

Multi-level Spatial Modeling for Stochastic Distributed Robotic Systems

Amanda Prorok[†], Nikolaus Corell[‡] and Alcherio Martinoli[†]

[†]Distributed Intelligent Systems
and Algorithms Laboratory
Ecole Polytechnique Fédérale
Lausanne, Switzerland
amanda.prorok@epfl.ch,
alcherio.martinoli@epfl.ch

[‡]Department of Computer Science
University of Colorado at Boulder
Boulder, CO, USA
nikolaus.correll@colorado.edu

Abstract

We propose a combined spatial and non-spatial probabilistic modeling methodology motivated by an inspection task performed by a group of miniature robots. Our models explicitly consider spatiality and yield accurate predictions on system performance. An agent’s spatial distribution over time is modeled by the Fokker-Planck diffusion model and complements current non-spatial microscopic and macroscopic models that model the discrete state distribution of a distributed robotic system. We validate our models on a microscopic level based on submicroscopic, embodied robot simulations as well as real robot experiments. Subsequently, using the validated microscopic models as our template, abstraction is raised to the level of macroscopic difference equations. We discuss the dependency of the modeling performance on the distance from the robot origin (drop-off location) and temporal convergence of the team distribution. Also, using an asymmetric setup, we show the necessity of spatial modeling methodologies for environments where the robotic platform underlies drift phenomena.

Keywords: distributed robotics, probabilistic modeling, swarm intelligence, distributed intelligent systems

1 Introduction

The resulting collective intelligence of groups of simple agents represents a powerful tool to solve tasks of high complexity by distributed problem solving (Bonabeau et al. 1999; Camazine et al. 2001). An important element in the process of the studies on distributed intelligent systems, especially those relying on self-organization as their main coordination mechanism, is the creation of models which aim to lead to a better understanding of the dynamics of the collective intelligence, as well as to accurate predictions of the distributed systems’

performances. Despite the deterministic control of the individual embedded systems, interaction with the environment is only statistically predictable (over numerous runs), and simplicity on the individual level invokes the use of basic, noisy sensors and actuators. Randomness being at the core of a self-organized system (Bonabeau et al. 1999), these factors together motivate the choice of probabilistic models, expressing a major contrast to the deterministic models of traditional robotics approaches to date (Choset 2001).

Recent work in swarm-robotic systems' modeling relies on the assumption that area coverage by the swarm agents is uniform, and non-spatial metrics are applied (Agassounon et al. 2004; Correll and Martinoli 2006c; Lerman and Galstyan 2002; Martinoli et al. 2004). A non-spatial metric is a performance metric that is independent of the actual spatial distribution of the robot swarm and solely a function of the discrete behavioral states of the robot and the environment. But often the constraints of the task do not fulfill this assumption, and spatiality must be taken into account. This is the case, for example, when the multiple agents are set in structured environments. Obstacles with regular or less regular shapes constrain or influence the agents' movements and affect the manner in which areas are explored. Correll and Martinoli (2006c) have shown that a structured environment may cause a constant drift in a given direction, and area coverage is not uniform (see Figure 1). Here, drift is caused by the geometry of the environment, but it can also be caused by a spatially biased robot controller such as a correlated random walk, or, more naturally, by wind or currents. The latter are of particular interest when studying large swarms of unmanned aerial vehicles (UAVs) (Chaimowicz and Kumar 2004), or miniature, underwater robots (Vasilescu et al. 2005). In (Correll and Martinoli 2006c) a swarm of robots is concerned with the inspection of a jet turbine engine model. This case study showed to be intrinsically different than earlier swarm robotic case studies. Foraging (Lerman and Galstyan 2002), aggregation (Agassounon et al. 2004), or distributed manipulation (Martinoli et al. 2004) all consider relatively long time spans and environments that allow the robots to distribute uniformly. In the jet turbine inspection model, however, short time spans invoke dependencies from the robot drop-off location, and environmental geometries hinder uniform distribution.

Our motivation is to develop new models which are able to take spatiality into consideration. Similar to our approach, Hamann and Wörn (2008) present a time- and space-continuous model based on the Fokker-Planck equation. Their method, however, is limited to the analysis of the distribution itself and does not consider robots being in different behavioral modes. Furthermore, Winfield et al. (2008) present a macroscopic model which describes the overall connectivity structure of a swarm of mobile robots. Similar to our approach, their model is incrementally constructed from microscopic models and low-level information, including an original, heuristic control algorithm. In contrast to our work, however, their work focuses on the estimation of state transition probabilities through a robot-centric approach, which symmetrically aggregates spatial distributions tailored on the underlying robot controller, operating in unbounded space. In our framework, we consider bounded space and combine spatially

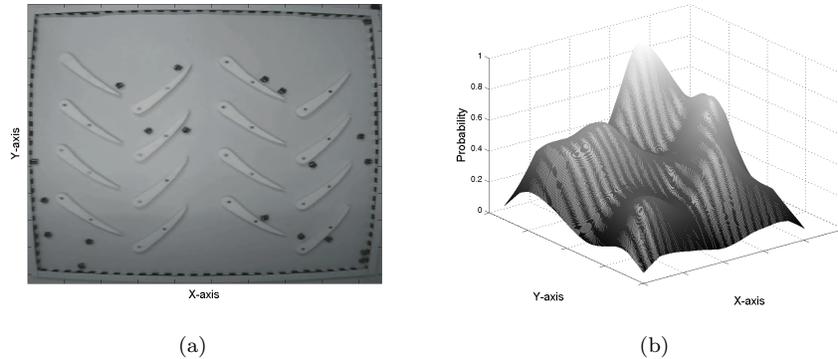


Figure 1: (a) Experimental Setup from (Correll and Martinoli 2006c). (b) Average distribution of the robot swarm in the arena over 2hrs of continuous inspection of the model jet turbine blades. The swarm’s motion is strongly biased by the structure of the environment.

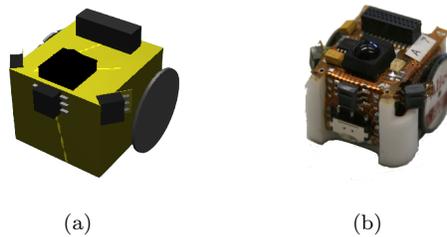


Figure 2: (a) A model of the Alice robot in the submicroscopic, embodied robot simulator Webots. (b) The Alice robot.

asymmetric and time-dependent effects with the dynamics of the whole swarm-robotic system that is modeled by a separate set of difference equations. Also, we show how discretization of time and space yields a common interface between models at different levels of abstraction, enabling accurate predictions on swarm performance over a larger problem space.

1.1 Outline and Contribution of this Paper

In this work, we consider a generic case study: the inspection of cell areas in a bounded, square arena. The simplicity of this case study allows us to focus on notions of spatiality and non-uniform coverage, at the same time allowing us to compare a non-spatial model (Correll and Martinoli 2006a) with new spatial models. First, we create a baseline by elaborating a non-spatial modeling methodology. We then introduce a basic microscopic spatial model (Section 3.1),

and subsequently increase the abstraction level by introducing a diffusion model (Section 3.2). The model predictions are evaluated by considering agreement amongst each other as well as with submicroscopic, embodied robot (see Figure 2(a)) simulations run with Webots (Michel 2004) and real robot (see Figure 2(b)) results obtained with the Alice robot (Caprari and Siegwart 2005). By developing the proposed model layers incrementally, we show how controller and environmental parameters can systematically be abstracted. This multi-level modeling approach is reviewed in Section 1.2. Finally, in Section 4, this paper proposes a spatial macroscopic modeling framework able to combine non-spatial macroscopic models (which capture the behavioral states of robots) with diffusion models (which capture the spatial distributions as a function of time).

Ultimately, since we consider scenarios including the inspection of real objects, which leads to asymmetric diffusion phenomena such as observed in Correll and Martinoli (2006c) (see Figure 1), we validate our spatial macroscopic modeling framework on an experiment with an underlying constant drift.

1.2 Multi-Level Modeling

Components of a distributed robotic system are physical artifacts endowed with sensing, computation, actuation, and communication means. Analyzing a collective of such individuals requires meaningful abstraction of the individuals and their interactions to capture specific metrics of interest. At the microscopic level we consider a single agent and its behavioral states. For instance, Camazine et al. (2001) discuss the example of a swarm of ants: a foraging ant’s microscopic states include exploring space in search for food, collecting food, and homing. Such a model considers the time spent in each state, such as the time it takes an ant to collect a portion of food. Interactions among different agents can then be studied by probabilistic simulation of a swarm where each member is governed by microscopic rules. Predictions are made, for example, on the number of ants currently homing, the number of ants in search of food, or the amount of food already collected. Such probabilistic finite state machines have been implemented by Martinoli (1999); Lerman and Galstyan (2002).

