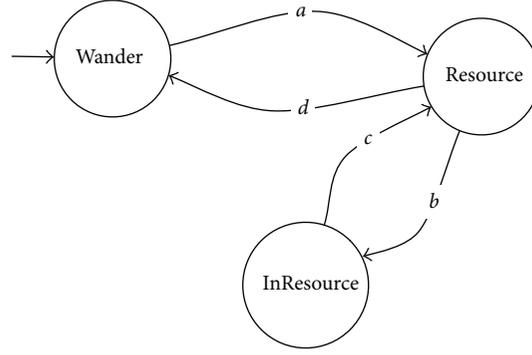


FIGURE 1: Finite state machine (FSM) that governs the operation of each agent. The initial state is Wander. The transition  $a$  is triggered when the agent's visual sensor detects an oil slick. Transition  $b$  occurs when the amount of oil detected is  $>80\%$  of the image. Transition  $c$  is triggered when the amount of oil is  $\leq 80\%$  of the image. Finally, transition  $d$  is triggered if no oil is detected.

$\gamma$  determines the direction of the velocity vector, depending on whether the robot is outside or inside de resource. The aim of this variable is, therefore, to keep the robot on the perimeter of the resource:

$$\gamma(S) = \begin{cases} 1 & \frac{\sum_{s \in S} s}{|S|} \leq \eta \\ -1 & \frac{\sum_{s \in S} s}{|S|} > \eta, \end{cases} \quad (7)$$

where  $\eta$  is a threshold that determines, from the quantity of resource detected (0 means no resource and 1 maximum quantity of resource), if the agent is located on the perimeter. If the main objective of the system is that drones cover the pollute slick, then  $\gamma(S)$  will be defined as 1 for any set of readings  $S$  at a given time.

$\mathbf{v}_o$  specifies an avoidance vector with respect to all robots detected at a given moment:

$$\mathbf{v}_o = \left\| \sum_{i=1}^{|R|} (\mathbf{pos}(r_i) - \mathbf{pos}(\text{rob})) \right\|, \quad (8)$$

where  $R$  is the set of detected robots,  $\mathbf{pos}(r_i)$  is the position of the detected robot  $i$ , and  $\mathbf{pos}(\text{rob})$  is the position of the current robot.

Moreover, 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:  $\mathbf{v}_r(t) = \mathbf{v}_r(t-1) + \mathbf{rand} \cdot \mu_2$ , where  $\mu_2$  is the coefficient of variability on the velocity.

Finally, when drones are located inside the spill and, therefore, the borders of the resource are not detected, we assume that drones will develop a random wander behaviour until they find water again because there is no other information about which direction is better to follow:

$$\mathbf{v}_s(t) = \mathbf{v}_s(t-1) + \mathbf{rand} \cdot \mu_3, \quad (9)$$

where  $\mu_3$  is the coefficient of variability on the velocity.

In this section we have defined a microscopic behaviour for the detection and tracking of an oil spill. Each agent of the swarm executes this behaviour locally. It is a simple behaviour that can be run in low processing capacity systems. This behaviour does not require any communication between agents or a global positioning system.

## 4. Experimentation

We will now present a set of experiments in order to test the operation of the proposed microscopic behaviour. As we will see, these experiments use simulations of oil spills based on real data.

We have simulated an oil spill in the coastal area of "Sant Antoni de Portmany," on Ibiza island, using an area of approximately  $350 \text{ km}^2$ . This area has been chosen because of the various currents that affect it and its proximity to the peninsula. Real weather data have been used for both sea and air currents, considering the start date of the spill on April 10, 2013, at 0:00 h. We use these real data to run a simulation for seven days from the start of the spill, simulating a ship loaded with 100000 barrels of oil. Figure 7 shows the map of the main area affected and the initial place of impact. In order to work with data generated by GNOME we use the average of the points of discharge generated by the application (More specifically,  $M(t)' = G(M(t) \oplus s_{\text{shape}})$  where  $M(t)$  is the map obtained with GNOME at time  $t$ ,  $G$  is a Gaussian filter with  $\sigma = 10|S|$ , and  $\oplus$  is a morphological binary operator) (*splots* in GNOME), making uniform the continuity between nearby contaminated areas and simplifying the simulation. This allows us to reduce the number of *splots* needed in the simulation:

Robots = 200

$\mu_1 = 0.3$

$\mu_2 = 0.3$

$\mu_3 = 0.3$

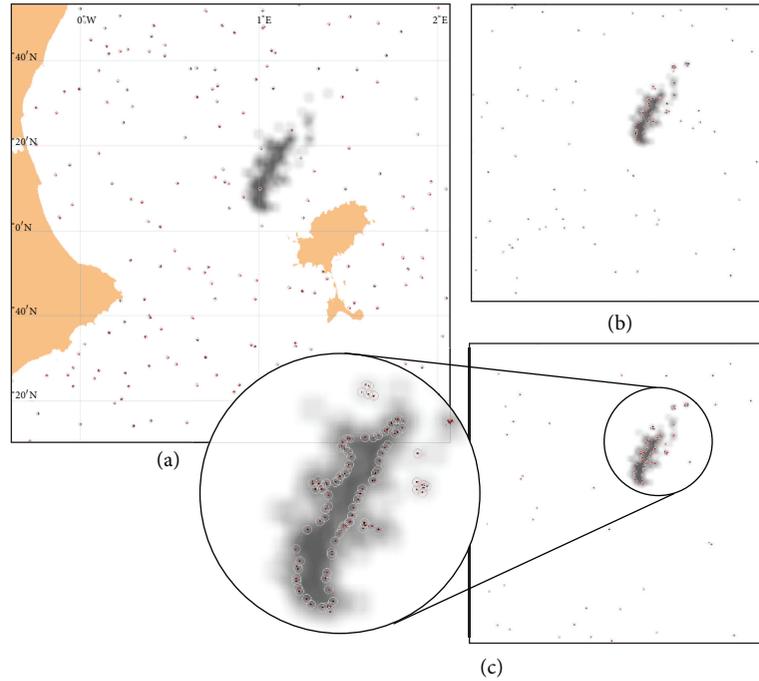


FIGURE 2: Distribution of agents with respect to time (in seconds) for the task of detecting the perimeter of an oil spill: (a) geographical location of the spill and initial position of agents, (b) distribution of agents at time  $t = 15000$  s, and (c) distribution of agents at time  $t = 30000$  s.

$$\begin{aligned}
 \eta &= 0.8 \\
 \alpha_1 &= 0.6 \\
 \alpha_2 &= 0.2 \\
 \alpha_3 &= 0.2.
 \end{aligned}
 \tag{10}$$

The previous equation shows default parameters used in simulations for the microscopic model. These parameters have been adjusted taking into account the microscopic behaviour experiments.

