Chapter Eight

Self-Organized Robotic Systems: Large-Scale Experiments in Aggregation and Self-Assembly using Miniature Robots

Grégory Mermoud, Amanda Prorok, Loïc Matthey, Christopher M. Cianci, Nikolaus Correll, and Alcherio Martinoli[1]

8.1 Introduction

Scientific and technological breakthroughs in the field of nano- and microengineering have steered the robotics community towards the realm of extreme miniaturization. Very small robots of a few centimeters in size can access environments that are beyond the reach of larger robots, with recent case studies including scenarios such as the inspection of the digestive tract [Nagy *et al.* (2008); Rentschler *et al.* (2008)] or complicated industrial machinery [Correll and Martinoli (2009)]. Further miniaturization down to the micro- or nanoscale holds even more exciting potential in a large variety of fields [Woern *et al.* (2006); Abbott *et al.* (2007); Dong and Nelson (2007)]. However, miniaturization comes at a price: such robots are likely limited to minimalist computational, sensing, actuation, and communication capabilities. These severe restrictions create the need for a collaborative approach towards the solving of tasks by leveraging perception and action at a collective level.

The need for collective perception and action generally emerges in systems that involve robots that are several orders of magnitude smaller than the environment in which they operate, or that have too limited sensorimotor capabilities for carrying out a given task (which, in nature, is often mere survival). For instance, in the case of environmental monitoring, sensor nodes of a few centimeters in size must observe an environment (e.g., a forest, mountain, or city) that might be up to several kilometers in size [Barrenetxea *et al.* (2008)]. In order to accomplish mon-

[1]This work was partially supported by SelfSys, a project sponsored by the Swiss research initiative Nano-Tera.ch.

itoring and inspection tasks in such scenarios, large-scale systems composed of hundreds or more individual nodes must be deployed [Howard *et al.* (2006a,b)]. Successful control schemes for large-scale systems can range from fully centralized (i.e., control algorithms are essentially carried out by a central computer, which then dispatches precise instructions to each individual robot, perhaps as probabilistic templates) [Michael *et al.* (2008); Milutinovic and Lima (2006)] to fully distributed (i.e., control algorithms run entirely on-board, and have generally access to only limited, local information) [Christensen *et al.* (2007)]. Centralized control is easier to formalize in a theoretical framework, and it often allows for achieving optimal performance, but it has usually high requirements in terms of communication bandwidth and computational resources at the central control unit. Centralized control schemes also suffer from limited scalability in terms of number of nodes, and are intrinsically characterized by a single point of failure (i.e., the central control unit). In contrast, distributed control is very attractive in terms of scalability and robustness, typically exhibiting a graceful degradation of system performance in the presence of one or more unit failures or malfunctions. However, distributed robotic systems, especially those consisting of a large number of autonomous mobile units, are generally very difficult to design and analyze. The complementary challenges of synthesis and analysis in such cases have been the focus of several recent research efforts within the domain of distributed robotic systems.

Centralized control algorithms are essentially carried out by a central unit, which then dispatches precise instructions of various form (e.g., probabilistic templates, high-level orders, motor commands) to each individual robot.
Distributed control algorithms run entirely on-board, and have generally access to only limited, local information about the environment and the state of the other robots.

8.1.1 *Self-Organization*

The first objective for this chapter is to review some of the techniques used for designing and analyzing distributed control strategies for large-scale systems. One of the coordination mechanisms that has proven very successful in addressing this type of problem is *self-organization*, particularly for very resource-constrained systems [Nagpal *et al.* (2006); Pfeifer *et al.* (2007); Baldassarre *et al.* (2006)].

Self-organization is the process whereby a global pattern emerges from multiple interactions among the lower-level components of the system, without any external guidance, combined with, and taking advantage of, the randomness inherent in the system.

The rules specifying these interactions[2] are executed without explicit reference

[2]One could associate other features to the concept of self-organization such as strong emergence or energy dissipation. While these features are commonly observed in many self-organized systems [Haken (2006)], they are much less relevant to the topic of this chapter, and therefore we do not discuss them in

to the global pattern, thus allowing the self-organized strategies to be extremely scalable. We will support the discussion with a series of real case studies that involve large groups of small mobile robots with minimal capabilities performing complex tasks (see Fig. 8.1.).

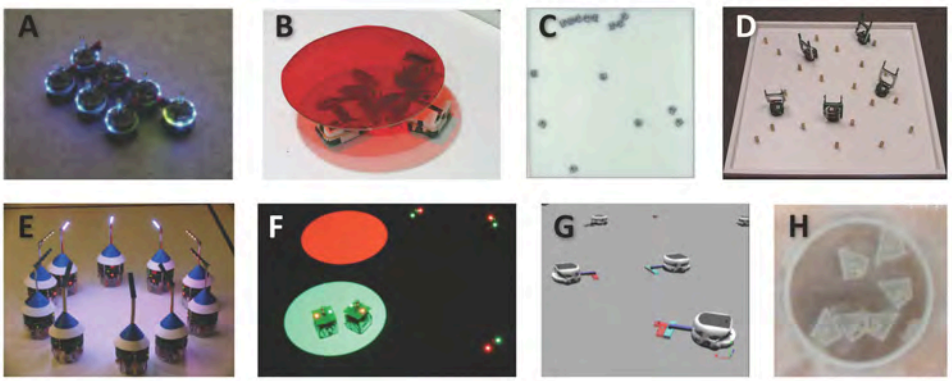


Figure 8.1. Examples of aggregation and self-assembly in real and simulated robotic systems. (**A**) Morphology control in a group of swarm-bot robots (10 centimeters in diameter) [Christensen *et al.* (2007)]. (**B**) Aggregation-mediated decision-making in mixed societies of robots and cockroaches [Halloy *et al.* (2007)]. (**C**) Self-assembly of Alice robots (2 centimeters in size) into chains of controllable size using minimalist local communication [Evans *et al.* (2010)]. (**D**) Clustering of objects in a swarm of mobile robots using only local perception (see [Martinoli *et al.* (1999)] and Sec. 8.3.1). (**E**) A group of networked e-puck robots forming a distributed lamp that can adapt to its environment (see Sec. 8.2). (**F**) Aggregation-mediated decision-making in a group of Alice robots endowed with noisy sensors (see [Mermoud *et al.* (2010)] and Sec. 8.3.2). (**G**) Distributed assembly of heterogeneous parts into planned structures using stochastic strategies in a swarm of mobile robots [Matthey *et al.* (2009)]. (**H**) Self-assembly of water-floating tethered units, called Tribolon, which are endowed with a vibrator for controllability purposes [Miyashita *et al.* (2008)].

Ranging from the clustering of small objects to collective decision-making, all of these tasks involve aggregation or the formation of specific spatial patterns, thus enabling a straightforward visualization and understanding of the case study. Although we focus here mainly on engineered systems rather than on natural ones, we may draw our inspiration from Nature when it comes to the design of large-scale systems and their control schemes, in particular when using self-organized strategies. Aggregation, for instance, is an efficient mechanism exploited by Nature for favoring interactions and information exchange between biological individuals, and thus enabling the emergence of complex collective behaviors [Garnier *et al.* (2008)] ranging from predator protection [Parrish and Edelstein-Keshet (1999)] to collective decision-making [Halloy *et al.* (2007)].

Similarly, a group of robots may exploit random encounters for sharing information collectively while communicating locally; they may also form spatial patterns and structures based only on local stochastic rules of interaction. This specific type of spatial self-organization is called *self-assembly*. Note that self-assembly does not necessarily involve a physical connection among the building blocks; especially in the context of dynamical self-assembly, where systems operate far

further details.

from equilibrium [Grzybowski and Campbell (2004)]. Although the concept of self-assembly originated in chemistry, components of any size (from molecules to galaxies) can self-assemble [Whitesides and Grzybowski (2002)], including engineered components such as passive building blocks [Boncheva *et al.* (2003)] or fully-fledged mobile robots [Groß and Dorigo (2008)].

instructions to each of the individual robots, perhaps as broadcast probabilistic templates) [Michael *et al.* (2008); Milutinovic and Lima (2006)] to fully distributed (i.e., control algorithms run entirely on-board, and have generally access to only limited, local information) [Christensen *et al.* (2007)]. Centralized control is easier to formalize in a theoretical framework, and it often allows for achieving optimal performance, but it has usually high requirements in terms of communication bandwidth and computational resources at the central control unit. Centralized control schemes also suffer from limited scalability in terms of number of nodes, and are intrinsically characterized by a single point of failure (i.e., the central control unit). In contrast, distributed control is very attractive in terms of scalability and robustness, typically exhibiting a graceful degradation of system performance in the presence of one or more unit failures or malfunctions. However, distributed robotic systems, especially those consisting of a large number of autonomous mobile units, are generally very difficult to design and analyze. The complementary challenges of synthesis and analysis in such cases have been the focus of several recent research efforts within the domain of distributed robotic systems.

