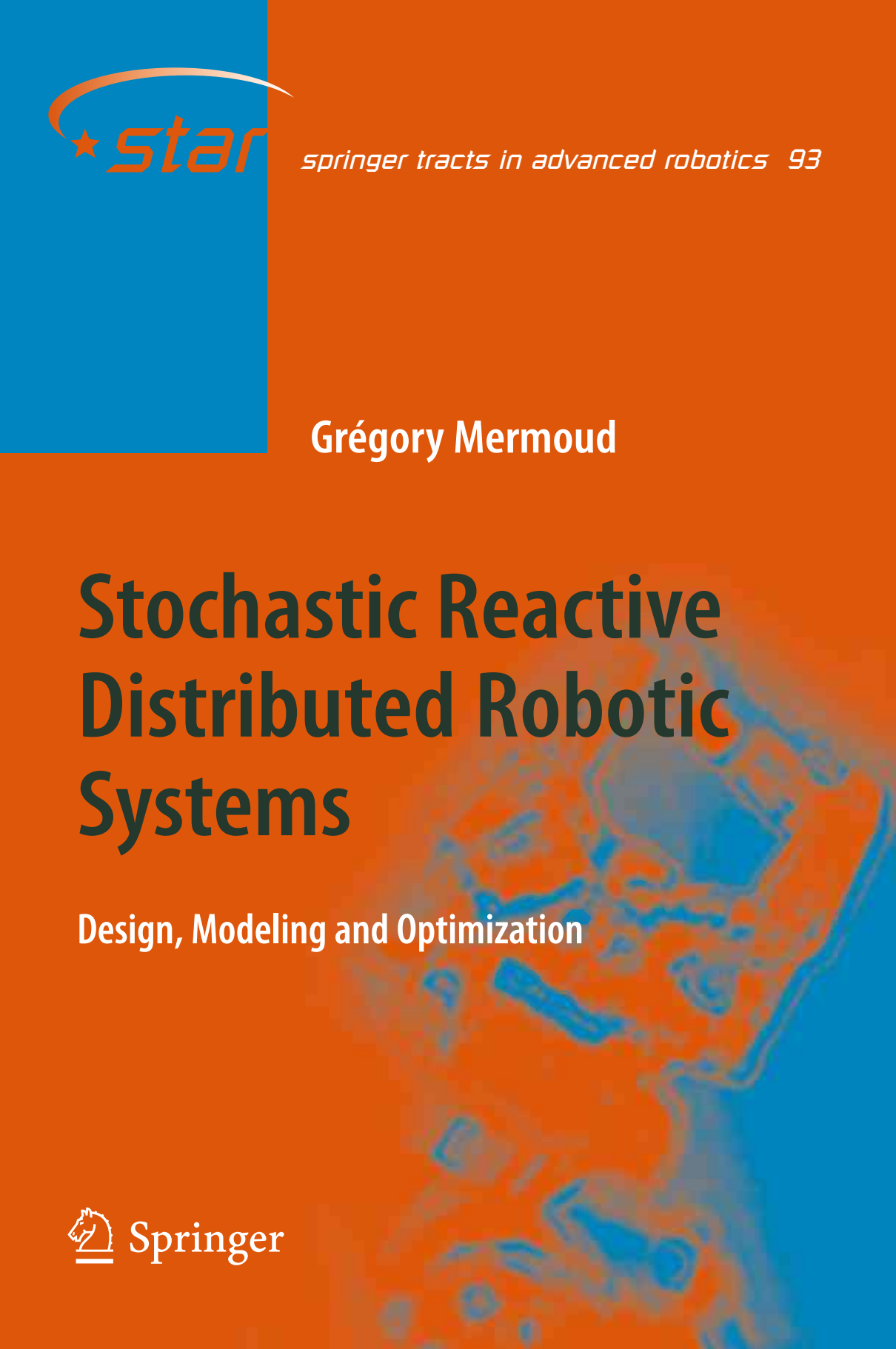
springer tracts in advanced robotics 93

Grégory Mermoud

Stochastic Reactive Distributed Robotic Systems

Design, Modeling and Optimization



Springer

Springer Tracts in Advanced Robotics 93

Editors

Prof. Bruno Siciliano
Dipartimento di Ingegneria Elettrica
e Tecnologie dell'Informazione
Università degli Studi di Napoli
Federico II
Via Claudio 21, 80125 Napoli
Italy
E-mail: siciliano@unina.it

Prof. Oussama Khatib
Artificial Intelligence Laboratory
Department of Computer Science
Stanford University
Stanford, CA 94305-9010
USA
E-mail: khatib@cs.stanford.edu

For further volumes:
http://www.springer.com/series/5208

STAR (Springer Tracts in Advanced Robotics) has been promoted under the auspices of EURON (European Robotics Research Network)

Grégory Mermoud

Stochastic Reactive Distributed Robotic Systems

Design, Modeling and Optimization

Springer

Grégory Mermoud
Cisco Systems
Av. des Uttins 5
1180 Rolle
Switzerland
gregory.mermoud@a3.epfl.ch

ISSN 1610-7438 ISSN 1610-742X (electronic)
ISBN 978-3-319-02608-4 ISBN 978-3-319-02609-1 (eBook)
DOI 10.1007/978-3-319-02609-1
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013949437

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Imagination is more important than knowledge.
Knowledge is limited; imagination embraces the entire world.
—Albert Einstein, 1879–1955

Every day you may make progress. Every step may be fruitful. Yet there will stretch out before you an ever-lengthening, ever-ascending, ever-improving path. You know you will never get to the end of the journey. But this, so far from discouraging, only adds to the joy and glory of the climb.
—Sir Winston Churchill, 1874–1965

À mes parents...

Acknowledgements

Success is the ability to go from one failure to another with no loss of enthusiasm.
— Sir Winston Churchill (1874 - 1965)

Getting a PhD is undoubtedly a great success. However, in hindsight, I can only recognize how deeply right Churchill was. The true challenge of such an endeavor lies in the ability to maintain a positive and enthusiast mindset in spite of the numerous difficulties and failures that one inevitably faces on the way. I am really proud that I could actually live by this quote, but this was made possible only by the support of many people. This dissertation is dedicated to each one of them.

First and foremost, I would like to thank Prof. Alcherio Martinoli for his guidance and unconditional support during this thesis. His formidable enthusiasm for distributed intelligent systems is contagious, and even four years of research did not immunize me against it. Alcherio has become a true "academic father" to me, and his influence on this work is considerable. I would also like to thank Prof. Juergen Brugger for accepting to co-supervise this thesis. Without his invaluable expertise, I could not have explored the profound connection between robotics and micro-engineering. His extraordinary creativity—very few people can come up with as many ideas as he does in a typical brainstorming session—inspired me both as a researcher and as an engineer.

Many people have been instrumental in shaping up this work, in particular my close collaborators and co-authors: Massimo Mastrangeli, who carefully revised this manuscript and contributed to many results and findings reported here; Loïc Matthey, who convinced me of the relevance of the Chemical Reaction Network formalism; Utkarsh Upadhyay, who nicely complemented my lack of mathematical rigor in several occasions and contributed to several theoretical developments of this work; Ezequiel di Mario, who developed the Lily modules and a novel trajectory-based calibration procedure; Chris Evans, who helped me tame the Alice robot, among other things; and Emmanuel Droz, for his invaluable support in mechatronic

development. I would also like to thank Loïc Jacot-Descombes, Dr. Maurizio Gullo, Jonas Wienen, Dr. Helmut Knapp, Laurent Sciboz, and all the team of the SELFSYS project. Furthermore, I am obliged to several researchers for their insightful scientific advice during my graduate studies: Prof. Nikolaus Correll, Dr. Vahid Fakhfouri, Dr. Jeroen Steen, Dr. Cristina Martin-Olmos, and Dr. Pierre Roduit.

Teaching has been an important and very enjoyable part of my graduate studies. I am grateful to Stefano Pennese, Loïc Matthey, Harsha Umesh Babu, Chris Evans, Ludovic Fardel, Elsa le Caignec, Ezequiel di Mario, Utkarsh Upadhyay, Hu Hainan, and Nicolas Sommer for their enthusiasm and hard work under my supervision.

I could not stress enough how important my friends have been during my doctoral studies. I would not dare attempt to cite them all, as I would risk forgetting some of them. Still, I would like to thank my old friends from the Lycée-Collége des Creusets, who are still there for me more than 10 years after we graduated; the hilarious people from the John's Club, with whom I had some much fun; my fellow musicians, who share one of my greatest passions; my current and former colleagues from SWIS, LMIS1, and DISAL, who made my working life so cheerful; my numerous roommates, some of which were great housekeepers, and some others were excellent party throwers; and all of the many people who contributed one way or the other to my well-being during this work.

I would like to thank the members of the jury who kindly accepted to review this manuscript and provided me with valuable feedback: Prof. Kay Severin, Prof. Auke Ijspeert, Prof. M. Ani Hsieh, and Prof. Jean-Louis Deneubourg.

My deepest gratitude goes to my better half, Corinne, who shares my life and my dreams. Writing a thesis is a very demanding process, and she has been an inexhaustible source of energy.

Finalement, je souhaite remercier ma famille: ma sœur, pour son grand cœur et sa bonne humeur communicative, et mes parents, pour leur soutien inconditionnel qui m'a toujours permis de me consacrer à la réalisation de mes rêves. Ce travail n'aurait probablement jamais vu le jour sans cet esprit de liberté et de passion qu'ils m'ont transmis au fil des ans.

Lausanne, 26, June 2013 Grégory Mermoud

Contents

1

Introduction

Qui règne et qui gouverne dans la cité? Où se cache la tête ou l'esprit, d'où émanent les ordres qui ne sont jamais discuté?
— Maurice Maeterlinck (1862–1949)
in *La Vie des Fourmis* (1930)

The idea that simple and resource-constrained individuals may cooperate so closely as to solve problems that are beyond the grasp of any single individual stemmed from the work of W.M. Wheeler about the social behavior of ants, only one century ago [1]. Considering that social insects such as ants, bees, and termites have co-existed with us since our apparition about 200 000 years ago[1], the recency of this idea may come as a surprise; in any case, it certainly pinpoints the counter-intuitive—and fascinating—nature of collective phenomena.

Take a system as simple as a simple pendulum—a weight suspended from a pivot by a rigid rod so that the whole can swing freely (Figure 1.1). The behavior of this system is rather intuitive to the human observer, in particular owing to its periodic nature. For instance, one could easily predict the weight's trajectory and grab it while it is swinging. Now, let us consider a double pendulum, that is, two simple pendulums attached to each other. This single, incremental change causes a dramatic difference in the system's behavior, leading in particular to the emergence of chaotic patterns, which preclude any long-term prediction of the weight's position—even by using extremely sophisticated simulations.

The essential property of the double pendulum that makes it so difficult to apprehend for the human observer arises from the presence of two *interacting* lower-level components whose trajectories in state space are coupled by positive and negative feedbacks, which may lead in turn to the amplification or the damping, respectively, of external perturbations. In the context of the double pendulum, the weights may exert a force on each other that is alternately in the opposite or the same direction as the gravitational force,

[1] In the most recent version of the timeline of evolution, earliest bees appear about 100 million years ago and the first ants about 80 million years ago. The first fossil records of the genus *Homo* date back to 2 million years before our era [2], and anatomically modern humans appeared in Africa about 200 000 years ago.

G. Mermoud, *Stochastic Reactive Distributed Robotic Systems*,
Springer Tracts in Advanced Robotics 93,
DOI: 10.1007/978-3-319-02609-1_1,

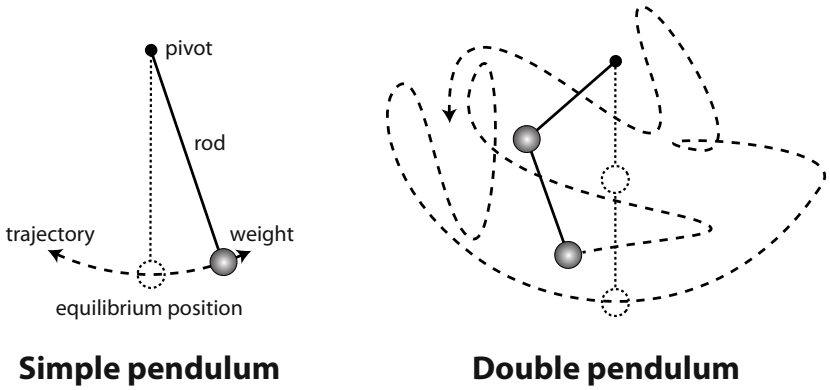


Fig. 1.1 Schematic illustration of a simple and double pendulum as well as a sketch of their typical trajectories. Note how the coupling of two simple systems can lead to complicated patterns.

thereby leading to the amplification or the damping of their respective oscillations. In chaotic regimes, these feedbacks will lead to an exponential increase in time of any initial error in the estimation of the pendulum's state.

As a matter of fact, feedbacks are not the only source of complexity in collective systems. A gas for instance can be described as a large number of rapidly moving particles that are constantly colliding with each other and with the container's walls. Because the system is isolated and collision times are assumed to be much shorter than the time between two successive collisions, no significant feedback mechanism is present in the system. Even in these circumstances, an accurate description—and *a fortiorti* an accurate prediction—of its dynamics is impossible since it would require to meticulously track the positions and velocities of the multitude of particles, and updating their momenta appropriately whenever they collide with each other.

In the 19th century, James Clerk Maxwell and Ludwig Boltzmann independently tackled this problem by formulating the first statistical description of a physical system—the famous Maxwell-Boltzmann distribution of molecular velocities in a gas. Rather than attempting to track each particle individually, they adopted a *macroscopic* approach in which the system is described in terms of a probability distribution over the statistical ensemble of microstates[2], which are too numerous to be explicitly enumerated. Of course, such an approach is suitable insofar as one is not interested in the position and velocity of any individual particle. As we shall see, this revolutionary

[2] Each microstate describes one possible arrangement of the system in terms of the positions and momenta of its constituent particles—even the slightest difference in position or momentum of a single particle would correspond to a different microstate.

approach based on statistical modeling is at the heart of the methodology developed in this thesis.

When one describes a collective system in such macroscopic terms, a fundamental question arises naturally: How many microstates correspond to a given macroscopic description? In other words, what is the *order* of the system in this configuration? Statistical entropy S is a direct measure of this quantity since $S = k_B \ln \Omega$ where k_B is the Boltzmann constant and Ω is the number of microstates corresponding to the same set of macroscopic variables (or macrostate). From a purely statistical standpoint, if all microstates are equiprobable—and there is no reason why it should not be the case—the system shall spontaneously tend towards the macrostate that involves the most microstates, that is, the macrostate of maximal statistical entropy, which we refer to as the *equilibrium state*. In the case of a gas, the Maxwell-Boltzmann distribution is precisely the one that maximizes the entropy of the system. As a matter of fact, this finding is a mere statistical interpretation of the second law of thermodynamics, which states that any isolated system shall become more disordered[3].

At this point, one may wonder how the notion of entropy may help in answering Maeterlinck's questions about insect societies. On the contrary, when ants form trails to optimize foraging or termites build complex tunnels in their nest for the purpose of thermoregulation, they form structures and patterns that tend to increase the system's order. There is no contradiction with the second law of thermodynamics, though, since insect colonies are not isolated systems: they exchange energy and matter with their environment, notably by foraging for food, dissipating heat, and expelling waste. However, this argument indicates that, during its life, the colony operates *far from equilibrium* by dissipating energy. In 1977, Prigogine won the Nobel Prize of Chemistry for proposing a theory of dissipative structures, that is, systems that trade energy and matter with their environment for an increase of their internal order—a phenomenon now referred to as *self-organization*.

Does it mean that the *spontaneous* formation of equilibrium structures or patterns, which we refer to as Self-Assembly (SA), is impossible? In supramolecular chemistry, highly ordered structures such as macrocycles form spontaneously at equilibrium without contradicting the second law of thermodynamics. Indeed, supramolecular complexes are energetically favorable, and their formation therefore dissipates some heat, which in turn increases the overall system's entropy by some larger amount than the decrease in entropy due to the structure formation. Similar arguments apply to self-assembling systems at larger scales. For instance, the spontaneous ordering of permanents magnets into chains or lattices is always accompanied by a dissipation of energy (e.g., due to mechanical friction and stress). In both cases, the ordering is *local*, and results from energetically favorable interactions that lead to more disorderly dispersal of the total energy of the system at a *global*

[3] The exact thermodynamical interpretation is that the system becomes more *energetically* disordered [3].

level. As a result, one can control, to some extent, the formed structures by properly tailoring the components' interactions, and adjusting the system's total energy.

This historical snapshot intends to illustrate the system- and scale-independent nature of self-organization. It might be then less surprising to observe the burgeoning of self-organized paradigms in disciplines as varied as computer science, biology, social sciences, arts, material science, and robotics.

1.1 Objectives of This Thesis

Self-organized and self-assembled systems may have intrigued scientists, but they have caused awe and envy among engineers. Indeed, these systems are robust to individual node failures (e.g., the death of an ant does not endanger the colony), they adapt to environmental changes, they are often nearly defect-free, and they may exhibit self-healing capabilities. All these properties are highly desirable for engineered systems, especially in robotics.

One objective of the present work is to leverage model-based methodologies for *engineering* self-organized and self-assembled systems. Ultimately, these methodologies shall lead to more robust, dependable, and inexpensive distributed systems. In the context of this dissertation, "distributed system" is an umbrella term for all engineered systems composed of multiple interacting lower-level components capable of collectively performing a task. Occasionally, we may use the term "collective systems" for denoting natural systems of the same type. This work focuses on a class of distributed systems that involve a specific type of constitutive components, which we refer to as Smart Minimal Particles (SMPs) [4]:

1. **Smart:** they are endowed with an internal state, which is often of physical nature (e.g., molecular conformation, energy level, biochemical composition, interaction configuration), even though it may also be logical in the case of robotic entities. State transitions result from interactions with other components and their environment, possibly in a time-delayed fashion. These interactions are characterized by an exchange of information, an applied force, or a physical contact.
2. **Minimal:** they are endowed with minimal resources in terms of sensing, actuation, computation, and communication. In particular, they are not capable of reasoning or anticipation, and they do not maintain an internal representation of their environment. Their dynamics are often noisy, partially unknown, or plainly non-deterministic.
3. **Particle:** the constitutive components are physically embodied, that is, they have a physical extent, and their physico-chemical and geometrical properties are an integral part of the characterization of the collective dynamics.

If not stated otherwise, the plural acronym SMPs designates an ensemble of such components. Examples of SMPs include but are not limited to supramolecular self-assembling systems [5], gene regulatory networks [6], bacteria and insects societies [7, 8], microscale self-assembling systems [9], and multi-robot systems [10]. Even though some of these systems cannot be strictly considered as engineered, they can all be tailored and controlled to some extent [11–13]. In this dissertation, we restrict our discussion to the two last categories, but our methodologies are in principle applicable to the whole spectrum of SMPs, as suggested by previous and on-going work of other researchers. Conversely, systems that exploit planning, mapping, deliberative strategies, and reasoning are not reactive, and cannot be considered as SMPs. Similarly, robotic systems endowed with complex and advanced sensing, actuation, computation, and communication capabilities, in particular if they exhibit a very strong robustness to noise and external perturbations, do not lie within the spectrum defined by SMPs.

Importantly, we hypothesize that SMPs form the *convergence* locus of two other classes of collective systems: (i) molecules and very small interacting particles, generally studied using the tools of *statistical mechanics*, and (ii) large robots with powerful computing, sensing, actuation, and communication capabilities, generally approached using *control theory*. This thesis attempts to bridge the gap between these two disciplines by tackling several key challenges. First, SMPs exhibit generally too much complexity and richness at the individual node level for being directly studied using the tools of statistical mechanics; at the same time, their stochasticity as well as their limited resources in terms of sensing, actuation and computation render tools of modern control theory inapplicable—or applicable at the expense of profound adaptations. Second, the systems studied in this thesis generally involve too few individual components for neglecting fluctuations, yet too many for dispensing with statistical modeling. Last, SMPs are characterized by the presence of discrete, intermittent, stochastic interactions—as opposed to the continuous, deterministic interaction that the two weights of a double pendulum exert on each other—that make the formalism of dynamical systems difficult to apply in this context.

1.2 Outline

The manuscript is composed of three distinct parts and a concluding chapter, which are described in details hereafter.

Part I aims at providing the reader with the pieces of information that are essential to the understanding of this manuscript such as references to prior work, theoretical concepts, or terminology definition. Chapter 2 combines an in-depth review of the state-of-the-art of the different topics of interest in the context of this thesis and an introduction to the core concepts used throughout the manuscript. Chapter 3 introduces the three

main experimental platforms as well as the computational tools used in this work. Chapter 4 describes the suite of five case studies that support the theoretical and methodological findings reported in this thesis.

Part II focuses entirely on the modeling of SMPs. Chapter 5 aims to discuss the theoretical fundamentals of modeling, and the specific role of stochastic processes for simplifying the description of SMPs, in particular in the context of multi-level modeling. Chapter 6 and Chapter 7 discuss respectively the construction and the calibration of models at multiple abstraction levels. Chapter 8 explores the validity of these models on a per assumption basis. Finally, Chapter 9 introduces a more systematic and generic methodology for constructing computationally efficient models of SMPs in an automated fashion.

Part III is dedicated to the model-based design and optimization of SMPs. Chapter 10 discusses the trade-offs that exist between various design choices with a specific emphasis on resource-constrained, miniature robots. Chapter 11 shows how one can use computationally efficient models to optimize design and control parameters of the system. Finally, Chapter 12 describes how generic, automatically generated models can be used for the real-time control of SMPs.

Finally, Chapter 13 summarizes the most important contributions and findings of this thesis and discusses the most promising lines of research for the future.

Part I

Preliminaries

2

Background

If I have seen a little further it is by standing on the shoulders of Giants.
— Sir Isaac Newton (1642–1727)

This chapter has two important objectives. First, it provides a thorough review of the literature relevant to this thesis. Second, it provides the reader with a brief reminder of the most important concepts used throughout the manuscript. As outlined in Chapter 1, we restrict our discussion to engineered distributed systems that lie between the micrometer and the centimeter scale, which we denote as SMPs. We suggest in particular that SMPs represent the natural convergence locus of two opposing tendencies observed in Micro-Electro-Mechanical System (MEMS) technology and distributed robotics. On one side of the continuum, an important body of precision manufacturing's research aims at producing small systems of increasing complexity and capability [9, 14–16]. On the other side of the continuum, many robotic researchers attempt to create increasingly robust, dependable, and inexpensive systems by transferring complexity from individual robots to collective dynamics, generally leveraging self-organization mechanisms and environmental surrogates (e.g., fluid flows for locomotion [17], stigmergy [18], templating [19]). However, as we shall see in this chapter, this convergence holds both great promises in terms of applications and considerable challenges in terms of modeling and design.

2.1 Fundamentals of Distributed Robotics

The key idea behind collective systems at large is to share labor among multiple agents. Social insects such as ants, termites, and bees, vertebrates such as migratory birds, lions and wolves, and, even more clearly, humans have been carrying out a number of tasks (e.g., preying, foraging, building, migrating) in a distributed manner ever since they existed. The motivation for doing so often comes from the need for *efficiency*, *robustness*, *scalability* and *adaptivity*, which are key concept in distributed robotics.

G. Mermoud, *Stochastic Reactive Distributed Robotic Systems*,
Springer Tracts in Advanced Robotics 93,
DOI: 10.1007/978-3-319-02609-1_2,

Efficiency is the property of a system to minimize the use of some resource such as energy or task duration while preserving its performance.

Robustness is the property of a system to be resilient to failures at the individual level or to unexpectedly adverse conditions of operation.

Scalability is the property of a distributed system to preserve or improve its performance as the number of its lower-level components increases.

Adaptivity is the property of a system to dynamically adjust its behavior to changes in environmental conditions in order to preserve its performance.

Before discussing how these concepts apply in the narrower context of robotics, we shall define more precisely the notion of *robot*, as it is used in this work.

A **robot** is a physical object whose properties and dynamics depend on an internal, logical state, which in turn depends on current or past external stimuli.

This definition is extremely general, and it encompasses a broad variety of agents, ranging from living cells to humanoid robots. Even agents without silicon-based logic can be considered as robots[1]. Because it does not impose any restriction on the sensing, actuation, computation, and communication capabilities of the agents, this definition is broader than that of SMP.

The need for multi-robot systems generally arises in scenarios that involve robots that are several orders of magnitude smaller than the environment in which they operate. 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 [21]. In order to accomplish monitoring and inspection tasks in such scenarios, large-scale[2] systems composed of hundreds or more individual nodes must be deployed [22]. This is typically the case of tasks that require highly miniaturized individual robots, like distributed inspection and sensing of industrial machinery [23], commercial pollination [24], or the human body [25].

Another important motivation for the use of multi-robot systems is their flexibility, which is especially important in scenarios such as the deployment of communication networks in disaster areas [26], the monitoring of urban environments [27], or the odor-based localization of victims in natural disasters [28]. Because of its inherent robustness, distributed robotics is increasingly envisioned as the solution to the operational challenges posed by

[1] The very notion of computation does not make any assumption in terms of substrate [20].

[2] The notion of scale pertains to the number of robots in the system here, and not to their size.

the harsh and adverse conditions encountered in scenarios such as nuclear disasters, space exploration, or battlefield support.

2.1.1 Core Concepts and Principles

Distributed robotics is a multidisciplinary field that encompasses mechatronic design, control theory, machine learning, applied physics, and statistical modeling. The reason for this richness lies in the intrinsic features of distributed robotic systems: hybridity, embodiment, mobility, complexity, and stochasticity. Hereafter, we discuss all these features in detail, as well as a few other common properties of distributed robotic systems.

Hybridity

Hybridity is the property of a robot to have a state with both a *continuous* and a *discrete* part, generally associated to physical properties (e.g., position in space, geometrical conformation, etc.) and logical properties (e.g., controller's state), respectively. While hybridity mostly pertains to the control of robotic systems, it has also implications in modeling, as we shall see later.

Embodiment

Embodiment is the property of a robot to have the physical body with its own geometric and physical properties[3]. Embodiment must be carefully accounted for so that robots can successfully operate in the physical world. As a result, all robotic systems heavily depend on a proper mechanical design; yet, the notion of embodiment pertains to all stages of the development of a robotic system, including control, design and modeling of the said system [29]. Some authors even suggest that the lack of embodiment precludes the development of intelligence altogether [30].

Mobility

Mobility designates the ability of a robot to move in space. The space of operation of a robot may be different from the regular, three-dimensional physical space: for instance, a modular robot may remain fixed in space, but reconfigure itself into some other structure, thereby *moving* in the space of its conformations. In this work, we distinguish between *controlled* mobility, which results from the robot's actuators, and *parasitic* mobility [31], which arises from the more or less controlled harnessing of environmental forces (e.g., wind, fluid flows, thermal agitation) or other agents' mobility (e.g., pollution sensors mounted on buses or taxis).

[3] As opposed to the assumption that individual robots are simple points in space, which is often found in the literature of distributed robotics.

Stochasticity

Stochasticity is the property of a system or a process to be intrinsically non-deterministic, that is, its subsequent state is determined by its current state (and, possibly, by some or all of its previous states) *and* by a random element. Stochasticity plays a key role in robotics since the behavior of a robot is never fully deterministic; both sensing and actuation are subject to noise, communication channels are intrinsically stochastic[4], and even computation, which is in principle the only fully deterministic component of the robot's behavior, may purposefully exhibit a stochastic component (i.e., probabilistic controllers).

Stochasticity is even more important in the context of distributed systems, whose behavior is largely determined by the interactions of the individual robots, which are essentially stochastic since they are mediated by mutual sensing, actuation, and communication—all subject to noise. As a result, these systems almost systematically need to be described using statistical models (also called probabilistic models), which can be then simulated using both deterministic and stochastic algorithms (see Chapter 5.4 for further details about modeling methods).