The simulation of the microscopic model has been developed using the software MASON [17]. We use a swarm of 200 agents randomly distributed by the environment that moves uniformly at 60 km/h. The simulation uses small size drones ( $< 3 \text{ m}^2$ ) that are able to visually detect an area of  $1 \text{ km}^2$ .

Using these parameters, several tests will be performed to verify the correct operation of our model at local level. Initially, first tests check the convergence of the swarm for a single static slick. Using the same example,  $\gamma$  will be modified to cover the slick instead of marking its perimeter. Next tests will verify the convergence of the swarm choosing an instant with several active slicks. Finally, we will check the tracking of the movement of the spill.

**4.1. Spill Detection.** First tests have been developed to detect and monitor a static slick. On the one hand, the swarm will mark the perimeter of the slick. Figure 2 shows the initial distribution of agents with their geographical location (a) and the position of agents at time  $t = 15000$  s and  $t = 30000$  s

((b) and (c)). At time  $t = 30000$  s the swarm completely surrounds the oil slick. Figure 3 presents the percentage of agents over an oil slick for ten independent simulations. As it can be appreciated, this percentage increases progressively.

Moreover, as it has been discussed in the definition of the microscopic model of the swarm, when  $\gamma(S) = 1$  the swarm will cover the slick instead of marking its perimeter. Figure 4 shows the results of the swarm for this case, where 10 independent simulations have been carried out.

The experiments in Figure 2 show that the operation of the swarm is correct: 50% of agents are able to detect the spill in less than 4 hours since the beginning of the execution, marking correctly the perimeter of the spill. However, in a real environment the appearance of several slicks is common. This case will be analysed in the next section.

**4.2. Several Spills.** When an oil spill occurs, the oil may spread across kilometres generating several oil slicks. For this reason, it is very important to verify the correct operation of the swarm with different slicks that can spread out over a wide area. In the simulation, at early hours, several areas containing large proportions of pollutant are produced. For example, underwater currents can move oil slick close to the coast, while the ship continues spilling oil into the sea.

In order to check the operation of the microscopic model in these cases, we have chosen an instant of the simulation where the slick has a complex structure. Figure 5 shows the evolution of the distribution of robots. As it can be seen in this figure, even in this complex case and without previous data that indicates the spill trend, our model is able to locate and mark the perimeter of the slicks. More specifically,

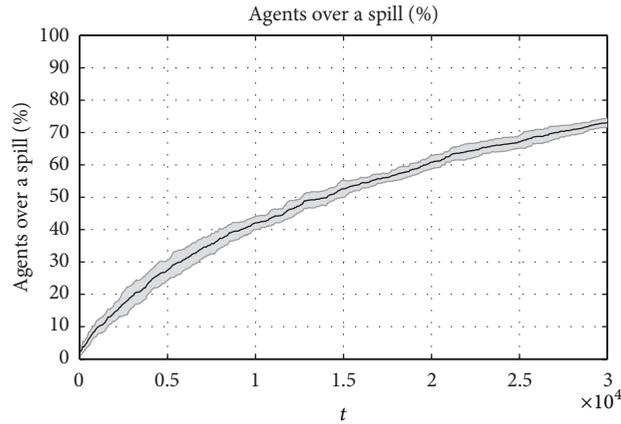


FIGURE 3: Percentage of agents on an oil slick with respect to time (in seconds). Ten different experiments have been carried out, showing the average and variance.

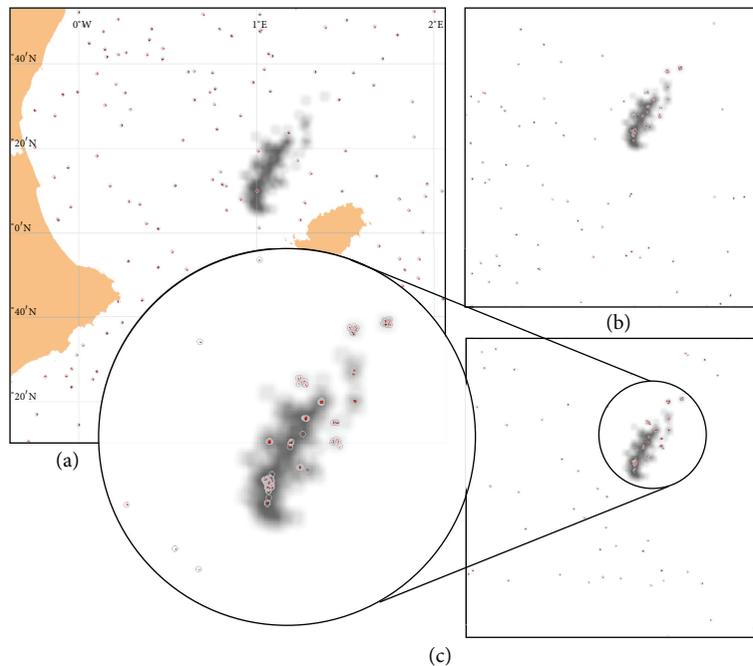


FIGURE 4: Distribution of agents for the task of covering the spill ( $\gamma(S) = 1$ ): (a) geographical location of the slick and initial position of agents, (b) position of agents at time  $t = 15000$  s, (c) position of agents at time  $t = 30000$  s.

Figure 6 presents the percentage of agents that are located on a polluted area. In this figure it can be seen that the number of agents increases with time, therefore distributing uniformly on the slick.

**4.3. Spill Movement.** Figure 5 tests have shown that the swarm is able to detect and mark the perimeter/area with complex slicks. However, in a real spill, finding the location of the spill is as important as effective monitoring is. In this section we will analyse the behaviour of the swarm taking into account that the spill will spread as time progresses.