When using robots in the process of model validation, multi-level modeling enables us to integrate the engineered (by programming the robot controller) robot states into a macroscopic model of the the robot system by incrementally raising the level of abstraction. The parameters introduced into the macroscopic model are essentially defined at the individual level (for example, the time an ant takes to collect food). Analyses can be made of the macroscopic model, enabling optimization of the parameters at the individual, agent level, in order to yield better system results on the collective level. The case study presented in this paper considers a system designed for inspection tasks. The metric we consider in purpose of evaluating our framework is the number of times a cell is ‘inspected’, or visited. However, further metrics could also be considered such as the inspection-completion time, the percentage of completed inspection, or the percentage of area covered.

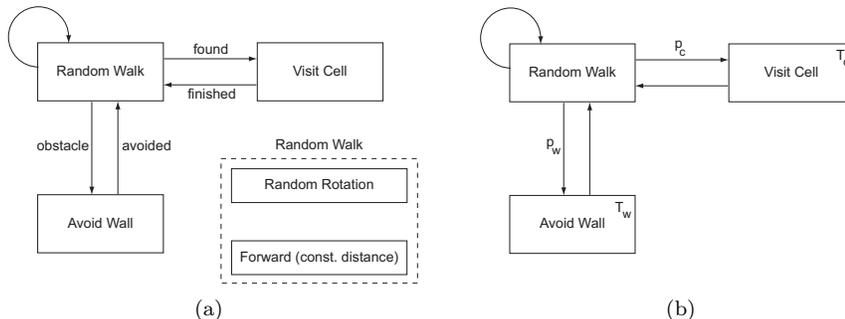


Figure 3: (a) A finite state machine (FSM) modeling robot behavior. The random walk state is modeled by two separate states. (b) A probabilistic FSM

1.3 Probabilistic Modeling: Non-spatial Models

Our modeling framework (Martinoli 1999) is based on the complementary use of models at the submicroscopic, microscopic, and macroscopic level. Whereas spatial effects were mainly investigated at the submicroscopic level using embodied realistic simulation, typical microscopic and macroscopic models required the assumption of uniform coverage, and did not capture information dependent on both space and time.

We define a non-spatial microscopic model, based on the robot controller - a single robot's behavior is modeled by a deterministic finite state machine (FSM). We introduce our current modeling methodology (Martinoli et al. 2004) using a simple inspection case study. The robot moves randomly within the arena. The arena is empty and contains M_0 cells that need to be visited (inspected) by a robot. At every time step it might either collide with the wall or move over the area of a defined cell, emulating its inspection. The corresponding schema is shown in Figure 3. At a higher abstraction level, the FSM provides the basis for the formulation of a non-spatial macroscopic model, in our case formalized as a set of difference equations. Each equation in the set describes the number of robots in a state at a given time, or in the case of a single robot, the probability for the robot to be in this state. The system dynamics can be described by exploiting the fact that the number of agents is constant. N_x is the number of robots in a certain state x — N_{rw} being the number in random walk, N_c the number visiting a cell, N_w the number avoiding a wall with M_0 the number of cells to inspect. Further, $M_c(k)$ is the number of times a cell c has been visited at iteration k , (or time $k \cdot T_{it}$, T_{it} being the sampling interval of the resulting time-discrete system). $\Delta_s(k)$ represents the number of agents entering a state s at iteration k and T_s is the average number of iterations an agent spends in state s . Thus we have:

$$N_{rw}(k+1) = N_{rw}(k) + \Delta_w(k - T_w) + \Delta_c(k - T_c) - \Delta_c(k) - \Delta_w(k) \quad (1)$$

$$N_w(k+1) = N_w(k) + \Delta_w(k) - \Delta_w(k - T_w) \quad (2)$$

$$N_c(k+1) = N_0 - N_{rw}(k+1) - N_w(k+1) \quad (3)$$

$$M_c(k+1) = M_c(k) + \Delta_c(k) \quad (4)$$

with

$$\Delta_w(k) = p_w \cdot N_{rw}(k) \quad (5)$$

$$\Delta_c(k) = p_c \cdot M_0 \cdot N_{rw}(k) \quad (6)$$

where p_w is the probability to encounter a wall and p_c the probability to encounter a cell. Equation 1, for example, states that the number of agents in search (random walk) at iteration $k + 1$ equals the number at iteration k , plus the number finishing wall-avoidance and cell-inspection, minus the number entering the same two states.

The above system can be solved for a given time-step k , assuming certain initial conditions. We have: $N_{rw}(0) = N_0, N_w(0) = 0, N_c(0) = 0$, thus all robots are initially in searching state, and $M_c(0) = 0$. Parameters to be defined according to the realistic setup are T_c, T_w, p_w and p_c . They are dependent on the arena geometry and on the sensor characteristics and robot controller. The parameters are calculated using heuristics as proposed by Martinoli (1999) and Correll and Martinoli (2006a), but can also be found by means of system identification methods (Correll and Martinoli 2006b).

We note that the system described in equations 1-6 is identical with the system used for modeling the turbine inspection scenario involving real robots (Correll and Martinoli 2006c). In this paper, however, we consider non-embodied inspection cells, in order to facilitate generalization of the proposed models.

2 Random Walk for Modeling Robot Movement in a Swarm

In this section, we briefly examine the notion of random walk in the context of robot controllers. As it forms an important element of our experiments conducted, we show by a simple experiment why our random walk model is appropriate for modeling an individual robot's movement.

Generally known, random walk is a formalization of the intuitive idea of taking successive steps, each in a random direction. A random walk can be simply constructed, according to the following rules

1. *There is a starting point.*
2. *The distance d_{step} from one point in the path to the next is a constant.*
3. *The direction from one point in the path to the next is chosen at random, and no direction is more probable than another.*

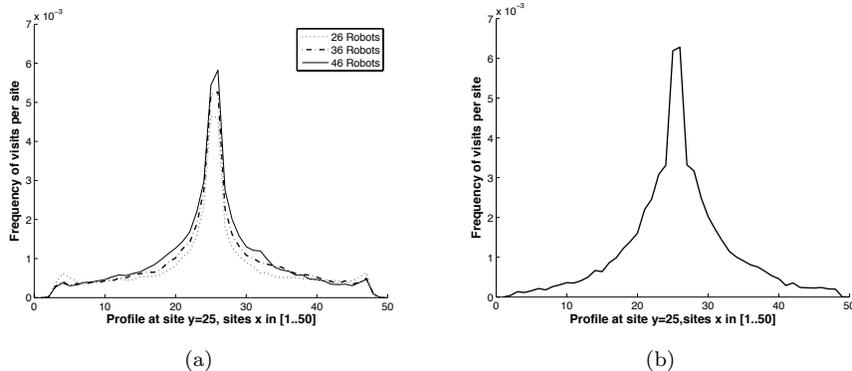


Figure 4: The plots show the normalized number of visits for all x , at a slice through the center of the area. (a) One robot is tracked in varying robot group sizes (26, 36, 46 robots), all moving through the arena in Braitenberg mode. Individual robots’ motion is biased by collisions with team mates and walls. (b) A single robot moves in a random walk.

In nature, there are many different manifestations of random walk (Berg 1983). Brownian motion is the most well known occurrence: a particle is knocked about by the molecules of a liquid or gas in the process of diffusing (Rudnick and Gaspari 2004). In mathematical terms, Brownian motion is the scaling limit of a random walk. As the step size of the random walk decreases, $d_{step} \rightarrow \epsilon$, and the number of steps is increased comparatively, the random walk converges to Brownian motion. Thus, a n -dimensional system which continuously undergoes a series of small, random changes may be modeled by an n -dimensional random walk.

We observe that robots are seldom controlled by an explicit random walk, but rather by reactive controllers (Arkin 2000; Braitenberg 1984), and due to sensor noise and other unpredictable parameters, exhibit randomized behaviors. Nevertheless, the explicit random walk model is practical, as it has precise formulations and ways of incorporating its dynamics.

We would like to show that the random walk model is suitable for our purposes. We design a set of two experiments, conducted using the embodied robot simulator Webots, a spatial, carefully calibrated submicroscopic model. We track a single robot performing an explicit random walk with $d_{step} = 2\text{cm}$ in an empty environment. Second, we track a single robot within a group of several other robots, with group size $G \in \{26, 36, 46\}$. In this scenario, all robots are controlled by a Braitenberg algorithm (Braitenberg 1984) for collision avoidance and move straight when no obstacle is detected. We note that whereas different obstacle avoidance controllers may lead to drastically different behaviors, they will nevertheless lead to a uniform distribution of the robots in the environment as long as they implement an uncorrelated random walk.