Complexity

The notion of complexity is often misunderstood; indeed, even though the words *complicated* and *complex* sound quite similar, their meaning is different. On the one hand, a complicated system is one that is difficult to understand and analyze because of its intricacy, the number and the variety of its constitutive components (the obscurity of its documentation); large pieces of software are an almost paradigmatic example of complicated entities. On the other hand, a complex system is one that exhibits unpredictable, yet not necessarily stochastic, dynamics[5]. Interestingly, simple systems can be complex, as exemplified by the double pendulum (Figure 1.1) or the three-body problem [32], which lead to the formulation of chaos theory. More generally, any system that involves multiple interacting lower-level components may, in principle, become complex under certain regimes. Such systems are almost systematically characterized by the presence of positive and negative feedbacks as well as non-linear response to (external and internal) perturbations.

[4] Owing to the remarkable developments of network coding, some communication channels offer such reliability that one might tempted to consider them as nearly deterministic. While this assumption might be acceptable in some settings (e.g., high-level information technology), most communication channels used in robotics still exhibit a strong stochasticity (e.g., packet loss, packet corruption, and communication delays).

[5] However, complex systems are often probabilistically predictable—this claim is one of the most important of this thesis.

However, they may not necessarily exhibit specific properties such as *chaos*[6] or *emergence*[7].

Adaptivity

Most robotic systems are designed to carry out a specific task, that is, a sequence of actions to be performed in some well-defined conditions. However, a desirable property of such systems is that they can deal with unknown conditions, and dynamically adjust their behavior accordingly. One may achieve adaptivity using *learning*, where each individual robot explicitly and dynamically adjust its behavior, generally by reinforcement of the most rewarding one. Alternatively, one may leverage self-organization and SA mechanisms such that the system exhibits adaptivity at the collective level. This latter approach is generally preferred over the former in distributed robotic systems since it favors simplicity at the individual level.

Coordination

When designing distributed robotic systems, one must account for the presence of other robots trying to carry out the same task. Indeed, as far as scalability is concerned, even for simple tasks such as foraging, adding robots to a task will generally not improve the performance of the system. In such cases, one needs to *coordinate* the different robots such that they do not interfere with each other, or, even better, such that they collaborate with each other in a way that improves the performance of the group.

2.1.2 Control

The notion of control is very general, and may designate any approach intended to constrain the behavior of a dynamical system. In particular, *control theory* is an interdisciplinary branch of engineering and mathematics, whose usual objective is to calculate solutions for the proper corrective action from a controller that result in system stability, that is, the system will hold a given set point, and not oscillate around it [35]. Control theory generally operates within the strict mathematical framework of dynamical systems [36], and therefore it makes critical assumptions such as the *determinism* of the underlying models (that are often based on differential equations).

[6] Chaos designates the behavior of dynamic systems that are highly sensitive to initial conditions, an effect which is popularly referred to as the butterfly effect. Small differences in initial conditions (such as those due to rounding or measurement errors) yield widely diverging outcomes, thereby making long-term predictions impossible [33].

[7] Emergence is the arising of higher-level structures, patterns and properties that are not explicitly encoded in the lower-level components, in particular in the context of self-organization [34].

In robotics, many successful approaches such as probabilistic robotics [37], behavior-based robotics [38], bio-inspired robotics [39], swarm intelligence [40], and hybrid systems [41] have emerged as potential alternatives to modern control theory for designing robotic controllers. Not only these methods generally yield similar or higher performance on selected benchmarks, but they are able to cope with the intrinsic uncertainty and stochasticity of robotic systems, especially when deployed in real settings. However, most, if not all, of them do not match the degree of analytical tractability offered by classic control theory, nor do they allow for a formal verification of their correctness (with the remarkable exception of hybrid systems, whose formal verification was pioneered by Henzinger [42]).

Successful control schemes for distributed robotic systems may range from fully centralized (i.e., control algorithms are essentially carried out by a central computer, which then dispatches precise instructions to each of the individual robots, e.g., as broadcast probabilistic templates) [43, 44] to fully distributed (i.e., control algorithms run entirely on-board, local to each individual robot) [45, 46]. Centralized control is usually easier to carry out in theory and often allows for achieving optimal performance, but it has usually high requirements in terms of communication bandwidth and computational resources of 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 attractive in terms of scalability and robustness, typically exhibiting a graceful degradation of system performance in the presence of one or more unit failures. However, distributed robotic systems, especially those consisting of large number of autonomous mobile units, are generally 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 instructions of varying precision and 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.

Another important distinction that we shall make concerns the synthesis of distributed controllers, which is commonly approached in either a bottom-up or top-down fashion [47]. In short, bottom-up approaches start with an implementation of the real system (or faithful simulations of this system), and then build up a series of increasingly abstract models, carefully validating each against those at lower abstraction levels [48, 49]. In contrast, top-down approaches [19, 27, 50] start with an abstract representation of the

system (e.g., based on graphs [27], potentials [19], hybrid automata [51]), which enables the use of powerful design, optimization, and analysis schemes (e.g., convex optimization [27], Padé approximants [52], optimal control [44], etc.), and then shape up the robots and their controllers such that the requirements of this idealized representation are fulfilled.

2.2 Small and Ultra-small Robots

Scientific and technological breakthroughs in the field of nano- and microengineering have steered the robotics community towards the realm of extreme miniaturization. As outlined earlier, 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 [53, 54] or complicated industrial machinery [23]. Further miniaturization down to the micro or nanoscale holds even more exciting potential in a large variety of fields such as biomedical engineering, environmental sensing, micromachinery inspection, and micromanipulation [55, 56]. However, miniaturization comes at a price: such robots are likely to be deprived in terms of sensing, actuation, and computation. These severe restrictions create the need for a collaborative approach towards the solving of tasks by leveraging perception and action at a collective level.

2.2.1 Existing Platforms

Attempts towards miniaturization are numerous in the field of robotics, and exciting new platforms have stemmed from this research. The following non-exhaustive list provides a few references of these platforms classified according to the size of an individual node (this list excludes the platforms specifically dedicated to aggregation and SA, which are discusses in Section 2.3.1): (i) centimeter-sized robots, such as Lipson's reconfigurable cellular robots (6 cm) [57], Wood's at-scale robotic insects (3-4 cm wing span) [58], Jasmine robot (3 cm, open-source platform) [59], the ultra low-cost Kilobot (2.5 cm) [60], the Alice robot (2 cm) [61], and the ARES project's swallowable modular robots (about 1 cm) [62]; (ii) millimeter-sized robots like the autonomous jumping robot proposed by Bergbreiter and Pister ($<$ 5 mm) [63], the I-Swarm robot (3 mm) [64], the Smart Dust motes (1-2 mm) [65]; (iii) micrometer-sized untethered robots, such as resonant soft-magnetic MEMS-based devices navigating a fluid-filled maze, as demonstrated by Nelson and colleagues (300 µm) [66], and steerable, untethered MEMS micro-robots (280 µm) [67]. However, intelligent and autonomous robots have yet to be demonstrated at the submillimeter scale. Indeed, endowing such small robots with capabilities sufficient to achieve closed perception-to-action loop control is a great challenge for two main reasons: (i) the fabrication processes of logic devices and other MEMS devices are

often incompatible (see Section 2.2.2), and (ii) power supply at these scales is still an on-going research topic [55].

2.2.2 Manufacturing Techniques

The manufacturing of small mechatronic devices is generally approached in two distinct ways: (i) conventional mechatronics based on a combination of mechanical and electrical engineering, in particular using printed circuit boards (PCBs) and mounted electrical components, and (ii) MEMS techniques based on the "depose, pattern, and etch" processes used in semiconductor device fabrication. While conventional mechatronics is much less suitable to extreme miniaturization than the latter, it offers one key capability that is still difficult to implement using photolithography, that is, the integration of *hybrid devices*. Namely, in conventional mechatronic design, almost any feature can be obtained by the integration of an appropriate component, which generally requires only to update the PCB layout and to solder the said component at the appropriate location. Whether the component is a CMOS chip, a sensor, an actuator, or a passive object is generally not relevant owing to the use of soldering. However, when it comes to endowing MEMS devices with multiple functionalities (e.g., sensing, actuation, computation, photonics, etc.), the problem of packaging and integration becomes important and truly critical. Indeed, the typical approach to MEMS manufacturing is photolithography, which consists of selectively removing parts of a thin film or the bulk of a substrate by transfering a geometric pattern from a photo mask to a light-sensitive polymer. A sequence of physico-chemical treatments then either engraves the exposure pattern into, or enables the deposition of other materials in the desired pattern upon, the material underneath the photoresist. The main limitation of this technique is that all materials used in the process need to be chemically and thermally compatible with each other. For instance, if one wants to etch a silicon wafer using potassium hydroxide (KOH) solutions at some point of the process, all exposed structures that need to be conserved after this step must be either made of a material resisting to KOH etching, or be covered by a protective resist. As a result, lithographic processes become increasingly complex as the variety of materials, functionalities, and length scales involved in the structure increases; in many cases, co-fabrication of heterogeneous devices is impossible.

To work around this problem, micro-engineers have developed a full breadth of assembly techniques for hybrid MEMS, including robotic pick-and-place methods and wafer-scale transfer, bonding and flip-chip methods to construct mechanical and electrical connections between different, individually fabricated sub-systems, or between MEMS devices and an interface board. Very recently, innovative approaches have started to consider the possibility of using SA to position and fix the parts on the target surface, or to assemble free-standing parts with each other [9]. This approach is

appropriate for situations in which the individual pieces are either too small or too numerous to be assembled using pick-and-place methods as it allows one to achieve high throughput and precise alignment of the parts.

In spite of these promising results, hybrid MEMS manufacturing is still subject to intense on-going research, and is not mature enough to allow for the manufacturing of truly autonomous robotic devices at the micrometer-scale.

2.2.3 Key Challenges

Small and ultra-small robots do not only pose manufacturing challenges, but their small size has important consequences in terms of design, control, and modeling. Hereafter, we describe the most prominent implications of miniaturization on these three tasks.

Force Scaling

An important property of our physical world is that all forces do not scale with size in the same way. Forces that depend on the volume (and, incidentally, the mass) of the object such as gravity and inertia scale with the cube of the size; forces that depend on the area of the object such as friction or drag scale with the square of the size; some forces even scale linearly with size, such as capillarity. Incidentally, the relative importance of these forces is completely modified upon scaling down a physical system. Typically, at scales below a few tens of micrometers, gravity and inertia become negligible whereas surface tension and capillarity dominate the force landscape.

As a result, some of the intuitive recipes for designing and controlling traditional robots are not applicable at small scales. For instance, the overwhelming prominence of adhesive and viscous forces make self-propelled submillimeter-sized robots difficult to realize. Typically, such systems rely on external forces (e.g., vibration, fluid flow, gravity, magnetic forces) to enable mobility of individual robots.

Stochasticity

Stochasticity is also directly connected to the notion of length scale. Indeed, physical phenomena that are essentially stochastic such as Brownian motion and turbulent flows appear only at certain length scales, and are completely negligible at others. For instance, the region in which Brownian motion is dominant—length scales of a few micrometers and below—is precisely the one where turbulences are completely negligible because of the absence of inertial forces and the prominence of viscous and drag forces. As a result, while fluidic phase mixing is a major challenge in microfluidics because of the absence of turbulence, small particles (less than 10 µm) are naturally agitated in fluidic environments.

Since inertia represents the resistance to change in motion of a physical object, the unfavorable scaling of inertial forces at small length scales makes small and ultra-small robotic systems very sensitive to stochasticity. Small grains of dust, surface roughness, and even small manufacturing defects are no longer negligible as the size of the robots decrease, and they should, in principle, be accounted for in models and control strategies of miniaturized robotic systems. In reality, these details are prohibitively difficult to model explicitly, and statistical approaches are therefore needed (e.g., stochastic differential equations such as Langevin equations).

The role of quantum uncertainties, while negligible in terms of physical interactions at the microscale, is critical in nanoelectronics, which will be instrumental in designing future ultra-small robotic systems. As a result, regardless of whether the world is deterministic or not, the practical implications of miniaturization are often an increased stochasticity, which one absolutely needs to account for when modeling, designing, and controlling small and ultra-small robotic systems.

Computational Resources

Another key challenge of miniaturized robots is their limited computational resources, which favor the use of simple, often reactive controllers. Indeed, as outlined earlier, in spite of our ability to manufacture small integrated circuits, their integration into submillimeter-sized robots is still hindered by a number of technical challenges. Even in a distant future it is unclear that microrobots will be endowed with logic circuitry similar to current centimeter-sized robots. This observation motivates the development of strategies that can circumvent these severe limitations.

Bacteria and, more generally, cells are, by far, the most advanced machinery of this size; yet, they exhibit behaviors that are not nearly as sophisticated as regular robots[8]. Still, large swarms of bacteria can in principle achieve regular computation [70], programmable pattern formation [71], and cooperation [72], thereby pinpointing the relevance of collective mechanisms for overcoming limitations at the individual level.

Energy Resources

Supplying energy to highly miniaturized devices is an important challenge from both a design and a manufacturing perspective. Several appealing solutions have been proposed in recent years, ranging from on-board power generators (e.g., thin-film batteries fabricated using semiconductor technology

[8] In his provocative book *Wetware* [68], cell biologist Bray goes as far as to suggest that biochemical systems that underlie the behavior of cells resemble neural networks, and are capable of "computation" and "learning". However, he uses these terms to designate quite different capabilities than those of regular robots [69]. Incidentally, his revolutionary vision of living cells strikingly resembles that of SMPs.

[73] or other cutting-edge micro-batteries [74]) to transducers scavenging energy from an often *ad hoc* engineered environment (e.g., MEMS-based power generators such as methanol fuel cells [75]). Yet, we believe that energy will remain a scarce resource at small scales, and one needs to consider very carefully this parameter when devising design and control strategies for highly miniaturized robots.

2.3 Self-organization, Aggregation, and Self-assembly

One of the coordination mechanisms that has proven successful in addressing the problem of controlling large distributed systems is *self-organization*, particularly if the system is resource-constrained [29, 76, 77]. There is an on-going debate regarding the exact definition of self-organization. In particular, Halley and Winkler [78] asserted that "self-organization implies a nonequilibrium process and SA is reserved for spontaneous processes tending toward equilibrium". The thermodynamic basis of this distinction makes its relevance to robotic systems questionable, though. Furthermore, while this distinction restricts the scope of self-organized systems, it broadens the definition of "self-assembly" to the extent that it nearly becomes a synonym of "aggregation" (because any aggregate, as disordered as it can be, is always more ordered than a perfectly mixed and randomized system). In this work, we shall therefore use more balanced definitions of self-organization, aggregation, and self-assembly, which pertains to a broader class of systems:

Self-organization is the spontaneous, decentralized and dynamic ordering of an initially disordered distributed system as a result of interactions of its constituent components.

Consistently with Halley and Winkler [78], we associate self-organization to a dynamic process that requires a constant source of energy for both *increasing* and *maintaining* the order of the system. The rules specifying the interactions are executed without explicit reference to the global pattern, thus allowing self-organized strategies to be very scalable.

Self-organization requires **four fundamental ingredients** to occur, which we illustrate here in the context of the stock market, which is a paradigmatic example of self-organized system (and voluntarily very different from those we investigate in this thesis):

1. **Positive feedback** is the reinforcement of an initial perturbation by its own effect, thus leading to an increase (generally exponential) of the magnitude of the perturbation. The price of a share on the stock market is strongly determined by the following positive feedback: as its price increases, the share becomes more attractive since people expect it to increase further, and therefore its price increases even further.
2. **Negative feedback** is the opposing force to exponential growths provoked by positive feedback; it is often characterized by the exhaustion

of resources, be they energetic, financial, or material, or by non-linear damping effects. The exponentially increasing price of a share rapidly reaches a value that investors consider (more or less objectively) as the real value of the underlying company, at which point they attempt to sell it.

3. **Stochasticity** allows the system to explore the solution space and to escape from local minima. The main source of randomness on the stock market is the investors' behavior, which depends on a large number of parameters (socio-economical of course, but also psychological, and even neurological [79]).
4. **Multiple interactions** among the lower-level components of the system is probably the most essential ingredient of self-organized systems. On the stock market, not only investors are continuously selling shares to each other, but they influence each other by their action.

Self-organized systems differentiate based on the substrate and mechanisms used to share information among the units. For instance, *stigmergy* [18] is a powerful indirect anonymous communication mechanism exploited by insect societies and some self-organized artificial systems reported in the literature [80–84]. Stigmergy relies on the specific signs left in the environment by the agents' actions, which in turn stimulates subsequent actions. It mediates the formation of complicated structures and spatial patterns, without need for any planning, control, or even direct communication between the agents. This type of control mechanism is simultaneously flexible and scalable, and therefore quite suitable to the control of large-scale systems. Self-organization has also attracted a lot of interest in the robotics community because of the confluent concept of emergence, which many researchers hope to exploit as a mechanism for improving both the adaptivity and the robustness of robotic systems. However, the wavering progress of artificial life outlines the difficulty of exploiting such a subtle and abstract principle in engineered systems, especially in concrete and applied case studies.

Whenever multiple components are allowed to move freely in a finite space, they may collide with each other and, in some cases, bond to each other. This phenomenon, widely referred to as *aggregation*, is certainly one of the most pervasive in our universe since it concerns objects as varied as atoms and molecules [85], microscale objects [86], animals [87], and galaxies [88]. Aggregation 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 [89] ranging from predator protection [90] to collective decision-making [11]. In robotics, aggregation has been demonstrated both with passive objects [82] and robots [89] as individual components.

Aggregation is the spontaneous and decentralized clustering of components into loosely ordered structures.

A group of robots may exploit random encounters for sharing information collectively while communicating locally; in such cases, aggregation can mediate self-organized processes such as collective decision-making [11] and consensus [91]. In some settings, the aggregating agents are passive objects moved by the robots, either one by one [50, 81], or in groups [80]. 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.

In many scenarios of aggregation, the bond stability depends on the local perceptions of the agents (e.g., the number of detected neighbors [92], the relative alignment of the robots [91, 93], light intensity [94]). In such cases, the overall stability of the aggregate also depends on its structure and geometry. The introduction of such local rules of interactions can allow for the formation of spatially and geometrically well-defined clusters. Although the concept of SA originated in chemistry, components of any size (from molecules to galaxies) can self-assemble[95], including engineered components such as passive building blocks [96], externally controlled parts [97], programmable parts [46], or full-fledged mobile robots [98]. Figure 2.1 illustrates a number of engineered self-assembling systems at all scales.

Self-assembly is the spontaneous and decentralized organization of components into spatially and geometrically well-defined structures, which are encoded in the specific, directional interactions among components.

Contrary to self-organization, aggregation and self-assembly systems do *not* require energy for maintaining their structure, consistently with Halley and Winkler [78]; they may however require energy for providing mobility to the constituent components and exploring the conformational space.

2.3.1 Existing Platforms

Among the recent implementations of self-assembling robotic systems at the macroscale, Gross and colleagues demonstrated SA in a group of Swarm-bots, a 15 cm-sized mobile robot equipped with a gripper [98]. Klavins and colleagues investigated in detail the use of graph grammars to control the SA of triangular robotic modules (12.5 cm in size) that slide passively on an air table [46]. The building blocks uses rotating permanent magnets as latching mechanism, and they execute a common graph grammar in order to determine their actions. At even smaller scale, Miyashita et al. [97] proposed simpler triangular robots called Tribolons (4.8 cm in size) that assemble at the surface of water and rely on a pantograph for both energy supply and control. Garnier et al. [89] successfully implemented bio-inspired probabilistic aggregation in a group of Alice robots (2 cm in size). Self-replication of

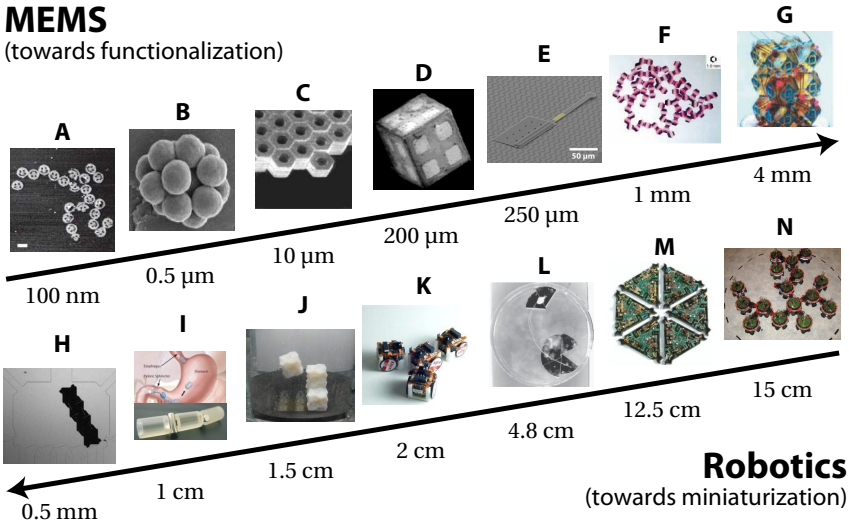


Fig. 2.1 Examples of engineered self-assembling systems reported in the MEMS literature (**A** to **G**) and in the robotics literature (**H** to **N**), thereby illustrating the clear convergence of these two disciplines. (**A**) Scaffolded DNA origami of 100 nm in diameter [99]. (**B**) Dense packing in small cluster of microspheres (844 nm in diameter) [100]. (**C**) Capillary SA of 10-µm-sized hexagonal plates into large crystal structures [101]. (**D**) Fabrication of 200-µm-sized patterned polyhedra by self-folding [102]. (**E**) Parallel assembly of stress-engineered MEMS microrobots into planar structures spanning several times the size of one robot (240-280 µm in length) [103]. (**F**) SA of millimeter-sized hexagonal plates using lateral capillary bond [104]. (**G**) Three-dimensional SA of millimeter-scale electrical networks [105]. (**H**) Dynamically programmable fluidic assembly of microtiles (500 µm in size) into arbitrary structures [106]. (**I**) Swallowable modular robots with *in vivo* self-assembling capabilities [53]. (**J**) Fluidic SA of centimeter-scale modules into arbitrary three-dimensional structures [107]. (**K**) SA of Alice robots (2 centimeters in size) into chains of controllable size using minimalist local communication [93]. (**L**) SA of water-floating tethered units, called Tribolon, which are endowed with a vibrator for controllability purposes [97]. (**M**) Self-assembling programmable parts that execute graph grammars as control strategy [46, 108]. (**N**) Morphology control in a group of swarm-bot robots (10 centimeters in diameter) [45].

robotic units, whose SA is a key ingredient, was also demonstrated by Griffith et al. [109] using two-dimensional coding strings as templates, and by Zykov et al. [110] with self-swiveling autonomous gripping blocks. Finally, Rus and colleagues have developed the Smart Pebble modules (1.2 cm in size), which self-assemble into a close-packed lattice of modules, which is subsequently shaped by allowing modules to communicate with each other and determine whether they must detach from the structure [111, 112]. While this system

exhibits impressive capabilities given its size, it still faces the problem of energetic autonomy—power is provided to a fixed, root module, and distributed through the system [111]. More generally, a full breadth of novel platforms are developed in the context of *modular robotics*, some of which are capable of autonomous locomotion and docking [113–116] (see [117] for a thorough review of the field), yet with a specific emphasis on reconfigurability rather than assembly from pre-existing scattered units.

Tolley et al. [17, 107] achieved three-dimensional fluidic SA of 1.5 cm-sized cubic modules using an active assembly substrate consisting of an array of fluidic ports with alignment patterns that match those of the cubes. Using a centralized control based on pressure feedback, their apparatus allows for the hierarchical assembly of relatively complex structures. Several research groups envision designing and fabricating modular surgical robots small enough (about 1 cm) for entering the human body through natural orifices (e.g., by ingestion [53]), and capable of configuring themselves into more complex devices within the stomach.

SA of sub-centimeter-sized robots is also being addressed in the literature. Tolley et al. [106] demonstrated the fluidic assembly of $500 \times 500 \times 30$ µm^3 silicon tiles into arbitrary two-dimensional structures by controlling the flow in a microfluidic assembly chamber. Donald et al. [103] have proposed MEMS-based robots (100 µm in size) that can be centrally controlled by a global and programmable electric field to dock compliantly with each other, thereby forming planar structures spanning several times the size of a single robot.

As researchers try to push the envelope of miniaturization, robots become more and more limited in terms of sensing, actuation, and computation, to the extent that one might be tempted to consider them as mere particles. However, self-assembling parts at the microscale are the result of intensive research and careful design; both their geometry and their surface properties are tailored for reacting in a specific and controlled manner with other parts. As a result, in spite of their relative simplicity as compared to conventional centimeter-sized robots, these devices ought to be considered as smart and reactive components.

Typical examples of such systems are the numerous self-assembling components reported in the MEMS literature [9, 118, 119]. These systems target a wide range of applications, including, e.g., three-dimensional electric circuits [105], flexible LED-based displays [120], integration of semiconductor devices onto plastic substrates [121], polyhedric containers [122], monocrystalline solar cells [123]. They exploit a broad spectrum of physical interactions, including (but not limited to) gravitational [124], hydrophobic [125], steric [126], dielectrophoretic [127], magnetic [128], capillary [104], DNA hybridization-mediated [129], or fluidic [130]. Interestingly, in the range of micrometric to nanometric scales, most of these interactions can be tuned to a reasonable degree [131, 132].

2.3.2 Design, Control, and Optimization

Controllers exploiting self-organization as coordination strategy are almost always synthesized in a bottom-up fashion. Indeed, top-down approaches are often inapplicable to such systems as they require a prior knowledge of the collective dynamics, which are generally either unknown (in particular when the system exhibits emergent properties) or difficult to translate into specifications at the individual node level. In the context of SA, this *inverse problem*—that is, finding the set of interactions that most robustly and spontaneously lead to a targeted many-particle configuration of the system for a wide range of conditions [133]—cannot be solved in polynomial time, in the general case[9]. In some simple cases, one can use linear programming [135], simulated annealing [136], or other stochastic optimization techniques [137] to solve this problem.

While the inverse problem is difficult *in general*, many of its variations can be solved efficiently if some critical assumptions are relaxed (e.g., by allowing building blocks to broadcast their state to their neighbors). The research groups of Mataric [138], Nagpal [139], and Requicha [140] have all demonstrated the SA of arbitrary structures by using active building blocks that can communicate with each other and execute a set of predefined rules in simulation. In both cases, the approach is completely decentralized, and the rules can be automatically generated by a compiler. While these works seem to contrast with—or even contradict—the claim that the inverse problem in SA is NP-complete, they actually make a critical assumption, that is, the ability of the building blocks to locate themselves in the structure (not necessarily explicitly, but also by propagating this information using directional and local communication channels). This requirement makes this type of algorithm difficult to implement in real, miniature robotic systems as directional, local communication is unreliable at small scales.

As a result, one generally focuses on the *forward problem*, which consists of tuning the design and control parameters of an existing system in order to favor the formation of a given target structure. Of course, this approach is much less powerful because prior knowledge of the space of feasible structures is needed. Only then can one optimize the system such that one target structure is favored over the others. From this perspective, the role of multilevel modeling (see Chapter 5.4) is critical for tackling the forward problem. Indeed, microscopic models are necessary to bridge the gap between the behavior of individual agents and the collective dynamics of the system as a whole, thereby allowing for the exploration of the search space. Then, the actual optimization of the system's control and design parameters requires computationally inexpensive macroscopic models [141, 142].

[9] A formal version of this general problem where the goal is to find the smallest tile system that uniquely produces a given shape has been proven to be NP-complete in general [134], therefore indicating that no efficient algorithm exists to solve this problem.