A simulation of 168 hours of the evolution of the spill has been performed, starting April 10, 2013, at 0:00 h. In

order to simplify the simulation, we capture an image of the state of the spill every 4.5 hours. Although the spill will experience changes in this period, the swarm can deal with these changes. Figure 7 shows the origin point of the spill and some captures of its evolution.

As in previous experiments, the swarm is distributed randomly by the environment. In this case, in the simulation process, the location of the resource obtained by GNOME is updated every 4.5 hours. Figure 8 shows the evolution of the swarm with respect to the state of the oil spill.

Figure 9 shows the percentage of drones that are on top of a resource at a given time. In order to produce this graph, 10 different simulations have been carried out. The graph shows

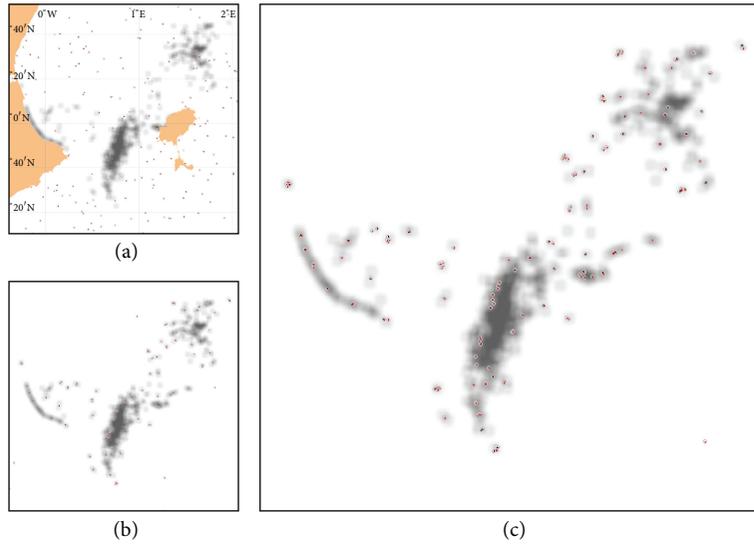


FIGURE 5: Distribution of agents for the task of detecting the perimeter of an oil spill on a map with several slicks: (a) geographical location of the spill and initial position of agents, (b) distribution of agents at time  $t = 15000$  s, and (c) distribution of agents at time  $t = 30000$  s.

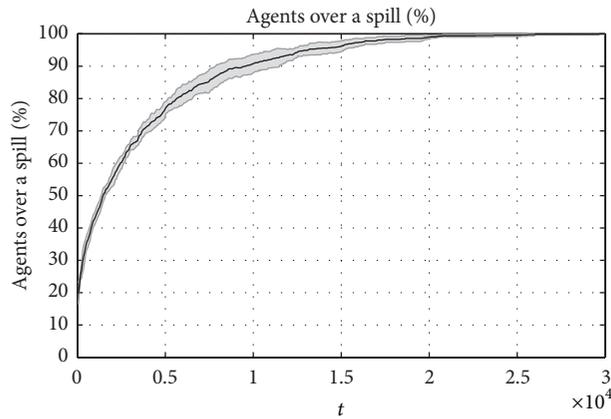


FIGURE 6: Percentage of agents on an oil slick with respect to time (in seconds) for a complex slick. Ten different experiments have been carried out, showing the average and variance.

the average and variance of these simulations. In this graph a series of steps produced artificially by the simulator (because of the sudden change from one state to the next) can be seen.

Nonetheless, the evolution of the swarm is correct. The swarm is able to track the spill and mark each of the slicks.

### 5. Macroscopic Model

Once the microscopic behaviour has been described, it is interesting to see the global behaviour of the swarm. There are several techniques used to analyse this behaviour [18] such as the use of recurrence equations, generated from a microscopic behaviour defined by a PFSM or the definition of differential equations. However, most of these methods allow only the analysis of the evolution of the state transitions globally.

In this work, we consider the framework proposed in [19] in order to obtain the probability distribution of the swarm

position for any time  $t$ . This will enable us to predict, in great detail, the behaviour of the overall system. As described by [19], once the microscopic behaviour has been defined, the global behaviour of the system can be calculated using the Fokker-Planck equation:

$$\frac{\partial \rho(\mathbf{r}, t)}{\partial t} = -\nabla \cdot (\mathbf{A}(\mathbf{r}, t) \rho(\mathbf{r}, t)) + \frac{1}{2} Q \nabla^2 (B^2(\mathbf{r}, t) \rho(\mathbf{r}, t)), \tag{11}$$

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

This equation provides a method to statistically model a swarm of robots based on modelling techniques of multi-particle systems from the field of quantum physics. From a Langevin equation that represents the behaviour of a single particle, the Fokker-Planck equation is derived for all the system.