Self-assembly is a specific class of self-organization whereby a set of pre-existing units autonomously form *spatial* patterns or structures without any external guidance.

Both self-organization and self-assembly usually rely on four fundamental ingredients, which we demonstrate here in the specific context of self-assembly:

(1) **positive feedback** is, in the context of self-assembly, an attractive force, or a binding interaction;
(2) **negative feedback** is generally a repulsive force, or an exhaustion of the building blocks;
(3) **randomness** is the property of a process whose realizations do not follow a predictable deterministic pattern, but are rather characterized by a probability distribution;
(4) **multiple interactions** is the fact that the different components of the system interact with each other often enough with respect to the duration of the process.

Self-organized systems also differentiate based on the substrate and mechanisms used to share information among the units. One example of such mechanism is stigmergy [Theraulaz and Bonabeau (1999)], which is a powerful indirect anonymous communication mechanism exploited by insect societies and some self-organized artificial systems reported in the literature [Beckers *et al.* (1994); Martinoli *et al.* (1999); Agassounon *et al.* (2004); Mamei and Zambonelli (2007); Werfel *et al.* (2006)].

Section 8.3.1 presents a case study that is a typical illustration of stigmergic coordination. In that particular case, the clustering of objects serves as a dynamic environmental template that guides the action of the swarm. This type of control mechanism is simultaneously flexible and scalable, and therefore quite suitable to the control of large-scale systems.

Stigmergy is an indirect anonymous communication mechanism, which relies on the specific signs left in the environment by the agents' actions, which in turn stimulates subsequent actions. It mediates the formation of complex structures and spatial patterns, without need for any planning, control, or even direct communication between the agents.

8.2 From Centralized to Distributed Control: The Case Study of A Distributed Table Lamp

With recent progress in embedded systems technology, increasing efforts are going towards the introduction of distributed, miniature robotic systems into everyday environments, with the ultimate goal of achieving seamless integration and disappearing technology. By leveraging new radio technologies, power-aware resource management, and controlled mobility, networked robotic systems are able to fulfill even more ambitious objectives. Still, as the results here will show, growing application requirements and system complexity pose challenges to centralized and distributed control strategies alike. The first case study presented in this chapter addresses the control and design of a physically distributed table lamp. This study is performed on a system of networked e-puck mobile robots, which—due to their small size and robustness—are ideal for prototyping robotic tools for everyday life [Cianci *et al.* (2008)].

The goal of this project was to explore the opportunities and challenges encountered in the development of a specific application—an interactive distributed table lamp. The intrinsically distributed nature of the system, embodied by a group of individual robots, exposed the system engineers (roboticists and interaction designers in close collaboration) to one essential challenge—what are the design choices to be made, in order for the system to be efficient (i.e., fast) and robust? In the following sections, we take the reader through the various elements of the system, and progressively show how this question was answered.

8.2.1 *The Configuration Problem*

Each of the robots in the system is equipped with a light, and the group is given the task of assuming various configurations as a function of user activity or instructions (see example in Fig. 8.2.(a)). There are several ways in which one could imagine approaching this type of coordination: through local rules for self-assembly [Klavins *et al.* (2006)], potential fields [Song and Kumar (2002)], or environmental templates [Correll and Martinoli (2006)]. In order to be effective, all of these methods will require some degree of coordination among the agents and

(a) (b)

Figure 8.2. (a) A networked multi-robot system forms a reconfigurable interactive table lamp, assuming various configurations based on user activity or input. (b) Interaction setup. Robot and user positions are tracked using two different tracking systems. The information is then combined and sent via radio packets to the robots, which take action accordingly.

with the user. Here, we focus on the inter-agent organization, and assume that the agent-user interface is handled by an independent process. Thus, the general objective of the system can be narrowed down to a *configuration* task, which consists of compelling the multi-robot system to move from a spatial pattern (e.g., randomly scattered, ordered configuration for a given activity) to another one (e.g., an ordered configuration for another activity). While the speed of the configuration process and the final accuracy of the configuration are the ultimate goals, our specific solution addresses *robustness* in particular, due to the typical resource boundedness of miniaturized robotic systems.

The setup consists of a collection of e-puck robots fitted with lamp turrets, a table with marked boundaries for them to interact on, an overhead camera for tracking the positions of the robots (using the SwisTrack multi-agent tracking software [Lochmatter *et al.* (2008)]), and a human-computer interface which senses and indicates which regions of the workspace are currently occupied (see Fig. 8.2.(b)). The target configuration of the distributed lamp is controlled through crude position and attitude tracking of users around the table. User and robot tracking are then integrated in software, and configuration and positioning information is then sent to the robots.

The goal of the multi-robot system is to reach a specific organized state as quickly and accurately as possible. More formally, let $p_i \in P$ be the position of robot i, C a goal configuration with c_j the individual placements, T_{max} the maximum time-to-completion allowed, and ϵ a precision parameter. Then , for N robots, the aim is to find $t_C < T_{max}$ such that $\forall t > t_C \ \forall i = 1 \ldots N$, $\exists c_j \in C$ such that the actual position lies within acceptable error bounds, i.e., $distance(p_i(t), c_j) < \epsilon$.

8.2.2 *System & Algorithms*

Absolute position messages are sent to the robots at approximately 1Hz, enabling the local computation of target positions. The position messages also enable a recalibration of the local odometry, increasing the accuracy of robot maneuvers at the individual level. Although certain global knowledge is available to the robots, the approach implemented for this project involves no explicit path planning.

In an intuitive approach towards simplifying the design of the algorithm, we decompose the configuration task into a set of independent subtasks, namely *position allocation, collective motion*, and *low-level control*. While each of the subtasks can be solved separately, the overall group behavior converges to the desired outcome. This resulting algorithm is given the name *Layered Nearest-Neighbor Control*. From a general perspective, the algorithm relies on two complementary ingredients: layered environmental templates broadcasted to each robot and local inter-robot interactions. This combination of centralized, broadcast-based control and local distributed control allows for both fast execution owing to the centralized commands, and enhanced robustness, mainly achieved through the anonymousness of robots (i.e., any robot is replaceable by a teammate in a given configuration) and local noisy interactions (avoidance of deadlocks) [Michael *et al.* (2008); Milutinovic and Lima (2006)].

Position allocation: The robots are assigned to their final positions within the configuration, which may be fixed or agreed upon at the beginning of the run or dynamically re-allocated during the run. This subtask is governed by a nearest-neighbor allocation algorithm: each robot attempts to move towards the closest unfilled position in the target configuration. This strategy may result in allocating more than one robot to the same destination within a given target configuration, however, once the position is filled the other robots will be automatically reassigned (*low-level control* handles the case where two robots arrive simultaneously).

Collective motion: Certain configuration architectures may lead to potential deadlock situations resulting from unfilled interior positions which may no longer be accessible due to other already placed robots. Thus, a *collective motion* directive is established by separating the list of target positions in the configuration into *layers*, such that exterior positions may not be filled (are considered as not available) until the interior ones are. Figures 8.3.(a) and 8.3.(b) depict two example configurations divided into layers. These specific layers are defined by exploiting symmetries in the shape to be constructed; algorithmic extensions to the solutions presented here should involve an automatic partitioning of the configuration into layers and a distributed implementation on the robotic platform.

Low-level control: This strategy comprises an individual robot's movement, and consists of a simple control layer combining obstacle avoidance with movement towards the target position. Both behaviors are computed locally, on the robot. Motor commands are determined straightforwardly by attempting to drive directly to the currently allocated target position; if an obstacle is encountered along the way, the robot will execute a random turn (in place) and a random back-off before re-attempting to drive straight towards its target. This approach was used specifically due to its nature of minimizing the accumulation of error in odometry, which is used to interpolate between reception of successive position messages. A pseudocode representation of the complete controller is shown in Algorithm 8.1.

Algorithm 8.1 Layered Nearest-Neighbor Control

Receive target configuration from tracker.
repeat
  Receive position information (x, y) from tracker.
  Select nearest unfilled configuration location $(\hat{x}, \hat{y}, \hat{\theta})$ as target.
  Attempt to move straight towards selected target.
  if Obstacle detected before target reached **then**
    Perform random turn / random backoff.
  else
    Estimate current orientation (θ) from difference between
      current and previous positions.
  end if
until Target location reached.