The robots run for a fixed time span, always starting at the center of a

bounded arena. Ultimately, spatial data is tessellated into 50×50 sites of a 2D histogram. Figure 4 shows slices of the corresponding histograms with the frequency a site was visited by the robot: In Figure 4(a) we perform 6000 runs for the time span of 1 minute, and for 4(b) we perform 2000 runs for the time span of 10 minutes.

A Braitenberg robot avoiding collisions in the swarm 4(a) displays the same spatio-temporal patterns as the single random-walking robot 4(b). Although far from representing distributed robotics controllers in general, this simple obstacle avoidance algorithm nonetheless exhibits the typical stochastic artifacts of non-deterministic robotics. Further, as time reaches sufficiently large values and tends towards infinity, the distribution in Figure 4 reaches the uniform distribution which—together with the fact that robots are initially uniformly distributed—leads to the assumption of Section 1.3. Based on these observations, we assume that our random walk model is a valid approximation of robot motion in simple, unsophisticated swarm-robotic tasks, and thus creates a baseline for our following theories.

3 Spatial Probabilistic Modeling

Given the assumption of spatial uniformity, as elaborated above in Section 1.3, our non-spatial microscopic and macroscopic modeling methodologies together enable us to accurately capture the dynamics of a swarm-robotic system. Here, we augment our set of probabilistic models with two complementary approaches that allow us to explicitly include spatio-temporal effects, *i*) a spatial microscopic model and *ii*) a diffusion model. Their ability of accurately capturing these effects is validated against real robot experiments as well as sub-microscopic embodied robot simulations. Later in this paper, we extend these models to include the effects of asymmetric environments under the influence of drift.

3.1 Spatial Microscopic Model

Spatial microscopic and Monte Carlo modeling methods applied on course-grained lattices have already been employed extensively in materials science (Chatterjee 2007) and catalytic chemistry (Makeev 2002; Gelten 1998). Here, we make an intuitive approach to spatial modeling by considering a Markov chain on a lattice of L^2 sites, ultimately representing a tessellation of space in a bounded area. We then consider that an agent moves at the speed of 1 site per iteration. Figure 5 (a) shows the eight possible transitions for the robots on the lattice. In the case of a random walk, all eight transition vectors are equally weighted, thus movements are random. Further, the probability to stay in the current site is zero. We measure the number of times each site is visited. The agent’s origin is (x_0, y_0) with $x_0, y_0 \in [0, L]$, and it iterates position transitions k_{max} times. The field is bounded by a ‘wall’, and T_w defines the number of iterations the agent is delayed on encountering these boundaries.

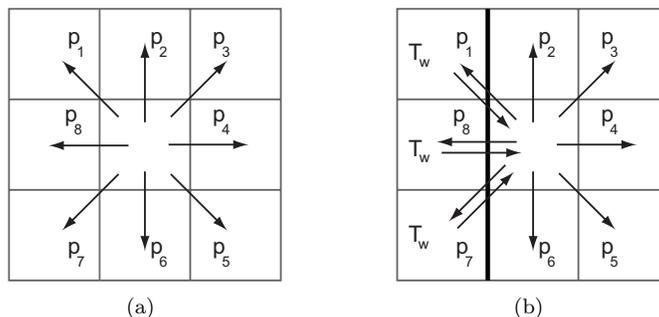


Figure 5: (a) An eight-cell neighborhood and eight transition probabilities. (b) If a transition crosses the boundary the agent is delayed by T_w .

3.2 Diffusion Model

Ferraro and Zaninetti (2001) propose analytical formulas which derive the mean number of times a site (on a lattice) has been visited by a random walker in a given time span. The derivations are based on particle diffusion theory and enable the inclusion of drift, which can lead to the modeling of asymmetric random walks. In the following, the key points of our proposed adaptation are elaborated.

3.2.1 Diffusion Model without Drift

Similarly as in the spatial microscopic model introduced earlier, the random walk takes place on a lattice. Let $P_k(s)$ be the probability that after k steps the walker is at the site s , after starting from the origin. Then, at iteration n we have $M_n(s) = \sum_{k=0}^n P_k(s)$, where $M_n(s)$ is the number of visits made on site s after n time steps. The standard, combinatorial method of solving $P_k(s)$ makes it infeasible to calculate for even moderately large values of k , and so an alternative solution for $P_k(s)$ is needed. The approximation chosen by Ferraro et al. is the solution of the Fokker-Planck equation. We thus use the continuous variable t in place of k , and assuming the continuity of x and y , $P_k(s)$ can be replaced with the solution $P(x, y; t)$ of the partial differential equation

$$\frac{\partial P}{\partial t} = \frac{1}{2}\sigma_x^2 \frac{\partial^2 P}{\partial x^2} + \frac{1}{2}\sigma_y^2 \frac{\partial^2 P}{\partial y^2} \quad (7)$$

with σ_x^2 and σ_y^2 the diffusion coefficients in x and y respectively. In order to obtain a solution for $P(x, y; t)$, we define initial and boundary conditions. We are interested in modeling a random walk on a lattice with reflecting boundaries. We have:

$$0 = \frac{\partial P(x=0, y; t)}{\partial x} = \frac{\partial P(x=L, y; t)}{\partial x} \quad (8)$$

$$0 = \frac{\partial P(x, y=0; t)}{\partial y} = \frac{\partial P(x, y=L; t)}{\partial y} \quad (9)$$

which represents the reflections at x or y equal 0 or L . And for an origin at (x_0, y_0) we have the initial condition

$$P(x, y; 0) = \delta(x - x_0, y - y_0) \quad (10)$$

where $\delta(\cdot)$ is the Dirac function and models the certainty of the agent's origin (drop-off location).

The integral $M(x, y; t) = \int_0^t P(x, y; \tau) d\tau$ of the solution $P(x, y; t)$ yields $M(x, y; t)$, which as in $M_n(s)$ above is the mean number of times position (x, y) was visited by the random walker. For further details of our derivation (which differs from the one of Ferraro et al.) see Appendix A.

3.2.2 Diffusion Model with Drift

We consider the possibility of having non-random motion in space (e.g., resulting from a correlated random walk, or due to an asymmetric environment) and reformulate the diffusion model from Section 3.2.1 to take into account the effects of a constant drift. A two-dimensional drift is defined by vector $[u_x \ u_y]$. Following the Fokker-Planck equation (see 14), Eq. 7 becomes

$$\frac{\partial P}{\partial t} = \frac{1}{2}\sigma_x^2 \frac{\partial^2 P}{\partial x^2} + \frac{1}{2}\sigma_y^2 \frac{\partial^2 P}{\partial y^2} - u_x \frac{\partial P}{\partial x} - u_y \frac{\partial P}{\partial y} \quad (11)$$

As shown in Appendix A, an analytical solution to this equation is non-tractable. Several numerical methods have been proposed and can be applied to solve this type of problem. The Green's function method is an important approach to the solution of boundary value problems, and has been applied to the solution of the diffusion equation (Barton 1989). Further methods have been developed which approximate the solutions of partial differential equations of parabolic type (Douglas 1961). For the current context, we have chosen the latter approach and elaborate the details in Appendix A.

3.3 Calibration of Models

In order to compare our model results among each other as well as with real-robot experiments, we need to define a comparison time scale. We introduce the quantity $T_{it}[\frac{sec}{iteration}]$, time per iteration.