Graph grammars are another promising tool for coordinating self-assembling robotic systems [46, 108], in particular owing to their sound theoretical grounding and the possibility of translating a given grammar into a set of hardware specifications. However, this approach requires the robots to share their internal states upon aggregation as well as to interpret graph grammars, which might hinder their implementation on very small robots. Other approaches to the control of robotic self-assembling systems at the centimeter-scale do not consider the intrinsic stochastic properties of these systems at smaller scales, and typically assume advanced sensing and computation capabilities such as fine-grained range and bearing and/or vision [45, 98], or centralized control feedback coupled with very subtle actuation techniques to achieve arbitrary structures [106, 107].

2.3.3 Modeling

As outlined earlier, self-assembling systems—and, more generally, all systems relying on self-organization as coordination mechanism—require efficient and accurate models to achieve tunable and optimal performance. However, because SA entails several physical phenomena occurring at various levels of detail and length scales, several different modeling techniques are often needed. In this respect, the literature of SA modeling can be split into two main categories, which we describe hereafter.

The first category focuses on the accurate modeling of the physico-chemical and geometrical properties that characterize the dynamics of a single particle in close proximity to its target position in the assembling structure. Analytical models (generally based on a combination of first-order approximations and first-principles equations [143–145]) and finite-element numerical simulations [146], often multi-physics, are well suited to this type of very detailed physical models. At the microscale, a substantial amount of rather case-specific works addressed the dynamics of capillary-mediated SA [145, 147, 148], but also other interactions [149]. At the macroscale, many experimental works also provide simple analytical or finite-element models of the underlying interactions they rely on, be they fluidic [146] or magnetic [53].

The second category adopts instead a more abstract and general multi-particle perspective which, while reducing the accuracy by sparing a substantial amount of details about the physical and geometrical details of the system, still captures meaningful information about the process dynamics, and possibly allows for a better generality and computational efficiency. In 1994, Hosokawa et al. [150] first proposed an explicit analogy between artificial, macroscopic self-assembling systems and chemical kinetics. The resulting model is essentially a Chemical Reaction Network (CRN), a modeling framework that has been studied extensively in the context of biochemical systems [151]. Similarly, Zheng and Jacobs [152] proposed a first-order population model for the SA of hybrid microsystems, which does not, however,

account for the reversibility of the process. Recently, Mastrangeli et al. [153] reported an Agent-Based Model (ABM) of the same process, thereby accounting for both the low copy numbers of the system and some important features such as spatiality and suboptimal mixing. Rigid body simulations endowed with simplified physics (e.g., approximated drag and friction forces, buoyancy, magnetic dipole interactions, etc.) also proved useful in studying self-assembling systems, especially for investigating large-scale systems that are difficult to study experimentally [17].

Even though an exact algorithm for solving CRNs is known since the late 70s owing to Gillespie [154], researchers in distributed robotics seem to have ignored it until recently. For instance, Hosokawa et al. [150] solved their population model deterministically (i.e., by numerical integration) in spite of the low copy numbers of the system. Early attempts to use stochastic modeling in robotics and systems biology have been carried out by Kumar and colleagues [12, 155]. More recently, Klavins and colleagues have used a similar framework for building a stochastic interpretation of their graph grammars [46]. Deterministic models of aggregation and flocking (which is conceptually similar to aggregation, but involves a coordinated motion of the aggregate) such as [156, 157] as well as graphical models of multi-robot systems such as [158] are interesting complements, from a system and control perspective, to stochastic modeling, but do not take explicitly into account the intrinsic randomness of SA processes.

More generally, few attempts have been made to construct a more comprehensive and systematic methodological framework for modeling self-assembling robotic systems at small scales. A notable exception are the works of Schweitzer [159] and Milutinovic [51]. Schweitzer introduced the concept of Brownian agent, which allows for an appealing harmonization of multi-agent modeling and statistical mechanics. In particular, it provides a set of relatively systematic and formal strategies to generate analytical models out of a description of the system in terms of Brownian agents. Milutinovic proposed a modeling framework that unifies the description of biological and robotic systems under the critical assumptions of large populations of cells or robots. In this context, he proposed an optimal control approach for maximizing the population density in a given region of the space.

Even though these contributions are essential steps towards more formal and systematic modeling methodologies for distributed systems, they still lack precise, algorithmic description for being carried out in a strictly automated fashion. Matarić and colleagues [160, 161] generated Augmented Markov Models (AMMs) in real time based on the sequence of behaviors executed by a robot. However, this approach requires a prior knowledge of the states of interest, and is limited to single-robot systems. Alternative methods for automated modeling adopt completely different strategies based on evolutionary computation [162, 163]. In spite of their attractive flexibility and versatility, these methods are computationally expensive, and they yield gray-box models whose structure and parameters are difficult to anchor back

to the original system. In particular, they rely on a single level of abstraction, even in the context of collective systems (e.g., biological and chemical reaction networks [164, 165]), thereby precluding any mapping between microscopic and macroscopic states.

Multi-level modeling was proposed by Martinoli and colleagues [48] in the context of swarm robotics as a method for fulfilling the competing requirements of the detailed, physical modeling of low-level interactions and the abstract, statistical modeling of robotic populations. Chapter 5 will discuss this approach in detail.

Summary and Conclusion

This chapter provides a thorough review of the literature relevant to this thesis as well as a brief reminder of the key concepts in distributed robotics. In essence, the essential motivation for using multi-robot systems comes from the need for efficiency, robustness, scalability, and adaptivity. Such requirements arise in particular in scenarios that require very high reliability, or that involve robots much smaller than the environment in which they operate. However, these appealing features come at the price of an increased complexity in terms of design, control, and modeling. We illustrate this trade-off with the specific case of SMPs, i.e., systems composed of mobile, reactive, physically embodied agents that compensate for their limited computational resources using specifically engineered interactions with each other and their environment. On the one hand, owing to their excellent scalability, both in terms of length scale and group size, SMPs hold exciting promises in many disciplines, especially in the field of small and ultra-small robots. On the other hand, important difficulties arise at all stages of their development: design, modeling, control, and manufacturing. Self-organization, aggregation, and self-assembly are scalable mechanisms that have proven useful in addressing these challenges, in particular when coupled with bottom-up model-based design methodologies.

3

Materials and Methods

See first, think later, then test. But always see first. Otherwise you will only see what you were expecting.
—Douglas Adams (1952–2001)
in *So Long, and Thanks for All the Fish*

This chapter provides a thorough description of the experimental platforms that we used for illustrating and validating our methodological framework. We also discuss the features and limitations of the main computational tools that served as basis for the analysis of experiments and the implementation of models.

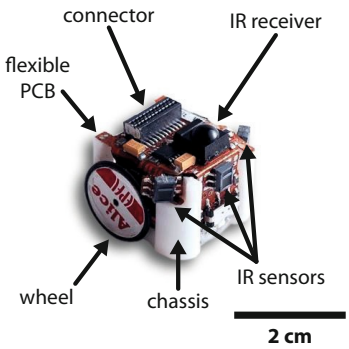
3.1 Experimental Platforms

Each piece of the methodological framework presented in this thesis is validated on three distinct experimental platforms. First, we use the Alice robot (Section 3.1.1), which is a conventional mobile robot based on regular mechatronic technology; yet, its small size, limited computational capabilities, and high sensitivity to noise (e.g., wheel slip, sensor noise, manufacturing and wearing heterogeneities) make it a suitable platform for studying SMPs. Second, we introduce a novel water-floating platform specifically targeted to the study of fluidic SA, called Lily (Section 3.1.2). It consists of passive modules of about 3 cm in size, whose design is tailored for allowing mutual latching and alignment. Third, we investigate the applicability of our methodological framework to microscale building blocks (Section 3.1.3). The latter platform is studied using realistic simulations as ground truth; the centimeter-sized platforms are studied using both real hardware and simulations.

3.1.1 The Alice Robot

The Alice 2002 mobile robot has a size of 21 mm×21 mm×21 mm, and is operated by a PIC 16LF877 microcontroller (4 Mhz, 384 bytes of RAM, 8kB of ROM). The Alice robot is endowed with four Infrared (IR) sensors that allow for a crude detection of passive obstacles up to 3 cm simultaneously

G. Mermoud, *Stochastic Reactive Distributed Robotic Systems*,
Springer Tracts in Advanced Robotics 93,
DOI: 10.1007/978-3-319-02609-1_3,



(a) Real Alice (b) Simulated Alice

Fig. 3.1 (a) The Alice 2002 mobile robot has a size of 21 mm×21 mm×20 mm and is equipped with four infrared sensors (rear sensor not visible on the picture) for crude proximity sensing and communication. (b) Simulated counterpart of the Alice robot implemented in Webots (see Section 3.2.1).

with 4 bps bidirectional communication up to 6 cm. It is also equipped with an IR receiver that allows the user to broadcast short messages to the group of robots; no other wireless communication mechanism is implemented in the basic configuration of the robot. The Alice is driven by a differential wheel drive based on stepper motors allowing for a maximal speed of 4 cm/s. The robot has been initially developed by Caprari et al. [61] at the Autonomous System Laboratory at EPFL, and several extension boards have been developed by other researchers [23].

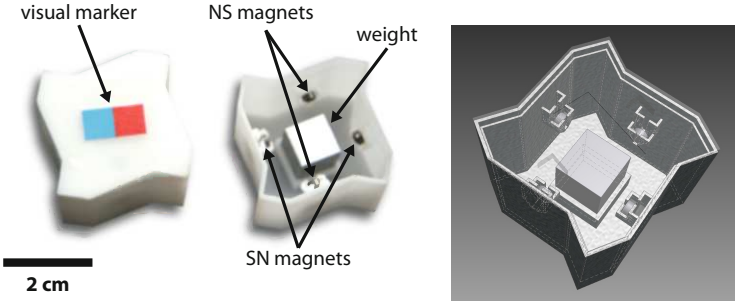
In the context of this thesis, we developed a series of extension boards endowed with epoxy-encapsulated photo-sensitive sensors[1] (A9950 11 photocell, Perkin Elmer, maximal spectral sensitivity at 530 nm), as well as two LEDs of different color (red and green) for tracking purposes.

The testing environment is a square arena (50 cm × 50 cm) equipped with an overhead camera for the purpose of data collection and control feedback, and, in some case studies, an overhead projector (more details are provided in Section 4.5). The group sizes vary between 4 and 20 Alice robots, depending on the case study.

3.1.2 The Lily Platform

Lily is a 3-cm-sized water-floating block endowed with four SmCo permanent magnets (one on each side's center) for mutual attraction and latching, as

[1] The number and the location of these sensors can be on at the center of the board, or two on each side, depending on the case study. More details are provided in Chapter 4.

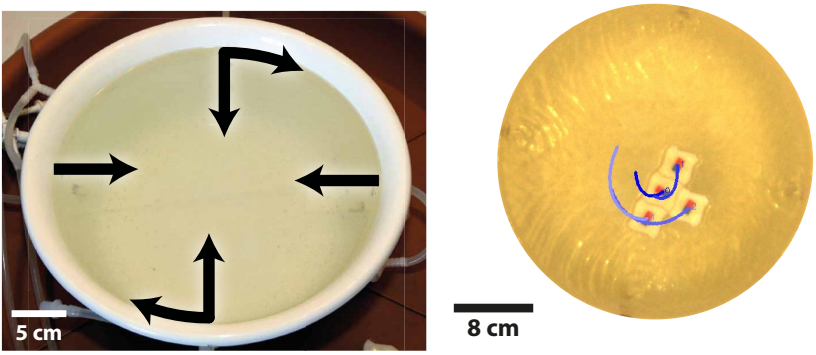


(a) Real Lily modules (b) CAD model

Fig. 3.2 (a) Details of a Lily module, including the latching mechanism composed of four permanent magnets with different pole orientation, north-south (NS) and south-north (SN), respectively. Each module is endowed with a 2-color visual marker for tracking purposes. (b) CAD model of the Lily module used as blueprint for 3D printing. The four permanent magnets attached to each side wall as well as the aluminum block at the bottom of the module are also shown.

well as with a 2-color visual marker for tracking purposes (see Figure 3.2a). Each module has a cuboidal, centro-symmetric shape specifically engineered for improving the relative alignment of the assembled blocks (Figure 3.2b); the main body and the cap of each module is manufactured using 3D printing. An extra aluminum block is added at the bottom of each module in order to reach an overall weight of about 17.3 g (the buoyancy limit is 21.9 g), and to improve flotation stability.

Importantly, Lily is not self-locomoted; instead, it is stirred by the fluid flow produced by pumps located along the tank perimeter (Figure 3.3a). The experimental setup consists of a circular water tank of approximately 30 cm in diameter, with six inlets connected to four diaphragm pumps. Four inlets are perpendicular to the wall, and the other two are as tangential as possible, allowing to create both radial and circular flows. Additionally, four outlets are placed at the bottom of the tank, to minimize the perturbations of the surface flow as much as possible. Each pump's flow rate can be controlled individually up to a maximal value of 600 ml/min. This flexible configuration allows us to investigate a variety of different flow patterns and associated robot trajectories (Figure 3.3b). Indeed, perpendicular inlets generate irregular trajectories, and allow for collisions in the middle of the tank, but they exhibit dead spots near the walls. Conversely, tangential inlets generate circular flows that prevent dead spots, but lead to regular, closed trajectories that do not favor collisions. As a result, the blocks describe trajectories with well-defined geometric features, yet with a strong stochastic component.



(a) Water-filled tank (b) Visual tracking

Fig. 3.3 The experimental setup of the Lily platform: (a) Water-filled tank with 6 inlets (4 orthogonal and 2 tangential to the wall, represented by black arrows); (b) Real-time visual tracking of four blocks during an experiment (the blue lines show a short history of the trajectory of each block).

3.1.3 MEMS Building Blocks

In order to investigate the applicability of our methodology to microscale fluidic SA, we fabricated building blocks whose size varies between 30 µm and 200 µm using photolithography. These devices are made of SU8, an epoxy-based, slightly hydrophobic, biocompatible, negative photoresist [166]. A large variety of such building blocks have been designed in the context of the SELFSYS project[2]. In each case, their surface properties and their geometry were designed such that they form more or less controlled assemblies once released in water. In this thesis, we focus on one particular class of building blocks that have a cylindrical shape and a rounded bottom, which lead to the formation of pairs (Figure 3.4a).

Microscale fluidic SA is mediated either (i) by surface tension (at the liquid/fluid interface) or (ii) by hydrophobic forces (in liquid phase). Hereafter, we illustrate these mechanisms based on prior theoretical and experimental works as well as qualitative insights gained from independent, orthogonal experiments carried out at the Laboratory of Microsystems at EPFL[3].

At the liquid/fluid interface, all objects causing a deformation of the interface (i.e., a meniscus forms close to the surface of the object) interact

[2] The SELFSYS project is funded by the Swiss funding initiative Nano-Tera. In a nutshell, the project aims at leveraging fluidic SA as an enabling technology for producing next-generation hybrid MEMS devices.

[3] These experiments were conducted by Loïc Jacot-Descombes, Maurizio Gullo, Cristina Martin-Olmos, and Juergen Brugger in the context of the SELFSYS project.

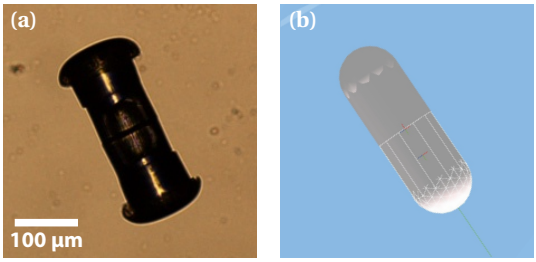
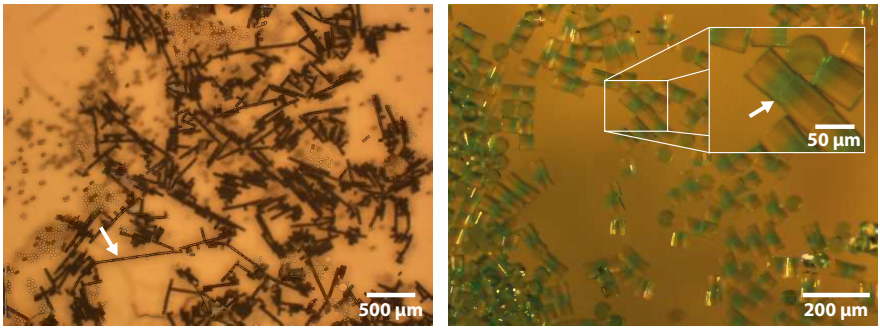


Fig. 3.4 (a) Optical microscopy images of two cylindrical building blocks (100 µm in diameter) assembled into a pair. The rounded bottom of the cylinders aims at preventing incorrect assembly configurations. (b) Simulated counterpart of these building blocks implemented in Webots (see Section 4.3 for more details about these simulations).

with each other through capillary forces, which are attractive if their menisci have the same convexity, and repelling otherwise [104]. Sharp features of the building blocks tend to attract each other because the local curvature of the liquid-air interface is larger around these regions, and so is the resulting interaction [167]. As a result, we expect cylinders floating at the water-air interface to experience stronger attractive forces along their rims, thereby leading to a preferential binding along their radial symmetry axis. Indeed, we observe experimentally the formation of long chains that can contain up to 18 building blocks with nearly perfect alignment (Figure 3.5a). Because the side walls of the building blocks are also hydrophobic, lateral forces are still present, thereby leading to undesired aggregates; a careful adjustment of the surface chemistry of the building blocks could remove these non-specific interactions [104].

In liquid phase, the absence of interface, and therefore surface tension, makes hydrophobic forces responsible for the interactions. One can tune these forces by adjusting the surface energy or the contact area of the building blocks. For instance, oxygen-plasma treatment increases the surface energy of SU8 [168], and therefore its hydrophilicity, thereby reducing the influence of hydrophobic interactions. Figure 3.5b shows the effect of oxygen-plasma treatment on the SA process; the non-treated, hydrophobic (binding) surface is on the colored end of the microcylinders. Other methods for adjusting the surface energy of the building blocks consist of depositing thin layers of other materials such as gold, and then either rely on their intrinsic surface properties (e.g., gold is naturally hydrophilic), or further tune these properties by applying a self-assembled monolayer (SAM) with hydrophilic or hydrophobic end groups. Contact area can be tuned by controlling the shape of the building blocks. Surface roughness and curvature decrease the contact area, and therefore the binding force, between building blocks. Similarly, complementary shapes may allow for specific interactions, which are difficult to obtain otherwise.

These results shown in Figure 3.5 were obtained in poorly controlled experimental conditions, and their use in the context of this thesis is hindered



(a) At the water-air interface (b) In liquid phase

Fig. 3.5 Optical microscopy images of the cylindrical building blocks whose self-assembling behavior is investigated in this thesis: (a) At the water-air interface, strong capillary forces lead to the formation of self-assembled chains (900 µm or 18 blocks long, white arrow) of uniform microcylinders (30 µm in diameter and 50 µm in length). (b) When immersed in water, surface-treated microcylinders (50 µm in diameter and length) tend to assemble with each other end to end, with a preferential attachment of their hydrophobic surfaces (colored end, white arrow). Images are courtesy of Loïc Jacot-Descombes.

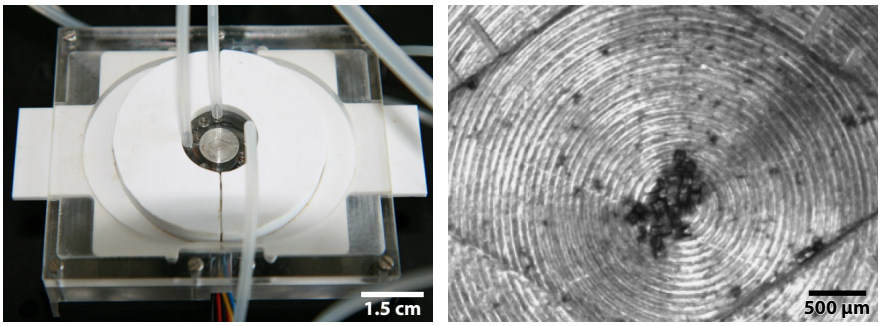
by two important factors. First, reproducibility and controllability of the process are critical to the validation of modeling and control methodologies. To tackle this problem, a sophisticated experimental setup based on a subtle combination of microfluidic and microacoustic mixing is currently under development at CSEM Center Central Switzerland[4] (Figure 3.6a). Second, process analysis and data collection require a high-speed camera and computer vision algorithms that can deal with the multiple difficulties encountered in imaging this type of systems: object cluttering, varying lighting conditions, occlusions, three-dimensional poses and trajectories (Figure 3.6b).

Because these developments are still on-going, this thesis does not include any experimental validation of models or control strategies of self-assembling systems at the microscale. Instead, the applicability of our methodological framework to such systems is studied based on realistic physical simulations of the building blocks shown in Figure 3.4.

3.2 Computational Tools

A large part of this work relies on computationally expensive models and simulations, which require both reliable and efficient software tools and high-performance computing infrastructures. On top of the usual software tools

[4] This experimental apparatus is developed by Jonas Wienen and Helmut Knapp in the context of the SELFSYS project.



(a) Close-up of the agitation chamber (b) Snapshot of an on-going experiment

Fig. 3.6 (a) Picture of an early prototype of the agitation chamber, which is the core unit of the experimental setup. The chamber is made of two glass slides sealed by a PDMS gasket, which includes microfluidic channels, cavities, filters, and valves. The whole is clamped onto a piezoelectric transducer, which is responsible for the agitation of the system through microacoustic mixing. (b) This snapshot of an on-going experiment illustrates the difficulty of imaging SA processes at the microscale, in particular due to object cluttering. Images are courtesy of Jonas Wienen.

used in research and engineering, we used a series of more domain-specific tools, which we describe in further details hereafter.

3.2.1 Webots

Webots is a development environment used to model, program, and simulate mobile robots [169]. It is commercially available and developed by the company Cyberbotics. Webots relies on an open-source library for simulating rigid body dynamics, and provides an OpenGL visualizer for both debugging and analysis purposes. Several features make Webots an attractive platform for the implementation of submicroscopic models (see Section 5):

1. Webots is a fast prototyping environment that allows for an easy visualization of the system's dynamics and the interactive editing of the robots and their environment. Also, the geometry of the robots can be directly imported from the CAD models used for their manufacturing.
2. Webots offers a vast library of pre-defined components such as sensors, actuators, or full-fledged robots; it also supports user-defined sensor and actuator characteristics, including non-linear and noisy response of the sensors and a tunable wheel slip (which is particularly important in the case of the Alice robot).
3. Even though it is mostly targeted to the development of realistic and detailed physical models of mobile robotic systems at the centimeter scale, Webots offers the possibility to develop custom physics plugins, thereby allowing to model a wide range of systems.

4. Webots allows one to implement hybrid simulations in which any object can be endowed with a software controller, which can be implemented in many different programming languages (e.g., C, C++, Java, Python, MATLAB™, etc.). A programmable supervisor enables a fine-grained control of the experimental conditions and a simple and efficient data collection, thereby making Webots simulations fully scriptable.
5. Webots can operate in two distinct modes: (i) in *physics mode*, the trajectory of the robots is the result of its physical interaction with the environment (e.g., the friction between the wheels and the floor); (ii) in *kinematics mode*, the robot's motion is calculated according to two-dimensional kinematic laws. The latter offers the advantage of being computationally cheaper, but it is available only for a limited range of robots (in the case of Webots, robots with two wheels and a differential steering), and its handling of collisions is very inaccurate due to the absence of friction.

Note that many competitors of Webots such as the Player/Stage/Gazebo suite [170], USARSim [171] or V-REP [172] are equally suitable to the development of submicroscopic models. The results presented in this thesis are in no way specific to the features offered by Webots.

3.2.2 StochKit

StochKit is an open-source extensible stochastic simulation framework developed in C++ [173]. It provides efficient implementations of several exact of Gillespie's Stochastic Simulation Algorithm (SSA), including the direct method [154], the optimized direct method [174], and the composition-rejection method [175], which enables the solving of the very large models generated by our automated modeling framework (Section 9). StochKit also provides an implementation of approximate methods such as the adaptive, explicit tau-leaping method [176, 177]. Furthermore, it supports user-defined propensity functions as well as triggers used to recreate changes in experimental conditions.

3.2.3 SwisTrack

All experiments presented in this study employ a system of overhead camera in conjunction with SwisTrack, an open-source object tracking tool targeted for multi-agent systems developed in our laboratory [178]. SwisTrack enables the tracking of both marked and markerless agents, with support for several different camera standards (e.g., USB, FireWire IEEE 1394, Gigabit Ethernet), including multi-camera systems. Its modular and extensible architecture makes SwisTrack suitable for agents of different type and size (e.g., robots, insects, small vertebrates such as chicken or fishes, and passive objects, including in principle MEMS devices). Furthermore, SwisTrack can

be easily interfaced with other software packages through a TCP/IP server that broadcasts the measured trajectories using the NMEA 0183 protocol.

3.2.4 Computational Cluster

In order to achieve fast and systematic simulations of the systems studied in this thesis, we rely on a computational cluster composed of 28 Intel Xeon-based nodes representing a total of 240 cores and 384 Gb of RAM. This cluster is primarily dedicated to the execution of numerous serial jobs, as its interconnect is relatively slow (regular Gigabit Ethernet).

In the context of this work, the cluster was primarily used to carry out a massive number of Webots simulations, and some computationally expensive data analysis. Numerous scripts and wrappers had to be developed in order to allow the reliable deployment of Webots on such a high performance computing platform, since this software package was not designed for this type of use.

Summary and Conclusion

This chapter introduces the three experimental platforms used in this work: (i) the Alice robot, a 2-cm-sized mobile robot based on regular mechatronic technology, (ii) the Lily module, a passive, water-floating block of about 3 cm in size specifically designed for the study of fluidic SA, and (iii) microscale building blocks (100 µm in diameter) made by photolithography, whose cylindrical shape and rounded bottom lead to the formation of pairs. The Alice robot lies at the "smart" end of the continuum spanned by SMPs: it is endowed with a PIC microcontroller, four IR sensors, and it is capable of self-locomotion. On the other side of this continuum, the microcylinders are passive devices with no sensing or actuation capability; yet, their shape and their surface properties are tailored such that they form more or less controlled assemblies once released in water. The Lily module lies between these two platforms: even though it is completely passive, its large size makes it significantly simpler to control than microscale building blocks. This chapter also describes a series of computational tools used to develop and solve the models presented in this thesis.

4

Case Studies

A pessimist sees the difficulty in every opportunity; an optimist sees the opportunity in every difficulty.
—Sir Winston Churchill (1874–1965)

This chapter describes the suite of five case studies that support our theoretical and methodological work. Each case study relies on one experimental platform described in Section 3.1, and aims at answering a precise research question that illustrates one or more features of our methodological framework. Each case study is associated to a mnemonic icon.

Case Study I is a paradigmatic example of aggregation using a group of Alice robots. Its simplicity and generality, along with the fact it has already been subject to rigorous investigation [92], make it an excellent baseline for illustrating and validating our modeling methodology.

Case Study II introduces the notion of order and geometry in aggregation processes by making Alice robots assemble into chains, thereby prefiguring the challenges relative to the modeling and the control of more advanced self-assembling systems.

Case Study III aims at illustrating the difficulties posed by three-dimensional and large-scale[1] systems. By investigating in simulation the pairwise SA of microcylinders, we provide some important insights into these two orthogonal challenges.

Case Study IV investigates the stochastic SA of Lily modules into a variety of different structures. Its aim is not only the rigorous validation of our methodological framework, but also the study of real-time control approaches that

[1] The notion of scale pertains to the number of building blocks in the system here, and not to their size.

G. Mermoud, *Stochastic Reactive Distributed Robotic Systems*,
Springer Tracts in Advanced Robotics 93,
DOI: 10.1007/978-3-319-02609-1_4,

Table 4.1 Transition probability p_n^{leave} from the resting state to the moving state as a function of the number n of neighbors within communication range. These values are taken from behavioral studies of cockroaches [179] (see [92] for more details).

n	0	1	2	3	4+
p_n^{leave}	N/A	1/49	1/424	1/700	1/1,306

can deal with the intrinsic stochasticity of these systems, as well as the lack of controllability at the individual level.