8.2.3 *Down to Reality*

The system was tested in reality for two specific configurations—the circle and triangle (see Fig. 8.3.(c) and Fig. 8.3.(d))—and in two variants: (i) robots are driven from an initially scattered state into a structured configuration, and (ii) robots are driven from one existing completed configuration into another. These test cases were performed in sequence (random $\rightarrow$ configuration $\rightarrow$ configuration) ten times for each combination ($\triangle \rightarrow \triangle$, $\triangle \rightarrow \bigcirc$, $\bigcirc \rightarrow \triangle$, $\bigcirc \rightarrow \bigcirc$). We recorded the time taken to accomplish the configurations as well as message loss: all experiments were completed successfully, with a time-to-completion of less than 150s, and a recorded message loss of 45%.

The results show us that even when including global information, optimal solutions are not necessarily guaranteed. Furthermore, theoretical approaches may not be tenable due to the non-negligible amount of message loss. In this respect, the current system demonstrated a high level of robustness. Yet, for an ultimately scalable system, even partial centralization will become inefficient as the conditions are worsened by the increasing number of agents. Decentralization will include node-to-node local communication and localized relative positioning methods using on-board sensors. However, these methods will introduce new challenges, as an accumulation of uncertainties at the collective level arise due to partial perception at the individual nodes in the system.

Many of these challenges may yet be addressed in the near future with additional advances in the area of networked robotic systems.

8.3 Self-Organized Strategies for Distributed Control

If very strict constraints are set on the individual robots (e.g., computational resources, reliability) because of miniaturization for instance, and if time pressure for achieving the task is not too high, fully distributed control strategies can be consid-

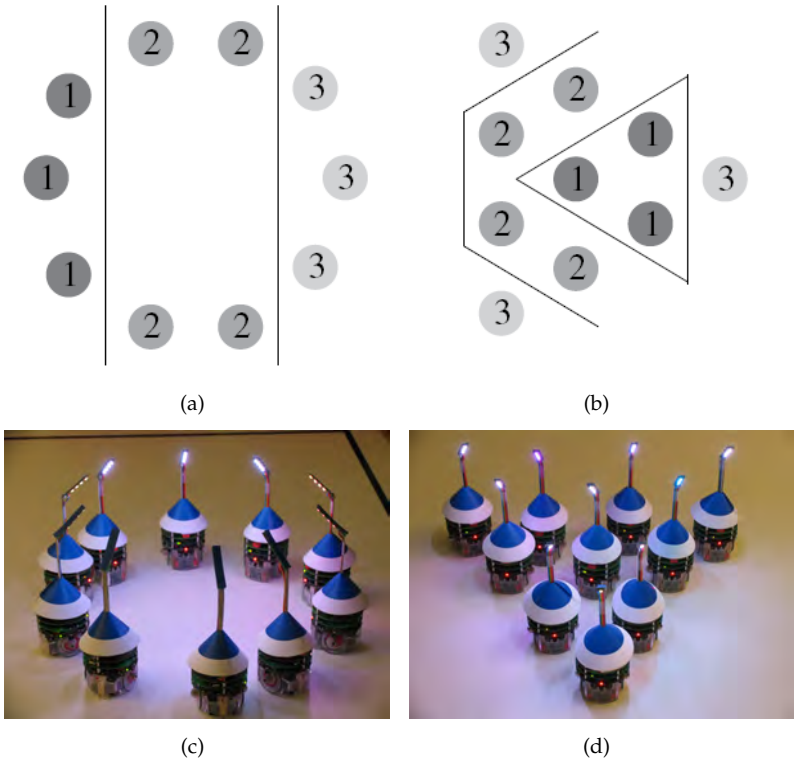
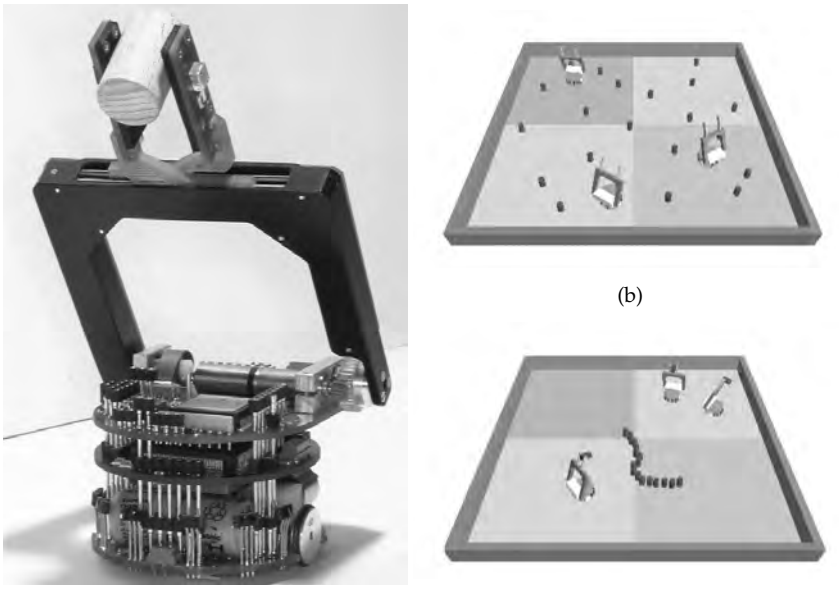


Figure 8.3. *Top:* examples of two configurations constructed from layers, so as not to unintentionally isolate robots from the positions they need to fill: (a) an open circle, and (b) a packed triangle. Positions are numbered by the layer that they belong to (e.g., all positions marked "1" must be filled before the positions marked "2" become available). *Bottom:* physical realization of these two configurations ((c) an open circle, and (d) a packed triangle) using ten e-pucks with floor sensors, radios, and lamp attachments. The "arena" (section of the table marked off with black lines) is 150 $\times$ 90 centimeters.

ered. Self-organization has proven to be very successful among the variety of coordination mechanisms used in distributed robotics. In this section, we support the discussion with two concrete examples illustrating how apparently complex problems can be solved with very simple reactive agents using self-organization. The first case study is concerned with the structured assembly of small objects using a group of Khepera I robots (see Fig. 8.4.(a)) [Martinoli *et al.* (1999)] . The second case study involves a team of miniature mobile Alice robots (depicted in Fig. 8.8.(b)) that must achieve collaborative screening of a noisy environment in order to identify and destroy undesirable objects [Mermoud *et al.* (2010)].

8.3.1 *Clustering of Objects*

In this experiment, the task is to collect small objects, referred to as "seeds", which are initially scattered throughout a square arena, and to gather them in a single in-line, structured aggregate using Khepera I robots equipped with grippers, and capable of distinguishing small objects to manipulate from obstacles to avoid with their frontal proximity sensors (see Fig. 8.4.). As the robots have only local sensing



(a) (b) (c)

Figure 8.4. (a) A Khepera robot holding a seed in its gripper. (b) Seed scattering at the beginning of a simulated experiment of aggregation with 3 robots, and (c) after about 4 hours of simulated time.

capabilities and do not exploit a global communication network, there is neither central nor global coordination among robots. In the experiments described in this chapter, robots do not exploit any form of specific wireless peer-to-peer communication; only stigmergic communication via the assembly process is considered. The type of stigmergic communication is in this case qualitative (or discrete) since the stimulating sign (a small object to manipulate) differs qualitatively from other perceptual stimuli (obstacles to avoid) rather than quantitatively (e.g., the spatial density of seeds).

The behavior of each robot is determined by a simple hand-coded program that can be represented with a standard flow chart or a Finite State Machine (FSM) (see Fig. 8.5.). In its default behavior, the robot moves straight forward within the working zone looking for seeds. When at least one of its six frontal proximity sensors is activated, the robot starts a discrimination procedure. Basically, two cases can occur: the robot might be in front of a large object (an arena wall, another robot, or a "wall" of contiguous seeds) or a small object (a seed or the tip of an in-line seed aggregate). In the first case, the object is considered to be an obstacle, and the robot avoids it. In the second case, the small object is considered to be an object to manipulate. If the robot is not already carrying a seed, it grasps the small object in front of it with the gripper, otherwise it drops the seed it is carrying close to the small object it has found; then in both cases, the robot resumes searching for seeds.