We consider the following elements of the real system: the size of the (bounded) arena and the robot's speed. In our models we tessellate the field into L^2 sites, defining a certain granularity. Applying this same granularity to our realistically modeled arena, we showed that (Martinoli 1999):

$$\frac{A_{det}}{v_{alice} \cdot w_{det}} \geq T_{it} \quad (12)$$

where, for a specific scenario, A_{det} is the detection area of the smallest object present, w_{det} is the detection width of the robot, and v_{alice} is the mean robot speed (see also (Correll and Martinoli 2006a) for further details). This formula

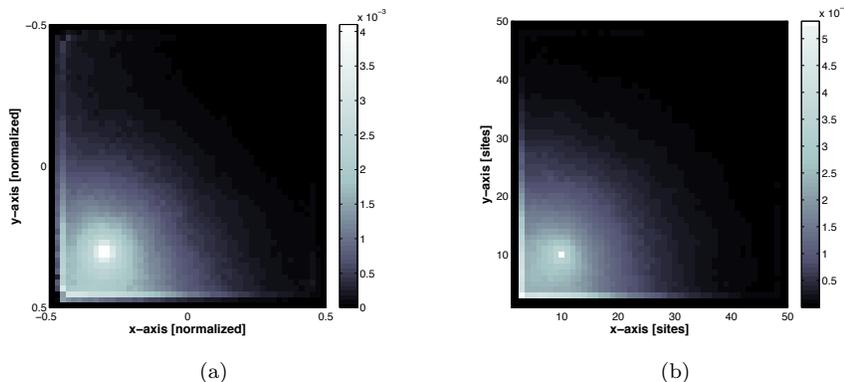


Figure 6: Spatial submicroscopic (Webots) and spatial microscopic models (for $t = 20\text{min}$), and origin at $x_0, y_0 = L/5$. (a) Submicroscopic model for 1500 runs. (b) Microscopic model taking the robot’s wall-interaction into account, for 5000 runs

excludes an under-sampling by choosing a too large value for T_{it} . Here, after T_{it} seconds the robot has covered an area at most equal to that of one site. In the limiting case, it moves at an average speed of one site per iteration. This calibration is applied to all our models. As shown before, the spatial microscopic model as well as the diffusion model with $\sigma_{x,y}^2 = 1$ take this into account. For a model lattice of $L = 50$ and for $v_{alice} = 0.01\text{ms}^{-1}$, we find an iteration time of $T_{it} = 1.33\text{s}$.

3.4 Multi-Level Comparison

In order to observe general model properties and to compare spatial submicroscopic, microscopic, and diffusion models, we designed the following experiment. We track a robot for a time span of 20min with a drop-off location at $x_0 = -0.195\text{m}$, $y_0 = 0.195\text{m}$ (for a field of length 0.65m and center at (0,0)). Figure 6 (a) shows the normalized number of visits per site in a 50×50 bin histogram.

An interaction with the wall becomes apparent and we observe that not all of the arena is covered. This reduces the actual space the robot moves in by 15%, which significantly affects the spatial probability distribution. In order to adapt our models to the robot’s behavior as shown in the submicroscopic embodied simulator (Webots), we implemented the spatial microscopic model as follows: the robot keeps a constant distance from the wall, here equivalent to 2 sites. Attempts to cross the arena’s boundary are penalized with a delay of $T_w = 2$ time steps, opposed to a default value of $T_w = 0$. These values result from measurements on the real robot running our obstacle avoidance controller as described in Section 2. Although the former values are carefully evaluated, the discretization of the field into sites may lead to differences in model predictions,

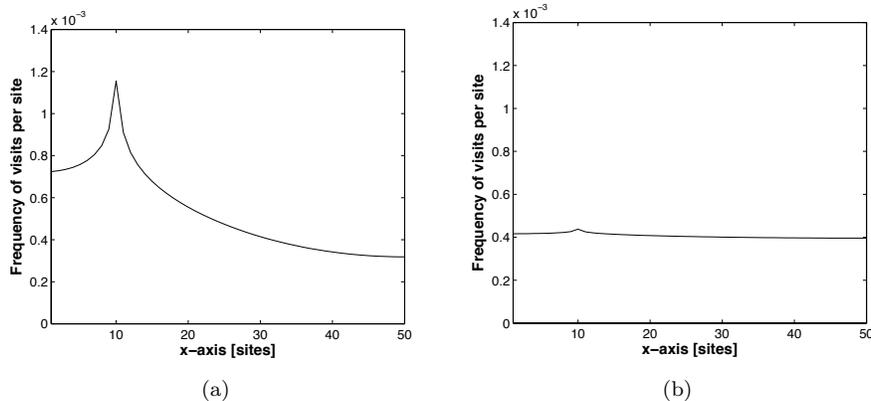


Figure 7: Predictions made by the diffusion model: a profile at $y = L/5$ of the normalized mean visits. (a) $t = 55\text{min}$ and (b) $t = 18\text{hrs}$.

as can be seen later in Section 3.6. Figure 6 shows the normalized number of visits per site for a submicroscopic model (Webots simulation) as well as the spatial microscopic model prediction.

We note that a (spatial) microscopic model allows us to approximate certain model parameters through simple measurements on the real system. As well, by forming the baseline for the creation of (spatial) macroscopic models, (spatial) microscopic models enable us to better grasp the abstract elements of a corresponding macroscopic system, validating the abstraction to higher modeling levels.

We now turn our attention to the diffusion model. As before, this model is applied on a lattice of L^2 sites. Essentially, the diffusion tensor defines the robot's speed in $\frac{\text{site}}{\text{iteration}}$. Therefore, in order to insure a quantitatively correct correspondence with the spatial microscopic model, where the agent moves forward one site per iteration, we have $\sigma_{x,y}^2 = 1$. As well, we define a time span of k_{max} , where k is equivalent to the number of iterations in the spatial microscopic model.

Figure 7 shows the normalized number of mean visits per site, $M(x, y; t)/t$, for an origin at $(x_0, y_0 = L/5)$, in (a) for a short time span ($t = 55\text{min}$), and in (b) for a longer time span ($t = 18\text{hrs}$). We notice how the reflecting boundaries of a bounded arena are taken into account. Further, we observe that for time approaching infinity, the robot distribution tends to be uniform. Analytically, the formula gives us

$$\lim_{\tau \rightarrow \infty} \frac{M(x, y; \tau)}{\tau} = \frac{1}{L^2} \quad (13)$$

which for $L = 50$ corresponds to $4 \cdot 10^{-4}$, and thus to the result in Figure 7 (b).

Figure 8 provides a graphical comparison between the diffusion model and the spatial microscopic model (without wall interaction). Both models were

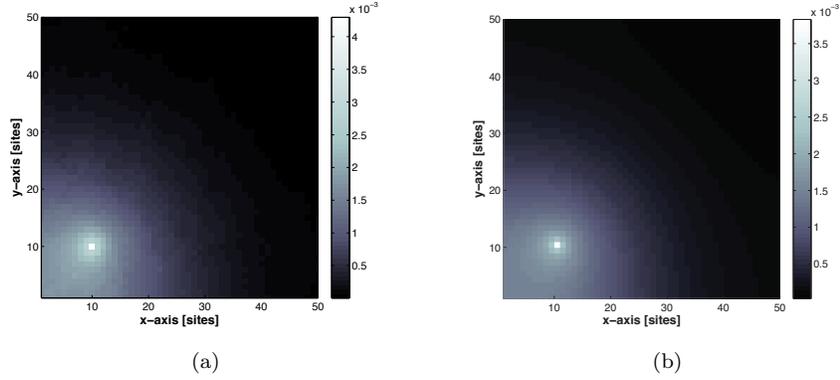


Figure 8: Spatial microscopic and diffusion models for $t = 13\text{min}$, dropoff location at site $x_0, y_0 = L/5$. (a) Spatial microscopic model for 5000 runs. (b) Diffusion model.

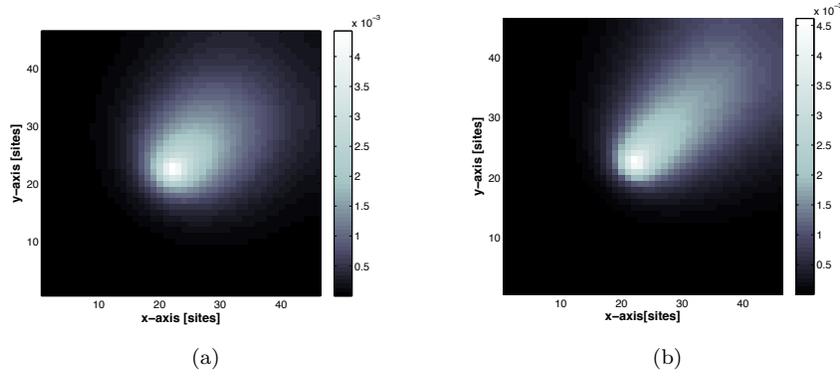


Figure 9: Diffusion model with non-zero drift. Drop-off location at the origin, $t = 3.3\text{min}$. (a) Drift $u_{x,y} = 0.1$ and (b) drift $u_{x,y} = 0.15$.

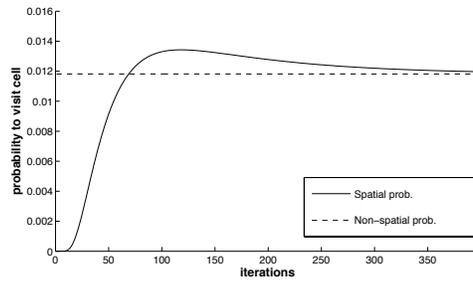


Figure 10: The time-dependent and time-independent probabilities to visit a cell of size $6.5\text{cm} \times 7.85\text{cm}$, at a distance of 21cm from the drop-off location

calculated for $t = 13\text{min}$. The models correspond well, qualitatively as well as quantitatively. Furthermore, we observe in Figure 9 how the diffusion model with non-zero u_x and u_y describes the frequency a site is visited for a given drift. The figure shows how a drop-off location at the center of the model lattice and model drift parameters $u_x = 0.1$, $u_y = 0.1$ create a diagonal drift towards the upper right hand corner of the field. We also observe how this bias increases for an increasing drift value.