Case Study V aims to show how self-organization, collective decision-making, and appropriate modeling methodologies can be exploited to overcome the intrinsic limitations of SMPs. In particular, we investigate the use of aggregation as a way of achieving consensus, thereby improving the performance of these systems. The case study also offers the opportunity to discuss the experimental challenges posed by top-down approaches to model-based design of distributed controllers, and the benefits of bottom-up approaches advocated by this thesis.

Table 4.4 on p. 55 provides a synoptic overview that summarizes how each case study is used to illustrate the different topics of this dissertation.

4.1 Case Study I: Aggregation of Alice Robots

We consider a system composed of N_0 robots 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 probability p^{leave}. This quantity 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.).

Similarly to Correll and Martinoli [92], we use a simple behavioral model in which every robot is either moving or resting (i.e., aggregated). In our case, however, the robots do not transition to the resting mode as soon as they establish a communication channel with another mate; rather, the robots aggregate *upon collision* (i.e., strong activation of their proximity sensors), and they use local communication only for estimating the number of neighbors. The transition probability p^{leave} from the resting state to the moving state is given by a non-linear function of the number of robots within communication range (Table 4.1).

This variation enriches the variety of cluster geometries, as the connectivity has no influence on the aggregation process. Only the stability of the cluster is influenced by the connectivity of its constitutive robots. As a result, accurate macroscopic models need to account for these very diverse

connectivity patterns, which is not the case in [92]. We will discuss such aspects in Chapter 9.

If not indicated otherwise, we consider a system of 12 Alice robots with a fixed communication radius of 4 cm. The experimental arena is circular and has a diameter of 1 meter ($A_{\text{total}} = 0.78$ m^2). In this particular case study, the ground truth is provided by realistic simulations implemented in Webots.

4.2 Case Study II: Self-assembly of Alice Robots

This case study is similar to the previous one, with the exception of the geometry of the aggregates. Indeed, rather than allowing for an uncontrolled aggregation, the robots may now decide not to aggregate with each other if the resulting bond does not satisfy some geometric constraints that are evaluated from sensory data. In particular, they require a strong activation of their front proximity sensor as well as a relatively low activation of their two side sensors (Figure 3.1a), thereby leading to the formation of chains of variable length.

One of the research questions tackled by this case study is whether one can control the size of these chains, and what is the best approach to do so. First, we consider a baseline controller in which the robots, once aggregated, never disaggregate (Figure 4.1). As discussed later (Section 10.2), this behavior invariably leads to an exponential-like distribution of chain sizes, which we call the *trivial distribution*. In order to tune the system in a non-trivial fashion, we propose two orthogonal approaches: (1) a *deterministic* controller where robots communicate with each other in order to determine the size of each chain, and adapt their behavior accordingly, and (2) a *probabilistic* controller where chain size is controlled by the probability that a robot will leave an aggregate.

In the deterministic case, the robots will choose not to aggregate with chains already equal to the target length. Each robot communicates on one sensor an incremented observation of the opposite side (see Figure 4.2). Thus, outer nodes will send a value of zero towards the center of a chain. This number is incremented as it passes through each robot in the chain, eventually communicating its total size to the outer nodes. Other Alice robots that attempt to bond at the ends of a completed chain will receive this value, causing them to fail to aggregate.

The probabilistic controller does not require any explicit communication between the robots. Instead of limiting the maximum size of a chain directly, the robots leave their current aggregate with probability p_1^{leave} or p_2^{leave} each second, depending on whether they have one or two neighbors, respectively. In other words, Alice robots in a pair or at the end of a chain will leave with probability p_1^{leave}, while robots in the middle of a chain will leave with probability p_2^{leave}.

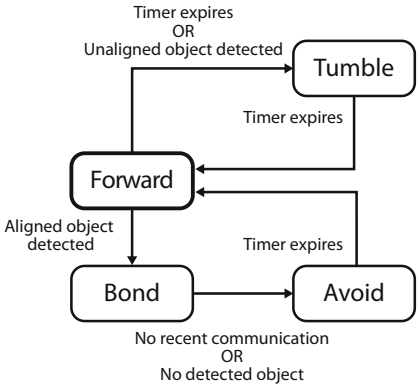


Fig. 4.1 Baseline controller used in Case Study II. The robot begins by moving in a straight path (*forward* state). After some period of time, or if an unaligned object or a wall is encountered, it begins a random tumble, before returning to the *forward* state. In this way, we achieve a better mixing by making robots "bounce" away from walls and misaligned robots. Should a well-aligned object be encountered, the robot stops in place and attempts to communicate. If communication is not returned, or the object "leaves", the robot performs a brief procedure of obstacle avoidance before returning to the *forward* state.

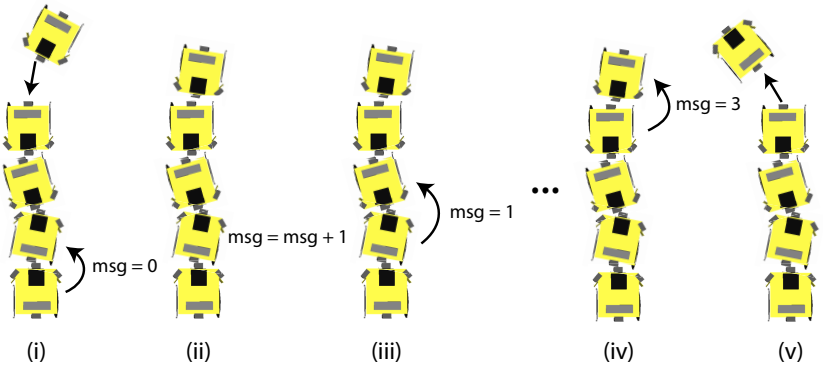


Fig. 4.2 A typical example of local communication in a system using the deterministic controller and a target chain size of four: (i) one robot joins a chain while the robot at the other end sends a value of zero into a chain. (ii) The adjacent robot reads an integer on one sensor, increments it, and (iii) communicates it on the opposite side. As time progresses, each robot learns its position within the chain and transmits to its next neighbor. (iv) Finally, the robot that has just aggregated receives a message equal to the chain size and (v) does not aggregate because this value is too large.

4.3 Case Study III: Self-assembly of Microscale Components

This case study investigates the SA of cylindrical building blocks with one hemispherical end (Section 3.1.3). Their geometry prevents the formation of unstructured aggregates or chains; rather, they tend to bind to each other in a pairwise fashion (see also Figure 4.4). Because of the experimental difficulties discussed in Section 3.1.3, we rely solely on simulations to investigate this type of system; yet, these numerical tools allow us to gain useful insights into the applicability of our methodological framework. We shall stress that these simulations serve as *baseline* for higher level models, this is why we describe them as an integral part of the case study.

For studying relatively small systems ($N_0 \leq 100$), we rely on dynamic rigid body simulations implemented in Webots (Figure 4.3). The underlying assumptions of these simulations are that both gravity and inertia are negligible (which is a safe assumption at these length scales, see Section 2.2.3), and that agitation is obtained by applying a zero-mean Gaussian stochastic force on each block at each time step. The latter assumption is a clear simplification of the underlying physical phenomenon, as we expect agitation to be provided by fluid flow, which would be more accurately captured using an approach similar to Case Study IV (Section 4.4). These simulations allow one to compute at each time step the force F_{ext} exerted on the joint by external perturbations such as the shear forces caused by the agitation and the collisions with other building blocks and the walls.

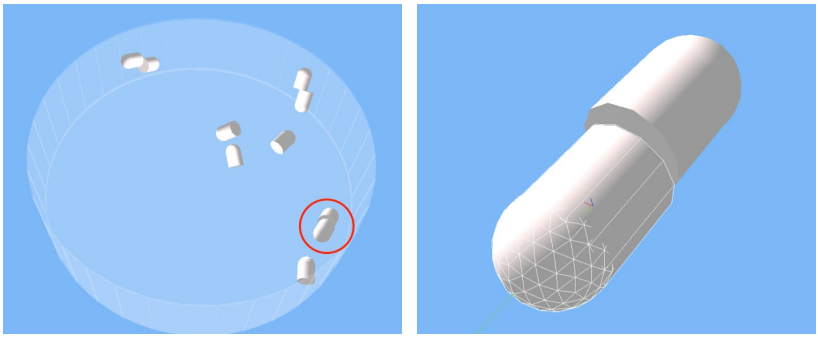
Large-scale systems ($N_0 > 100$) are studied using a two-dimensional ABM implemented in Netlogo, a powerful multi-agent simulation environment [180]. In this case, we assume that building blocks are radially symmetric bodies with a mass m, a position $\mathbf{x} \in \mathbb{R}^2$ of their center of mass, a velocity $\mathbf{v} \in \mathbb{R}^2$, an orientation $\theta \in [0, 2\,\pi]$, and a radius r, which undergo stochastic perturbations their trajectory[2] of, which can be described by a Langevin equation:

$$m\,\dot{\mathbf{v}} = -\,\gamma\,\mathbf{v} + \mathcal{N}(0, \eta\,\nu_s^2) \tag{4.1}$$

where $\mathbf{v}$ is the velocity of the particle, γ is a drag constant for small Reynolds number, and $\mathcal{N}(0, \eta\,\nu_s^2)$ is a Gaussian random variable of variance $\eta\,\nu_s^2$, which is proportional to the agitation of the system represented by the unitless value ν_s^2 ($\eta = 1$ N is a constant term in Newton). Two building blocks B_1 and B_2 of radius r_1 and r_2, respectively, located at a distance d from each other, collide if and only if $d < r_1 + r_2$. Furthermore, we simplify one of

[2] For the sake of simplicity, we assume that an aggregate undergoes the same stochastic perturbations as its building blocks. This assumption is not verified in the case of large aggregates, but is acceptable here as we limit our study to the formation of pairs. Equation 4.1 is relevant to building blocks that undergo Brownian motion; yet, one could use another equation of motion without restricting the validity of the approach.

the most computationally expensive aspect of the realistic simulations, that is, the calculation of the external force F_{ext}. Instead, we use a stochastic treatment similar to a Monte-Carlo Method (MCM), that is, each bond has a probability p^{leave} of breaking up in the next time interval $[t, t+\Delta t)$. The ABM model captures the conformation of each aggregate by keeping track of the relative positioning of each building block. Each building block is connected, if part of an aggregate, to other building blocks, called its *neighbors*, through a set of *bonds*.



(a) Global view (b) Close-up

Fig. 4.3 Screenshots of the rigid body simulation implemented in Webots; (a) Global view of an on-going simulation of 10 building blocks. (b) Close-up of the encircled aggregate under a slightly different angle.

In the context of this case study, the role of the bond model is crucial, as it determines almost completely the dynamics of the SA process. More particularly, the stability of the bond, which generally depends on alignment of the building blocks and the type of interaction, is crucial. Here, we consider three variants of the case study that involve bonds with zero, one, and two degrees of freedom (DOF), respectively.

In the **first variant**, the binding surfaces are modeled as idealized connectors that attract, align, and bond to each other in an irreversible manner whenever they are in close proximity. Strong, long-range interactions (e.g., capillary or magnetic forces) typically give rise to this type of bonds. This simplified 0-Degree of Freedom (DOF) bond model is studied using realistic simulations.

In the **second variant**, we simulate more carefully the dynamics of binding by using a 1-DOF bond model that enables misalignments among assembled building blocks. Upon collision of two building blocks, if one contact point belongs to the binding surface of both building blocks and their relative bearing is smaller than a given threshold, then both building blocks align along

the average direction common to both their axes, and a fixed joint is attached to them. The resulting bond force is computed as follows:

$$F_{bond} = F_{area} \cdot A(b, R) \tag{4.2}$$

where F_{area} is the surface force per unit area, and $A(b, R)$ is the overlap area[3] between two disks of radius R and inter-center distance b (see Proposition 1 of [144] and Figure 4.4**B**). The condition $F_{ext} > F_{bond}$ is checked at each time step, and if it is not verified, then the bond is destroyed. This 1-DOF bond model is studied using realistic simulations.

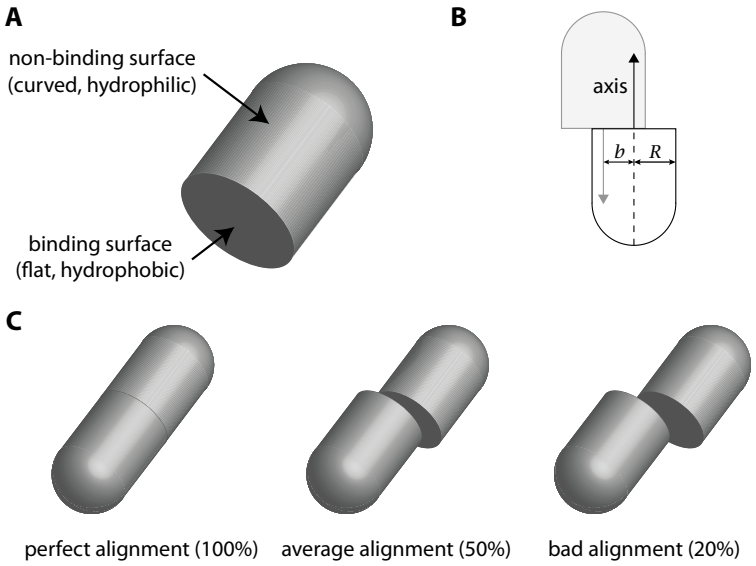


Fig. 4.4 Schematic illustrations of the 1-DOF bond model of Case Study III. (**A**) Binding and non-binding surfaces of a building block. (**B**) Cross-sectional view of two building blocks of radius R assembled with an offset b. (**C**) Three instances of self-assembled structures with different alignment. The 0-DOF variant of the case study leads only to self-assembled structures with perfect alignment, but it is less computationally expensive.

In the **third variant**, we consider a 2-DOF bond model that is characterized by an energy ΔE given by a Gaussian-like function of the relative alignment $\xi = (\theta_1 \ \theta_2)^T$ of the colliding building blocks:

[3] Since b is the only variable parameter of $A(b, r)$, the bond has indeed a single DOF, even though building blocks may move with respect to each along two distinct dimensions.

$$\Delta E(\xi) = -E_{\text{bond}} \cdot \exp\left(-\frac{\theta_1^2}{2\,\sigma_{\theta_1}^2} - \frac{\theta_2^2}{2\,\sigma_{\theta_2}^2} \right) \tag{4.3}$$

where E_{bond} is the maximal bond energy, σ_{θ_1} and σ_{θ_2} are parameters denoting the misalignment tolerance of each building block, and θ_1 and θ_2 are the angles depicted in Figure 4.5. Note that E_{bond}, σ_{θ_1}, and σ_{θ_2} are design parameters that depend generally on the physico-chemical and geometrical properties of the building blocks. This energy function implies that aggregates with properly aligned building blocks have maximal bond energy (in absolute value). For the sake of simplicity, we study this bond model in two dimensions using an ABM implemented in Netlogo. However, we expect this model to be flexible and meaningful enough to be applied to a large variety of target systems.

In this case study, we assume that all building blocks have the same tolerance of misalignment, and the bond energy can therefore be written

$$\Delta E(\xi) = -E_{\text{bond}} \cdot \exp\left(-\frac{\theta_1^2 + \theta_2^2}{2\,\sigma_\theta^2} \right). \tag{4.4}$$

To determine the policy of bond destruction, we use a model similar to the well-known *law of mass action* in chemistry. We know that the equilibrium constant K of a reaction

$$A + B \underset{k_-}{\overset{k_+}{\rightleftharpoons}} AB \qquad \text{with } K = \frac{k_+}{k_-} \tag{4.5}$$

can be written as a function of the energy ΔE of the formed bond:

$$K = \frac{k_+}{k_-} = \exp\left(\frac{-\Delta E}{E_B} \right) \tag{4.6}$$

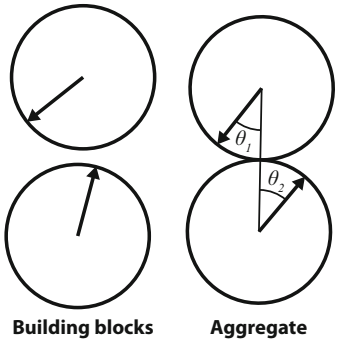


Fig. 4.5 Sketch of the 2-DOF bond studied using a two-dimensional ABM in the third variant of Case Study III

where k_+ and k_- are the forward and backward rate constants, respectively, and E_B is the mean energy of the particles in the system, if they are thermally randomized to the Maxwell-Boltzmann distribution.

One can interpret the rate constants k_+ and k_- as probabilities of bond formation and destruction. We assume that all collisions lead to an aggregation regardless of the energy of the formed bond, which is similar to setting $k_+ = 1$. Therefore, the probability that one bond of energy ΔE will break up in the next time interval $[t, t+T]$ can be written:

$$p^{\text{leave}} \equiv k_- \cdot T = \exp\left(\frac{\Delta E}{E_B}\right) \cdot T. \tag{4.7}$$

In the systems we are investigating, one cannot assume that the kinetic energies of the building blocks are randomized to any specific distribution. Furthermore, at the microscale, agitation is not necessarily strictly thermal. Hence, the term E_B can be replaced by a parameter $\alpha\,\nu_s^2$, that is, an energy term ($\alpha = 1$ J is a constant term in Joule) proportional to the agitation of the system, in its most general sense (see also Equation 4.1). Finally, the breakup probability p^{leave} of a bond of energy $\Delta E(\xi)$ can be written:

$$p^{\text{leave}}(\xi) = \exp\left(\frac{\Delta E(\xi)}{\alpha\,\nu_s^2}\right) \cdot T. \tag{4.8}$$

The performance metric is the same in all variants, and is given by the yield

$$Y(t) = \frac{X_a(t)}{X_s(t) + X_a(t)}, \tag{4.9}$$

where $X_a(t)$ and $X_s(t)$ denote the number of assembled aggregates and single building blocks at time t, respectively. In the third variant, we also study the average misalignment

$$M(t) = \frac{1}{X_a(t)} \sum_{i=1}^{X_a(t)} ||\xi_i(t)||^2, \tag{4.10}$$

where $\xi_i(t)$ is the relative alignment of aggregate i at time t.

4.4 Case Study IV: Self-assembly of Lily Modules

This case study investigates the SA of N_0 Lily modules into discrete aggregates with different geometries. The combined effects of mutual magnetic forces and block shape geometry lead to the precise, pairwise assembly and alignment of Lily modules in close proximity (about 0.5 cm), when not hampered by fluidic drag forces. We consider the special case where $N_0 = 4$ because it enables a

simple exhaustive enumeration of the assemblies that can be formed by the system, as well as the associated reactions (see Figure 4.6).

The inter-modules bonds are designed to be reversible, accounting for the interplay between the magnetic force (about 16 mN per bond according to finite element analysis) and the torque induced by hydrodynamic forces acting on the aggregated modules, the latter being modulated by the fluid flow regimes. As a result, the stability of all assembled structures corresponding to *local* minima of the system energy can be controlled by tuning the agitation of the system, whereas the *global* minimum, which corresponds to the 2-by-2 square structure—assembly D in Figure 4.6—is irreversible and acts as an absorbing state in the system dynamics.

In a perfectly mixed system, each structure has an intrinsic probability of being formed, which depends not only on its own geometry, but also on the tank geometry and the agitation parameters. For instance, the assembly A (Figure 4.6) is unlikely to be observed in a small tank because of the high probability of encountering between the blocks. Similarly, one may expect F_3 to be favored over F_2 in a small tank because of its smaller footprint. On top of these design considerations, one can favor the formation of a given species by controlling appropriately the agitation of the system.

In the present case study, we consider only two modes of agitation, denoted m_0 and m_1, corresponding to two different pump configurations that lead to different agitation schemes. In mode m_0, the fluid flow induces smooth and regular block trajectories and only marginal difference in their relative velocities, thereby favoring a high stability of the formed aggregates, but relatively few interactions. In mode m_1, the blocks exhibit a much more erratic movement, dominated by the stochastic perturbations of the water surface (Faraday waves) caused by pumps-induced tank vibrations. The resulting higher kinetic energy of the blocks increases the encountering rate, but also the instability of the aggregates.

4.5 Case Study V: Collaborative Spot-Destruction Using Alice Robots

This case study is a generalization of the stick-pulling experiment [48], in which pairs of robots must collaborate to pull sticks out of the ground. Here, sticks are replaced by N circular patches of light (referred to as *spots*) of diameter d_{spot} randomly scattered throughout a bounded arena of area A_{total}. A group of R robots wander about the arena looking for these spots, and attempt to destroy them. Each time a spot is destroyed, it is immediately recreated at another location within the environment. We consider two variants of this case study:

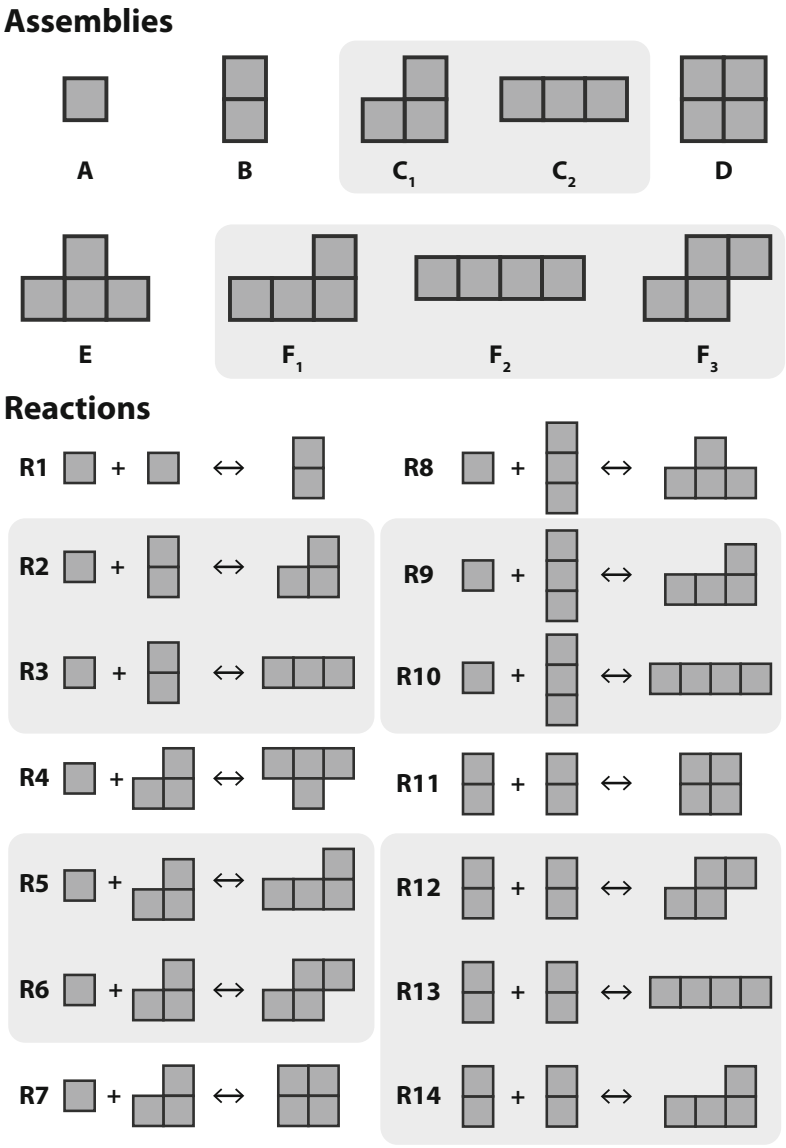


Fig. 4.6 Graphical representation of all the assemblies that can be formed out of four Lily modules and the 14 corresponding reversible reactions. Chiral copies of assemblies F_1 and F_3 are not included. The shaded rectangles denote the categories of isomeric assemblies in a 4-neighbor topology (i.e., two assemblies are isomeric if their corresponding graphs are isomorphic), and the resulting grouping of reactions.

- **Variant A**: a typical destruction scenario that involve two types of spot: *good* spots (in green), which must be preserved by the group of robots, and *bad* spots (in red), which need to be destroyed (Figure 4.7).
- **Variant B**: a more complex scenario that involves only bad spots[4] with random characteristic thresholds $\kappa \in [1, \ldots, N_\kappa]$ denoting the number of robots required to destroy the spot.

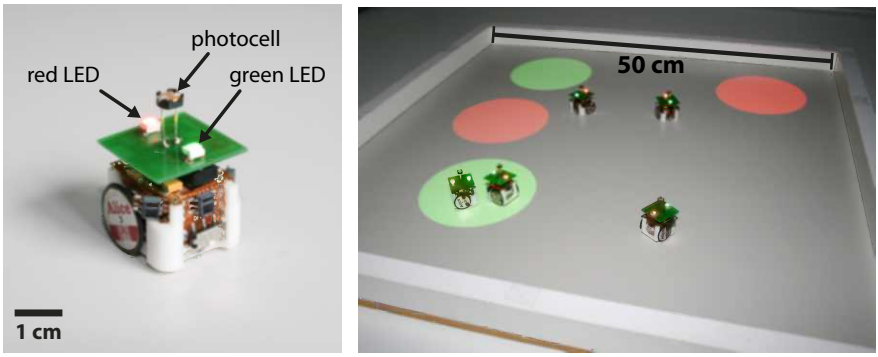
Obviously, the robots cannot directly interact with the spots; as a result, we employ an overhead camera, in conjunction with SwisTrack (Section 3.2.3), to emulate this interaction. In order to obtain an accurate measure of both the position and the orientation of the robots, we use two-color active marker composed of two LEDs. A combined camera-projector system integrates the trajectory of each robot, and updates the environment accordingly (i.e., it detects the destruction of spots and modifies the display by relocating the destroyed spots).

In real applications, the robots would need an actual cell-destruction mechanism, which can come in many forms. In many scenarios, more than one robot may be required to successfully destroy a cell, either because the effect of a single individual's action is too small or because the destruction mechanism is too complex (e.g., chemical reaction involving more than one reactant). Collaborative strategies may also improve the robustness of the system in presence of noise. For instance, variant A introduces a simple and reliable mechanism based on aggregation to share information among the robots, and therefore to overcome the sensing limitations of individual robots.

4.5.1 Variant A: Two-Type Spots

In this variant, the Alice robots are equipped with a unique light sensor (Section 3.1.1) that can be used to assess a spot's type. However, the measure provided by the light sensor is noisy (see Figure 4.9a); therefore, the robots may mistakenly trigger the destruction of a good spot. We denote $p_{w,good}$ the probability that a robot believes a good spot to be a bad one (false positive, from a destruction perspective) and $p_{w,bad}$ the probability that a robot believes a bad spot to be a good one (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 make deterministic decisions based on a single measurement, based on a decision threshold whose value is $t_d = (\mu_{good} + \mu_{bad})/2$ where μ_{good} and μ_{bad} are the average sensor response in good and bad spots, respectively. We assume that the robots can always determine whether they are exploring a spot or not in a perfect manner.

[4] Even though bad spots are supposed to be red, they are all green in this variant, with different intensities. The reason for this discrepancy is strictly technical: the A9950 11 photocell is more sensitive in the green region of the light spectrum.



(a) Close-up of an Alice robot (b) On-going real experiment

Fig. 4.7 (a) Close-up of an Alice robot and its extension board endowed with one photocell and two LEDs of different color (red and green) for tracking purpose. (b) Picture of an on-going real experiment with 5 robots and 4 spots in a 50 cm square arena.