This simple individual behavior has three consequences: (1) the team of robots is able to gather objects in aggregates of increasing size, (2) aggregates have a precise structure and are built in line, and (3) eventually, the aggregation process will

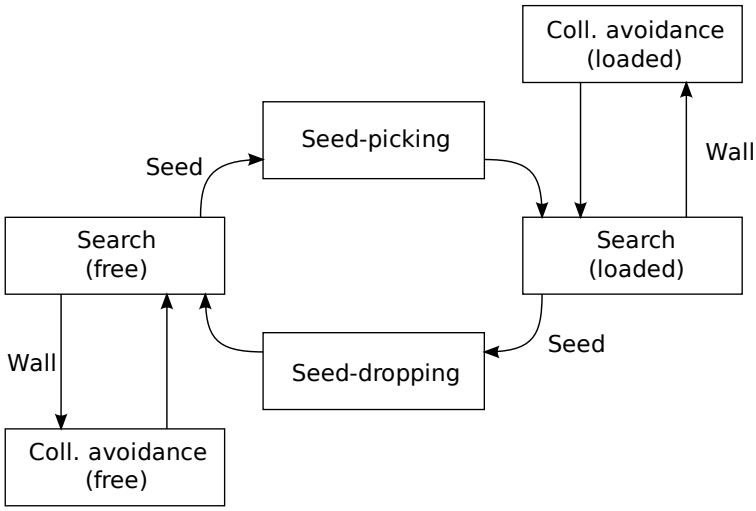


Figure 8.5. FSM representing the seed-aggregation controller. Transitions between states are deterministically triggered by sensor measurements (i.e., detection of a seed or a wall).

result in a single-aggregate configuration.

The two first consequences are due to geometry (see Fig. 8.6.). First, because the probability of decrementing an aggregate is always smaller than that of incrementing it, except for isolated seeds, aggregates tend to grow. Second, seeds that belong to an aggregate are perceived as seeds (as opposed to obstacles) only when they are at the tip of the structured aggregate; therefore, seeds are always dropped in a more or less regular line.

The third consequence is slightly less intuitive. Because large aggregates tend to be more stable than small aggregates (i.e., the probability of an aggregate being decremented is inversely proportional to its size), the aggregation process will eventually result in a single-aggregate configuration, which can be seen as the lowest energy configuration of the system.

One could perhaps imagine a situation where seeds are constantly exchanged between two aggregates of the same size, and no isolated seeds remain in the arena. In this scenario, both aggregates would have equal probabilities of being decremented and incremented; a potential deadlock in a non-optimal energy configuration. However, this situation can never occur because of the following reasons. First, the intrinsic randomness of the process prevents the system from remaining stuck in such situations; even if both aggregates have identical geometric stability, this type of 2-aggregates configuration can be seen as an *unstable fixed point* of the system. Second, the probabilistic distribution over all the possible sizes of aggregate are typically asymmetric either because of subtle non-modeled spatial effects (e.g., aggregates more or less close to each other, dependence of robots' trajectory) or deliberate design choices (e.g., in the experiment of [Martinoli *et al.* (1999)], aggregates of one seed are irreversibly removed and never generated again). These asymmetries lead to cluster instabilities for some non-negligible periods of time,

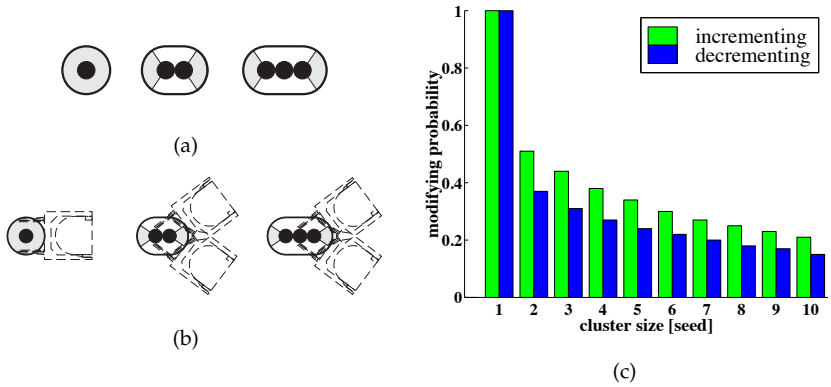
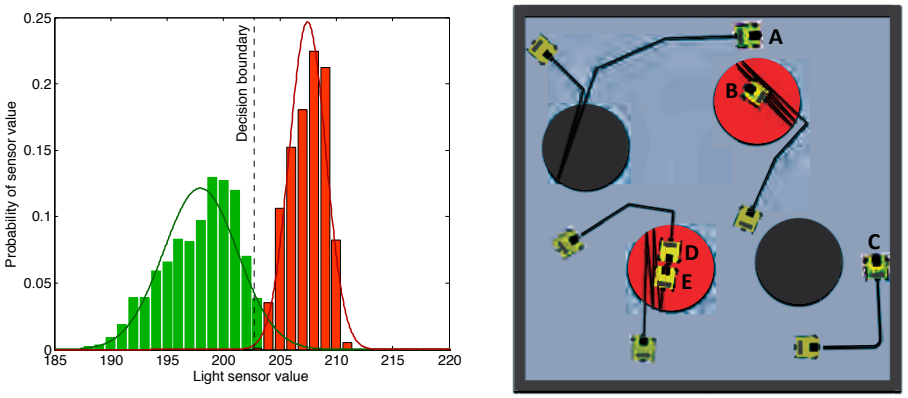


Figure 8.6. (a) Geometrical representation of the aggregate incrementing probability. The ratio between the identification perimeter (arc delimiting the grey zone) and the total detection perimeter represents the probability to increment the aggregate by one seed. (b) Geometrical representation of the aggregate decrementing probability. Due to mechanical constraints, the angle from which a seed can be successfully grasped by a robot is slightly smaller than its detection angle. (c) The numerical values of both probabilities as a function of the size of the aggregate.

are accentuated by high robots-to-object ratios, generate aggregates being more favored than other, and eventually promote a single-aggregate configuration.

8.3.2 *Collaborative Decision-Making in Presence of Noise*

The case study investigated in this section is an example of how aggregation can be used to overcome perception limitations in swarms of robots with only low-bandwidth communication. In particular, we show how physical contact (i.e., aggregation) can be used as a positive feedback mechanism for collective decision-making in a swarm of minimalist robots, namely 2-cm-sized Alice robots [Caprari and Siegwart (2005)]. The environment is populated with N_s spots which can be either *good* or *bad*, and N_r robots whose goal is to search for and destroy bad spots while preserving the good ones. Each time a spot is destroyed, another is immediately created at a different location within the environment. In our current setup, the spots are colored circles of diameter d_{spot} drawn on an arena by an overhead projector; good spots are green and bad spots are red (see Fig. 8.8.(a)). The robots are equipped with a light sensor that can be used to assess the type of a spot. However, the measurement data provided by the light sensor is noisy (see Fig. 8.7.(a)); therefore, it is possible for the robots to mistakenly trigger the destruction of a good spot. We denote $p_{w,good}$ the probability that a robot believes a good spot to be bad (false positive) and $p_{w,bad}$ the probability that a robot believes a bad spot to be good (false negative). Depending on the distribution of light sensor measurements, these probabilities can be different. Since we assume the robots to be purely reactive, they form their belief on the basis of a single measurement (made upon entering or leaving the spot), and in a purely deterministic manner, by using a simple decision threshold $t_d = (\mu_{good} + \mu_{bad})/2$, with μ_{good} and μ_{bad} the average light intensity in good and bad spots, respectively. We assume that the robots can always determine the presence (or absence) of a spot in a perfect manner.

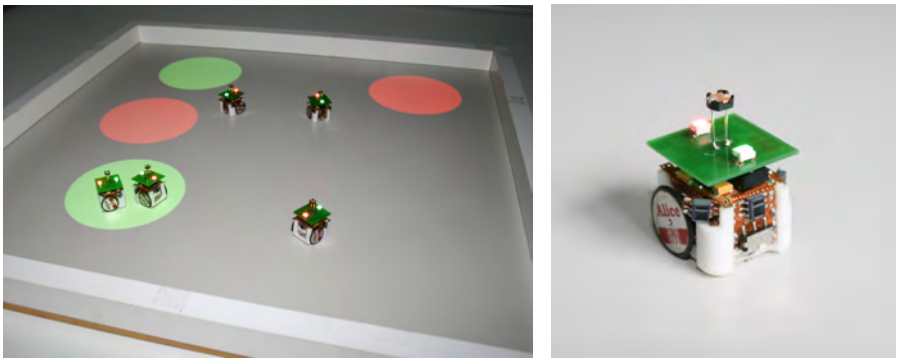


(a) (b)

Figure 8.7. (a) Histogram of light sensor measurements (1000 values) in good spots (green), and in bad spots (red). Fitted Gaussian distributions are also shown (continuous lines). Hereafter, we study the scenario with $p_{w,good} = 0.271$ and $p_{w,bad} = 0.2224$. (b) Sketch of a typical experiment with 4 spots and 5 robots for $k = 2$, i.e., two robots are required to trigger the destruction of a spot. Trajectories of the robots are denoted by black lines. Robot A explored a good spot, made one wrong decision (by performing a U-turn at the border of this spot), but eventually left the spot. Robot B is exploring a bad spot, waiting for a teammate. Robot C avoided an obstacle while exploring the environment. Robots D and E encountered each other in a bad spot, and decided to aggregate; this spot is therefore about to be destroyed, and re-created at some other location in the arena.