Later in this paper we exploit the diffusion model’s property of calculating time-dependent probabilities. This is in contrast to non-spatial models, which predict a constant event probability defined solely by geometric characteristics of the environment. Figure 10 shows the probability to visit a cell at a distance of 21cm from the agent’s origin, as a function of time for both the non-spatial (independent of time, see Section 1.3) and diffusion models. An agent is initially at the origin (center of model lattice). With time, the spatial probability becomes uniform, and thus asymptotically approaches the value predicted by the non-spatial model.

3.5 Real Robot Experiment

We now validate the previously established models by quantitatively comparing the various model predictions with each other and also with real robot data. We consider an empty, square arena ($0.65\text{m} \times 0.65\text{m}$) bounded by four walls. An Alice II miniature autonomous robot (Caprari and Siegwart 2005), is dropped off at the center of the bounded area and explores the field in a random walk. We superimpose the areas of 12 cells over the field: the placing of one cell is defined by the parameters $x_{c_i,min}, y_{c_i,min}, x_{c_i,max}, y_{c_i,max}, i \in \{1..12\}$, specifying its lower and upper bounds. The agent, having no prior knowledge of the cells’ locations, will explore the arena. By moving over the area specific to one cell, the robot emulates the inspection of that cell.

We measure the number of times each cell is inspected in a given time span. Figure 11 is a picture of the setup captured by an overhead camera. It shows the bounded arena with the Alice robot and the 12 marked cells. In order to track the robot and detect if a cell has been visited, we use an overhead camera and the SwisTrack tracking software¹(Lochmatter et al. 2008).

As introduced in Section 2, the robot is controlled by a random walk with step length $d_{step} = 2\text{cm}$. Nevertheless, at the boundaries, the random walk is subsumed by a Braitenberg obstacle-avoidance behavior. We group the 12 cells into 3 subgroups, according to their distances to the origin. The first group contains the four cells nearest to the origin, at a distance of 19.2cm, the second contains the next four cells at a distance of 22.1cm, and the third contains the farthest cells at a distance of 34.9cm. This allows us to analyze the spatial characteristics of the setup as a function of the distance from the drop-off location.

We performed two series of experiments. A first series, for the short time

¹SwisTrack: <http://en.wikibooks.org/wiki/SwisTrack>

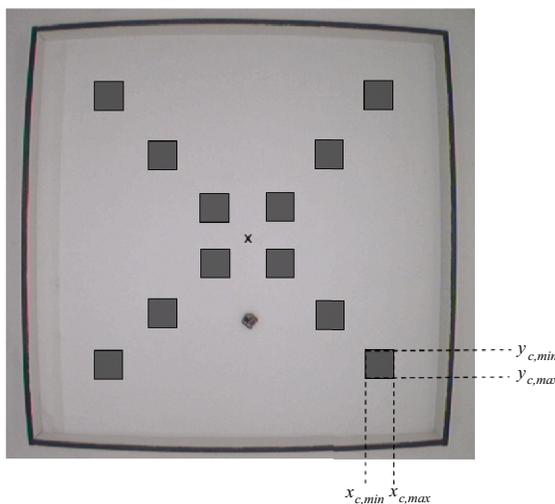


Figure 11: Overhead camera shows a bounded arena and an Alice robot. 12 cell areas are superimposed. The drop-off location is marked by an x

span of 8 minutes, and a second series for a longer time span of 10 hours. We consider both spatial and non-spatial models. For the short experiment, we validate model predictions with real experiments on the Alice robot as well as with an Alice model in the submicroscopic embodied robot simulator Webots. We performed 50 runs with the Alice robots, and 1000 runs in Webots. For the long experiment, we base our validations on the submicroscopic simulator. We performed 20 runs in the embodied simulator and 1000 runs using the spatial microscopic model.

3.6 Results

The results are evaluated by considering agreement among the different models (spatial and non-spatial) and by comparison to the real and simulated robot experiments. Results are presented in Figure 12, (a) for the short time span experiment and (b) for the long time span. We plot the standard error for iterated runs. We notice that for a short time span, our real and simulated robot experiments show a decreasing number of visits for an increasing distance to the origin. This prediction is obtained with both spatial models, the diffusion and spatial microscopic models. Moreover, for sake of reference, we consider the prediction of our non-spatial models (microscopic or macroscopic) described in Section 1.3. Since they do not differentiate cell locations, they predict a constant value for all three groups. Further, we are able to validate the equivalence of real robot and simulated (in Webots) robot results.

For the 10 hour experiment, we observe model predictions of a near to constant mean number of visits per group, for all three groups. This corresponds to

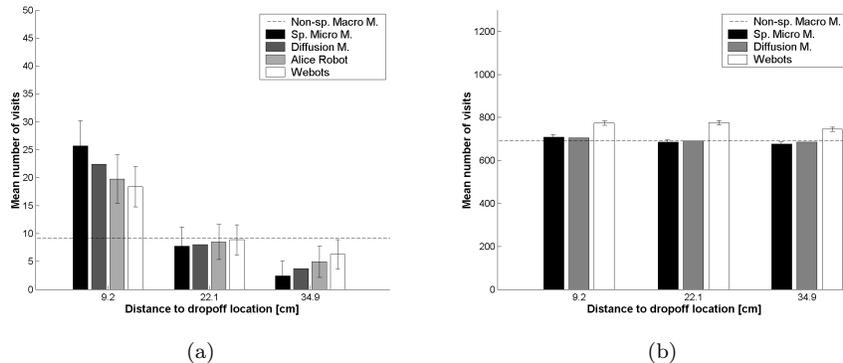


Figure 12: Mean number of visits made to three cell groups in function of the distance to the dropoff location. (a) $t = 8\text{min}$. (b) $t = 10\text{hrs}$

results obtained with the submicroscopic simulator, with a slight offset due to the coarse discretization of the field (as explained in Section 3.4). In this case, non-spatial models predict accurate values and dependency on the distance from the drop-off location can be neglected.

4 A Spatial Macroscopic Model

In this section we elaborate a spatial macroscopic model—a union of the non-spatial macroscopic model (Section 1.3) and the spatial diffusion model (Section 3.2). The result is a more powerful modeling tool, combining macroscopic population models with the spatiality of the diffusion model.

4.1 General Model Framework

In the current non-spatial modeling methodology, we make the hypothesis of uniform robot distribution, and therefore the probability to encounter a cell p_c is constant. Here, we limit the macroscopic system in a bounded space, in which the robot’s entry point is defined. The probability to encounter a cell is no longer constant, but for each consecutive time step k newly evaluated. The diffusion model provides us with $P(x, y; t)$, which is the solution of the Fokker-Planck equation. We implement it as $P(x, y; k)$, the probability a site (x, y) , is visited at time k . We integrate the probability of visiting site (x, y) over the area of a cell, defined by the bounds $c_{x/y,min}$ and $c_{x/y,max}$ and thus obtain an exact encountering probability $p_c(k)$ for each cell, as a function of time.

The system is modeled by the standard set of difference equations as first shown in equations 1-4. In this case, the potential presence of a drift suggests us to differentiate between the individual cells (symmetry cannot be assumed). $M_c(k)$ becomes $M_c(k, i)$, $\Delta_c(k)$ becomes $\Delta_c(k, i)$ and $p_c(k)$ becomes $p_c(k, i)$

where i indexes the cells, $i \in 1..M_0$.
Our system is now

$$\begin{aligned}
N_{rw}(k+1) &= N_{rw}(k) + \Delta_w(k - T_w) + \sum_{i=1}^{M_0} \Delta_c(k - T_c, i) - \sum_{i=1}^{M_0} \Delta_c(k, i) - \Delta_w(k) \\
N_w(k+1) &= N_w(k) + \Delta_w(k) - \Delta_w(k - T_w) \\
N_c(k+1) &= N_0 - N_{rw}(k+1) - N_w(k+1) \\
M_c(k+1, i) &= M_c(k, i) + \Delta_c(k, i), \quad i \in 1..M_0
\end{aligned}$$

with

$$\begin{aligned}
\Delta_w(k) &= p_w(k) \cdot N_{rw}(k) \\
\Delta_c(k, i) &= p_c(k, i) \cdot N_{rw}(k) \quad i \in 1..M_0
\end{aligned}$$

The above system is solved for a given time step k . Again, the initial conditions are: $N_{rw}(0) = N_0, N_w(0) = 0, N_c(0) = 0$, with $M_c(0, i) = 0, \forall i \in 1..M_0$ and $M_0 = 12$. We will consider the case for one robot, and so $N_0 = 1$. We express probabilities $p_w(k)$ and $p_c(k, i)$ in function of time steps, with p_c specific to each cell. We have

$$\begin{aligned}
p_w(k) &= \sum_{(x,y) \in A_w} P(x, y; k) dx dy \\
p_c(k, i) &= \sum_{(x,y) \in A_{c_i}} P(x, y; k) dx dy, \quad i \in 1..M_0
\end{aligned}$$

where A_w is the area the wall is detected in and A_c the area of one cell bounded by $c_{x/y, min}$ and $c_{x/y, max}$. The area $dx dy$ corresponds to that of one site.