The robots are endowed with a reactive controller shown in Figure 4.8, and minimal sensory and communication capabilities. Namely, nearby robots can detect each other (and distinguish other robots from obstacles), but they cannot actually communicate, nor can they carry out complex computation. Also, the robots do not have any reference to a global coordinate system. The robots explore the environment by performing a simple random walk with collision avoidance. After entering a spot, a robot will remain inside of it, "bouncing" off of its edges by performing a U-turn. At each bounce, a robot will decide to leave the spot with probability $p_{\text{good}}^{\text{leave}}$ or $p_{\text{bad}}^{\text{leave}}$, depending on whether it believes to be in a good or bad spot, respectively. As a result, a robot can change its belief about the type of spot it is in at regular intervals, i.e., upon bouncing off the edges of the spot. When a robot encounters an obstacle while exploring a spot, it assumes that it is another robot, and its next action will depend on the destruction threshold 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. For $k = 2$, the spot is destroyed as soon as a robot aggregates with another robot (Figure 4.9b 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 $p_{\text{aggr}}^{\text{leave}}$, that is, the probability that a robot leaves the aggregate, if it does no longer detect a nearby obstacle.

In order to have a quantitative method of reporting system performance, we define a performance metric in terms of the rate of destruction of good and bad spots:

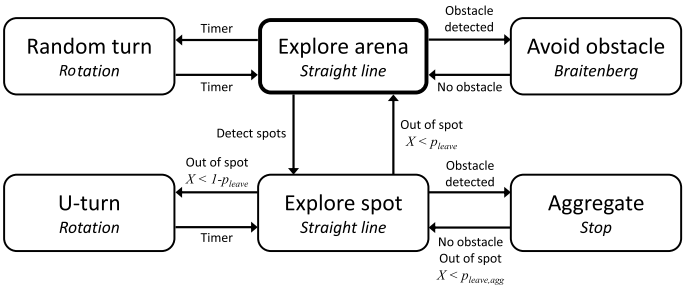
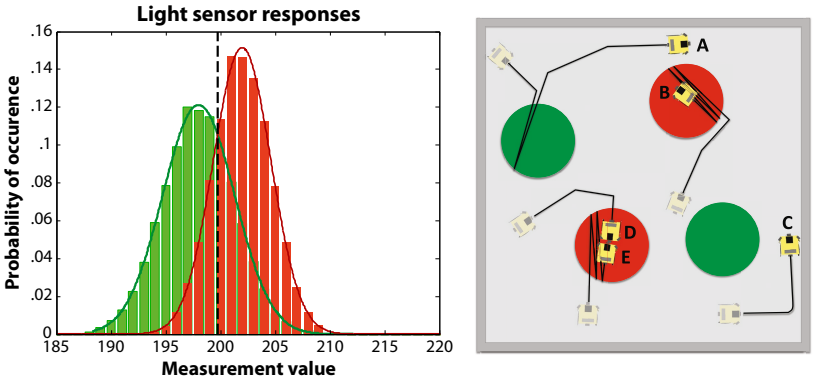


Fig. 4.8 The FSM representing the robots' controller in the variant A of Case Study V. The initial state is indicated in bold, and each state has an associated action (in italic). Some transitions are probabilistic (i.e., implemented using an on-board random number generator); in this case, the value of p^{leave} is either $p^{\text{leave}}_{\text{good}}$ or $p^{\text{leave}}_{\text{bad}}$ depending on the robot's estimate.



(a) Histogram of light sensor measurements (b) Sketch of a typical experiment

Fig. 4.9 (a) Histogram of light sensor measurements (1000 values) in good spots (in green), and in bad spots (in red). Fitted gaussian distributions are also shown (continuous lines). These data yield $p_{w,good} = 0.252$ and $p_{w,bad} = 0.259$. (b) Sketch of a typical experiment with 4 spots and 5 robots for $k = 2$. Trajectories of the robots are denoted by black lines. Robot A explored a good spot, made one wrong decision, 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-located at some other location in the arena.

Table 4.2 Control and design parameters used in variant A of Case Study V

Variable	Default value	Description
N_{spots}	4	Number of spots
N_0	5	Number of robots
$p_{\text{good}}^{\text{leave}}$	1.0	Probability that a robot leaves a spot that it believes to be good
$p_{\text{bad}}^{\text{leave}}$	0.0	Probability that a robot leaves a spot that it believes to be bad
$p_{\text{agg}}^{\text{leave}}$	0.001	Probability that a robot leaves an aggregate
$p_{w,good}$	0.252	Probability that a robot believes a good spot to be a bad one
$p_{w,bad}$	0.259	Probability that a robot believes a bad spot to be a good one

$$Perf. = \frac{\Lambda_{bad}}{(\Lambda_{good} + 1)^{\alpha}} \tag{4.11}$$

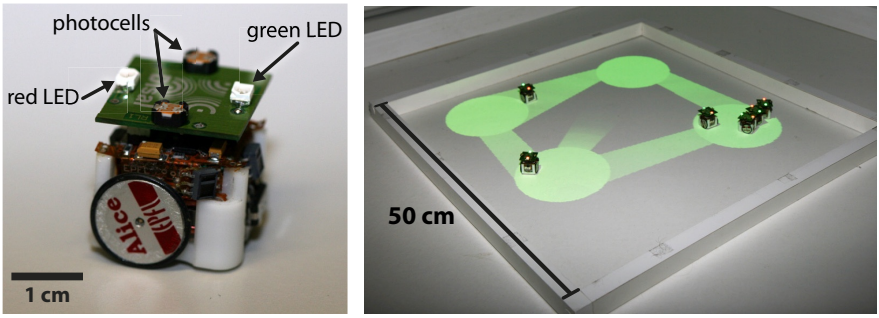
where Λ_{bad} and Λ_{good} is the rate of destruction of bad and good spots destroyed, respectively. The coefficient α may be balanced according to the penalty one wishes to associate with the destruction of a good spot; the higher the coefficient, the higher the penalty. Hereafter, we always set $\alpha = 2$ since we want to emphasize the ability of the group to discriminate between good and bad spots.

4.5.2 Variant B: k-Type Spots

In this variant, the intensity of a spot is a Gaussian random variable with mean I_κ and variance σ_κ^2, which both depend on the characteristic threshold κ. As a result, there is a probability $\chi(\kappa, \hat{\kappa})$ that a spot with threshold κ is measured with threshold $\hat{\kappa}$. Furthermore, spots are not destroyed instantaneously if $x_i \geq \kappa_i$ (where x_i is the number of robots at spot i; see Table 4.3 for all notations used in this variant), but they rather undergo a destruction process at a constant rate ρ, i.e., they have a "health level" that starts at 1, and decreases at constant rate ρ as long as κ robots (or more) are present in the spot. Upon destruction (i.e., when its health reaches zero), the spot is relocated at a random location in the arena, and the experiment continues. The performance of the system is given by the weighted rate of destruction of the spots

$$Perf. = \sum_{\kappa=1}^{N_\kappa} \kappa \cdot \Lambda_\kappa \tag{4.12}$$

where $\Lambda_\kappa = \rho \sum_{i=1}^{N_{\text{spots}}} \mathbf{1}_{\{x_i \geq \kappa_i \text{ and } \kappa_i = \kappa\}}$ is the destruction rate of spots with threshold κ.



(a) Close-up of an Alice robot (b) On-going top-down experiment

Fig. 4.10 (a) Close-up picture of an Alice robot with its extension board (2 LEDs for tracking purposes, and 2 light sensors for gradient detection and climbing). (b) Picture of an on-going top-down experiment with six Alice robots and four spots.

Table 4.3 Notation used in variant B of Case Study V. Note that the notation of variable x without mentioning the time (by opposition to the notation $x(t)$) is a simplified way to write $x(\infty)$, i.e., the value of the variable after convergence has been attained.

Variable	Default value	Description
N_{spots}	4	Number of spots
N_0	6	Number of robots
N_κ	3	Number of different thresholds a spot can have
$\mathbf{x}^d = (x_i^d)$	–	Desired distribution of robots
$x_i(t)$	–	Number of robots at spot i
$\mathbf{1}_{\{i,j\}}(t)$	–	1 if robot j is at spot i at time t
$\chi(\kappa, \hat{\kappa})$	–	Probability that a spot with threshold κ is measured as a spot with threshold $\hat{\kappa}$
Λ_κ	–	Destruction rate of spots with threshold κ

This variant is used as a testbed for comparing top-down and bottom-up model-based methodologies (see Section 10.1). In this context, we need a method for broadcasting navigation information to the robots. We propose an original solution based on augmented reality to solve this problem using a simple, nearly reactive, algorithm. Light gradients depicted on the arena by a video projector allow a central planner to "steer" the robots in the environment (see Section 10.1.2 for more details).

Table 4.4 Synoptic overview of the case studies (abbreviated CS in the table) and how they are used to illustrate the different topics of this dissertation. Dashes indicate that the case study is not featured for this particular topic, and that no relevant reference to previous work is available.

Topic	Details	CS I	CS II	CS III	CS IV	CS V
Submicroscopic	Construction	6.1.1	[93]	4.3	6.1.2	[91, 181]
	Calibration	7.1.1		–	7.1.2	
Microscopic	Construction	–	–	6.2.2	–	6.2.1
	Calibration	–	–	7.2	–	7.2.2
Macroscopic	Construction	6.3.1	[93]	6.3.2	9.5	6.3.3
	Calibration	9.5.1	7.2.1	–		7.2.2
Model Validation	Embodiment	–	8.1, 9.5.4	–	9.5.4	8.1
	Spatiality	9.5.1	–	8.2	9.5.2	8.2
	Macro-determin.	–	8.3	8.3	–	8.3
	Scalability	–	–	9.5.3	–	8.4
	Sensitivity	–	–	–	8.5	–
Synthesis	Design	–	10.2	–	–	10.1
	Optimization	–	–	11.1	12	11.2

Summary and Conclusion

This chapter provides a detailed description of the different case studies investigated in this thesis. More importantly, we discuss how each case study contributes to answering key research questions. Case Study I investigates the aggregation of Alice robots, and serves as baseline for illustrating and validating our modeling methodology. Case Study II introduces the notion of order and geometry in aggregation processes by making Alice robots assemble into chains. Case Study III investigates in simulation the pairwise SA of microcylinders, thereby illustrating the challenges arising in three-dimensional and large-scale systems. Case Study IV investigates the stochastic SA of Lily modules into a variety of structures; it also serves as testbed for real-time, model-based control strategies of highly stochastic distributed systems. Case Study V illustrates how self-organization, collective decision-making, and multi-level modeling enable SMPs to achieve complex tasks; in particular, we investigate collaborative detection and destruction of undesirable objects using Alice robots.

Part II

Modeling

5

Fundamentals of Modeling

All knowledge degenerates into probability.
— David Hume (1711-1776)
in *A Treatise on Human Nature* (1740)

Engineering always starts with a model. It may be an inaccurate diagram roughed out on a notepad, yet it serves the same purpose as any of the models discussed in this thesis: helping the designer understand and predict the dynamics of the system he is building. In the most general sense, a model is anything used in any way to represent anything else [182]. Some models are physical objects used to represent another physical object at a more convenient scale. However, most models used in science and engineering are abstract constructions that rely on mathematical concepts and language to describe an underlying system.

A **model** is an abstract representation of an underlying system using mathematical concepts and language. The level of abstraction of the model is generally tunable as a function of both the properties of the system and the intended use of the model.

Note that this definition pertains to a wide range of models ranging from detailed and realistic simulations to abstract and compact set of equations. Indeed, for any given system, one can formulate many different models with different levels of abstraction. Section 5.1 illustrates this claim by describing, in very general terms, a suite of models for a group of Brownian ferromagnetic particles. This example presents the advantage of involving strictly passive particles and rather simple interactions (i.e., magnetic dipole interactions, and elastic collisions)—leaving aside some of the difficulties posed by the modeling of more complicated features of SMPs, such as hybridity, complex geometry, and suboptimal mixing. Section 5.2 proposes a formal classification of the different models that can be used for describing SMPs in general. Then, Section 5.3 provides some key concepts of the theory of stochastic processes, and it outlines their fundamental role in multi-level modeling. Finally, Section 5.4 describes the Multi-Level Modeling Methodology (MLMM), which provides a set of recipes for building up models at increasing levels of abstraction in an efficient, albeit not automated manner.

G. Mermoud, *Stochastic Reactive Distributed Robotic Systems*,
Springer Tracts in Advanced Robotics 93,
DOI: 10.1007/978-3-319-02609-1_5,

5.1 Modeling Simple Collective Systems

Consider the case of small ferromagnetic particles (a few micrometers in size) floating in a glass of water (Figure 5.1). They exhibit complicated and irregular trajectories as a result of random collisions with the lighter molecules of the fluid, which cause the velocity of the particles to vary by a large number of small and presumably uncorrelated jumps—a phenomenon called Brownian motion. Upon getting close to one another, the influence of magnetic interactions become dominant, and the particles will eventually align and aggregate.

In principle, one could attempt to model the system by simulating the deterministic, physical movements of atoms constituting the ferromagnetic particles and molecules of the surrounding fluid (Figure 5.1**A**). At this level of detail, the dynamics of the particles are the implicit result of a multitude of interactions: the collisions between water molecules and atoms of the ferromagnetic particles, the hydrogen bonds forming between water molecules, and the magnetic arrangement of atoms of each particle. Such models are generally intractable due to the immense difference in length scale between the system of interest and the constituent atoms of the magnetic particles. We therefore need to *simplify* the model, thereby increasing the level of *abstraction*.

The first simplification that one can make consists in *averaging* the influence of the multitude of magnetic dipoles that constitute each particle into a single magnetic dipole (Figure 5.1**B**). This *mean-field approximation* is appropriate as long as the fluctuations are negligible—or irrelevant to the dynamics of the system of interest. Because each particle contains several trillion atoms, this approximation is indeed appropriate. As we will see, we rely on similar simplifications in our methodology: whenever too many individual elements contribute to a phenomenon, it is often useful to encapsulate them into a single, average contribution.

A similar reasoning can be made about the molecules of the surrounding fluid. In particular, one might be tempted to follow the same mean-field approach as described above: since the system is presumably symmetric and well-mixed (i.e., there is no particular reason why molecules of the fluid would hit more strongly or more often one specific spot of the particle), the net force exerted on a particle is *on average* zero. However, experiments show that the particles move erratically, i.e., they undergo Brownian motion. This contradiction can be explained by the fact that the necessary condition for using a mean-field approach is not fulfilled as *fluctuations are not negligible*. That is, the process does *not* consist of a large number of small random perturbations; Brownian motion occurs precisely in regimes where the ratio between the number of collisions and their effect on the particle's trajectory is not large enough.

Does this mean that one needs to simulate each of these collisions independently? Fortunately not. These innumerable fluctuations can be

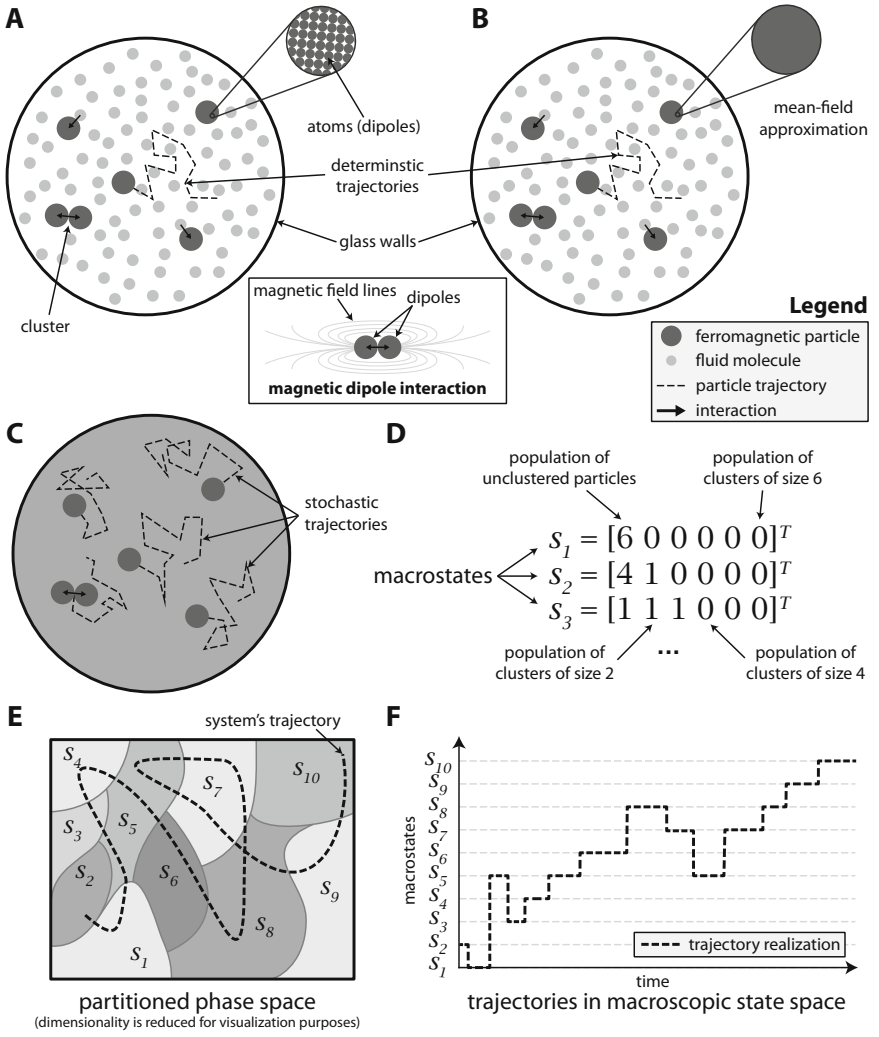


Fig. 5.1 Sketch of the different simplifications that can be made in a suite of models at multiple abstraction levels of a group of Brownian ferromagnetic particles. (A) Deterministic simulation of the atoms constituting the ferromagnetic particles and the molecules of the surrounding fluid. (B) Mean-field approximation of the magnetic dipoles of each particle as a single magnetic dipole. (C) Trajectories of individual particles are modeled as stochastic processes. (D) Macrostates are defined based on population vectors, and they form (E) a partition of the phase space of the system. (F) Based on this partition, the system's trajectory can be modeled as a jump process in macroscopic state space.

modeled using a stochastic process. More precisely, one can use a Wiener process $W_X(t)$, which is characterized by three properties: (i) $W_X(0) = 0$, (ii) $W_X(t)$ is almost surely continuous, and (iii) $W_X(t)$ has independent increments with $W_X(t) - W_X(s) \sim \mathcal{N}(0, t-s)$ and $0 \leq s < t$ where $\mathcal{N}(\mu, \sigma^2)$ denotes the normal distribution with expected value μ and variance σ^2. These two parameters correspond to drift and diffusion, respectively[1]. Note that this description is purely phenomenological, as it relies on the central limit theorem to justify its use of the normal distribution, but makes no specific assumption about the actual distribution of the discrete fluctuations caused by each collision event. As a result, one important challenge related to the use of stochastic processes as abstraction mechanism is the calibration of their parameters; in physics, this is typically achieved by using a combination of first principles, mathematical approximations, and empirical fitting. At this point, our model no longer captures any detail internal to individual particles, and their trajectories are no longer the implicit result of other entities' dynamics. Instead, they are modeled explicitly as a stochastic process with well-defined mathematical properties (Figure 5.1C).

Nevertheless, if the system involves a large number of particles, this model might still not be appropriate—or even tractable—in spite of all the simplifications we described earlier. Fortunately, the modeler might not be necessarily interested in spatial distribution of all particles at any arbitrary time t; instead, one might want to track properties such as the number of unclustered particles at time t or the number of clusters of size three at time t. Answering this type of questions does not require to track the position of each individual particles. Instead, one may describe the system by *counting* the number of unclustered particles or the number of clusters of a given size. In other words, the state of the system is no longer a vector of positions and momenta of its constituent particles, but a vector of natural numbers (called populations) (Figure 5.1D). From a modeling perspective, this strategy can be compared to a lossy compression method. In the previous, microscopic model, the system was described as a collection of N_0 stochastic processes defined on a n-dimensional continuous state space (the physical space, in our case), giving rise to a $N_0 \times n$-dimensional phase space. In this novel, macroscopic model, the system is described as a single stochastic process whose state space is a *partition* of this $N_0 \times n$-dimensional phase space. Each element of the partition contains all the possible arrangements (i.e., microstates) that satisfy a given macroscopic description, that is, a given population vector (Figure 5.1E). This macroscopic stochastic process then takes the form of a jump process (Figure 5.1F), whose derivation is discussed in Section 5.3; at this point, we can only say that its dynamics result from the aggregation of all individual stochastic processes.

[1] In accordance with the symmetry of the system, there is no drift in this case; diffusion is proportional to the temperature, and inversely proportional to the viscosity of the liquid.

Interestingly, the mean-field approach that failed to capture the dynamics of individual particles may now allow to simplify even further the macroscopic description of the system. While it is clear that the outcome of an experiment will vary from trial to trial, an average trend may exist. For instance, a system composed of 100 particles may not always end up with a single cluster of 100 particles, but exhibit relatively moderate fluctuations around this average behavior. For instance, a few particles may still remain unclustered at the end of the experiment, but this behavior is still different from the situation where 50 clusters of two particles are formed. When fluctuations are moderate and one wishes to distinguish only between such extreme cases, one may convert the stochastic process that describes the macroscopic dynamics of the system into a deterministic rate equation that describes the average behavior of the system. Because this description is based on ordinary differential equations, a series of analytical tools can be in principle leveraged, such as steady state analysis.

This example intends to illustrate that appropriate simplifications allow for an improvement of the computational and analytical tractability of models. However, it also outlines the challenges of constructing abstract models while maintaining their accuracy, especially in the context of SMPs, which may exhibit significantly higher complexity and richness than the system considered in this section. In particular, one need to find a proper partition of the phase space, such that the resulting macroscopic model is compact enough to be accurately calibrated, yet captures enough details for answering the modeler's questions. These challenges motivate a combination of multiple levels of abstraction, ranging from detailed, realistic models up to abstract population models, into a consistent multi-level modeling framework.

5.2 A Taxonomy of Models

A precise taxonomy of the models investigated in this thesis consists mainly of three axes (Figure 5.2): (1) the level of abstraction, (2) the representation of the state space, and (3) the representation of time. Because a model is essentially an abstract representation of the underlying system, the most important axis of our taxonomy is the level of abstraction. While the distinction between microscopic and macroscopic models is widely used in science and engineering, the notions of sub-microscopic models (borrowed from the field of transportation engineering), and the essential difference between macro-stochastic and macro-deterministic models, are often overlooked in the literature. As we shall see, we adopt a strict definition of *microscopic model*, which is an essential piece of the methodological framework presented in this thesis, and which will be rigorously formalized in Chapter 9. The representation of the state space plays an important role in the process of abstraction, as illustrated in Section 5.1. Indeed, both accuracy and computational complexity of macroscopic models critically depend on a proper partitioning of the phase

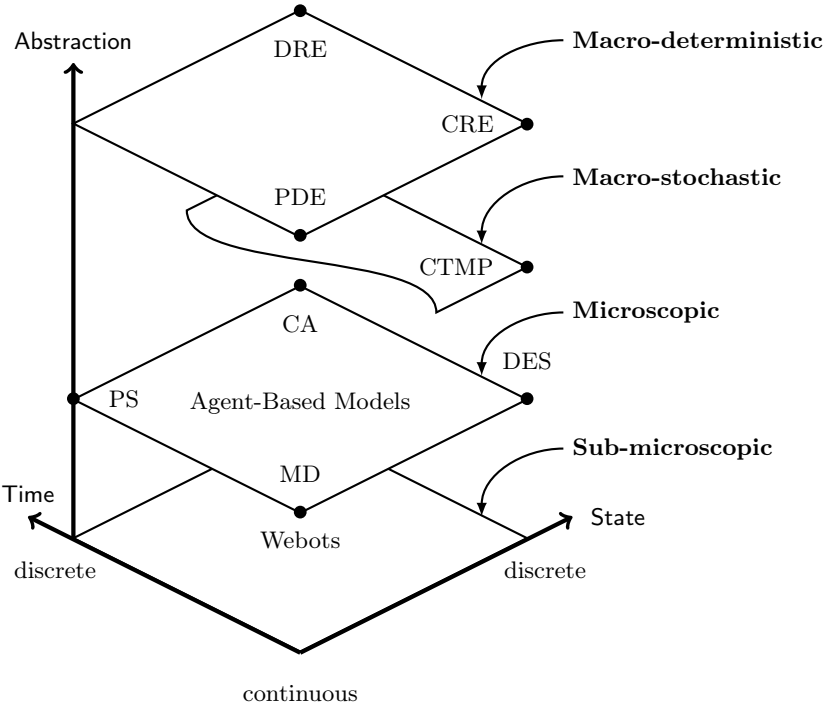


Fig. 5.2 Schematic view of the ecology of models defined in this thesis, classified according to (1) their level of abstraction, (2) their representation of the state space, and (3) their representation of time. The missing portion of the plan corresponding to continuous-state macro-stochastic models indicates that no such model exists. Acronyms: Point Simulations (PS), Molecular Dynamics (MD), Discrete Event Simulations (DES), Cellular Automata (CA), Continuous-Time Markov Processes (CTMP), Markov Chains (MC), Partial Differential Equation (PDE), Continuous-time Rate Equation (CRE), Discrete-time Rate Equation (DRE).

space. The distinction between time-discrete and time-continuous models is not as important as the others, and has mainly implications in terms of the implementation and the calibration of these models.

Submicroscopic models[2] constitute the most detailed models in the taxonomy. In particular, they attempt to capture as accurately as possible the physico-chemical and geometrical properties of the individual agents. The models **A** and **B** illustrated in Figure 5.1 are examples of submicroscopic models. At this level, the system's dynamics are generally *implicit*, that is, they result from the interplay of the different constituent modules and components of the agent (e.g., at the submicroscopic level, the trajectory of a robot is not

[2] We apologize in advance to the reader with a background in natural sciences or etymology, who might get annoyed by this abuse of terminology.

described by a kinematic equation, but it results from motor commands and the friction friction of the wheels with the ground). In principle, submicroscopic models make as few approximations as possible; in reality, the heterogeneity of the robots (due to manufacturing and wearing differences) are not taken into account and some sources of noise need to be modeled using idealized distributions (e.g., wheel slip, sensor noise, etc. are often modeled using Gaussian or uniform noise distributions). Submicroscopic models generally use a hybrid representation of the state space in order to capture both the physical and the logical state of the robots.

Microscopic models explicitly track the state of each individual agent as an atomic entity, that is, *all internal details are hidden*. More precisely, all internal details of the agents are lumped into a single state vector[3], and a (often stochastic) motion model governs the agents' trajectories in state space. Ideally, microscopic models would track the time evolution of the Probability Density Function (PDF) over the $N_0 \times n$-dimensional phase space of the system. However, because this endeavor is both mathematically and computationally intractable in the general case (in particular because of the large number of dimensions and the difficulty of writing governing equations), microscopic models almost systematically take the form of ABMs. In this thesis, we adopt a broad definition of ABMs, which encompasses all computational models that generate realizations of individual agents' trajectories in state space. Microscopic models are computationally cheaper than their submicroscopic counterpart, and they may be state-continuous, state-discrete, or hybrid.

Macroscopic models no longer track the state of each individual agent, but instead attempt to capture the time evolution of the PDF over a *partition* of the phase space of the system. The partition may be arbitrary, but it generally arises from a macroscopic description of the system. In physics, each combination of macroscopic variables such as temperature, volume and pressure correspond to one cell of the partition. In robotics, macroscopic models generally rely on countable partitions whose cells can be associated to population vectors. This approach presupposes an underlying characterization of the system into discrete "species". Furthermore, the partition of the phase space needs to fulfill two essential requirements: (i) it must be fine enough to allow the modeler to gain insights into its properties of interest, and (ii) it must be such that the system is well-behaved within each macrostate, that is, transitions from one state to the other can be somehow described as a jump process—one often tries to find a partition that yields a Markov process. Often, the system needs to be well-mixed (see Section 5.3.5), isotropic, and homogeneous within each cell of the said partition.