One can draw an analogy with different natural systems that are responsible for identifying and neutralizing pathogens in a given environment (e.g., the human immune system, or bacteria purifying environmentally polluted regions). Importantly, this task must be carried out in a reliable manner: the system must attack pathogens while preserving healthy actors of the environment. Similarly, in our case study, the environment contains two types of spots ("good" and "bad"), which differ from each other in an observable fashion. However, light intensity measurements are corrupted by both the intrinsic noise of the photocell and the lighting variations of the projector, thus making identification of the cell-type unreliable.

As mentioned earlier, collective decision-making is one way of overcoming the limitations of the individual agents in terms of sensing. The question is, how can we achieve collective decision-making without explicit communication? Aggregation allows us to solve this problem by replacing the transmission of a message by the detection of a physical presence (which can be thought also as a form of implicit communication). Here, we use local infrared beaconing as a way of discriminating between obstacles and other robots, a mechanism easily replicated at smaller length scales (e.g., by an electrical contact or pressure sensor). Then, we can exploit aggregation as an implicit communication scheme that allows the robots, uniquely through their physical presence, to share their estimate of the type of the spot they are in. When two robots encounter each other in a spot, they form an aggregate only if both believe that this spot is a bad spot; otherwise they perform obstacle avoidance, and eventually leave the spot.



(a) (b)

Figure 8.8. (a) Picture of a real experiment with 5 robots and 4 spots. (b) Close-up of an Alice 2002 robot, which has a size of 2cm×2cm×2cm and is equipped with four infrared sensors for environment sensing and communication as well as an extension board with one photocell and two colored LEDs (red and green) for tracking purposes.

One important parameter of our controller is k, which denotes the number of aggregated robots required to trigger the destruction of a spot. For $k = 1$, there is no collaboration: a single robot can destroy the spot it is exploring by itself. For $k = 2$, the spot is destroyed as soon as a robot aggregates with another robot (Fig. 8.7.(b) depicts a typical experiment with $k = 2$). For $k = 3$, an aggregate can remain in a spot for a while without triggering its destruction, which therefore introduces a further parameter $p_{leave,aggr}$, that is, the probability that a robot leaves the aggregate it is part of.

The optimal value of k depends on the difficulty of the task, i.e., the amount of noise characterizing the light sensor measurements in good and bad spots, as well as their separability. Note that even in absence of noise, i.e., when the probability of false positives and false negatives is zero ($p_{w,good} = p_{w,bad} = 0$), more than one robot may be required to trigger the destruction of a spot (e.g., when individual robots are too small or limited for carrying out the task on their own). In order to have a quantitative method of reporting system performance, we define an arbitrary metric function M in terms of the number of good and bad spots destroyed:

$$M = \frac{D_{bad}}{(D_{good})^{\alpha} + 1} \tag{8.1}$$

where D_{bad} is the number of bad spots destroyed, D_{good} is the number of good spots destroyed, and α a coefficient that may be balanced according to the penalty one wishes to associate with the destruction of a good spot; the higher the coefficient, the higher the penalty.

Given the intrinsically stochastic nature of the investigated processes, a large number of runs is required in order to obtain statistically relevant data, which results in extremely time-consuming experiments if real robots are used. We provide experimental results (see Table 8.1) that suggest the relevance of collective perception and action as a mechanism for coping with unreliable sensing at the individual level. In spite of the high variability of the results obtained with real robots ($> 100\%$ of variability on the performance metric), collaboration seems to

Table 8.1. Summarized results of two experiments (with and without collaboration) using 5 real Alice robots and 4 spots (2 of each kind). Destruction rates are given in number of spots destroyed per minute. The performance of the swarm (calculated using Eq. (8.1) with $\alpha = 2$) is two orders of magnitude higher when collaboration is introduced.

Destruction rate	No collaboration ($k = 1$)		Collaboration ($k = 2$)	
	Bad spots	*Good spots*	*Bad spots*	*Good spots*
Run 1	4.93	3.85	0.68	0.09
Run 2	5.28	2.68	0.55	0.00
Run 3	5.12	2.95	1.56	0.20

Performance	No collaboration ($k = 1$)	Collaboration ($k = 2$)
Run 1	$2.9 \cdot 10^{-2}$	2.8
Run 2	$7.0 \cdot 10^{-2}$	9.0
Run 3	$5.8 \cdot 10^{-2}$	1.76

provide a non-negligible performance gain, up to two orders of magnitude in these particular experiments using our specific metric (Equation (8.1)) with $\alpha = 2$.

8.4 Modeling Self-Organized Distributed Robotic Systems

In this section, we show how one can go beyond local heuristic reasoning by exploiting model-based approaches to design and control self-organized large-scale robotic systems. One of the main difficulties in modeling such systems, and particularly those involving aggregation and self-assembly, is the inherent randomness and complexity of the dynamical process. These challenges motivate a combination of multiple levels of abstractions, ranging from detailed, realistic, submicroscopic models up to macroscopic models, into a consistent multi-level modeling framework. On the one hand, one needs submicroscopic models that are able to capture low-level details of a robotic node and its modules (e.g., sensors, actuators, body shape). On the other hand, one is also interested in models that can yield accurate numerical predictions of collective metrics, and investigate, possibly formally, macroscopic properties such as the sizes, types, and proportions of the resulting aggregates. Multi-level modeling allows fulfillment of both requirements in a very efficient way by building up models at incrementally increasing levels of abstraction in order to capture the relevant features of the system.

Hereafter, we show how one can model, at different abstraction levels, systems composed of N_0 agents that move randomly throughout an arena of area A_{total}, and, upon collision, aggregate into clusters of different sizes and shapes. Clusters are generally not persistent because robots might leave them with a certain probability p^{leave}, which is a control parameter of the robots' behavior that can be tuned as a function of the local perception of the agent (e.g., the presence of neighbors, light intensity, etc.). The overall stability of an aggregate (i.e., the probability p^{split} that it splits up into different sub-aggregates) is a function of the number of robots in it, and their respective leaving probability p^{leave}.

In many scenarios, p^{leave} may depend on the local perception of the robot such as the number of detected neighbors [Correll and Martinoli (2007)] or their rela-

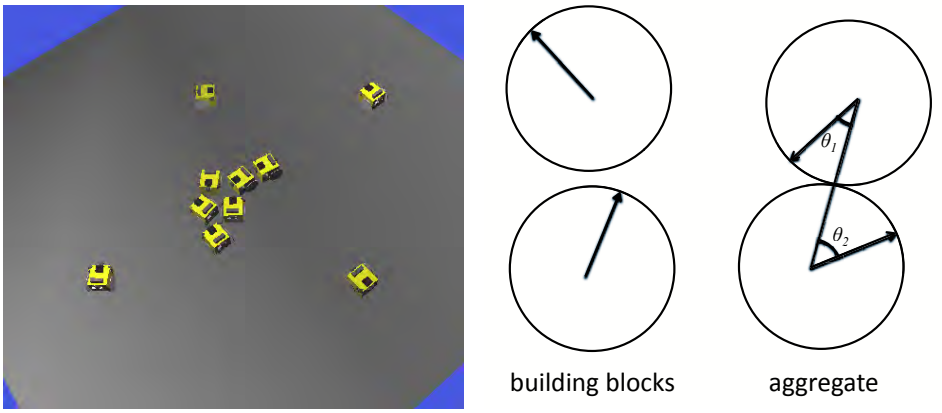
tive alignment [Mermoud *et al.* (2009)]. In such cases, p^{split} also depends on the structure and the geometry of the aggregate, which are very difficult to accurately capture at high abstraction levels. In some settings, the aggregating agents are passive objects moved by the robots, either one by one [Matthey *et al.* (2009); Martinoli *et al.* (2004)], or in groups [Beckers *et al.* (1994)]. Also, aggregates can either remain still (as it is often the case with passive objects or non-holonomic robots), or move throughout the environment, and therefore aggregate with each other just like individual agents do. All these features have profound implications on the models, especially at higher abstraction levels.

Models of distributed robotic systems can be classified into three main categories: (i) **submicroscopic models**, in which the state of each individual and its sub-components (e.g., sensors, actuators, body pose) is captured (see Sec. 8.4.1); (ii) **microscopic models**, in which the state of each individual in the system is captured, but the details about its sub-components are abstracted (see Sec. 8.4.2), and (iii) **macroscopic models**, in which all individuals in a given state are aggregated into one state variable (see Sec. 8.4.3).