4.2 A Case Study: Modeling a Spatial Drift

An environment under the influence of drift will cause a random walk to become asymmetric. Pursuing the initial motivation of modeling biased motion of mobile swarm-robotic systems, we use our adapted diffusion model which takes into account the influence of drift (Section 3.2.2), and subsequently introduce this model into our spatial macroscopic modeling framework (Section 4.1). Finally, we create an asymmetric experimental setup and study the correspondence of the predictions of this combined model with actual system performances.

4.3 Experiment

We validate our modeling approach as in Section 3.5. We wish to discuss the effects of drift on modeling system performance. The twelve cells are grouped into four groups, depending on the quadrants they lie in. The center of the arena being the origin of our coordinate system, the axes divide the arena into

four quadrants, each including three cells. We performed experiments with the time span of 8 minutes, once with real robots and once with the embodied robot simulator Webots, and compare the results to macroscopic model predictions. In the following, the two individual setups are elaborated in more detail.

4.3.1 Real Robot Experiment

We designed a real robot setup induced with a drift. We tilted the robot arena at an angle, creating an inclination on the arena’s floor. Due to the smooth arena surface, the higher the inclination, the higher the wheel slip, and the higher the drift in downhill direction. Figure 13 shows the arena being lifted on the left side by a height of 10cm. We performed two series of experiments,



Figure 13: The robot arena is tilted to an angle of $\alpha_2 = 4.5^\circ$.

one with lift $h_1 = 5\text{cm}$, the other with lift $h_2 = 10\text{cm}$. The arena box has a width of $w = 128\text{cm}$, and our inclination angles are $\alpha_1 = \sin^{-1}(h_1/w) = 2.2^\circ$, and $\alpha_2 = \sin^{-1}(h_2/w) = 4.5^\circ$. Corresponding to our coordinate system, this lift produces a drift in direction of the x-axis ($u_x \neq 0$), and no drift in the direction of the y-axis ($u_y = 0$).

We performed 40 8-minute runs for each of the two setups. The robot was placed at the center of the arena at the beginning of each run, its orientation repeatedly random.

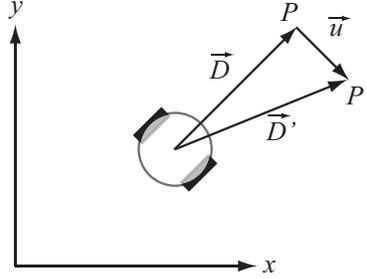


Figure 14: The integration of drift \vec{u} into the robot controller

4.3.2 Simulated Robot Experiment

Also, we introduce drift into our submicroscopic model: we integrate drift directly into the robot's random walk, enabling a straightforward evaluation of model predictions.

Figure 14 shows how we integrate drift into the robot's movements. At regular steps of $d_{step} = 2\text{cm}$, the robot rotates a random angle defining the direction for its next step. Let \vec{D} be this planned step, and P the resulting position. The two-dimensional drift vector \vec{u} is added to the planned step \vec{D} , and we have $\vec{D} + \vec{u} = \vec{D}'$, with $\vec{u} \ll \vec{D}$. So \vec{D}' is the step the robot will make under the influence of drift \vec{u} . When the robot approaches a wall, the random walk continues to be subsumed by obstacle avoidance behavior, and is resumed as soon as the obstacle is no longer detected.

Further, we elaborate the conversion from the model drift parameter $u_{x,y}$ to the drift parameter used in our submicroscopic model (Webots) $u_{x,y,web}$. In the diffusion model, $u_{x,y}$ is expressed in terms of $[\frac{sites}{it}]$. Our vector $\vec{u} = [u_{x,web} \ u_{y,web}]$ used in the robot controller is expressed in meters. For conversion, we define the following equivalence

$$\frac{u_{x,y}}{T_{it} \cdot L} = \frac{u_{x,y,web}}{t_{step} \cdot \Lambda}.$$

Here $t_{step} = d_{step}/v_{alice}$, where $d_{step} = 0.02\text{m}$, and $v_{alice} = 0.01\text{ms}^{-1}$. Further, the field has a length $\Lambda = 0.65\text{m}$ with $L = 50$ sites, and $T_{it} = 1.33\text{s}$ as presented above in Section 3.3.

We performed 1000 8-minute runs, once for a drift of $u_{x,web} = 0.003$, $u_{y,web} = 0$ and once for $u_{x,web} = 0.0035$, $u_{y,web} = 0$. The robot was repeatedly placed at the center of the arena at the beginning of each run with a random orientation.

4.4 Results

Figure 15 presents results obtained, (a) for the real robot experiment, and (b) for the experiments performed with our submicroscopic model (Webots) and the corresponding spatial macroscopic model predictions. Clearly, for u_x positive,

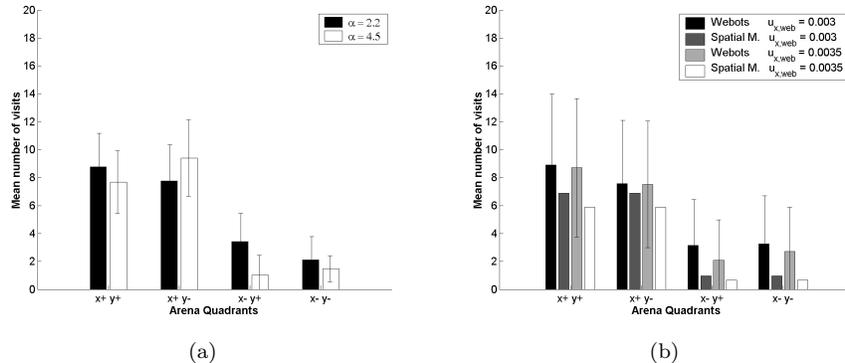


Figure 15: Mean number of visits made to cell groups according to the four quadrants. (a) Alice experiment for $\alpha = 2.2^\circ$ and $\alpha = 4.5^\circ$. (b) Spatial submicroscopic model (Webots) and spatial macroscopic model for drift parameter $u_{x,y,web} = 0.003$ and $u_{x,y,web} = 0.0035$

quadrants along the positive x-axis are visited more frequently than quadrants along the negative x-axis. For an increasing drift, we observe the decreasing number of visits to the cells in x-negative quadrants for both the real Alice and submicroscopic model experiments. Furthermore, the spatial macroscopic model predictions and submicroscopic model results suggest a decreasing number of visits to x-positive quadrants as well.

5 Discussion

5.1 Scaling Properties of the Proposed Model

We are interested in models describing the spatio-temporal distribution of a robot swarm as a function internally (robot controller) or externally (environment) induced drift phenomena. We incrementally derived a macroscopic model based on the Fokker-Planck diffusion equation, while validating model prediction with experimental data at every abstraction step. Although we validated our framework using only a single robot, because of the correspondence outlined in Section 2, our formalism has the potential to scale for larger number of robots. However, this approach is limited when the area used by the robots comes close to the total area of the environment—for our approach to hold, we must have $N_0 \cdot p_c(k) \ll 1, \forall k$. Also, interaction among the robots (collision avoidance) promotes their diffusion, which corresponds to a higher diffusion coefficient on a macroscopic level. In environments without drift, this will lead to a faster convergence towards a uniform distribution of the robots in the environment (compare Figure 7 (b) and Equation 13), encouraging the use of non-spatial probabilistic models for scenarios considering long enough time spans (Section 1.3). In scenarios with drift however, we have seen that the steady-state distri-

bution of the probability density function is non-uniform (Figure 1). Here, the impact of collision avoidance between the robots might be lower, as drift acting upon the robots will counter-effect the diffusion caused by collision avoidance.