[3] As a result, the distinction between *microscopic* and *submicroscopic* is somewhat dependent on the object of interest. For instance, Molecular Dynamics may alternately be considered as a microscopic model if the molecules themselves are the object of interest or as a submicroscopic model if used for studying interactions of larger objects (e.g., nanoparticles, nanodroplets, etc.).

Ideally, macroscopic models should track the time evolution of the complete PDF over the partitioned phase space of the system. However, it turns out that analytical derivations are impossible in the general case, except for some special cases (e.g., branching processes [183]), and numerical solutions are generally intractable. When the model is Markovian, one may try to solve its associated master equation, which is often an equally difficult problem, both analytically and numerically (see Section 5.3.2 and [183] for general considerations about master equations). Some numerical approximations were recently proposed for systems whose PDF remains mainly contained within a small region of the space [184–187].

In this work, we rely mainly on more conventional methods for solving master equations. First, *macro-stochastic* models rely on MCMs, in particular the Gillespie's algorithm [154] and its optimized or approximate variants [188], to generate *exact realizations* of the system's trajectory in partitioned phase space (Figure 5.1**F**). A large number of runs allows one to obtain significant statistics about the time evolution of the PDF. Second, *macro-deterministic* models adopt a *mean-field* approach based on rate equations, to yield an *approximation* of the average system's trajectory in phase space. The fundamental assumption of macro-deterministic models is that fluctuations are negligible or irrelevant to the metric of interest.

Macro-stochastic models strictly require a discrete representation of the state space as they *count* the exact number of individuals in each state. Therefore, a continuous representation of the state space at the macro-stochastic level is impossible possible as it would boil down to a microscopic treatment. Macro-deterministic models can in principle track "densities" of agents in one or more continuous dimensions of the state space by relying on Partial Differential Equations (PDEs) such as the Fokker-Planck equation [189], the advection-diffusion equation [24], or reaction-diffusion systems [159]. As it turns out, computational methods for solving PDEs rely either on an internal discretization (e.g., finite element methods [190]) or on MCMs (e.g., smoothed particle hydrodynamics [24]) in any case.

5.3 Stochastic Processes in Multi-Level Modeling

As illustrated in Section 5.1, the essence of probabilistic modeling is to describe an underlying system as a collection of coupled stochastic processes. In many prior works, the link between probabilistic modeling and stochastic processes is unclear because macro-deterministic models are derived directly from individual behaviors [48, 92, 191]. Hereafter, we outline the fundamental role of stochastic processes in multi-level modeling, and the formal relationship that exists between microscopic and macroscopic models.

First, we introduce some key concepts of the theory of stochastic processes. This part of the manuscript is largely inspired from the celebrated textbook *Stochastic Processes in Physics and Chemistry* by van Kampen [183] and

the milestone paper *Stochastic Simulation of Chemical Kinetics* by Gillespie [188]. In particular, the mathematical formalism, most definitions, and some accompanying explanations are directly taken from these two essential works.

A **stochastic process** $Y_X(t)$ is a function of two variables, that is, the time t, and a random (or stochastic) variable X:

$$Y_X(t) = f(X, t). \tag{5.1}$$

One can obtain a *realization* of the process $Y_X(t)$ upon replacing X by one of its possible value x:

$$Y_x(t) = f(x, t), \tag{5.2}$$

which is an ordinary function of the time t.

Often, one is interested in knowing the PDF for the stochastic process $Y_X(t)$ to take the value y at time t, which we write

$$P_1(y, t) = \int \delta\{y - Y_x(t)\}\, P_X(x) dx \tag{5.3}$$

where $\delta\{\cdot\}$ represents the Dirac delta function and $P_X(\cdot)$ is the PDF of the stochastic variable X. One can also construct the joint PDF that $Y_X(t)$ takes the values $y_1, y_2, \ldots, y_n$ at times $t_1, t_2, \ldots, t_n$, respectively:

$$\begin{aligned} &P_n(y_1, t_1; y_2, t_2; \ldots; y_n, t_n) \\ &\quad = \int \delta\{y_1 - Y_x(t_1)\}\, \delta\{y_2 - Y_x(t_2)\} \cdots \delta\{y_n - Y_x(t_n)\}\, P_X(x)\, dx. \end{aligned} \tag{5.4}$$

Importantly, the infinite hierarchy of joint PDFs $P_n (n = 1, 2, \ldots)$ defines completely the stochastic process $Y_X(t)$, and it is often more physically relevant than Equation 5.1. In general, however, one would like to rely on stochastic processes that are more practical to construct; this is the case of Markov processes.

5.3.1 Markov Processes

The Markov property indicates that a given stochastic process is memoryless, i.e., its future dynamics depend only on its present state, and not on its past.

A stochastic process has the **Markov property** if the conditional PDF of future states of the process depends only upon the present state, not on the sequence of events that preceded it. A stochastic process with this property is called a **Markov process** [192].

More formally, the *conditional probability* $P_{1|1}(y_2, t_2|y_1, t_2)$ is the probability density for Y to take the value y_2 at t_2 given that its value at t_1 is y_2. One can write for the generalized conditional probability

$$P_{l|k}(y_{k+1}, t_{k+1}; \ldots; y_{k+l}, t_{k+l} | y_1, t_1; \ldots; y_k, t_k), \tag{5.5}$$

which describes the joint probability density for Y to take the values $y_{k+1}, \ldots, y_{k+l}$ at l times $t_{k+1}, \ldots, t_{k+l}$ given that its values at k previous times $t_1, \ldots, t_k$ are $y_1, \ldots, y_k$.

Then, a Markov process is formally defined as a stochastic process for which the following property holds:

$$P_{1|n-1}(y_n, t_n | y_1, t_1; \ldots; y_{n-1}, t_{n-1}) = P_{1|1}(y_n, t_n | y_{n-1}, t_{n-1}) \tag{5.6}$$

for any n successive times $t_1 < t_2 < \cdots < t_n$. In other words, the conditional PDF at time t_n is entirely determined by the value y_{n-1} at t_{n-1}, and is not affected by any knowledge of the values at earlier times [183]. $P_{1|1}$ is called the *transition probability*. As a result, a Markov process is fully determined by two functions: (i) the PDF $P_1(y_1, t_1)$ of the initial state y_1 at time t_1, and (ii) the *transition probability* $P_{1|1}(y_{n+1}, t_{n+1} | y_n, t_n)$, thereby making its definition and its use substantially easier.

The Markov property is undoubtedly the most widespread assumption made in the field of statistical modeling. First, many natural phenomena can be characterized by Markov processes (e.g., Brownian motion). Second, mathematical developments are greatly simplified by assuming that the Markov property holds. In particular, one can compute the conditional PDF at time t_3 given the value y_1 at t_1 using the Chapman-Kolmogorov equation:

$$P_{1|1}(y_3, t_3 | y_1, t_1) = \int P_{1|1}(y_3, t_3 | y_2, t_2)\, P_{1|1}(y_2, t_2 | y_1, t_1)\, dy_2. \tag{5.7}$$

Using this equation, one can derive the joint PDF of the stochastic process at any arbitrary time sequence $t_1 < t_2 < \ldots < t_n$.

5.3.2 The Master Equation

An equivalent form of the Chapman-Kolmogorov equation for Markov processes is the *master equation*. The underlying assumption is that one can discretize the range of values taken by the underlying stochastic process Y into a set of discrete states s. Then, the master equation is a gain-loss equation for the probabilities $p_s(t)$ of the discrete states s [183], which has the form:

$$\frac{dp_s(t)}{dt} = \sum_{s'} \Big\{ W_{s's}\, p_{s'}(t) - W_{ss'}\, p_s(t) \Big\} \tag{5.8}$$

where $W_{ss'}$ is the transition rate constant from state s to s'.

The master equation has a direct physical interpretation owing to the notions of discrete state and transition rate. In particular, the quantity $W_{ss'}\, \Delta T$ is the probability of transition from s to s' during a short time ΔT, which can be computed, for a given system, from first principles or experimentally measured in some cases.

Master equations can be derived for both microscopic and macroscopic models. Indeed, let us consider a large system consisting of a collection of N_0 individual agents that may be in N_s different states $q_1, \ldots, q_{N_s}$. At the microscopic level, the state s of the system is given by the vector

$$\mathbf{X}^{\text{micro}}(t) = \big[Q_1(t), Q_2(t), \ldots, Q_{N_0}(t)\big], \tag{5.9}$$

where $Q_i(t) = q_1, \ldots, q_{N_s}$ is the state of agent i at time t. At the macroscopic level, one is merely interested in the number of agent of each state; therefore, the state s of the system becomes

$$\mathbf{X}^{\text{macro}}(t) = \big[N_1(t), N_2(t), \ldots, N_{N_s}(t)\big], \tag{5.10}$$

where $N_i \in \mathbb{N}_{\geq 0}$ is the number of agents in state q_i at time t.

Both equations 5.9 and 5.10 are the basis for constructing a complete enumeration of the possible microstates and macrostates of the system, and from there the associated master equations describing the probability balance among these states. However, solving these master equations is generally difficult because of the large size of the associated state space. Indeed, the total number of possible states grows exponentially with the number of individual agents in the case of the micro-states (more precisely, in $O\big(N_s^{N_0}\big)$) and in the order of $\binom{N_s+N_0-1}{N_s-1}$ in the case of the macrostates. This observation advocates strongly for the use of macroscopic modeling when dealing with large distributed systems. A well-established example of macroscopic master equation is the Chemical Master Equation (CME), which describes the evolution of a population of reacting molecules as a Markov process. However, even macroscopic master equations such as the CME are difficult to construct and to solve in general. These difficulties can be partially overcome by the use of numerical approximations [184–187]. These approaches are based on the idea of restricting the analysis of the model to a subset of states that have "significant" probability, and cannot be applied to systems that have a large variance.

5.3.3 Chemical Reaction Networks

Master equations are powerful representations of Markov processes, but they are difficult to derive because they require an exhaustive enumeration of the states of the system and the transitions among them. Hereafter, we introduce the CRN formalism, which is essentially a compact description of the CME, which is widely used for modeling large biological networks [193]. The essential feature of the CRN formalism is that it allows one to describe the CME from the perspective of the agents, by listing the states they can be in, and by describing how state transitions depend on their mutual interactions.

Similarly to previous works [184, 188, 194], we define a CRN $(\mathcal{R}, \mathcal{S})$ as a set of reactions $\mathcal{R} = \{R_1, \ldots, R_{N_R}\}$ acting on a set of species $\mathcal{S} = \{S_1, \ldots, S_{N_S}\}$. Each reaction R is defined as two vectors of nonnegative integers specifying

the stoichiometry of the reactants, $\mathbf{r}_R = [r_{R,1}, \ldots, r_{R,N_S}]$, and the products, $\mathbf{p}_R = [p_{R,1}, \ldots, p_{R,N_S}]$, respectively. The stoichiometry denotes how many copies of a given reactant or product is required or produced, respectively, when a reaction takes place. For example, assume a CRN with $\mathcal{S} = \{A, B, C\}$, the reaction $A + 3\,B \rightharpoonup A + 2\,C$ is represented by the following vectors:

$$\mathbf{r} = [1\ 3\ 0]$$
$$\mathbf{p} = [1\ 0\ 2]$$

The CRN being a *population* model, it keeps track of the *number* individuals of each species are present in the system at a given time. The state of the CRN is therefore given by the vector $\mathbf{X} \in \mathbb{N}_{\geq 0}^{N_s}$, whose elements specify the number of individuals of each species. A reaction R may occur iff the number of reactants is sufficient, that is, $\mathbf{X} \geq \mathbf{r}_R$ element-wise. When reaction R occurs, the new state $\mathbf{X}'$ is given by:

$$\mathbf{X}' = \mathbf{X} - \mathbf{r}_R + \mathbf{p}_R = \mathbf{X} + \boldsymbol{\nu}_R \tag{5.11}$$

where $\boldsymbol{\nu}_R = \mathbf{p}_R - \mathbf{r}_R$ is the net change in population caused by R. Therefore, a CRN can be summarized into a $N_S \times N_R$ matrix $\mathbf{S}$, called a *stoichiometry matrix*, whose columns are the population change $\boldsymbol{\nu}$ of each reaction of the system. For instance, the following CRN

$$\begin{aligned} R_1 &: \quad 3\,B \rightharpoonup A + 2\,C \\ R_2 &: \quad A + 2\,C \rightharpoonup B \\ R_3 &: \quad C \rightharpoonup A \end{aligned} \tag{5.12}$$

can be represented by a stoichiometry matrix

$$\mathbf{S} = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} R_1 & R_2 & R_3 \end{array} \\ \begin{bmatrix} 1 & -1 & 1 \\ -3 & 1 & 0 \\ 2 & -2 & -1 \end{bmatrix} \end{array} \tag{5.13}$$

where S_{ij} is the stoichiometric coefficient of the i-th species in the j-th reaction. Positive and negative coefficients denote products and reactants of the reaction, respectively.

Another important characterizing quantity for a reaction R is its propensity function a_R, which is defined such that $a_R(\mathbf{x}, \cdot)\, dt$ is the probability that one reaction R will occur in the next time interval $[t, t + dt)$, given that the current state of the system is $\mathbf{X}(t) = \mathbf{x}$ [188]. In principle, one may consider a generalization of the CRN framework wherein propensity functions may depend on a variety of quantities (in addition to the state of the network $\mathbf{x}$). For instance, in the semi-Markovian case, the propensity functions have the form $a_R(\mathbf{x}, t_w)$ where t_w is the time since the last occurrence of reaction R, that allows for capturing arbitrary distributions of waiting time. In this work

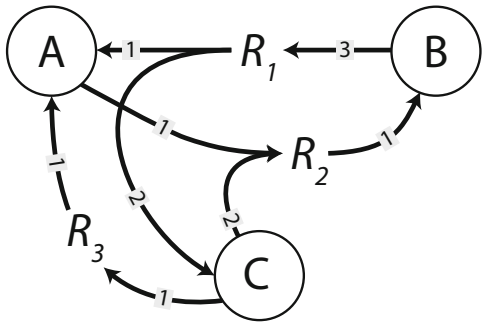


Fig. 5.3 Graphical representation of the CRN given by Equation 5.12. Species are encircled and reactions are denoted by splitting and merging arrows. Gray-shaded labels indicate the stoichiometric factors.

however, a_R depends only on the current state of the system, and the waiting times are therefore exponentially distributed.

Based on the above definitions, one can readily derive the CME:

$$\frac{\delta P(\mathbf{x}, t|\mathbf{x}_0, t_0)}{\delta t} = \sum_{j=1}^{N_R} \Big(a_{R_j}(\mathbf{x} - \boldsymbol{\nu}_{R_j})\, P(\mathbf{x} - \boldsymbol{\nu}_{R_j}, t|\mathbf{x}_0, t_0) - a_{R_j}(\mathbf{x})\, P(\mathbf{x}, t|\mathbf{x}_0, t_0) \Big) \tag{5.14}$$

where $P(\mathbf{x}, t|\mathbf{x}_0, t_0)$ denotes the probability that the process $\mathbf{X}(t) = \mathbf{x}$ given that $\mathbf{X}(t_0) = \mathbf{x}_0$.

As outlined in Section 5.3.2, both numerical and analytical solutions of Equation 5.14 are prohibitively difficult to compute, except for very simple cases. As a result, one often need to *simulate* the process $\mathbf{X}(t)$ using the SSA proposed by Gillespie [154], which constructs exact numerical realizations of the process. This algorithm belongs to the category of macro-stochastic models, as defined in Section 5.2.

5.3.4 The Macro-deterministic Approximation

Macro-stochastic models yield realizations of the system dynamics; as a result, over a large number of runs, one can obtain a good approximation of the distribution of trajectories. Alternatively, one can in principle convert any CRN (Section 5.3.3) into a macro-deterministic model, that is, a non-linear[4] system of ODEs:

$$\frac{d\mathbf{X}^{\infty}(t)}{dt} = \mathbf{S} \cdot \mathbf{a}\big(\mathbf{X}^{\infty}(t)\big) \quad \text{with } \mathbf{X}^{\infty}(t) \approx \langle \mathbf{X}(t) \rangle, \tag{5.15}$$

where $\mathbf{S}$ is the stoichiometry matrix (Equation 5.13), and $\mathbf{a}\big(\mathbf{X}^{\infty}(t)\big)$ is a *non-linear* function mapping population to propensity vectors. The deterministic

[4] If the CRN involves only unimolecular reactions, then the resulting system of Ordinary Differential Equations (ODEs) is linear.

trajectory $\mathbf{X}^{\infty}(t)$ *approximates* the average trajectory $\langle \mathbf{X}(t) \rangle \in \mathbb{R}_{\geq 0}^{N_s}$. There exists two *sufficient* conditions for this approximation to be accurate: (i) *the absence of fluctuations*, that is, the underlying process is completely deterministic[5], or (ii) the system operates in the *thermodynamic limit*, i.e., the number of particles and the volume approach infinity together such that the particle density remains constant. It turns out however that many previous works rely on the macro-deterministic approximation for modeling systems that do not fulfill these conditions [92, 152, 195], thereby raising the question of what are the *necessary* conditions for these models to be accurate.

Before proceeding further, we shall distinguish between two distinct notions of accuracy. First and foremost, we are interested in the ability of the macro-deterministic model to predict the average behavior of the system, which is given by the distance between $\mathbf{X}^{\infty}(t)$ and $\langle \mathbf{X}(t) \rangle$. We refer to this distance as *the first-order error* $\mathcal{E}_1$. Second, assuming that the first-order error is small, we may look at the role of fluctuations around this well-predicted, average trajectory, which we refer to as *the second-order error* $\mathcal{E}_2$.

The first-order error becomes small as one scales the system such that the reaction rates become large, and their effect small, i.e., the system implies a large number of small changes [196, 197]. As a result, the validity of this approximation is not necessarily correlated with the number of agents in the system, but rather with the number and the nature of their interactions. More precisely, the macro-deterministic approximation could remain valid for systems that involve few agents that interact often, as long as the individual interactions do not affect too much the state of the ensemble.

As regards the second-order error, we provide a formal characterization of its expectation as a function of the finiteness of the population. This error typically grows as the number of agents decreases. More formally, one can write

$$\begin{aligned} \mathbf{X}^{\infty}(t) &= \left(x_i^{\infty}(t) \right) \\ \mathbf{X}^{N_0}(t) &= \left(x_i^{N_0}(t) \right) \quad i \in \mathcal{S} \end{aligned}$$

the distributions yielded by macro-deterministic models (with an infinite number of agents) and macro-stochastic models (with N_0 agents), respectively. Therefore, the distance between these distributions is the error caused by the assumption that the system involves an infinite number of agents. Using the l^2-norm as a measure of the distance between the distance between distributions, the second-order error $\mathcal{E}_2$ of macro-deterministic models is:

$$\mathcal{E}_2(t) = \left\| \mathbf{X}^{\infty}(t) - \mathbf{X}^{N_0}(t) \right\|^2 = \sum_{i=1}^{N_S} \left(x_i^{\infty}(t) - x_i^{N_0}(t) \right)^2 \tag{5.16}$$

[5] This condition is never verified in practice, but it is necessary in the mathematical developments that lead to Equation 5.15 [188].

and as $t \to \infty$, we obtain[6]:

$$\mathcal{E}_2 = \frac{1 - \sum_{i=1}^{N_S} (x_i^{\infty})^2}{N_0}. \tag{5.17}$$

Therefore, the actual distribution of agents is always different, on average, from the distribution predicted by the macro-deterministic model. The distance between these two distributions can be predicted for any given number of agents using Equation 5.17. Figure 5.4 illustrates this result by comparing the expected and actual distance between predictions of macro-stochastic and macro-deterministic models of Case Study V.

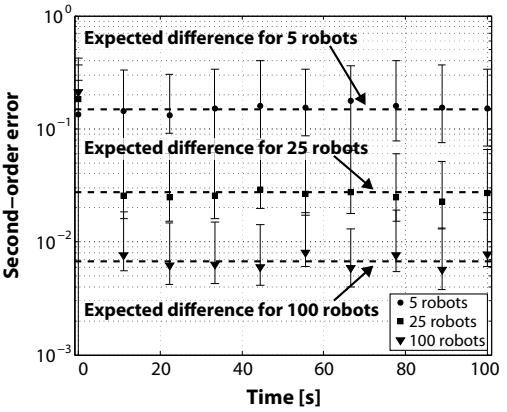


Fig. 5.4 Second-order error $\mathcal{E}_2$ (on a logarithmic scale) of a macro-deterministic model with respect to its corresponding macro-stochastic model for different number of robots (Case Study V, 30 runs, marker denotes the mean, and error bars the 95% confidence interval). The dashed horizontal lines denote the expected distance according to Equation 5.17. We note a strong agreement between our numerical simulations and the analytical prediction.

Fixed Points: Existence, Uniqueness, Multiplicity, and Stability

Macro-deterministic models allow in principle for the study of formal properties such as the presence of fixed points and their stability. A fixed point correspond to a state $\mathbf{X}^{\infty}$ such that

$$\frac{d\mathbf{X}^{\infty}}{dt} = \mathbf{S} \cdot \mathbf{a}\big(\mathbf{X}^{\infty}\big) = 0. \tag{5.18}$$

However, because $\mathbf{a}(\cdot)$ is non-linear and of high dimensionality, Equation 5.18 is difficult to solve analytically. The Chemical Reaction Network Theory (CRNT), which has been developed over the last 30 years, establishes several theorems about systems of non-linear ODEs derived from CRNs, assuming that they use mass-action kinetics [198, 199]. The theory introduces the concept of *deficiency* of a CRN, and builds upon this notion to derive conditions for the existence, uniqueness, multiplicity, and stability of fixed points.

[6] See Appendix C for the proof of Equation 5.17.

CRNT relies solely on the *structure* of the CRN to infer the existence and uniqueness of fixed points [200]. As a corollary, the method is unable to deal with bifurcating systems—whose qualitative steady state behavior depends on their parameters. Recent results based on algebraic geometry supplement the predictions of CRNT by locating the steady states of CRNs [201]. More generally, CRNT is still an on-going research area, and many recent works have proposed theorems applicable to larger classes of networks [202–204]. In future, these efforts may lead to simple and general methods for inferring in detail the steady state behavior of systems described using the CRN formalism.

5.3.5 The Well-Mixed Property

The epithet "Markovian" is often used abusively to describe physical phenomena; in fact, the Markov property is defined as a mathematical property that pertains to stochastic processes, and *not* to real physical systems (Section 5.3.1). However, one might be interested in defining the physical properties that allow a system to be accurately modeled by a collection of Markov processes. Hereafter, we consider the notion of "mixing", which can be informally thought of as a *spatial* memoryless property. In particular, a system is *well-mixed* if the position and velocity of a given agent at time $t + \Delta t$ are random quantities that do not depend on their values at time t. In practice, this property holds as long as the agents move fast enough relatively to the size of the environment and the various time scales of the system. For instance, if the average waiting time between two interactions is Δt, the average distance traveled by an agent in Δt must be in the order of the size of the environment.

The well-mixed property is an important justification of the use of macroscopic modeling in statistical physics. Indeed, since the system's state is completely defined by a population vector, the positions and velocities of individual particles must not influence the reaction rates, which translates into assuming that any pair of particles of species s_1 and s_2, respectively, has an identical probability of reacting in the next time interval $[t + \Delta t)$ (Figure 5.5). This in turn translates into stating that any particle of species s_1 has the same probability of encountering any other particle of species s_2 in the time interval $[t + \Delta t)$.

However, SMPs are often *not* well-mixed in the sense of the above definition. For instance, when the density of the system is large, reaction rates become too large for allowing individual agents to achieve mixing between two firing events (Section 8.2). Also, when dealing with more deterministic agents such as robots, and in absence of explicit randomization mechanisms (e.g., periodic tumbling), trajectories need to be randomized through non-reactive interactions. For instance, a system constituted of robots bouncing back and forth between obstacles in a deterministic fashion will not achieve

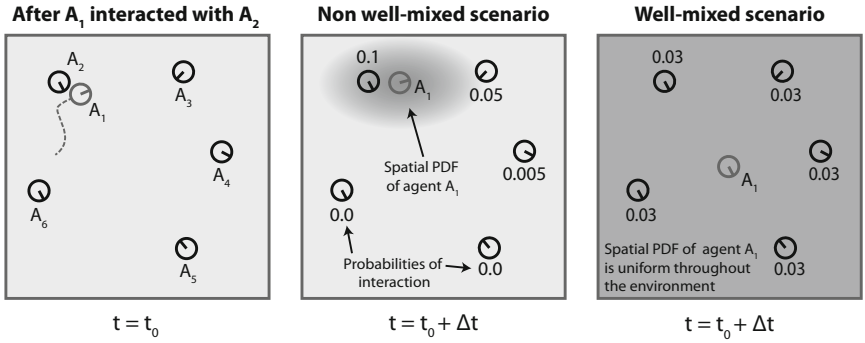


Fig. 5.5 Graphical illustration of the well-mixed property in an idealized scenario with 6 agents of the same species. For the sake of clarity, we assume that only agent A_1 is moving. At $t = t_0$, agent A_1 has just interacted with agent A_2. In a non well-mixed scenario, the probability for an agent A_i ($i \in [2, \ldots, 6]$) to be the next interaction partner of A_1 in the next time interval $[t + \Delta t)$ depends on their relative position at time t. In a well-mixed scenario, this probability is the same for all possible interaction partners.

good mixing because they will remain in a subpart of the arena. From this point of view, sensor and actuator noise, as well as randomization of the robots' initial positions, are important ingredients of good mixing multi-robot systems. Finally, long-range forces tend to hinder the mixing of SMPs by increasing the likelihood that recent interaction partners interact again (Section 9.4.2).

5.4 The Multi-Level Modeling Methodology (MLMM)

This section builds up on the two previous sections, and proposes a more robotics-focused approach to the construction of models at multiple abstraction levels. To this end, we rely on the seminal work of Martinoli and colleagues [48], who introduced the MLMM, a bottom-up approach to the construction of probabilistic models of distributed robotic systems, which relies on two fundamental governing principles. First, the robots' controller—more precisely, the associated FSM—serves as blueprint for building a hierarchical suite of models at increasing level of abstraction. Second, a consistent set of parameters—those used in the calculation of the system performance metric—are conserved throughout the whole process, and shared at all abstraction levels, thereby guiding the process of abstraction. While these principles were intuitively postulated by Martinoli and colleagues, it turns out that they address one important challenge outlined in previous sections, that is, the partitioning of the phase space. Indeed, the states of the FSM that governs the robots' behavior form essentially a specific partition of their state

space. Also, the conservation of parameters ensures that this partition is fine enough to allow for the evaluation of a performance metric at all abstraction levels. However, these principles provide no guarantee that the system is well-behaved within each macrostate (that is, transition times from and to this state follow an exponential distribution), which makes refinement of the partition often necessary.

One essential requirement of the MLMM is that a preliminary, non-optimized specification of the system is available at the beginning of the modeling process. Based on this prior "draft" of the system one can then build an initial model, generally at the submicroscopic level. For the most part, this first step consists in reproducing as accurately as possible the relevant features of the system in simulation. Of course, one needs in principle to decide which features are relevant to the system's dynamics; however, the capabilities of the simulation tool and the available computational resources have often more influence on this decision than any other consideration. This initial submicroscopic model is then used both for constructing and for calibrating more abstract and efficient models (Figure 5.6). As the level of abstraction increases, a series of approximations are made, thereby reducing the computational cost and, often, enhancing the analyzability of the model.