Hereafter, we describe a typical multi-level modeling framework that consists of different models at different abstraction levels. The governing principle of our modeling methodology is to build the suite of models from the bottom up while conserving a consistent set of parameters that are shared at all abstraction levels (i.e., the joining and leaving probabilities, the number of agents, the arena size, etc.). Collisions are either explicitly simulated using more or less complex collision routines in spatial models, or probabilistically emulated by a geometric approximation in non-spatial models.

8.4.1 *Submicroscopic Models*

The most detailed level of modeling is provided by physics-based simulation implemented in robotic simulation engines such as *Webots* [Michel (2004)]), which accurately models each module of a robotic node (e.g., a sensor, an actuator, a transceiver), including often its nonlinear transfer functions and noise distributions. These simulations faithfully account for a subset of physical phenomena such as friction and inertia, which are considered most relevant to the dynamics of conventional mobile robots. One great advantage of using a physics engine is the fact that it also provides accurate collision detection between robots as well as between robots and obstacles. Also, they allow one to easily visualize a robot's behavior and dynamics (see Fig. 8.9.(a)). Therefore, at this level of abstraction, it is fairly easy to capture all the different aspects of aggregation with a great deal of flexibility. However, these simulations are also extremely computation- and memory-intensive, in particular when studying systems that involve large numbers of robots.



(a) (b)

Figure 8.9. (a) Screenshot of an aggregation experiment with Alice robots simulated in Webots, a physics-based mobile robotics simulation. (b) In the microscopic setting, many details, such as the shape of the robot and most physical effects, are abstracted away. Here, robots are radially symmetric bodies with preferred binding directions. The relative alignment of two aggregated robots is denoted by two angles $\theta_1 \in [0, \pi]$ and $\theta_2 \in [0, \pi]$, i.e., the bearing of each building block with respect to the other.

8.4.2 *Microscopic Models*

Even though microscopic models capture the state of each individual robot in the system, their state vector is significantly smaller than their correspondingly submicroscopic counterpart. This state reduction is typically obtained through appropriate aggregation of the state variables, which can be more or less important as a function of the desired level of detail. Hereafter, we describe two types of microscopic models: (1) a spatial agent-based model, (2) a non-spatial Monte Carlo model.

Spatial Agent-Based Model: Physics play a very important role in most self-organized systems, but many physical effects can often be neglected without impacting the particular system parameter being studied. In spatial agent-based models, the kinematics of each individual robot is still captured, but a lot of intra-node, submicroscopic details about the modules and their specific interaction with the environment (e.g., wheel slip, sensor noise, etc.) are abstracted away, resulting in significantly faster simulations. We assume that our robots are radially symmetric, with a position $\vec{x} \in \mathbb{R}^2$ of their center of mass, a velocity $\vec{v} \in \mathbb{R}^2$, an orientation θ, and a radius r (see Fig. 8.9.(b)). The environment has a finite surface area A_{tot} and has toroidal boundary conditions. Each robot R_j follows a precise trajectory determined by the kinematic laws of interest. Two robots R_1 and R_2 of radius r_1 and r_2, respectively, located at a distance d from each other, collide if and only if $d < r_1 + r_2$.

Nonspatial Monte-Carlo Model: Spatial models offer an interesting modeling framework for multi-agent systems, but they are still expensive both in terms of memory and computation. Indeed, these models store the position and the orien-

tation of each agent as well as the precise structure of each aggregate. Also, they must determine at each iteration and for each pair of agents whether a collision occurred or not. One can go even further in the process of abstracting details that are not particularly relevant to the dynamics of the main process under investigation (e.g., aggregation, self-assembly); hereafter, we describe a Monte Carlo approach, which does not capture spatiality, i.e., it does not keep track of the position and orientation of each individual. It can be considered a stochastic microscopic model that, in contrast to the macroscopic models developed later (see Sec. 8.4.3), does not rely on a mean-field approach, i.e., it does not "aggregate" discrete entities into real-valued state variables that describe *averaged* quantities. However, the model assumes that the individual behavior of each robot and that of the environment can be represented by intertwined Markov chains, i.e., the probabilistic transition from one state vector A to another state vector B depends only on the information contained in the state vector A.

The particular model that we describe hereafter assumes that agents have only one binding site, thus intrinsically limiting the size of the formed aggregates to pairs. Our model keeps track of only one property of the aggregates, that is, the relative alignment of their building blocks (see Fig. 8.9.(b)). On one hand, since our model is non-spatial, collisions are no longer deterministic, but are randomly sampled from a Poisson distribution of mean $\lambda = p^{join} N_s$ (see Eq. (8.2)). On the other hand, each aggregate resulting from these collisions is individually captured: a random relative alignment $\xi_i = (\theta_{1,i}, \theta_{2,i})$ is generated and stored in a list Ξ_a (see Algorithm 8.2). One very interesting feature of this type of models is that they store only relevant pieces of information about the aggregates, which can range from the number of building blocks to a fully-fledged graph-based representation of the aggregate's topology.

One subtlety in building non-spatial models of aggregation is to accurately capture the encountering probabilities. Here, we assume a constant encountering probability p^{join} that is determined using a geometric approximation:

$$p^{join} \sim \frac{\widehat{v}\, T\, w_d}{A_{tot}} \tag{8.2}$$

where $\widehat{v}$ is the average velocity of the robot, w_d its diameter, T the sampling time, and A_{tot} is the total area of the arena [Martinoli *et al.* (2004)]. In more complicated scenarios, one would also account for encountering probabilities that depend on the size and the geometry of the aggregates.

8.4.3 *Macroscopic Models*

The models described in the previous sections were all stochastic models, which provide a single realization of the time evolution of the system at each run, and do not scale well with the number of robots. As a result, one must usually perform a large number of computationally expensive runs in order to obtain statistically meaningful results.

Hereafter, we describe a non-spatial macroscopic model of aggregation, which allows one to overcome these limitations, but at the price of further approxima-

Algorithm 8.2 Pseudo-Code of Monte-Carlo simulation.

Algorithm 1.2 Pseudo-Code of Monte-Carlo simulation

Initialize $N_s = N_0$ and $N_{2,3,\ldots} = 0$
for all t in tspan **do**
– Sample n_c the number of collision events from a Poisson distribution of mean $\lambda = p^{join}\, N_s$
– Generate and append to Ξ_a a random vector of n_c relative alignments $\Xi_c = (\xi_1, \ldots, \xi_{n_c})$ with $\xi_i = (\theta_{1,i}, \theta_{2,i})$ and $\theta_{d,i} \sim U(0, \pi)$
– Generate a random vector $\mathbf{X}^s = (x_1^s, \ldots, x_{N_a}^s)$ with $x_i^s \sim U(0,1)$ and $N_a = \texttt{size}(\Xi_a)$
– Compute n_b the number of aggregates in Ξ_a with ξ_i such that $x_i^s < p^{leave}(\xi_i)$ and remove them from Ξ_a
– Let $N \leftarrow N + 2\, n_b - 2\, n_c$
end for

tions. Our model is a time-discrete system of difference equations, where k denotes the current iteration (time step) and kT the actual time, with T the sampling time, which is left out in the equations for the sake of simplicity, and should be chosen small enough in comparison to the time constants and dynamics of the system. We can summarize the average state transitions of each individual dynamical system, and thus keep track of the number of aggregates of size 1 to N_0. The ensemble of individuals, including structural properties, is now represented by a difference equation, which keeps track of the average number of individuals in each state[3]. Inflow and outflow of each state represent the average fluctuation between states and are given by the probability for a state transition to occur and the number of robots in other states. Using Definition 1 and following a mean-field approach, the average number $N_j(k+1)$ of aggregates of size j (with $1 < j < N_0$) at time $k+1$ is then given by the following difference equation:

$$\begin{aligned} N_j(k+1) = N_j(k) &+ f_{in,j}(P^{join}(k), P^{leave}(k), N_i(k)) \\ &- f_{out,j}(P^{join}(k), P^{leave}(k), N_i(k)) \\ &\text{with } i = 1, \ldots, N_0 \text{ and } i \neq j \end{aligned} \tag{8.3}$$

where functions $f_{in,j}$ and $f_{out,j}$ denote the inflow and the outflow of the state $N_j(k)$, i.e., the number of aggregates of size j being formed or destroyed at time k. The matrices $P^{join} = (p_{i,j}^{join})$, and $P^{leave} = (p_{i,j}^{leave})$ denote both the connectivity and the transition probabilities between states N_i, N_j, and N_{i+j}. Namely, aggregates of size i can form aggregates of size $i+j$ by joining an aggregate of size j with probability $p_{i,j}^{join}$. Inversely, aggregates of size $i+j$ can split into aggregates of

[3]We denote this model as *macro-continuous* since it keeps track of averaged, real-valued populations, by opposition to *macro-discrete* models that account for the discreteness of the population, but need to be solved using stochastic simulations (see [Gillespie (2007)] for more details about stochastic simulations, and [Evans *et al.* (2010); Mermoud *et al.* (2010)] for examples of macro-discrete models of robotic systems).

size i and j with $p_{i,j}^{leave}$. If there is no interaction between aggregates of size i and j, then $p_{i,j}^{join} = p_{i,j}^{leave} = 0$.