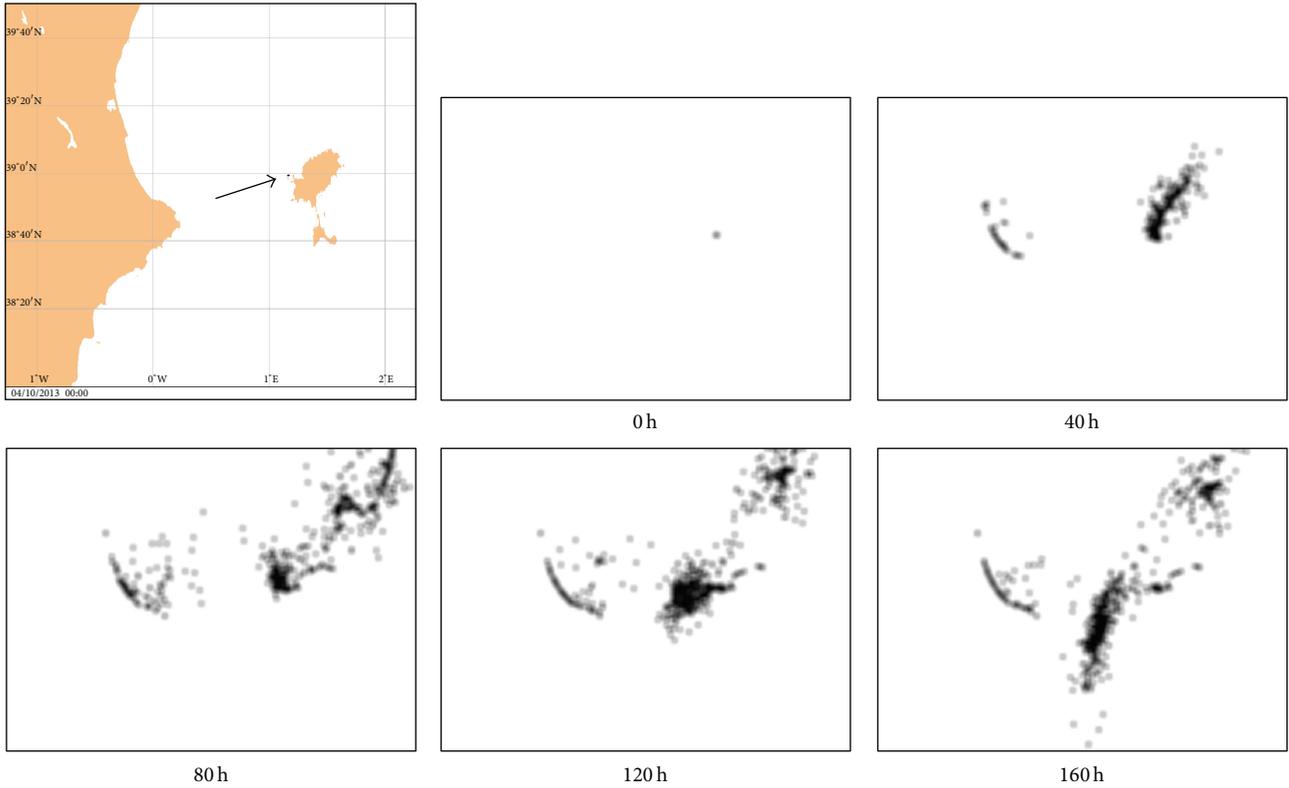


FIGURE 7: Origin and temporal evolution of the spill. Several snapshots that show the position of the spill at different instants, measured in hours from the beginning of the spill, are shown.

As we have already seen in [20], the Fokker-Planck equation implements the necessary abstraction of microscopic details as described above and treats rapidly changing parameters such 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.

Initially, the swarm designer must specify the functions  $\mathbf{A}$  and  $B$  of (11), in accordance with the desired microscopic behaviour. 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 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.

The most difficult task is the definition of these functions. This design is not unique and it requires finding two functions that correctly describe the behaviour of the swarm. These functions will be analytically presented and their performance will be tested in following sections.

$\mathbf{A}$  function determines the displacement of the swarm.  $\mathbf{A}$  depends primarily on a vector representing the directional information. A potential field  $P$  is commonly used to define it. In our case, we need to establish a function that takes

into account the following things based on the proposed microscopic model: the random motion states of the robot, the probability that a movement of an agent fails in its execution (e.g., due to a collision), and the potential field where the robots move. Although it is possible to model a probability distribution for each state, as our microscopic model has no interaction between agents (except purely physical, as collisions) and the behaviour of states is relatively simple, the macroscopic behaviour of the swarm can be comprised in a single distribution:

$$\mathbf{A}(\mathbf{r}, t) = (1 - b_r) \cdot (1 - \rho(\mathbf{r}, t))^{\mu_4} \cdot (\Gamma(P(\mathbf{r}, t)) \cdot K(\nabla P(\mathbf{r}, t))), \quad (12)$$

where  $P$  is directly related to the sensor readings in point  $\mathbf{r}$  at time  $t$  and  $\mu_5$  is a normalization term:

$$P(\mathbf{r}, t) = \mu_5 \cdot s(\mathbf{r}, t). \quad (13)$$

In (12), a function  $\mathbf{A}$  that takes into account previous aspects is proposed. When the density of agents increases, the probability of collision also increases and, therefore, this situation reduces the rate at which robots are directed by vector specified by  $\mathbf{A}$ . More specifically, consider the following.

$b_r$  is the probability that comprises the influences of all states that develop a random or pseudorandom state.