Interestingly, these approximations can be classified in a relatively systematic way. The transition **from submicroscopic to microscopic models** involves generally physical approximations: robots' trajectories are governed by simple kinematic laws rather than being the result of physical interactions, collisions are handled in an idealized fashion, and the robots are represented using elementary shapes or mere points. In non-spatial models, collisions are no longer explicitly simulated, but probabilistically emulated based on a series of geometric approximation (Section 7.2). Therefore, this transition from spatial to non-spatial microscopic models yields a collection of probabilistic FSMs. Often, an *ad hoc* coupling is required to account for the synchronicity of state transitions when they result from inter-robot interactions. The transition **from microscopic to macroscopic models** consists in aggregating all individual probabilistic FSMs into a single CRN, using a relatively systematic procedure described in Chapter 6. Finally, the resulting CRN can be readily used to derive either a macro-stochastic or a macro-deterministic model. Table 5.1 lists a typical hierarchy of models, in order of increasing level of abstraction, and the associated approximations that are typically used in the context of distributed robotic systems.

One important limitation of the MLMM as described in [48] is that the process of converting a given FSM into a hierarchy of models is mainly guided by the controller of the individual robots. However, in our case, the robots do not necessarily have a controller in the usual sense (e.g., ultra-small robots, cells, molecules, inert parts), or the relevant pieces of information about the robots' state are hidden at the controller level (e.g., the robots do not know the size of the aggregate they belong to in Case Study I). In such cases, the original MLMM supposes that the modeler will manually include

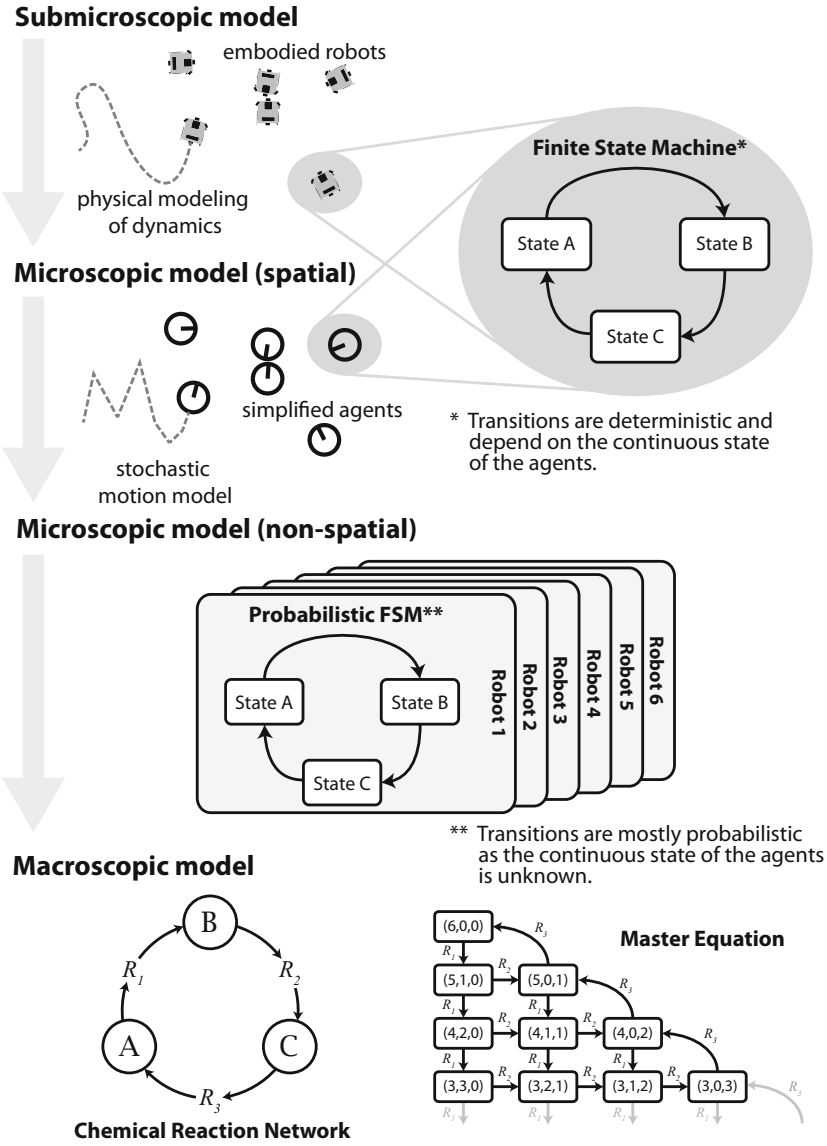


Fig. 5.6 Sketch of a typical procedure of model construction using the MLMM. Note in particular how the structure of the robots' controller is re-used at each abstraction level, up to the macroscopic level.

Table 5.1 A typical hierarchy of models and their respective approximations in the context of distributed robotics

Model	Approximations
Submicroscopic	Homogeneous group of robots, neglected physical phenomena, idealized noise models
Microscopic (spatial)	Simple kinematics, elastic collisions, abstraction of the robots' embodiment
Microscopic (non-spatial)	Well-mixed system, geometric approximation of probabilities
Macro-stochastic	Aggregation of individual states, semi-Markovian or pure Markovian process
Macro-deterministic	Negligible fluctuations, continuous populations

additional, relevant states based his understanding of the underlying systems. This methodological limitation is addressed in Chapter 9.

Summary and Conclusion

This chapter describes in detail the notion of model, and the many variants that can be used in the context of SMPs. In particular, we show how the richness of their dynamics motivate a combination of models at multiple levels of abstraction, as advocated by the MLMM, which was introduced in prior works. First, submicroscopic models capture the detailed, physico-chemical properties of the individual agents (e.g., shape, material, surface chemistry, charge, etc.), which determine the nature and the magnitude of their interactions. Second, microscopic models track the individual state of each agent, but they encapsulate all the details into a more compact state vector of lower dimensionality. Finally, macroscopic models aggregate all individual state vectors into a single stochastic process that describes the time evolution of the PDF over the phase space of the system. One may further distinguish macro-stochastic models—which account for all the moments of the PDF—and macro-deterministic models—which adopt a mean-field approach, and keep track only of the average trajectory of the system. Macroscopic models allow one to investigate, sometimes analytically, the collective properties of the system and their dependence on its design and control parameters.

As the level of abstraction increases, the role of stochastic processes as abstraction and simplification mechanism becomes more prominent. In microscopic models, stochastic processes are used to model the complicated and irregular trajectory of the individual agents in state space whereas interactions are handled deterministically. At the macroscopic level, however, every bit of the system is modeled probabilistically, thereby leading to a drastic reduction of the system's state space. However, this incremental, bottom-up approach to the construction of compact, computationally efficient models

poses two fundamental challenges. First, one needs to come up with a discretization of the state space that (i) does not prevent the performance metric of interest from being estimated, and that (ii) fulfills the underlying assumptions of the model (good mixing in particular). Second, one needs to calibrate the free parameters of the model, in which many lower-level, physico-chemical details are lumped.

The MLMM attempts to tackle both challenges at once by advocating an incremental, bottom-up construction of the models. In particular, the robots' controller serves as blueprint for partitioning the phase space of the system (Chapter 6) whereas the model's calibration is based on a series of geometrical approximations coupled with orthogonal experiments carried out at lower abstraction levels (Chapter 7).

6

Model Construction

The first principle is that you must not fool yourself,
and you are the easiest person to fool.
—Richard Feynman (1918–1988)
Caltech commencement address (1974)

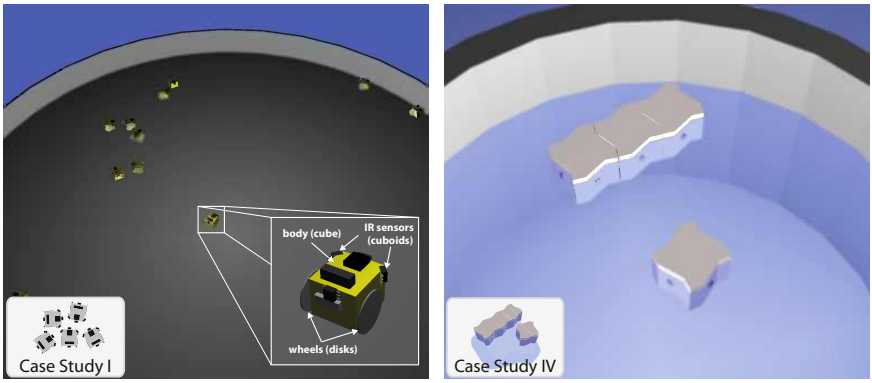
This chapter describes in detail the construction of models at multiple abstraction levels. Each section describes one given level of abstraction in general terms, using selected examples from the case studies described in Chapter 4. In particular, the variant A of Case Study V is featured at each abstraction level to illustrate concretely the incremental, bottom-up aspect of the MLMM.

6.1 Submicroscopic Models

As outlined in Section 5.4, the MLMM starts with a preliminary, non-optimized specification of the system. The first step consists therefore in translating this specification into a submicroscopic model, which is then used as baseline for devising and calibrating more abstract models. This process is greatly facilitated by the use of development environments such as Webots (Section 3.2.1), which enables the user to create custom robots and environments, and program them in a variety of languages (e.g., C, C++, Java, Matlab, Python). Hereafter, we discuss the construction of submicroscopic models using Webots in two scenarios:

1. Case Study I, which illustrates a standard development process.
2. Case Study IV, which features several difficulties related to the modeling of fluid dynamics.

G. Mermoud, *Stochastic Reactive Distributed Robotic Systems*,
Springer Tracts in Advanced Robotics 93,
DOI: 10.1007/978-3-319-02609-1_6,



(a) Alice robots (b) Lily modules

Fig. 6.1 Screenshots of submicroscopic models of (a) Case Study I (unstructured aggregation of Alice robots) and (b) Case Study IV (stochastic SA of Lily modules). Both models are implemented using Webots in spite of their very different underlying physics.

6.1.1 Example from Case Study I

Webots allows the user to add robots and solid objects to the simulation by importing them either from a pre-defined library or from custom CAD models. Then, the user needs to define their physical and visual properties (e.g., pose, shape, color,texture, mass, inertia matrix, friction coefficient, etc.). In Case Study I for instance (Figure 6.1a), the experimental arena is modeled as a solid disk (i.e., the floor) encircled by a series of narrow rectangles (i.e., the outer wall). Each Alice robot is modeled as as a set of solid objects forming the body: the main part is represented as a yellow cube; small cuboids represent the IR proximity sensors, the connector, and the IR receiver; two disks represent the wheels. Furthermore, each Alice robot is endowed with sensing and actuation modules (i.e., four proximity sensors and a differential steering) and a controller written in C.

Even though submicroscopic models are intended as very detailed and faithful representations of the system, many aspects of reality are left out in the submicroscopic models of Case Study I. We list hereafter a few examples of such approximations:

- The Alice robot is quite fragile and subject to rapid wear, such that each robot has a distinct and unique behavior. This heterogeneity is not reproduced at all in submicroscopic models, where all robots are exact copies of each other.

- Because Alice robots are so small and lightweight, even small objects such as grains of dust, hairs, or surface asperities that are not modeled at the submicroscopic level can affect their mobility.
- The two side IR sensors of the real Alice robot are mounted on a flexible circuit board while they are captured as fixed, solid cuboids in simulation.
- All IR proximity sensors are sensitive to lighting conditions, especially for communication purposes; this aspect is not accounted for at the submicroscopic level.

In general, these approximations have little influence on the accuracy of the simulations. However, in some scenarios (e.g., Case Study II), they can lead to important discrepancies (see Section 8.1 for more details).

6.1.2 Example from Case Study IV

In Case Study IV, the CAD model used for manufacturing the Lily modules is directly imported into Webots (Figure 6.1b), and their inertia matrix is computed using Autodesk Inventor™. However, an accurate simulation of the system requires to capture both the flow generated by the pumps *and* the interactions between the robots and the flow. Since Webots does not natively support fluidic dynamics modeling, a possible approach would consist in coupling Webots with a Computational Fluid Dynamics (CFD) tool such as the Lattice Boltzmann method [205]. However, this approach is not only computationally expensive, but also difficult to implement in order to ensure numerical stability of both simulations. Also, our system being essentially stochastic and subject to important asymmetries and manufacturing imperfections that may significantly affect the flow, an approach based on conventional CFD techniques is not a suitable choice.

As a result, we decided to adopt a completely different approach. Instead of trying to predict the flow resulting from a certain pump configuration, we use a method inspired by Particle Image Velocimetry (PIV) to extract the flow velocity field, and then use it in our submicroscopic simulations. This approach dramatically decreases the computational cost as compared to CFD methods. However, it requires a prior knowledge of the flow velocity field generated by each pump configuration of interest. Also, when extending the simulation to multiple modules, it does not account for the mitigation of the fluid flow due to the presence of other modules.

More concretely, we record the trajectory of a single Lily module during 30 minutes, and then construct a discrete regular velocity field by discretizing the trajectory plane into a regular grid, and averaging the observed velocity vectors at each cell of the grid. Choosing the number of divisions of the grid is a compromise similar to choosing the number of bins in a histogram: a too coarse grid may hide important features of the data, while a too fine grid could result in very few samples per division, and therefore less statistical

significance. We used a discretization of about 50 cells in each dimension for our arena (30 cm in diameter).

Once the velocity field is extracted, a correction is applied to compensate for the inertia of the block. This correction is based on a simple dynamical analysis considering the drag force, which is the only force acting in the horizontal plane. First, we compute the Reynolds number Re, which determines the flow regime of our system:

$$Re = \frac{\rho V L}{\mu} \sim 2000 \tag{6.1}$$

where $\rho = 10^3$ kg/m^3 is the density of water, $V \sim 6.4$ cm/s is the mean velocity of the object relative to the fluid (experimentally measured), $L = 3$ cm is the characteristic length of the Lily module, and $\mu = 8.90 \cdot 10^{-4}$ Pa·s is the dynamic viscosity of water. This value of Re is quite higher than the typical values ($Re < 10$) present at smaller scales [55], therefore prescribing the use of a quadratic drag force:

$$|\mathbf{F}_{\text{drag}}| = \frac{1}{2} \rho A C_x |\mathbf{v}_{\text{block}} - \mathbf{v}_{\text{flow}}|^2 \tag{6.2}$$

where $\mathbf{v}_{\text{block}}$ is the block's velocity, $\mathbf{v}_{\text{flow}}$ is the flow's velocity, A the block's cross sectional area to the flow, and C_x a dimensionless drag coefficient.

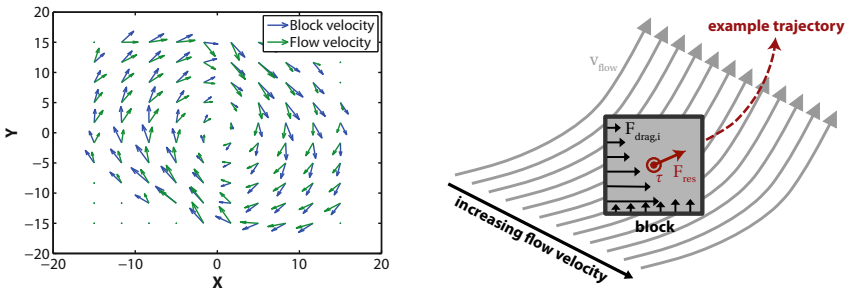
The direction of the drag force is opposed to the velocity of the block relative to the flow, as stated by:

$$\frac{\mathbf{F}_{\text{drag}}}{|\mathbf{F}_{\text{drag}}|} = -\frac{\mathbf{v}_{\text{block}} - \mathbf{v}_{\text{flow}}}{|\mathbf{v}_{\text{block}} - \mathbf{v}_{\text{flow}}|} \tag{6.3}$$

Assuming $A = 9$ cm^2 (i.e., the area of one block's side, since the block is almost completely immersed) and the drag coefficient of a cube $C_x = 1.05$, and taking into account that the mass m of the block can be measured, and its velocity $\mathbf{v}_{\text{block}}$ and acceleration $\mathbf{a}_{\text{block}}$ can be obtained from the tracked trajectories, the only unknowns are the x and y components of the flow velocity $\mathbf{v}_{\text{block}}$, which are given by the following equations:

$$v_{\text{flow},i} = v_{\text{block},i} + \frac{m \cdot a_{\text{block},i}}{\sqrt{\frac{1}{2} \cdot \rho \cdot A \cdot C_x \cdot m \cdot ||\mathbf{a}_{\text{block}}||}} \qquad i = x, y. \tag{6.4}$$

It must be noted in Figure 6.2a that the computed flow velocity has a centripetal component that generates the observed circular trajectory. If the correction of Equation 6.4 is not applied, the block constantly crashes onto the walls, which is not consistent with experimental observations. Note that, importantly, our simulation is completely independent from the method used for measuring the flow velocity; in this particular case, we used the same object for measuring the flow velocity and in our simulation, but this needs not to be the case in principle. For example, a regular PIV method could also be used to determine the flow velocity field.



(a) Velocity field and resulting flow (b) Torque and force due to a non-uniform fluid flow

Fig. 6.2 (a) Velocity field of the tracked block and calculated flow. (b) Torque τ and force $\mathbf{F}_{res}$ on a block in a non-uniform fluid flow.

Rigid body dynamics is not sufficient to simulate the motion of the blocks given the flow velocity. To solve this problem, we developed a physics plugin for Webots; this plugin applies the Archimedes' force, which is the weight of the fluid displaced by the block, and the drag force based on the flow velocity at the location of the block. The drag force is integrated over each face of the block to account for rotational effects. To this end, each face is divided into N planes, and the drag force is computed for each plane according to Equation 6.2 using appropriate values of A and C_x. On the one hand, the area A becomes the plane's cross sectional area to the flow (which depends on the plane's orientation relative to the flow). On the other hand, C_x is now the drag coefficient for each of the N planes. This coefficient is unknown and becomes a free parameter to be calibrated, which we denote C_d for the sake of clarity. The calculated force is not added to the block's center of mass but to the center of each plane, which allows for capturing the vorticity of the flow, e.g., the block will rotate when faced with a flow whose strength is increasing across one face, as shown in Figure 6.2b.

Importantly, the physics plugin also adds a stochastic force F_{stoch} to the center of mass of each block in order to account for non-modeled effects (e.g., physical irregularities, turbulences). The stochastic force F_{stoch} is a gaussian random variable with zero mean and standard deviation σ_F. As a result, the simulation has two parameters: the dimensionless drag coefficient C_d and the standard deviation σ_F of the stochastic force. Section 7.1.1 describes a trajectory-based method for calibrating these parameters.

6.2 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 that denote the internal properties of the individual robots. Hereafter, we illustrate the construction of two microscopic models: (1) a spatial ABM and (2) a non-spatial Discrete Event Simulation (DES), which were both developed[1] in the context of Case Study V (variant A). At the end of this section, we introduce a hybrid model of Case Study III that lies at the intersection of the microscopic and the macroscopic levels.

6.2.1 Example from Case Study V

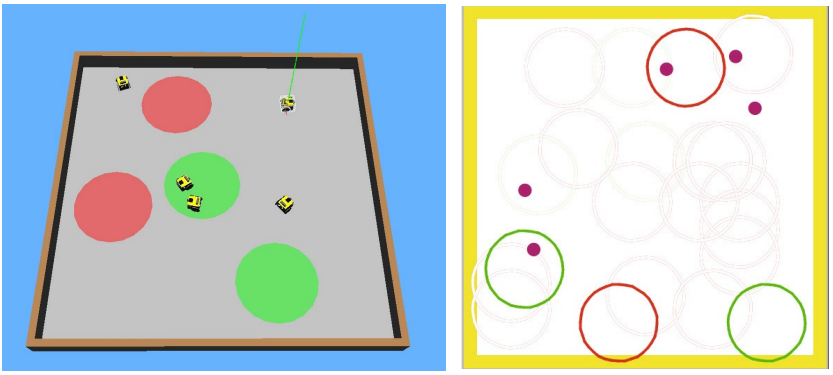
The MLMM prescribes that all control parameters that are relevant to the performance metric under investigation are conserved throughout the abstraction process. In our particular case, the leaving probabilities $p_{\text{good}}^{\text{leave}}$, $p_{\text{bad}}^{\text{leave}}$, and $p_{\text{agg}}^{\text{leave}}$ appear explicitly in both models presented hereafter. Also, the FSM associated to the robots' controller (Figure 4.8 on p. 52) is directly used to define the behavior of the agents in the ABM, and the state space in which the DES operates.

Agent-Based Model

While simplified as compared to its submicroscopic counterpart, the two-dimensional ABM represents the experimental setup and the robot cooperation rules with good accuracy. Particularly, the model accounts explicitly for the spatial extent of the robots, the spots, and the arena (see Figure 6.3). However, the exact shape of the robots is abstracted away; instead, they are represented as disks. In contrast to the submicroscopic model in which the physical units (e.g., space, time, force, velocity, etc.) are explicitly characterized, all quantities in this ABM are dimensionless, but scaled consistently with respect to the dimension of the agents. While the model captures the noise of the light sensors, some other sources of noise are neglected at this level, such as the noise on proximity sensors and wheel slip. Similarly, individual sensors and actuators are not captured. Instead, the neighborhood of each robot is screened for proximity detection, with sensing radii and coverage angles that accurately reproduce the overall sensing zone of an Alice robot. As a consequence, collision detection and aggregation are completely deterministic and are not characterized by a probability of encountering as it is the case at higher level of abstraction.

The model does not account for the specific kinematic constraints of the robots (i.e., they are assumed to be holonomic). Therefore, the obstacle avoidance procedure is implemented as a purely elastic collision in this model;

[1] These models were developed by Massimo Mastrangeli and William Chris Evans, respectively.



(a) Submicroscopic model (b) Spatial microscopic model

Fig. 6.3 (a) Screenshot of a submicroscopic model of Case Study V (variant A) implemented in Webots. (b) Screenshot of the spatial ABM described in Section 6.2.1. The robots are represented by magenta-filled dots. The large colored circles are the good (green) and bad (red) spots. Lighter circles are previously destroyed spots. The yellow band represents the limits of the arena.

this approach allows for a significant decrease of the overall computational cost. Except for the details discussed above, the agents behave according to the FSM depicted in Figure 4.8 (p. 52). The model was implemented using the open-source software package NetLogo [180].

Discrete Event Simulation

At a higher level of abstraction, we developed a DES that generates trajectories of individual robots in *discrete state space* by calculating the time until the next state transition occur according to the current state of the robots and their environment. In contrast to SSA [154], which is a macroscopic model, the DES presented here captures the state of each individual robot. The events are kept in a temporally sorted priority queue and then processed in order. We track dependencies between state changes in order to appropriately remove outdated events, e.g., when a robot leaves a spot, robots that remain in the spot should no longer collide with it. This approach allows us to, for the most part, ignore spatiality, yielding a significantly faster model (see Section 8.4).

More concretely, when a robot is placed in the arena, we calculate the time until its encounter with each spot as an exponentially distributed random variable with rate equal to the compounded spot encountering rate r_i^e (all rates discussed here are calibrated in Section 7.2). While we do not simulate spatiality explicitly, spots are tracked individually and robots only exist inside of one spot at a time. We use a rough geometric approximation to simulate

the time that a robot takes to cross a spot. While in more abstract models all events must follow a Poisson distribution, here we are able to use arbitrary, possibly parametric, distributions more appropriately suited to the specific phenomenon of interest. For instance, we compute the time t_b for a robot to cross a spot of diameter d_{spot} as follows:

$$t_b \sim U\left(\frac{d_{spot}/2}{v}, \frac{d_{spot}}{v}\right), \tag{6.5}$$

where v is the robot velocity, and $U(\cdot, \cdot)$ is the uniform distribution. Robots may leave a spot as usual at each bounce according to $p^{\text{leave}}_{\text{good}}$ and $p^{\text{leave}}_{\text{bad}}$.

Robots inside a spot will collide with each other much like they collide with spots, but at rate r^a_i or r^t_i for pairs and triplets, respectively. When the requisite number of robots has aggregated inside a spot it is destroyed, i.e., all robots are removed from the spot and placed back into the arena.

6.2.2 A Hybrid Monte-Carlo Model

In this section, we describe a MCM of Case Study III that keeps track, on one hand, of the population X_s of single building blocks in the system (macroscopic component) and, on the other hand, of the alignment of each individual aggregate (microscopic component). More concretely, since our model is non-spatial, collisions are not handled deterministically, but are randomly sampled from a Poisson distribution of mean $\lambda = p^{\text{join}} X_s$ where p^{join} is the probability that a given robot collides with another robot per unit time (see Section 7.2 for more details about the calibration of p^{join}). Each aggregate resulting from these collisions is individually captured using a Monte Carlo procedure: a random relative alignment $\xi_i = (\theta_{1,i}, \theta_{2,i})$ is generated and stored in a list $\boldsymbol{\Xi}_a$ (see Algorithm 1 below).

Algorithm 1. Pseudo-code of the hybrid Monte Carlo model described in Section 6.2.2.

Initialize $X_s = N_0$, $\boldsymbol{\Xi}_a = \emptyset$, and $t = 0$
while $t \leq t_{max}$ **do**
– Sample n_c the number of collision events from a Poisson distribution of mean $\lambda = p^{\text{join}} X_s (X_s - 1)$
– Generate and append to the list $\boldsymbol{\Xi}_a$ a set of n_c random relative alignments $\{\xi_1, \ldots, \xi_{n_c}\}$ with $\xi_i = (\theta_{1,i}, \theta_{2,i})$ and $\theta_{d,i} \sim U(0, \pi)$
– Generate N_a uniform random variates $r^s_i \sim U(0, 1)$ with $i = 1, \ldots, N_a$ and $N_a = \text{size}(\boldsymbol{\Xi}_a)$
– Compute n_b the number of aggregates in $\boldsymbol{\Xi}_a$ with ξ_i such that $r^s_i < p^{\text{leave}}(\xi_i)$ and remove them from $\boldsymbol{\Xi}_a$
– Let $X_s \leftarrow X_s + 2\,n_b - 2\,n_c$ and $t \leftarrow t + 1$
end while

This hybrid treatment allows for a great deal of flexibility. On one hand, the microscopic treatment captures only the relevant pieces of information about the individual aggregates, which can range from the number of building blocks to a full-fledged graph-based representation of the aggregate's topology. On the other hand, since the individual state of the single building blocks is not necessary (assuming that the system is well-mixed), a macroscopic treatment is sufficient for this species.

6.3 Macroscopic Models

The microscopic models described in the previous section are already the result of significant simplifications and state reductions with respect to their submicroscopic counterpart. However, they still suffer from some important shortcomings. First, they do not scale well with the number of robots. Second, their analytical tractability is poor. Macroscopic models address these intrinsic limitations of microscopic modeling.

As outlined in Section 5.4, the applicability of the original MLMM is limited when dealing with systems in which the relevant pieces of information about the robots' state are completely hidden at the controller level. In such cases, one usually tries to infer the structure of the macroscopic model directly from the collective dynamics, without referring to an underlying controller structure [23, 92]. Aggregation and SA are nearly paradigmatic examples of systems in which the robots have a limited knowledge of the quantities of interest to the modeler, such as the size or the geometry of the aggregates. Hereafter, we demonstrate the construction of several macroscopic models that capture these quantities by using two specific techniques: (i) state space augmentation and (ii) state space discretization.

6.3.1 State Space Augmentation

The behavior exhibited by the robots in Case Study I is simple: they are either moving or resting. The resting state can be further characterized by the number of robots within communication range, which in turn determines the probability p^{leave} of transition to the moving state. Yet, this piece of information is insufficient to track the size or the geometry of the clusters, which are generally of interest to the modeler in this type of scenario. Therefore, rather than constructing our macroscopic model based on the regular state space of the robots

$$\mathcal{S}^{\text{robots}} = \{\text{moving}, \text{resting}, \}$$

we use an *augmented* state space

$$\mathcal{S}^{\text{clusters}} = \{\text{moving}, \text{resting in cluster of size } 2, \ldots, \text{resting in cluster of size } N_0\}$$

that captures the size of the clusters.