Therefore, the functions $f_{in,j}$ and $f_{out,j}$ may have a different number of terms depending on the properties of the aggregation process, but their form remains identical.

$$\begin{aligned} f_{in,j}(\ldots) = & \sum_{i=1}^{j-1} p_{i,j-i}^{join}(k)\, N_{j-i}(k)\, N_i(k) \\ & + \sum_{i=j+1}^{N_0} p_{i-j,j}^{leave}(k)\, N_i(k) \end{aligned} \tag{8.4}$$

$$\begin{aligned} f_{out,j}(\ldots) = & \sum_{i=1}^{N_0-j} p_{i,j}^{join}(k)\, N_i(k)\, N_j(k) \\ & + \sum_{i=1}^{j-1} p_{i,j-i}^{leave}(k)\, N_j(k) \end{aligned} \tag{8.5}$$

Terms of the form $p_{i,j}^{join}(k)\, N_i(k)\, N_j(k)$ correspond to the number $N_i(k)$ of aggregates of size i that join one aggregate of size j at time k with a probability $p_{i,j}^{join}(k)\, N_j(k)$, and form an aggregate of size $i+j$. Terms of the form $p_{i,j}^{leave}(k)\, N_{i+j}(k)$ denotes the number of aggregates of size $i+j$ that split into aggregates of size i and j at time k with probability $p_{i,j}^{leave}(k)$.

In the general case, one should take into account all possible $N_0 - 1$ pairwise combinations of aggregates that lead to the formation of an aggregate of size j. However, in many cases, one assumes that the robots remain still once aggregated (e.g., because wheeled robots are often non-holonomic); there are then only two ways of forming an aggregate of size j[4] :

$$\begin{aligned} N_{j-1} + N_1 &\rightarrow N_j \\ N_{j+1} &\rightarrow N_j + N_1 \end{aligned} \tag{8.6}$$

In such cases, we have that $p_{i,j}^{join} \neq 0$ if and only if $i = 1$ or $j = 1$. Of course, this assumption dramatically simplifies the complexity of the model, both in computational complexity and memory requirements. Figure 8.10. depicts the state transition diagram of this model.

Macroscopic models can also track properties of the aggregates other than their size (i.e., the number of building blocks), such as their geometry. In the above model, we assume that aggregates have basically no geometry. For instance, robots cannot be stuck in the middle of an aggregate, surrounded by other robots. Also, nearby aggregates never connect with each other because of one robot joining them; similarly, aggregates never split into two sub-aggregates because one robot left. These scenarios, depending on the structure of the aggregates, may happen in reality, but the particular model depicted in Fig. 8.10. does not account for them.

Yet, it is possible to capture some simple geometric features of the aggregates at the macroscopic level. To achieve that, one conventional approach is to discretize selected state variables into several sub-variables, essentially going through a state expansion process. For instance, in order to capture the alignment of pairs of building blocks, one can discretize the state variable N_2 into M sub-variables $N_{2,i}$ that

[4] We neglect the case of aggregates that merge when growing, which is a safe assumption in the case of non-crowded scenarios. Also, we assume that only one robot joins and leaves the aggregate in a given time step, if the model is time-discrete.

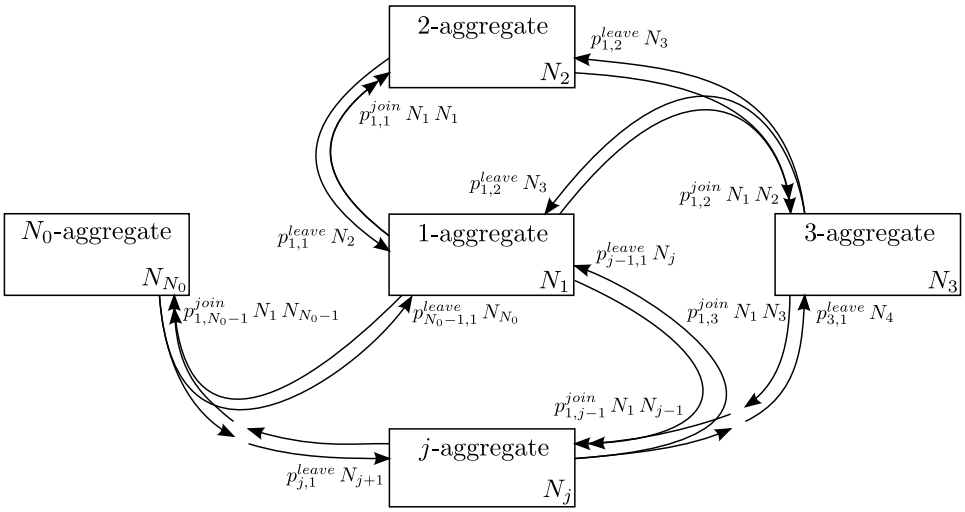


Figure 8.10. State transition diagram of the aggregation model when aggregated robots remain stationary. Only single robots (N_1) can interact with each other, and with aggregates (N_i with $i = 2, \ldots, N_0$).

denote the number of aggregates with an average alignment ξ_i with $i = 1, \ldots, M$. Obviously, such a discretization leads to a M-fold increase of the number of states, and therefore an exponential increase of the number of equations, making the model rapidly intractable. In [Mermoud *et al.* (2009)], we present in detail a model that captures alignment of building blocks at the macroscopic level by using this approach, but with an explicit limitation on the size of the aggregates to pairs.

Last, but not least, one should keep in mind some of the limitations of macroscopic models. In particular, since macroscopic models "aggregate" discrete entities into real-valued state variables that describe *averaged* quantities, both the discreteness of the entities and the potentially non-uniform behavior of the system under consideration are lost. Therefore, macroscopic models rely on an approximation, called the *ODE approximation*, which assumes that the system involves a large number of small changes, i.e., the model becomes exact if we scale the system such that the reaction rates become large and the effects of those reactions small. The validity of the approximation does not only depend on the number of robots in the systems, though; the very structure of the network and the number of interactions also plays a key role. From this perspective, discretization of state variables is generally a source of inaccuracy, because it tends to lower the reaction rates while increasing their effects.

8.5 Conclusion

Devising control strategies for distributed robotic systems is a difficult problem *per se*, and enabling scalability of these control strategies is an even harder challenge. The thrilling promises of micro-robotic systems in a broad variety of disciplines, such as biomedical engineering, pervasive and ambient information technology,

environmental engineering, and space exploration are not going to be fulfilled if we do not successfully overcome two crucial obstacles: (i) manufacturing and integrating these ultra-miniaturized robots, and (ii) devising suitable control strategies for large-scale distributed robotic systems at this size range. The former is currently the subject of various intense research efforts, but the latter remains largely unaddressed. Indeed, there is no evidence whatsoever that even cutting-edge distributed control strategies are scalable and reliable enough to be successfully applied to these future robotic platforms, which will be at the same time massively distributed and extremely miniaturized. The main reason for this situation is of course the lack of proper experimental platforms for validating those strategies targeted to massively distributed systems, but also the lack of a suitable theoretical framework for analyzing these strategies in a formal and rigorous manner. In particular, the development of an efficient modeling framework for large-scale distributed robotic systems is, in our opinion, a crucial step towards an *actual* application of those systems to *real-world* engineering problems.

Bibliography

Abbott, J. J., Nagy, Z., Beyeler, F. and Nelson, B. J. (2007). Robotics in the small - part i: microrobotics, *Ieee Robot Autom Mag* **14**, pp. 92–103.

Agassounon, W., Martinoli, A. and Easton, K. (2004). Macroscopic modeling of aggregation experiments using embodied agents in teams of constant and time-varying sizes, *Auton Robot* **17**, 2-3, pp. 163–192.

Baldassarre, G., Parisi, D. and Nolfi, S. (2006). Distributed coordination of simulated robots based on self-organization, *Artif Life* **12**, 3, pp. 289–311.

Barrenetxea, G., Ingelrest, F., Schaefer, G., Vetterli, M., Couach, O. and Parlange, M. (2008). Sensorscope: Out-of-the-box environmental monitoring, *Information Processing in Sensor Networks, 2008. IPSN '08. International Conference on* , pp. 332 – 343.