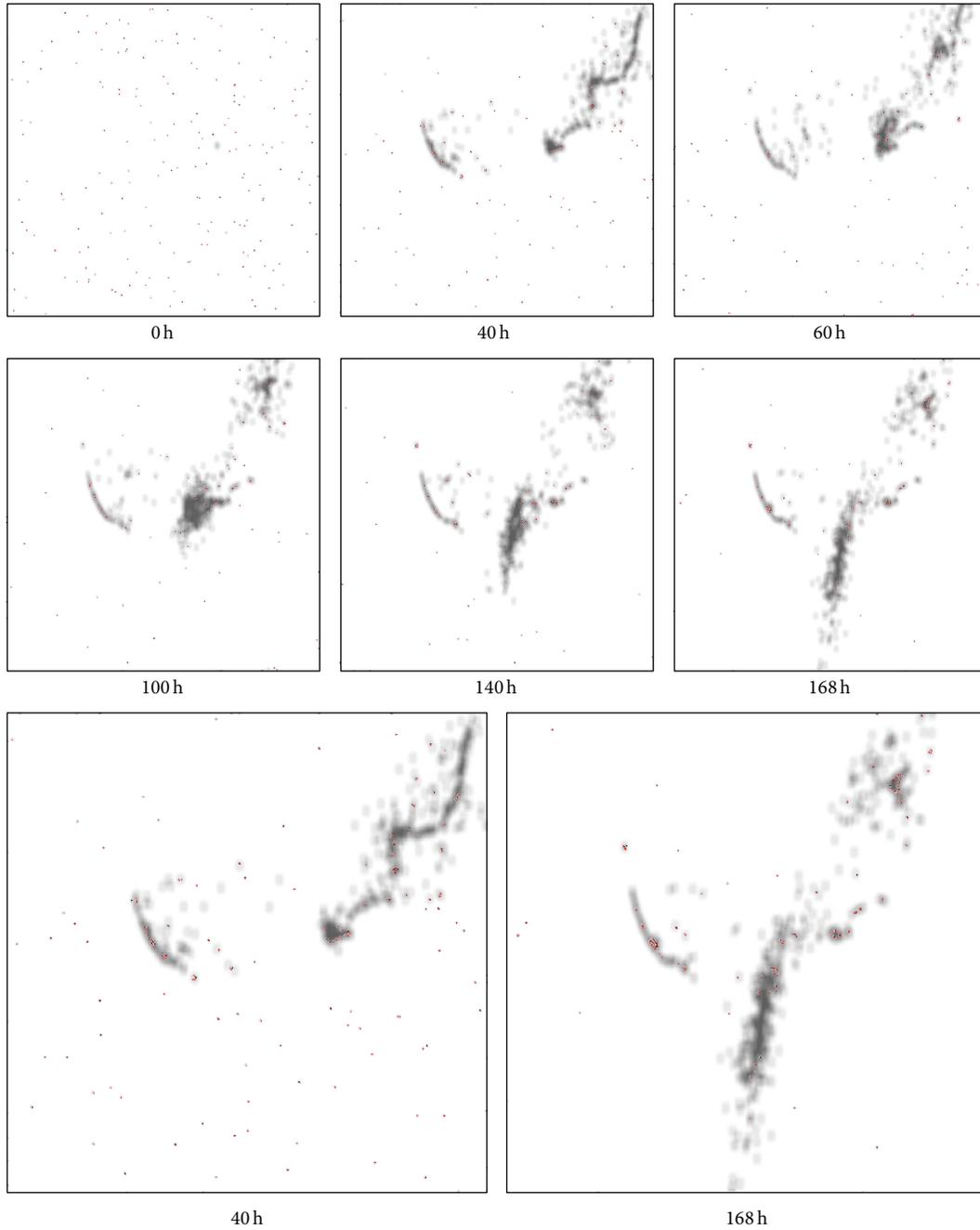


FIGURE 8: Evolution of the swarm with respect to the oil slick. Multiple snapshots at different instants in time (measured in hours) are shown.

$P(\mathbf{r}, t)$  as commented before, is defined as the probability of encountering a robot at position  $\mathbf{r}$  at time  $t$ .

$\Gamma$  is a function applied to the potential field that produces, for each position of  $P$ , the direction to be taken by drones, as a single agent would do using the function  $\gamma$ . In our case, we use a “sliding-neighbourhood” filter as commented, for example, in [21] to perform the same calculation on  $P$  as on the microscopic model, changing the sign of the displacement when a ratio greater than 80% of pollutant is detected.

$K$  is a convolution operator. Being  $G$  a square gaussian kernel of size  $d$ , generated with  $\sigma = 4 \cdot |S|$ , and assuming two-dimensional coordinates so that  $\mathbf{r} = (i, j)$ , the sum of the components of this kernel is  $\sum_{i=1}^d \sum_{j=1}^d G(i, j)$ .  $F$  is defined as this sum if it is different from 0; otherwise, it will be defined as 1. In this way,

$$K_{i,j} = \frac{1}{F} \cdot \sum_{i=1}^d \sum_{j=1}^d G(i, j) \cdot \nabla P((i, j), t). \quad (14)$$

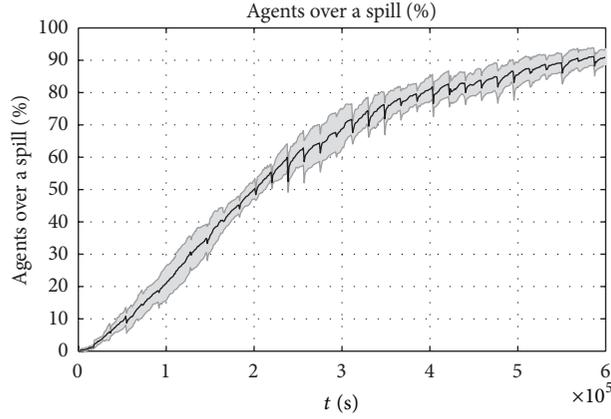


FIGURE 9: Percentage of agents over an oil slick with respect to time (in seconds). Ten different experiments have been carried out, showing the average and variance.

Function  $B$  describes the nondeterministic motion and, therefore, it takes into account the random motion of agents. Two forces, which must be considered, take part in the microscopic behaviour. On the one hand, some influences derived from agents that are on *Wander* and *InResource* states. These states have a random motion, depending on the intensity of parameters  $\mu_1, \mu_3$ . On the other hand, the behaviour itself causes that the environment has areas with a higher density of agents. In these areas the probability of collision can be increased depending on the density of agents at a given time:

$$B(\mathbf{r}, t) = b_r \cdot \rho(\mathbf{r}, t)^{\mu_4}. \quad (15)$$

Thus, in (15) two terms can be observed:  $b_r$  comprises the influences of all states that develop a random or pseudorandom state, as previously commented, and  $\rho(\mathbf{r}, t)^{\mu_4}$  is a term that defines the connection between the density of robots and their probability of collision.

**5.1. Spill Detection.** The operation of the microscopic behaviour has been analysed in the previous section. It has been shown that the swarm is able to locate and mark the perimeter of an oil spill with relatively simple rules. Several tests have been developed to verify the validity of the presented model in various cases.

In addition to the tests above, it is possible to establish, for a given discharge, the areas of the map with the highest probability of containing a robot, independently of the number of agents (on condition that you use enough of them), by using the macroscopic model. This provides a more accurate visualization of the behaviour for large swarms without being limited by the number of agents to be simulated.

This section, using the previously presented macroscopic definition, presents how the swarm is able to locate and mark, in a given time, an oil spill:

$$\begin{aligned} \text{Robots} &= 200 \\ \mu_4 &= 5 \\ \mu_5 &= 0.95 \\ b_r &= 0.5. \end{aligned} \quad (16)$$

The previous equation shows default parameters used in simulations for the macroscopic model. These parameters have been adjusted taking into account the microscopic behaviour experiments. We will present the analysis of the macroscopic behaviour of the swarm for two time instants  $t = (140 \text{ h}, 168 \text{ h})$ .