5.2 Limitations of our Approach

In Section 2, we experimentally showed why we chose the concepts of random walk as a mathematical foundation for modeling the motion of an individual within a swarm. We note that this approximation only holds for extremely simple robots, endowed with simple reactive controllers, as for instance those used in (Correll and Martinoli 2006c) and in this paper. Generally speaking, macroscopic models are based on aggregate functions in order to obtain mathematically tractable, compact models. Depending on the modeling purpose and the targeted performance metric of the swarm system, a given aggregate function might or might not be appropriate and therefore might or might not deliver quantitatively correct predictions within a certain bound. Thus, if robots did not follow a random walk, this would not necessarily mean that the overall behavior resulting from many interactions could not be captured statistically and incorporated into a macroscopic model. This would simply mean that parametric aggregate functions based on well-studied random walk models could probably no longer be used. As a consequence, in such scenarios, appropriate distributions (sometimes only approximated by parametric formalisms) should be defined case by case. Finally, if even the latter correspondence does not hold, the spatiality of a single robot can be captured parametrically and compared with the spatiality resulting from multiple interacting robots. This method additionally enables the calibration of models with a single robot, as has been done in this work.

Further, our notion of drift as formulated in this paper only accounts for a constant drift over the arena. Although this might be the case for a regular structure as in (Correll and Martinoli 2006c), constant drift is seldom the case in nature (the current in an ocean or a river, or wind in an airspace). Nevertheless, it might be a valid assumption over time spans and areas that extend experimental time and space respectively. Also, it is worth mentioning that nothing prevents us from solving the diffusion equations numerically for a varying drift in function of time or space (compare Eq. 11). Further, we consider the calibration of the model parameters, in particular the calculation of drift parameters. The assumption of a constant drift (as provided for instance by a regular structure) allows us to obtain measurements for a sub-set of the experimental space, for instance, in the case of turbine-blade inspection, an environment consisting of only 3×3 blades (compare Figure 1). Parameters reproducing the observed probability density function in such a setup can be identified by means of system identification (Correll and Martinoli 2006b).

6 Conclusion and Outlook

In this work, we develop spatial microscopic and macroscopic models motivated by drift effects experimentally observed in a swarm of miniature robots tasked with the inspection of regular structures. We show how to combine state-of-the-art methods which do not capture spatial distributions with a spatial model, based on the solution of the Fokker-Planck equation borrowed from particle physics. We show how a single robot can effectively represent an individual member of a larger swarm and validate our modeling approach for scenarios with and without drift. Using the models developed in this paper, we are able to formulate the assumptions being made for the non-spatial methodology (Martinoli et al. 2004) more precisely: for time spans large enough, and in environments without drift, the spatial model predicts a uniform distribution of agents in space. Further, the combination of population based models with spatial models opens new possibilities to the modeling of stochastic, distributed robotic systems as well as to the modeling of real, biological systems. Still, some questions remain open on how to calibrate parameters of our models, and future research is necessary in order to find efficient methods to synthesize model parameters from experimental conditions. Lastly, a final validation of our proposed macroscopic model with a team of real robots is necessary and a generalization of the mapping between multiple-robot versus single-robot behavior should be envisioned.

7 Acknowledgements

The authors would like to thank the referees for all their helpful feedback. This work has been partially supported by a Swiss National Science Foundation grant (contract Nr. PP002-116913).

A Solution to the Fokker-Planck equation

The result presented in Equation 5.2 of Ferraro and Zaninetti (2001) is based on the Fokker-Planck equation. This equation describes the time evolution of the probability density function of position and velocity of a particle. The general form of the Fokker-Planck equation for N variables is

$$\begin{aligned} \frac{\partial W}{\partial t} = & - \sum_{i=1}^N \frac{\partial W}{\partial x_i} D_{1,i}(x_1, \dots, x_N) + \\ & \sum_{i=1}^N \sum_{j=1}^N \frac{\partial^2 W}{\partial x_i \partial x_j} D_{2,ij}(x_i, \dots, x_N) \end{aligned} \quad (14)$$

Here, D_1 is the drift vector and D_2 is the diffusion tensor. We will consider the one-dimensional case. Thus, the Fokker-Planck equation is formulated as

$$\frac{\partial P}{\partial t} = \frac{1}{2}\sigma_x^2 \frac{\partial^2 P}{\partial x^2} - u_x \frac{\partial P}{\partial x} \quad (15)$$

where $\frac{1}{2}\sigma_x^2$ is the diffusion tensor and u_x the drift vector.

In the following, we derive the solution to this partial differential equation. In contrast to Ferraro and Zaninetti (2001), we explicitly model the initial distribution by taking into account the robot origin (drop-off location, see Equation 19). Furthermore, we also show the solution to the two-dimensional case.

First, we consider the one dimensional case for reflecting boundaries. As variables x and t are independent, they can be separated as

$$P(x, t) = f(t)g(x). \quad (16)$$

We can thus express 15 in terms of $f(t)$ and $g(x)$ and their corresponding derivations $\frac{f(t)}{dt} = \dot{f}$, the time derivative, and $\frac{g(x)}{dx} = g'$, the space derivative:

$$\dot{f}g = \frac{1}{2}\sigma_x^2 f g'' - u_x f g'$$

By multiplying by $\frac{1}{fg}$ one obtains:

$$\frac{\dot{f}}{f} = \frac{1}{2}\sigma_x^2 \frac{g''}{g} - u_x \frac{g'}{g} = -\lambda$$

where λ is constant.

We then obtain the following two ordinary linear differential equations

$$\dot{f} = -\lambda f \quad (17)$$

$$\frac{1}{2}\sigma_x^2 g'' - u_x g' = -\lambda g \quad (18)$$

Under the following boundary and initial conditions, the system can be solved for λ . The boundary condition for reflecting boundaries is

$$\begin{aligned} \frac{\partial P(x=0, t)}{\partial x} &= \frac{\partial P(x=L, t)}{\partial x} = 0 \\ \Rightarrow \frac{\partial g(0)}{\partial x} &= \frac{\partial g(L)}{\partial x} = 0 \end{aligned}$$

and the initial condition

$$P(x, 0) = \delta(x - x_0) \quad (19)$$

where $\delta(x - x_0)$ is the Dirac function and represents the initial presence of an agent at one-dimensional coordinate x_0 .

System 17 and 18 can be solved with respect to t and x separately. In the following, first a solution for 18 is elaborated, and then one for 17.

Solving $g(x)$ by 18:

We consider the Ansatz (20) for linear differential equations and its characteristic equation (21):

$$g(x) = e^{ikx} \quad (20)$$

$$-k^2 \frac{1}{2} \sigma_x^2 - ik u_x + \lambda = 0 \quad (21)$$

The characteristic equation is solved and its solution is

$$k = \frac{-iu_x}{\sigma_x^2} \pm \sqrt{\frac{-u_x^2}{\sigma_x^4} + \frac{2\lambda}{\sigma_x^2}}$$

Here we consider the simple case without drift, that is for $u_x = 0$. Thus we introduce the solution k and $u_x = 0$ into 20.

$$g(x) = a \cos \sqrt{\frac{2\lambda}{\sigma_x^2}} x + b \sin \sqrt{\frac{2\lambda}{\sigma_x^2}} x$$

The boundary condition yields $b = 0$ and so we have

$$\begin{aligned} \sqrt{\frac{2\lambda}{\sigma_x^2}} L &= m\pi \\ \lambda &= \frac{1}{2} \frac{m^2 \pi^2 \sigma_x^2}{L^2}, \quad m \in \mathbb{N} \end{aligned}$$

leading to the following solution

$$g(x) = \sum_{m=0}^{\infty} \alpha_m \cos\left(\frac{m\pi}{L} x\right) \quad (22)$$

Solving $f(t)$ by 17:

Solving $f(t)$ is straightforward. From 17 one can derive $\frac{df}{f} = -\lambda dt$ and therefore $\ln(f) = -\lambda t$. Thus we have:

$$f = e^{-\lambda t} = e^{-\frac{1}{2} \frac{m^2 \pi^2 \sigma_x^2}{L^2} t}$$

Merging $f(t)$ and $g(x)$:

According to 16, the intermediary results now lead to

$$P(x, t) = \sum_{m=0}^{\infty} e^{-\frac{1}{2} \frac{m^2 \pi^2 \sigma_x^2}{L^2} t} \alpha_m \cos\left(\frac{m\pi}{L} x\right)$$

which can be reformulated as a Fourier Series in cosine (Dacorogna and Tanteri 2002) as

$$F_c f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos\left(\frac{\pi n}{L} x\right)$$

$$a_n = \frac{2}{L} \int_0^L f(y) \cos\left(\frac{\pi n}{L} y\right) dy.$$

By this theorem, we have equation 19 which yields

$$P(x, 0) = \sum_{m=0}^{\infty} \alpha_m \cos\left(\frac{m\pi}{L} x\right) = \delta(x - x_0)$$

$$\alpha_m = \frac{2}{L} \int_0^L \delta(x - x_0) \cos\left(\frac{\pi m}{L} x\right) dx$$