Beckers, R., Holland, O. and Deneubourg, J.-L. (1994). From local actions to global tasks: Stigmergy and collective robotics, in R. Brooks and P. Maes (eds.), *Proc. of the Fourth Workshop on Artificial Life*, pp. 181–189.

Boncheva, M., Bruzewicz, D. and Whitesides, G. (2003). Millimeter-scale self-assembly and its applications, *Pure Appl. Chem* .

Caprari, G. and Siegwart, R. (2005). Mobile micro-robots ready to use: Alice, in *Proc. of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, pp. 3295– 3300.

Christensen, A. L., O'Grady, R. and Dorigo, M. (2007). Morphology control in a multirobot system - distributed growth of specific structures using directional self-assembly, *IEEE Robotics & Automation Mag.* **14**, 4, pp. 18–25.

Cianci, C., Nembrini, J., Prorok, A. and Martinoli, A. (2008). Assembly of configurations in a networked robotic system: A case study on a reconfigurable interactive table lamp, in *Proc. of the 2008 IEEE Symp. on Swarm Intelligence (SIS 2008)*.

Correll, N. and Martinoli, A. (2006). Collective inspection of regular structures using a swarm of miniature robots, in M. H. A. Jr. and O. Khatib (eds.), *Proc. of the Ninth Int. Symp. on Experimental Robotics (ISER)*, *Springer Tracts in Advanced Robotics*, Vol. 21 (Singapore), pp. 375–385.

Correll, N. and Martinoli, A. (2007). Modeling self-organized aggregation in a swarm of miniature robots, in *Proc. of the IEEE 2007 International Conference on Robotics and Automation Workshop on Collective Behaviors inspired by Biological and Biochemical Systems*.

Correll, N. and Martinoli, A. (2009). Multirobot inspection of industrial machinery, *IEEE Robotics & Automation Mag.* **16**, 1, pp. 103 – 112.

Dong, L. and Nelson, B. J. (2007). Robotics in the small - part ii: nanorobotics, *IEEE Robotics and Automation Mag.* **14**, pp. 111–121.

Evans, C., Mermoud, G. and Martinoli, A. (2010). Comparing and modeling distributed control strategies for miniature self-assembling robots, in V. Kumar (ed.), *Proc. of the 2010 IEEE Int. Conf. on Robotics and Automation (ICRA 2010)* (Anchorage, Alaska, USA), pp. 1438 – 1445.

Garnier, S., Jost, C., Gautrais, J., Asadpour, M., Caprari, G., Jeanson, R., Grimal, A. and Theraulaz, G. (2008). The embodiment of cockroach aggregation behavior in a group of micro-robots, *Artificial Life* **14**, 4, pp. 387–408.

Gillespie, D. T. (2007). Stochastic simulation of chemical kinetics, *Annual Review of Physical Chemistry* **58**, pp. 35–55.

Groß, R. and Dorigo, M. (2008). Self-assembly at the macroscopic scale, *Proceedings of the*

IEEE **96**, 9, pp. 1490 – 1508.

Grzybowski, B. A. and Campbell, C. (2004). Complexity and dynamic self-assembly, *Chemical Engineering Science* **59**, 8-9, pp. 1667–1676.

Haken, H. (2006). *Information and self-organization: a macroscopic approach to complex systems*, 3rd edn., Springer series in synergetics (Springer, Berlin), ISBN 3540330216 (hbk.).

Halloy, J., Sempo, G., Caprari, G., Rivault, C., Asadpour, M., Tache, F., Said, I., Durier, V., Canonge, S., Ame, J. M., Detrain, C., Correll, N., Martinoli, A., Mondada, F., Siegwart, R. and Deneubourg, J.-L. (2007). Social integration of robots into groups of cockroaches to control self-organized choices, *Science* **318**, pp. 1155–1158.

Howard, A., Parker, L. and Sukhatme, G. (2006a). Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection, *Int. J. Robotics Research* **25**, 5-6, pp. 431–448.

Howard, A., Parker, L. and Sukhatme, G. (2006b). The sdr experience: Experiments with a large-scale heterogeneous mobile robot team, *Experimental Robotics IX* **21**, pp. 121–130.

Klavins, E., Ghrist, R. and Lipsky, D. (2006). A grammatical approach to self-organizing robotic systems, *IEEE Trans. Automat. Contr.* **51**, 6, pp. 949–962.

Lochmatter, T., Roduit, P., Cianci, C., Correll, N., Jacot, J. and Martinoli, A. (2008). Swistrack - a flexible open source tracking software for multi-agent systems, in *Proc. of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2008)*, pp. 4004–4010.

Mamei, M. and Zambonelli, F. (2007). Pervasive pheromone-based interaction with rfid tags, *Transactions on Autonomous and Adaptive Systems (TAAS)* **2**, 2.

Martinoli, A., Easton, K. and Agassounon, W. (2004). Modeling swarm robotic systems: A case study in collaborative distributed manipulation, *Int. J. Robot Res.* **23**, 4-5, pp. 415–436.

Martinoli, A., Ijspeert, A. and Mondada, F. (1999). Understanding collective aggregation mechanisms: From probabilistic modelling to experiments with real robots, *Robotics and Autonomous Systems* **29**, 1, pp. 51–63.

Matthey, L., Berman, S. and Kumar, V. (2009). Stochastic strategies for a swarm robotic assembly system, *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on* , pp. 1953–1958.

Mermoud, G., Brugger, J. and Martinoli, A. (2009). Towards multi-level modeling of self-assembling intelligent micro-systems, *AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems* **1**, pp. 89–96.

Mermoud, G., Matthey, L., Evans, C. and Martinoli, A. (2010). Aggregation-mediated collective perception and action in a swarm of miniature robots, in M. Luck, S. Sen, W. van der Hoewk and G. Kaminka (eds.), *Proc. of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)* (Toronto, Canada), pp. 599–606.

Michael, N., Fink, J. and Kumar, V. (2008). Experimental testbed for large multirobot teams, *IEEE Robotics and Automation Mag.* **15**, 1, pp. 53 – 61.

Michel, O. (2004). Webots: Professional mobile robot simulation, *Journal of Advanced Robotics Systems* **1**, 1, pp. 39–42.

Milutinovic, D. and Lima, P. (2006). Modeling and optimal centralized control of a large-size robotic population, *Robotics, IEEE Transactions on* **22**, 6, pp. 1280–1285.

Miyashita, S., Kessler, M. and Lungarella, M. (2008). How morphology affects self-assembly in a stochastic modular robot, *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on* , pp. 3533 – 3538.

Nagpal, R., Zambonelli, F., Sirer, E., Chaouchi, H. and Smirnov, M. (2006). Interdisciplinary research: roles for self-organization, *IEEE Intelligent Systems* **21**, 2, pp. 50–58.

Nagy, Z., Oung, R., Abbott, J. and Nelson, B. (2008). Experimental investigation of magnetic self-assembly for swallowable modular robots, *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on* , pp. 1915–1920.

Parrish, J. and Edelstein-Keshet, L. (1999). Complexity, pattern, and evolutionary trade-offs

in animal aggregation, *Science* **284**, 5411, pp. 99–101.

Pfeifer, R., Lungarella, M. and Iida, F. (2007). Self-organization, embodiment, and biologically inspired robotics, *Science* **318**, 5853, pp. 1088–1093.

Rentschler, M., Platt, S., Berg, K., Dumpert, J., Oleynikov, D. and Farritor, S. (2008). Miniature in vivo robots for remote and harsh environments, *IEEE Trans. on Information Technology in Biomedicine* **12**, 1, pp. 66–75.

Song, P. and Kumar, V. (2002). A potential field based approach to multi-robot manipulation, in *Proc. of the 2002 IEEE Int. Conf. on Robotics and Automation (ICRA 2002)*, Vol. 2 (IEEE, Washington DC, USA), pp. 1217–1222.

Theraulaz, G. and Bonabeau, E. (1999). A brief history of stigmergy, *Artificial Life* **5**, 2, pp. 97–116.

Werfel, J., Bar-Yam, Y., Rus, D. and Nagpal, R. (2006). Distributed construction by mobile robots with enhanced building blocks, *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on* , pp. 2787 – 2794.

Whitesides, G. and Grzybowski, B. A. (2002). Self-assembly at all scales, *Science* **295**, 5564, pp. 2418–2421.

Woern, H., Szymanski, M. and Seyfried, J. (2006). The i-swarm project, *Robot and Human Interactive Communication, 2006. ROMAN 2006. The 15th IEEE International Symposium on* , pp. 492 – 496.