The simulation process is simple, once the functions  $A$  and  $B$  of Fokker-Planck equation have been defined. Initially, an origin point for the swarm must be established. Although in microscopic simulations it is possible to establish different origin points (as setting randomly the position of agents), Fokker-Planck equation requires a single point. Nonetheless, several simulations for different origin points have been performed, observing that the results for large values of  $t$  are very similar: only slight variations can be seen if the origin point is located just on an oil slick. In this case, the probability of this slick in relation to the rest of the spill can be higher.

The same origin point has been used for all tests, discretizing the simulation area in  $100 \times 100$  units. The origin point is established in Fokker-Planck equation as  $(x, y) = (40, 25)$ .

Once the Fokker-Planck equation has been defined, the probability distribution that a certain agent is in a position of the environment at a given time can be obtained in an iterative way. This distribution can be calculated iterating for each instant of time  $t$  all the positions of the map, updating for each position the new probability described by the equation.

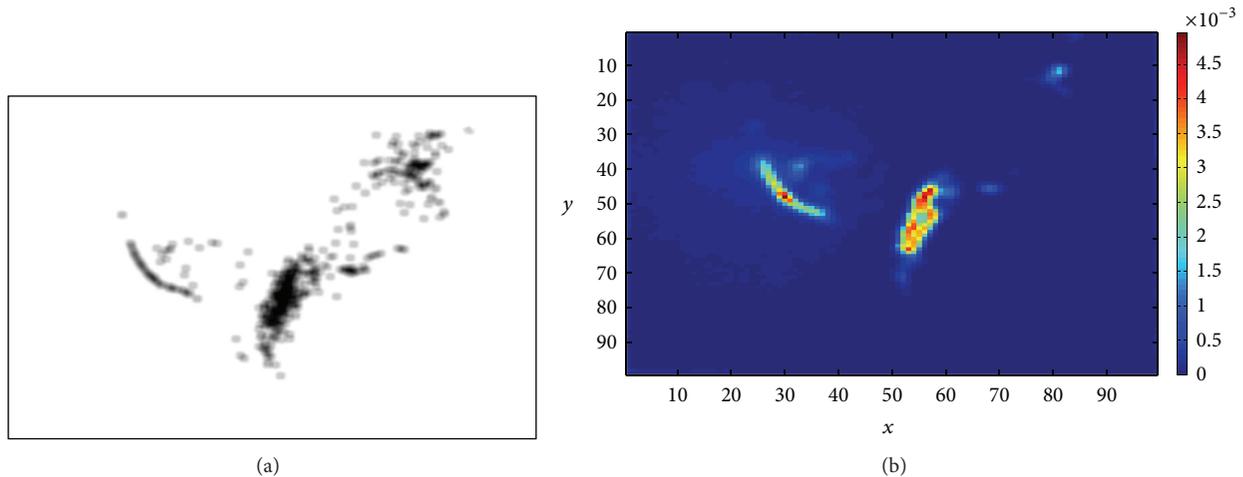


FIGURE 10: (a) Map ( $M$ ) generated from GNOME data for  $t = 140$  h. (b) Probability that a robot is in a certain position of the space at  $t = 140$  h. Sampled using the macroscopic model of the swarm with the Fokker-Planck equation.

The macroscopic state of the swarm is presented in Figure 10 for  $t = 140$  h. A clear convergence in the perimeter of the spill can be observed. A three-dimensional representation of the probability distribution that an agent is in a certain position of the environment at  $t = 168$  h is presented in Figure 11. As it can be observed, the macroscopic model correctly predicts the behaviour presented at the microscopic level of the swarm.

**5.2. Model Comparison.** In the previous section the behaviour of the macroscopic model and how this model predicts the overall behaviour of the swarm have been presented. Now, we will compare predictions of microscopic model and macroscopic model for a specific case.

Figure 12(a) shows a probability distribution obtained from the microscopic model. In order to do this, we have discretized the simulation environment and, with the same parameters used in previous sections, we have developed a simulation with 200 agents at  $t = 140$  h. Along the simulation, we save the number of agents that pass on a discrete cell (in order to compare it with the macroscopic model that we have also discretized in  $100 \times 100$  units) and then we calculate the probability that an agent is in this cell at a given time.

In the same way, Figure 12(b) shows the probability distribution that predicts the macroscopic model. In this figure it can be seen that the area of interest is covered in both models. There are minor differences within the models, due to the deficiencies of microscopic simulation that, among other things, depends on the number of agents used in the simulation.

Nevertheless, we can compare both approaches by multiplying both distributions. In this way only high probabilities remain and, therefore, it is easier to observe if the areas of the spill are correctly identified in both models. Bearing this in mind, we have slightly rectified the microscopic distribution. When we use a limited number of agents in the simulation, we discover, in some cases, that high probabilities hide important information in the distribution. In order to avoid this loss of

information, we have used the square root of the distribution in order to compare both models.

The product of macroscopic distribution and the square root of microscopic distribution are presented in Figure 12(c). This figure shows how the most important parts of the spill are detected with both distributions, predicting the macroscopic model the same perimeter areas detected by the microscopic model.

## 6. Discussion

This paper describes a microscopic model that is able to locate and mark the perimeter of an oil spill. The microscopic behaviour presented does not require direct communication between agents. This limitation can cause that the convergence of the swarm on the spill takes more time, depending on the number of drones and the size of the spill. However, this behaviour is versatile and easy to implement and develop, even in areas without GPS coverage. It is important to highlight that a swarm system, which requires direct communication between agents, is a limited system because of the maximum range of each agent and the saturation of the radiofrequency space if the system needs a large number of agents.

Moreover, we have demonstrated that the process of locating and marking the perimeter of the spill without communication is robust and efficient. We have shown that the swarm system is able to completely delimit the spill if the number of agents is sufficient. In order to achieve this task, an agent must be able to detect drones that are nearby. There are several ways, as, for example, using a camera or transmitting the GPS position.