One can now extract a_0 from the sum term which leads to

$$P(x, 0) = \frac{1}{L} + \frac{2}{L} \sum_{m=1}^{\infty} \cos\left(\frac{m\pi x}{L}\right) \cos\left(\frac{m\pi x_0}{L}\right)$$

and finally the solution in function of x and t :

$$P(x, t) = \frac{1}{L} + \frac{2}{L} \sum_{m=1}^{\infty} e^{-\frac{1}{2} \frac{m^2 \pi^2 \sigma_x^2}{L^2} t} \cos\left(\frac{m\pi x}{L}\right) \cos\left(\frac{m\pi x_0}{L}\right)$$

Following the same methodology we obtain for two dimensions

$$P(x, y; t) = \frac{1}{L^2} + \frac{2}{L^2} \sum_n^{\infty} e^{-\frac{n^2 \sigma_y^2 \pi^2 t}{2L^2}} \cos\left(\frac{n\pi y}{L}\right) \cos\left(\frac{n\pi y_0}{L}\right)$$

$$+ \frac{2}{L^2} \sum_m^{\infty} e^{-\frac{m^2 \sigma_x^2 \pi^2 t}{2L^2}} \cos\left(\frac{m\pi x}{L}\right) \cos\left(\frac{m\pi x_0}{L}\right)$$

$$+ \frac{4}{L^2} \sum_n^{\infty} \sum_m^{\infty} e^{-\frac{n^2 \sigma_y^2 \pi^2 t}{2L^2}} \cos\left(\frac{n\pi y}{L}\right) \cos\left(\frac{n\pi y_0}{L}\right)$$

$$\cdot e^{-\frac{m^2 \sigma_x^2 \pi^2 t}{2L^2}} \cos\left(\frac{m\pi x}{L}\right) \cos\left(\frac{m\pi x_0}{L}\right)$$

where $[x_0, y_0]$ is the initial position. For the case of a non-zero drift, one obtains a series composed of both sine and cosine functions, that cannot be transformed into a general Fourier series. For this work, we opted for the numerical solution as described below.

A two dimensional drift is defined by vector $[u_x u_y]$. Following the Fokker-Planck equation (see 14), Eq. 7 becomes

$$\frac{\partial P}{\partial t} = \frac{1}{2} \sigma_x^2 \frac{\partial^2 P}{\partial x^2} + \frac{1}{2} \sigma_y^2 \frac{\partial^2 P}{\partial y^2} - u_x \frac{\partial P}{\partial x} - u_y \frac{\partial P}{\partial y} \quad (23)$$

We apply the same initial conditions as for Eq. 7 (Eq. 8-9). We consider the factors d, c and a , and our probability function P . The generic parabolic equation is

$$d \frac{\partial P}{\partial t} - \nabla \cdot (c \nabla P) + aP = f \quad (24)$$

with the initial value $P_0 = P(t_0)$. The Fokker-Planck equation can be written in these terms and numerically solved, resulting in probability $P(x, y; k)$ with non-zero drift. We set the constants as follows

$$\begin{aligned} d &= 1 \\ c &= \frac{1}{2} \cdot \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix} \\ a &= 0 \end{aligned} \quad (25)$$

and f is thus

$$f = \begin{pmatrix} u_x \\ u_y \end{pmatrix} \cdot \nabla P \quad (26)$$

References

- W. Agassounon, A. Martinoli, and K. Easton. Macroscopic modeling of aggregation experiments using embodied agents in teams of constant and time-varying sizes. *Autonomous Robots*, 17(2-3):163–191, 2004.
- R. Arkin. *Behavior-Based Robotics*. The MIT Press, Cambridge, MA, USA, 2000.
- G. Barton. *Elements of Green's functions and propagation: potentials, diffusion and waves*. Oxford Science Publications. Clarendon Press, Oxford, 1989.
- H. C. Berg. *Random Walks in Biology*. Princeton University Press, 1983.
- E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. SFI Studies in the Science of Complexity, Oxford University Press, New York, NY, USA, 1999.
- V. Braitenberg. *Vehicles. Experiments in Synthetic Psychology*. MIT Press, Cambridge, Mass., 1984.
- S. Camazine, J.-L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau. *Self-Organization in Biological Systems*. Princeton Studies in Complexity. Princeton University Press, 2001.

- G. Caprari and R. Siegwart. Mobile micro-robots ready to use: Alice. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3295 – 3300, 2005.
- L. Chaimowicz and V. Kumar. Aerial shepherds: Coordination among UAVs and swarms of robots. In *7th International Symposium on Distributed Autonomous Robotic Systems (DARS)*, pages 231–240, Toulouse, France, June 23-25 2004.
- A. Chatterjee. An overview of spatial microscopic and accelerated kinetic monte carlo methods. *Journal of Computer-Aided Materials Design*, 14:253, 2007.
- H. Choset. Coverage for robotics - a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31:113–126, 2001.
- N. Correll and A. Martinoli. Modeling and Optimization of a Swarm-Intelligent Inspection System. In *Int. Symp. on Distributed Autonomous Robotic Systems (DARS)*, Distributed Autonomous Robotic Systems, Springer Verlag, pages 369–378, 2006a.
- N. Correll and A. Martinoli. System Identification of Self-Organizing Robotic Swarms. In *The 8th Int. Symp. on Distributed Autonomous Robotic Systems (DARS)*, Distributed Autonomous Robotic Systems, pages 31–40. Springer Verlag, 2006b.
- N. Correll and A. Martinoli. Collective Inspection of Regular Structures using a Swarm of Miniature Robots. In *The 9th Int. Symp. on Experimental Robotics (ISER)*, Springer Tracts in Advanced Robotics, pages 375–385, 2006c.
- B. Dacorogna and C. Tanteri. *Analyse avancée pour ingénieurs*. Presses Polytechniques et Universitaires Romandes, Lausanne, 2002.
- J. Jr. Douglas. A survey of numerical methods for parabolic differential equations. *Advances in Computers*, 2:pp. 1–52, 1961.
- M. Ferraro and L. Zaninetti. Number of times a site is visited in two-dimensional random walks. *Physical Review E*, 64(056107), 2001.
- R. J. Gelten. Monte carlo simulations of a surface reaction model showing spatio-temporal pattern formations and oscillations. *The Journal of Chemical Physics*, 108:5921–5935, 1998.
- H. Hamann and H. Wörn. A framework of space-time continuous models for algorithm design in swarm robotics. *Swarm Intelligence*, 2(2-4):209–239, 2008.
- K. Lerman and A. Galstyan. Mathematical model of foraging in a group of robots: Effect of interference. *Autonomous Robots*, 2(13):127–141, 2002.

- T. Lochmatter, P. Roduit, C. Cianci, N. Correll, J. Jacot, and A. Martinoli. SwisTrack - A Flexible Open Source Tracking Software for Multi-Agent Systems. In *IEEE/RSJ 2008 International Conference on Intelligent Robots and Systems (IROS)*, pages 4004–4010. IEEE, 2008.
- A. G. Makeev. Coarse bifurcation analysis of kinetic monte carlo simulations: a lattice-gas model with lateral interactions. *The Journal of Chemical Physics*, 177:8229–8240, 2002.
- A. Martinoli. *Swarm Intelligence in Autonomous Collective Robotics: From Tools to the Analysis and Synthesis of Distributed Collective Strategies*. PhD thesis, DI-EPFL, Lausanne, Switzerland, 1999.
- A. Martinoli, K. Easton, and W. Agassounon. Modeling of swarm robotic systems: A case study in collaborative distributed manipulation. *Int. Journal of Robotics Research*, 23(4):415–436, 2004.
- O. Michel. Webots: Professional mobile robot simulation. *Journal of Advanced Robotic Systems*, 1(1):39–42, 2004.
- J. Rudnick and G. Gaspari. *Elements of the Random Walk*. Cambridge University Press, Cambridge UK, 2004.
- I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke. Data collection, storage, and retrieval with an underwater sensor network. In *Proceedings of the 3rd international conference on Embedded networked sensor systems (SenSys)*, pages 154–165, 2005.
- A. Winfield, W. Liu, J. Nembrini, and A. Martinoli. Modelling a wireless connected swarm of mobile robots. *Swarm Intelligence*, 2:241–266, 2008.