We propose the use of signal intensity (at a given frequency) for obstacle avoidance tasks. This strategy may show some problems (we have implemented it by using a reactive behaviour); however, it has several advantages. Many countries require that drones broadcast a continuous signal indicating their position. Europe uses 433 MHz frequency

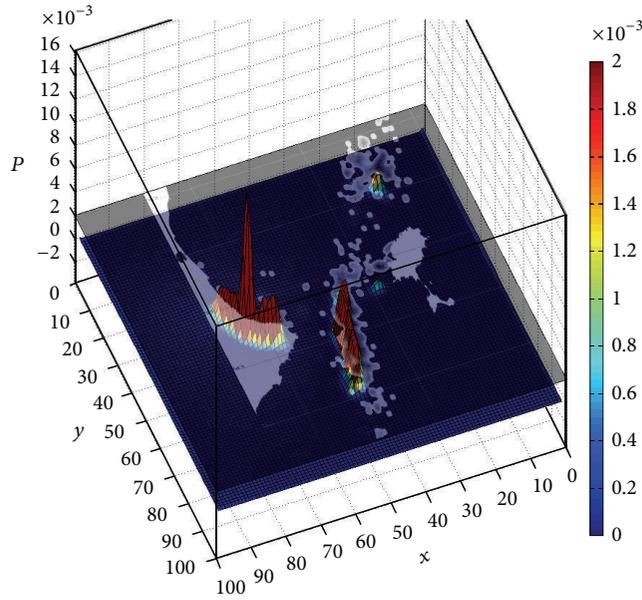


FIGURE 11: Three-dimensional representation of the probability that a robot is in a particular position at  $t = 168$ h. Sampled using the macroscopic model of the swarm with the Fokker-Planck equation. A map for the probability 0.002 where the environment and the state of the spill at  $t = 168$  h are represented is added.

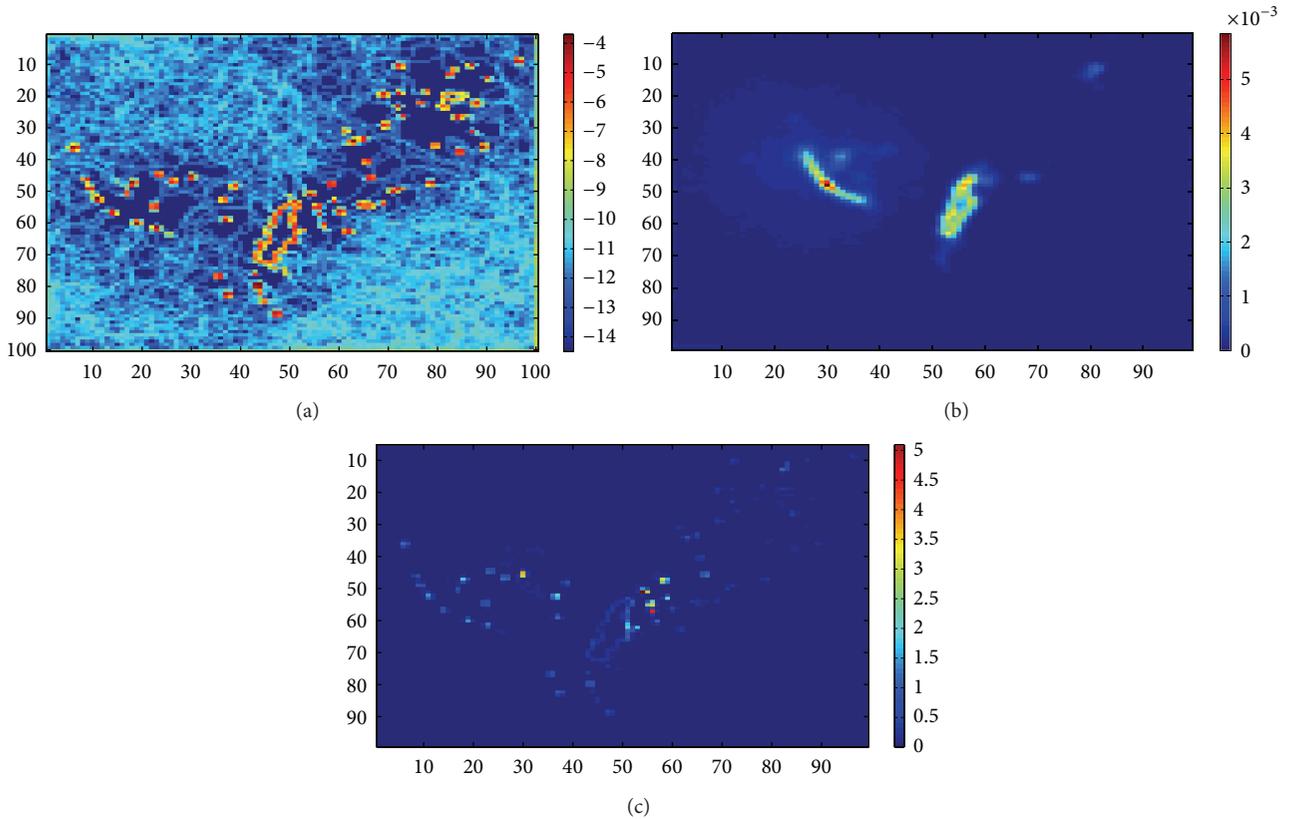


FIGURE 12: Comparison of the microscopic prediction and macroscopic model for the map  $M$  at  $t = 140$  h. (a)  $(\log(P_{\text{micro}}))$  Logarithm of the probability distribution obtained simulating 200 agents for 30,000 seconds. Logarithmic distribution is used to highlight the states with low probability. (b)  $(P_{\text{macro}})$  Probability distribution obtained with Fokker-Planck equation. (c) Product of both distributions, decreasing the importance of high values in the microscopic simulation (specifically  $P_{\text{macro}} \times P_{\text{micro}}^{1/2}$ ).

for this purpose. The intensity of the signal in a particular area can be detected by using the same infrastructure. If the intensity of the signal grows with the movement of the agent, this agent must change its direction. We emphasize that, as a swarm approach, this is not a communication between agents but simply a *beacon* that we can use, if necessary, to know the position of drones.

The proposed macroscopic model demonstrates that the tendency of the swarm, for a sufficient number of drones, is the same that can be perceived in the microscopic model. The connection of both models has been tested for a complex spill, generated with GNOME. These experiments have shown that the fundamental characteristics of the behaviour (detection and monitoring) are reflected in both models. It is advisable not to forget the differences between the two models.

The microscopic model defines the individual behaviour. Because of this it is easy to understand at local level. However, this model does not define the behaviour of all the swarm. In order to analyse the global behaviour, a set of tests can be defined for a large number of agents. However, these tests can be expensive and difficult and are not exempt from problems.

The macroscopic model defines the global behaviour of the swarm. It allows us to verify the emergent behaviour from the interaction between all agents that run the microscopic model. The macroscopic model demonstrates the tendency of the swarm for a large number of agents. The analysis of this model is complex, because of the use of differential equations that, for example, force us to choose a single point to start the simulation. Even so, this model has remarkable advantages [20], for example, continuous analysis for any point of the environment, time of the probability that an agent is located in a given location, and simulation time negligible compared to microscopic model.

We are currently working on the implementation of this system in a real swarm of drones. Our immediate future research focuses on this real swarm, since it allows us to adjust the algorithms for a real system.

We are already in the testing phase for small swarms (5 drones), obtaining satisfactory results in our preliminary tests. We are using low-cost, custom-developed hexacopters to test this behaviour. The low computational needs required by this behaviour make it possible to use cheap Arduino control boards to control the UAVs. We have developed several tests for our swarm behaviors using artificial color landmarks. While flying our agents can localize (by wandering) a landmark and follow it (if moving) in real time.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgment

This work has been supported by the Spanish Ministerio de Ciencia e Innovación, Project TIN2009-10581.

## References

- [1] J. I. Medina-Bellver, P. Marín, A. Delgado et al., "Evidence for in situ crude oil biodegradation after the prestige oil spill," *Environmental Microbiology*, vol. 7, no. 6, pp. 773–779, 2005.
- [2] S. Díez, J. Sabaté, M. Viñas, J. M. Bayona, A. M. Solanas, and J. Albaigés, "The prestige oil spill. I. Biodegradation of a heavy fuel oil under simulated conditions," *Environmental Toxicology and Chemistry*, vol. 24, no. 9, pp. 2203–2217, 2005.
- [3] J. J. González, L. Viñas, M. A. Franco et al., "Spatial and temporal distribution of dissolved/dispersed aromatic hydrocarbons in seawater in the area affected by the prestige oil spill," *Marine Pollution Bulletin*, vol. 53, no. 5–7, pp. 250–259, 2006.
- [4] F. Nirchio, M. Sorgente, A. Giancaspro et al., "Automatic detection of oil spills from sar images," *International Journal of Remote Sensing*, vol. 26, no. 6, pp. 1157–1174, 2005.
- [5] M. N. Jha, J. Levy, and Y. Gao, "Advances in remote sensing for oil spill disaster management: state-of-the-art sensors technology for oil spill surveillance," *Sensors*, vol. 8, no. 1, pp. 236–255, 2008.
- [6] C. Brekke and A. H. S. Solberg, "Oil spill detection by satellite remote sensing," *Remote Sensing of Environment*, vol. 95, no. 1, pp. 1–13, 2005.
- [7] M. Dorigo, E. Tuci, R. Groß et al., "The swarm-bot project," in *Kunstliche Intelligenz*, pp. 31–44, Springer, Berlin, Germany, 2005.
- [8] M. Dorigo and E. Şahin, "Autonomous Robots: guest editorial," *Autonomous Robots*, vol. 17, no. 2–3, pp. 111–113, 2004.
- [9] G. Beni, "From swarm intelligence to swarm robotics," in *Swarm Robotics*, pp. 1–9, Springer, Berlin, Germany, 2005.
- [10] E. Şahin, "Swarm robotics: from sources of inspiration to domains of application," in *Swarm Robotics*, vol. 3342 of *Lecture Notes in Computer Science*, pp. 10–20, 2005.
- [11] O. Soysal and E. Şahin, "A macroscopic model for self-organized aggregation in swarm robotic systems," in *Swarm Robotics*, pp. 27–42, Springer, Berlin, Germany, 2007.
- [12] W. Liu, A. F. T. Winfield, and J. Sa, "Modelling swarm robotic systems: a case study in collective foraging," in *Proceedings of the Towards Autonomous Robotic Systems (TAROS '07)*, pp. 25–32, 2007.
- [13] United States. National Ocean Service. Office of Response and Restoration, *General NOAA Operational Modeling Environment (Gnome) Technical Documentation*, U.S. Dept. of Commerce, National Oceanic and Atmospheric Administration, National Ocean Service, Office of Response & Restoration, 2012.
- [14] K. Wojtaszek, *Application of transport model for building contingency maps of oil spills on the North Sea [M.S. thesis]*, Delft University of Technology, 2003, A-1, [http://risk2.ewi.tudelft.nl/research-and-publications/doc\\_download/70-katarzynawojtaszekthesispdf](http://risk2.ewi.tudelft.nl/research-and-publications/doc_download/70-katarzynawojtaszekthesispdf).
- [15] P. D. Boehm, D. L. Flest, D. Mackay, and S. Paterson, "Physical-chemical weathering of petroleum hydrocarbons from the ixtoc I blowout: chemical measurements and a weathering model," *Environmental Science and Technology*, vol. 16, no. 8, pp. 498–505, 1982.
- [16] K. Karantzalos and D. Argialas, "Automatic detection and tracking of oil spills in SAR imagery with level set segmentation," *International Journal of Remote Sensing*, vol. 29, no. 21, pp. 6281–6296, 2008.
- [17] S. Luke, C. Cioffi-Revilla, L. Panait, and K. Sullivan, "Mason: a new multi-agent simulation toolkit," in *Proceedings of the Swarm Fest Workshop*, vol. 8, 2004.

- [18] L. Bayindir and E. Şahin, "A review of studies in swarm robotics," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 15, no. 2, pp. 115–147, 2007.
- [19] H. Hamann and H. Wörn, "A framework of space-time continuous models for algorithm design in swarm robotics," *Swarm Intelligence*, vol. 2, no. 2–4, pp. 209–239, 2008.
- [20] H. Hamann, *Space-Time Continuous Models of Swarm Robotic Systems: Supporting Global-to-Local Programming*, vol. 9, Springer, Berlin, Germany, 2010.
- [21] J. L. Semmlow, *Biosignal and Medical Image Processing*, Signal Processing and Communications, Taylor & Francis, 2004.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