We can then construct a macro-deterministic model that is capable of tracking the average number of robots in each state $s \in \mathcal{S}^{\text{clusters}}$, represented by a set of continuous variables $X_1, \ldots, X_{N_0}$. The model is based on a system of difference equations, where k denotes the current iteration (time step) and kT the actual time with T the sampling time[2]. Inflow and outflow of each state represent the proportion of robots switching to and from this state. They are given by the probability for a state transition to occur and the number of robots in other states. It is more natural—and mathematically equivalent—to track the average number of clusters of size j rather than the average number of robots in such clusters.

Following a mean-field approach, the average number $X_j(k+1)$ of clusters of size j (with $1 < j < N_0$) at time $k+1$ is therefore given by the following difference equation:

$$\begin{aligned} X_j(k+1) - X_j(k) = {} & f_{in,j}\big(\mathbf{P}^{\text{join}}(k), \mathbf{P}^{\text{leave}}(k), X_i(k)\big) \\ & - f_{out,j}\big(\mathbf{P}^{\text{join}}(k), \mathbf{P}^{\text{leave}}(k), X_i(k)\big) \\ & \qquad \text{with } i = 1, \ldots, N_0 \text{ and } i \neq j \end{aligned} \tag{6.6}$$

where functions $f_{in,j}$ and $f_{out,j}$ denote the inflow and the outflow of the state $X_j(k)$, i.e., the number of clusters of size j being formed or destroyed at time k. The matrices $\mathbf{P}^{\text{join}} = \big(p^{\text{join}}_{i,j}\big)$, and $\mathbf{P}^{\text{leave}} = \big(p^{\text{leave}}_{i,j}\big)$ denote both the connectivity and the transition probabilities between states X_i, X_j, and X_{i+j}. Namely, two clusters of size i and j, respectively, can aggregate into a cluster of size $i+j$ with probability $p^{\text{join}}_{i,j}$. Inversely, a cluster of size $i+j$ can split into two clusters of size i and j with probability $p^{\text{leave}}_{i,j}$. If there is no interaction between clusters of size i and j, then $p^{\text{join}}_{i,j} = p^{\text{leave}}_{i,j} = 0$.

As a result, 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:

$$f_{in,j}(\ldots) = \sum_{i=1}^{j-1} p^{\text{join}}_{i,j-i}(k)\, X_{j-i}(k)\, X_i(k) + \sum_{i=j+1}^{N_0} p^{\text{leave}}_{i-j,j}(k)\, X_i(k), \tag{6.7}$$

$$f_{out,j}(\ldots) = \sum_{i=1}^{N_0-j} p^{\text{join}}_{i,j}(k)\, X_i(k)\, X_j(k) + \sum_{i=1}^{j-1} p^{\text{leave}}_{i,j-i}(k)\, X_j(k). \tag{6.8}$$

Terms of the form $p^{\text{join}}_{i,j}(k)\, X_i(k)\, X_j(k)$ correspond to the number $X_i(k)$ of clusters of size i that join one cluster of size j at time k with a probability $p^{\text{join}}_{i,j}(k)\, X_j(k)$, and form a cluster of size $i+j$. Terms of the

[2] Hereafter, we leave T out of the equations for the sake of simplicity. Note that it should be chosen small enough in comparison to the time constants of the system.

form $p_{i,j}^{\text{leave}}(k)\, X_{i+j}(k)$ denotes the number of clusters of size $i+j$ that split into clusters of size i and j at time k with probability $p_{i,j}^{\text{leave}}(k)$.

In the general case, one should take into account all possible N_0-1 pairwise combinations of clusters that lead to the formation of a cluster of size j. Fortunately, in Case Study I, the robots remain still once aggregated, and there are therefore only two ways[3] of forming a cluster of size j:

$$X_{j-1} + X_1 \rightarrow X_j \qquad\qquad X_{j+1} \rightarrow X_j + X_1. \tag{6.9}$$

In such cases, we have that $p_{i,j}^{\text{join}} \neq 0$ if and only if $i = 1$ or $j = 1$. Of course, this assumption dramatically simplifies the complexity of the model, both in terms of computation and memory usage. Figure 6.4 depicts the state transition diagram of this specific model.

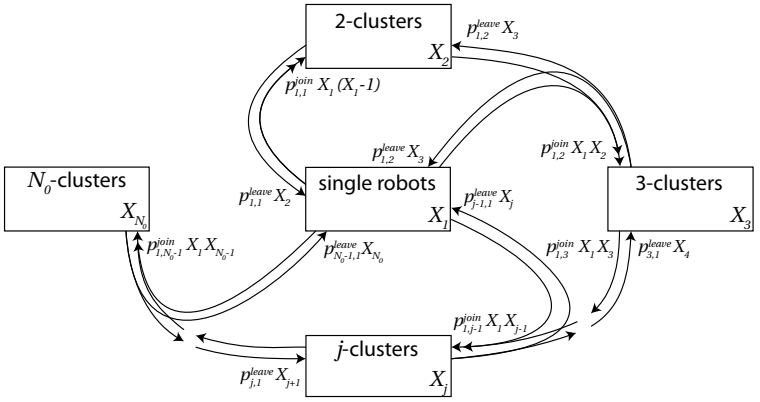


Fig. 6.4 State transition diagram of the aggregation model with stationary clusters. Only single robots (state variable X_1) can interact with each other, and with clusters (state variables X_i with $i = 2, \ldots, N_0$).

Macroscopic models can also track properties of the clusters other than their size (i.e., the number of building blocks), such as their geometry. In the above model, we assume that clusters have no embodiment (see Section 2.1.1). For instance, robots cannot be stuck in the middle of a cluster, surrounded by other robots. Also, nearby clusters never connect with each other because of one robot joining them; similarly, clusters never split into two sub-clusters because one robot left. These scenarios, depending on the structure of the clusters, may happen in reality, but the particular model depicted in Figure 6.4 does not account for them.

[3] We neglect the case of clusters 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 cluster in a given time step, if the model is time-discrete.

6.3.2 State Space Discretization

In some scenarios, the only available description of a part (or the whole) of the system is an abstract, continuous law. For instance, in distributed coverage scenarios, an overarching continuous law may describe the steady-state spatial distribution $f(x, y)$ of the robots. In such cases, one needs to find an appropriate set of discrete states and corresponding dynamic equations such that this continuous law is reproduced (e.g., following a discretization of the environment, one needs to find the transition rates between cells that will eventually lead to $f(x, y)$). In other words, the level of abstraction with respect to the initial description of the system is lowered, thereby allowing for the incorporation of more details, if available. For instance, transition rates can be parameterized by the velocity and the mobility pattern of the robots. This procedure is relevant in the context of microscale systems, whose low-level properties are often either unknown or lumped into abstract, empirical distributions. By moving down the hierarchy of models, one actually aims to infer lower-level properties of the underlying system, which can then be calibrated and, to some extent, validated using these models. This thesis just scratches the surface of this different approach to multi-level modeling.

In Case Study III, the stability of an aggregate is described by a continuous function of the alignment of its building blocks—more specifically, the bond energy $\Delta E \in \mathbb{R}_{\leq 0}$ (Equation 4.4). As a result, the notion of directionality plays a key role in the system's dynamics, and cannot be abstracted away at the macroscopic level. Therefore, we need to discretize the state space of the aggregates in order to account for the different levels of bond energy they can be in. The fraction of formed aggregates that have a particular energy level, and the transition rates between energy levels (through self-alignment, see [206]), typically depend on details such as the geometry and the surface properties of the building blocks, which may be gradually incorporated into the models as their level of abstraction *decreases*.

The state of an aggregate is fully determined by the relative positioning ξ of its building blocks, which is a two-dimensional vector in the 2-DOF variant. Fortunately, the symmetry of Equation 4.4 allows one to simplify this definition to a scalar, that is, the norm of the relative positioning, denoted

$$\theta^2 = ||\xi||^2 = \theta_1^2 + \theta_2^2 \in [0, 2\pi^2], \tag{6.10}$$

which can be easily discretized into a set of K averaged values $\widehat{\theta_i^2}$ given by

$$\widehat{\theta_i^2} = \left(i - \frac{1}{2}\right) \cdot \frac{2\pi^2}{K} \quad \text{with } i = 1, 2, \ldots, K. \tag{6.11}$$

Therefore, one can define $\mathcal{S}_d = \{s_0, s_i, \ldots, s_K\}$ as the discretized space of aggregate's types, with s_0 representing single building blocks and s_i aggregates with an average relative alignment $\widehat{\theta_i^2}$ and the following bond energy:

$$\Delta E(s_i) = E_{\text{bond}} \cdot \exp\left(\frac{\widehat{\theta}_i^2}{2\,\sigma_\theta^2}\right). \tag{6.12}$$

The average proportion of single building blocks x_s is then given by the following difference equation:

$$x_s(k+1) - x_s(k) = 2\,\langle \mathbf{p}^{\text{leave}}\, \mathbf{x}_p(k)\rangle - p^{\text{join}} \cdot x_s(k)^2 \tag{6.13}$$

with

$$\mathbf{p}^{\text{leave}} = \left[p^{\text{leave}}(s_1), \ldots, p^{\text{leave}}(s_K)\right]^T$$
$$\mathbf{x}_p(k) = \left[x_1(k), \ldots, x_K(k)\right]^T$$

where $\langle \cdot\ \cdot \rangle$ denotes the scalar product, and $p^{\text{leave}}(s_i)$ is the probability that an aggregate of type s_i is destroyed:

$$p^{\text{leave}}(s_i) = \exp\left(\frac{\Delta E(s_i)}{\alpha\,\nu_s^2}\right). \tag{6.14}$$

The scalar term $\langle \mathbf{p}^{\text{leave}}\, \mathbf{x}_p(k)\rangle$ is the average proportion of aggregates that were destroyed at iteration k. The term $p^{\text{join}} \cdot x_s(k)^2$ is the average proportion of building blocks that collided and formed a pair at iteration k. Similarly, the proportion of pairs of type s_i with $i = 1, \ldots, K$ is given by the following difference equation:

$$x_i(k+1) - x_i(k) = f(i) \cdot \frac{p^{\text{join}} \cdot x_s(k)^2}{2} - p^{\text{leave}}(s_i) \cdot x_i(k) \tag{6.15}$$

where $f(i) : \mathbb{Z}_+ \to [0,1]$ is a function that denotes the fraction of formed aggregates that are actually of type s_i; $p^{\text{leave}}(s_i) \cdot x_i(k)$ is the average proportion of pairs of type s_i that broke up at iteration k; $p^{\text{join}} \cdot x_s(k)^2$ is the average proportion of building blocks that collided and formed an aggregate at iteration k. Since two building blocks are needed to form an aggregate, this term is divided by two.

The function $f(i)$ shall be a discretized version of the PDF of the random variable Z, which denotes the probability that a formed aggregate has a relative positioning norm $\theta^2 \in [0, 2\pi^2]$. Now, since $\theta^2 = \theta_1^2 + \theta_2^2$, one can write $Z = X^2 + X^2$, where X is a random variable that denotes the probability that θ_1 or θ_2 take a specific value in $[0, \pi]$. We assume that X is uniformly distributed, i.e. $X \sim U(0, \pi)$. Therefore, according to [207], we have $X^2 \sim \text{Beta}(0, \pi, \frac{1}{2}, 1)$ and its PDF is given by

$$f_{X^2}(x) = \frac{(x/\pi)^{-1/2}}{\int_0^\pi (u/\pi)^{-1/2}\,du} = \frac{1}{2}\left(\frac{x}{\pi}\right)^{-1/2} \quad \text{with } x \in [0, \pi^2].$$

The PDF of $Z = X^2 + X^2$ is given by the convolution of the PDF of X^2 with itself [207]:

$$f_Z(z) = f_{X^2}(z) \star f_{X^2}(z) = \frac{1}{4} \int_{-\infty}^{\infty} \left(\frac{z-\tau}{\pi} \right)^{-1/2} \left(\frac{\tau}{\pi} \right)^{-1/2} d\tau. \tag{6.16}$$

Since $z \in [0, 2\pi^2]$, we need to normalize $f_Z(z)$ such that

$$f_Z(2\pi^2) = 0 \quad \text{and} \quad \int_{-\infty}^{\infty} f_Z(z)\, dz = 1,$$

which leads to

$$f_Z(z) = \begin{cases} \frac{1}{4\pi} & \text{if } z \in [0, \pi^2] \\ \frac{1}{\pi^2} \arctan\left(\frac{\pi}{\sqrt{z-\pi^2}} \right) - \frac{1}{4\pi} & \text{if } z \in [\pi^2, 2\pi^2] \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, the function $f(i)$ is given by

$$f(i) = \int_{l(i)}^{u(i)} f_Z(z)\, dz \quad \text{with} \quad \begin{cases} l(i) = \frac{2\pi^2}{K}(i-1) \\ u(i) = \frac{2\pi^2}{K} i \end{cases} \tag{6.17}$$

with $l(i)$ and $u(i)$ the lower and upper bound, respectively, of the i-th subinterval of $[0, 2\pi^2]$.

6.3.3 From Finite State Machines to Chemical Reaction Networks

In Case Study V, the robots are endowed with a full-fledged controller, thereby allowing us to transform the corresponding FSM into an equivalent CRN by converting states and state transitions into species and reactions, respectively (Figure 6.5). The states of the FSM that are irrelevant to the system's dynamics (because the robot spends a negligible amount of time in them, or because they are not relevant to the performance metric under investigation) are *merged* into a unique species of the CRN. For instance, even though obstacle avoidance is a state of the robots' controller, it is not explicitly modeled as such at the macroscopic level; rather, we assume that a robot seamlessly switch from wandering to obstacle avoidance state.

Since we are interested in the ability of aggregation to deal with the noise in decision-making, we explicitly incorporate belief representation in our model. More specifically, the state of a robot is not only defined by its internal state, i.e., whether it is wandering, exploring a spot, or part of an aggregate, but also whether its internal state is a correct representation of reality. To this end, we refine the CRN following an approach similar to those described in Section 6.3.1. For instance, since the model needs to track the number of robots in each spot, and whether their estimate is correct or not, the species *in spot* is split into $2\,N_{\text{spots}}$ sub-species.

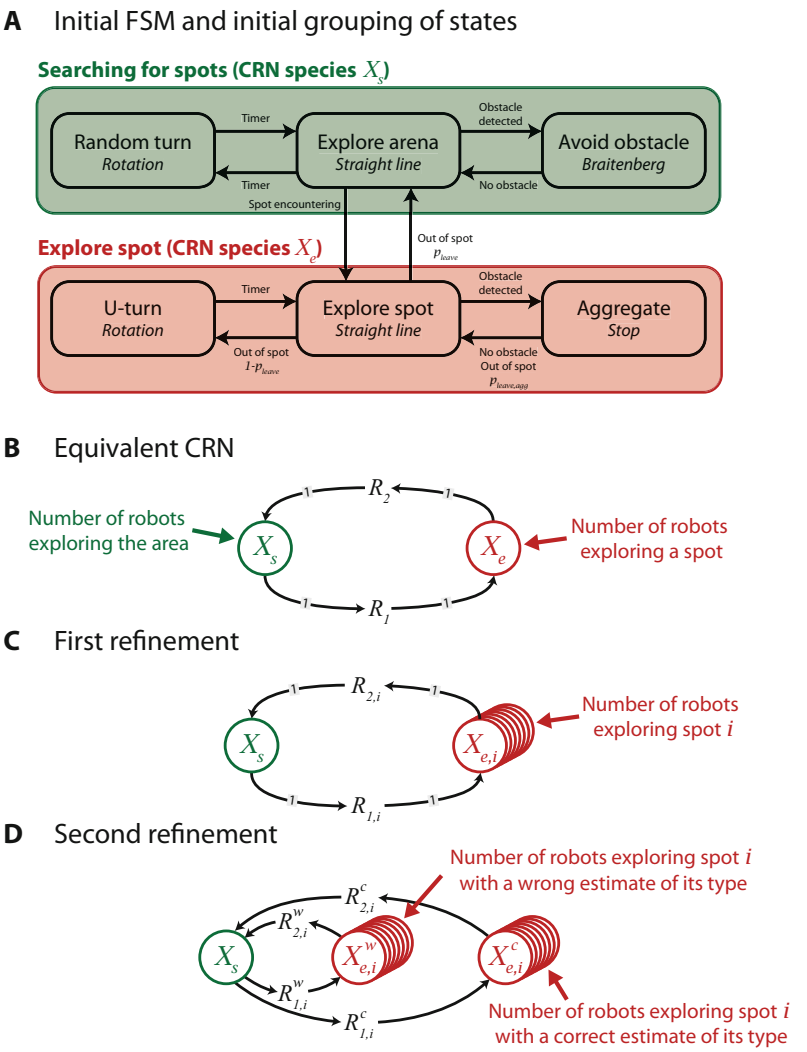


Fig. 6.5 Sketch of the translation of a FSM into a CRN, which consists in grouping the states that are not relevant to the system's dynamics (step **A**) and constructing the equivalent CRN (step **B**). For instance, the states *u-turn*, *explore spot*, and *aggregate* are merged into a single species X_e. Then, one can iteratively refine the CRN by splitting the species that need to account for states that are hidden at the controller level into multiple sub-species. In this case, the model needs to track the number of robots in each spot, and whether their estimate is correct or not. Therefore, the species X_e is split into $2 \cdot N_{\text{spots}}$ sub-species $X^w_{e,i}$ and $X^c_{e,i}$ (steps **C** and **D**).

Based on the FSM depicted in Figure 4.8 (p. 52), and using the approach described earlier, we build up models of the collective perception experiment, each corresponding to a different value of k. Since these models are built in an incremental fashion, we shall start by describing in details the model for $k = 1$, which is the basis for other models where $k > 1$.

In the case of $k = 1$, a robot can trigger the destruction of a spot on its own. Therefore, we shall distinguish between the following species:

- Robots searching for spots: X_s
- Robots in spot i with a correct estimate of its type: $X_{c,i} \quad \forall i \in \mathcal{S}$
- Robots in spot i with a wrong estimate of its type: $X_{w,i} \quad \forall i \in \mathcal{S}$

where $\mathcal{S}$ denotes the set of spots in the system. Equation 6.18 provides a complete view of the $4 \cdot |\mathcal{S}|$ reactions of the CRN:

$$\begin{aligned} X_s &\xrightarrow{r^e_{c,i}} X_{c,i} & X_{c,i} &\xrightarrow{r^l_{c,i}+r^d_{c,i}} X_s \\ X_s &\xrightarrow{r^e_{w,i}} X_{w,i} & X_{w,i} &\xrightarrow{r^l_{w,i}+r^d_{w,i}} X_s \qquad \forall i \in \mathcal{S} \end{aligned} \tag{6.18}$$

where $r^e_{c,i}$ and $r^e_{w,i}$ are the rates at which a robot encounters spot i and correctly or wrongly identifies its type, respectively. Similarly, $r^l_{c,i}$ and $r^l_{w,i}$ is the rate at which a robot leaves the spot it is exploring and believes to be of a given type, either correctly or wrongly, respectively, without destroying it. $r^d_{c,i}$ and $r^d_{w,i}$ are the same as $r^l_{c,i}$ and $r^l_{w,i}$, except that the robot destroys the spot in this case.

When collaboration is introduced ($k = 2$), one needs to account for the fact that two robots are needed to trigger the destruction of a spot. It has two main implications: (i) robots can change their belief about the type of the spot they are exploring at a rate $r^m_{w,i}$ (from a correct to wrong belief), and $r^m_{c,i}$ (from a wrong to a correct belief), and (ii) aggregation introduces non-linear reactions, i.e., reactions that involve more than one robot, whose rate is r^a_i. Equation 6.19 provides a complete view of the CRN:

$$\begin{aligned} X_s &\xrightarrow{r^e_{c,i}} X_{c,i} & X_{c,i} &\xrightarrow{r^l_{c,i}+r^d_{c,i}} X_s & X_s &\xrightarrow{r^e_{w,i}} X_{w,i} \\ X_{w,i} &\xrightarrow{r^l_{w,i}+r^d_{w,i}} X_s & X_{c,i} &\xrightarrow{r^m_{w,i}} X_{w,i} & X_{w,i} &\xrightarrow{r^m_{c,i}} X_{c,i} \\ 2\,X_{c,i} &\xrightarrow{r^a_i} 2\,X_s & 2\,X_{w,i} &\xrightarrow{r^a_i} 2\,X_s & X_{c,i} + X_{w,i} &\xrightarrow{2\,r^a_i} 2\,X_s \end{aligned} \tag{6.19}$$

for all $i \in \mathcal{S}$.

Finally, if more than two robots are needed to trigger the destruction of a spot, it means that those robots that are part of a pair $X_{a,i}$ may remain idle in spot i, waiting for a third robot to join up, which is an event that happens at a rate r^t_i. The robots can also decide to disaggregate at a rate r^d_i, thus leading to the following CRN:

$$
\begin{array}{lll}
X_s \xrightarrow{r^e_{c,i}} X_{c,i} & X_{c,i} \xrightarrow{r^l_{c,i}+r^d_{c,i}} X_s & X_s \xrightarrow{r^e_{w,i}} X_{w,i} \\
X_{w,i} \xrightarrow{r^l_{w,i}+r^d_{w,i}} X_s & X_{c,i} \xrightarrow{r^m_{w,i}} X_{w,i} & X_{w,i} \xrightarrow{r^m_{c,i}} X_{c,i} \\
2\,X_{c,i} \xrightarrow{r^a_i} X_{a,i} & 2\,X_{w,i} \xrightarrow{r^a_i} X_{a,i} & X_{c,i}+X_{w,i} \xrightarrow{2\,r^a_i} X_{a,i} \\
X_{a,i} \xrightarrow{r^d_i} 2\,X_s & X_{c,i}+X_{a,i} \xrightarrow{2\,r^t_i} 3\,X_s & X_{w,i}+X_{a,i} \xrightarrow{2\,r^t_i} 3\,X_s
\end{array}
\tag{6.20}
$$

for all $i \in \mathcal{S}$. Note that, while we do not study models for $k > 3$, these are relatively easy to derive from the model for $k = 3$.

Summary and Conclusion

This chapter outlines both the benefits and the limitations of the MLMM for constructing models at multiple abstraction levels: (i) the crucial role of the modeler's expertise at each step of the modeling process, and (ii) the tight correlation between abstraction and probabilistic modeling, which in turn stresses the importance of stochastic processes as simplification and abstraction mechanisms.

Submicroscopic models generally proceed from a direct translation of the target system's specifications. While some details are abstracted away throughout this process, there is generally no methodological motivation for doing so; rather, the absence of support in the modeling tool, or the prohibitive difficulty in implementing these details are the primary motivation. Often, these simplifications have little or no impact on the model's accuracy; for instance, the absence of CFD support in Webots does not pose any problem when modeling a group of Alice robots (case studies I, II and V). However, when dealing with fluidic SA (case studies III and IV), one needs to account for the most important hydrostatic and hydrodynamic effects observed in such systems. Stochastic processes may become necessary when these effects are too difficult to capture explicitly (e.g., physical irregularities and flow turbulences are captured by a stochastic force in Section 6.1.2).

At the microscopic level, the abstraction process is no longer governed by implementation-related considerations. The construction of a microscopic model essentially boils down to determining a state vector of the individual robots significantly more compact than its submicroscopic counterpart. As a matter of fact, the MLMM does not provide any algorithmic method for determining this state vector (this limitation is addressed in Chapter 9). As a rule of thumb, we shall retain only those pieces of information that are required by the robot controller. However, many features that are hidden at the controller level still need to be conserved at the microscopic level because the system's dynamics depend on them implicitly. For instance, robots may ignore their position in space, but microscopic models still need to account for spatiality in order to deal with robots' interactions deterministically. As a

corollary, one may design non-spatial microscopic models by using a stochastic treatment of robots' interactions when appropriate (i.e., when the system is well-mixed).

Macroscopic modeling pushes the envelope of abstraction by assuming that the whole system can be described accurately using stochastic processes. The MLMM prescribes a nearly direct translation of the FSM of the robots' controller into an equivalent CRN. The species and reactions of the CRN represent the different states and state transitions of the FSM, respectively. The CRN is then iteratively refined either by lumping the species that are not relevant to the system's dynamics, or by splitting the species that need to account for states that are hidden at the controller level into multiple sub-species.

7

Model Calibration

We know what [...] to do experimentally to measure this number very accurately, but we don't know what [...] to do on the computer to make this number come out, without putting it in secretly!
—Richard Feynman (1918–1988)
on the numerical value of α, the fine-structure constant
in *QED : The Strange Theory of Light and Matter* (1985)

Any model, regardless of its nature, is characterized by parameters, i.e., unknown quantities that need to be determined in order to solve the model. Often, parameters are directly measurable on the target system (e.g., the weight, the size, or the speed of a robot), they can be obtained from data sheets or computed from physical laws (e.g., friction and restitution coefficients, inertia matrices, etc.). However, more abstract models might involve so-called *free* parameters that result from the lumping of several phenomena, some of which might be difficult to model explicitly, thereby rendering direct approaches inapplicable. In such cases, more sophisticated numerical methods and machine learning techniques are necessary.

The MLMM prescribes that model parameters shall be conserved throughout the whole process of abstraction, thereby yielding macroscopic models with zero free parameter [48]. Even upon moving from spatial to non-spatial models, one can use closed-form geometrical approximations based on known properties of the system (Section 7.1.2). However, in scenarios such as Case Study IV, where the underlying physics cannot be fully accounted for even at the submicroscopic level, one needs to introduce free parameters, which then require special care in terms of calibration.

7.1 Calibration of Submicroscopic Models

Because of their high level of detail, submicroscopic models have generally a large number of parameters, which are however relatively simple to calibrate owing to their direct anchoring to reality. Hereafter, we illustrate briefly the calibration of physical parameters of Case Study I using direct measurements and orthogonal experiments (Section 7.1.1). Then, we describe a novel

G. Mermoud, *Stochastic Reactive Distributed Robotic Systems*,
Springer Tracts in Advanced Robotics 93,
DOI: 10.1007/978-3-319-02609-1_7,

calibration method based on the principle of minimizing the distance between *simulated* and *real* trajectories. This method is used for determining the drag coefficient C_d and the standard deviation σ_F of the stochastic force in Case Study IV (Section 7.1.2).

7.1.1 Example from Case Study I

Because we use Webots in physics mode, one need to carefully adjust the physical properties of all objects forming the Alice robots and the environment. For instance, the weight of the Alice robot as well as the friction coefficients between the wheels and the ground are particularly important for a faithful reproduction of phenomena such as wheel slip and obstructions. The IR sensor characteristics (aperture, range, and non-linear transfer functions) of the Alice robot are also critical in the context of aggregation and SA experiments. These parameters have been carefully calibrated in previous works using a combination of direct measurements and systematic experiments with real robots [208].

7.1.2 Example from Case Study IV

When it comes to characterizing the trajectory of a robot, it is often practical to use one of these alternative assumptions: (i) the robot follows a deterministic trajectory defined by a given control law (e.g., [19]), or (ii) the robot performs a random walk with some known average speed that can be mapped to some diffusion coefficient (e.g., [209]). However, there are many situations, such as in Case Study IV, where the reality lies between these two extremes.

In two dimensions, sampled trajectories can be viewed as sequences of points in a two-dimensional space: there is no need to consider the temporal dimension as we are using a constant sampling rate. Furthermore, trajectories do not have a common frame because there are no reference points or pre-established paths, just the blocks reacting to the environment. Therefore, it is not appropriate to compare trajectories in the Euclidean space because, even though they are generated by the same agitation mechanism and should therefore exhibit common traits, the sequence of points will vary greatly depending on initial conditions and random collisions with walls. A solution proposed by Roduit [210] is based on the Correlated Random Walk (CRW) model, which represents trajectories as a succession of steps whose directions are correlated with each other (Figure 7.1). A step is defined as the segment that connects two points of the trajectory, sampled at time t and $t + \Delta t$, respectively. Then, any trajectory can be represented as an ensemble of step lengths S_i and step angles A_i, which in turn can be modeled as a two-dimensional histogram (Figure 7.2).

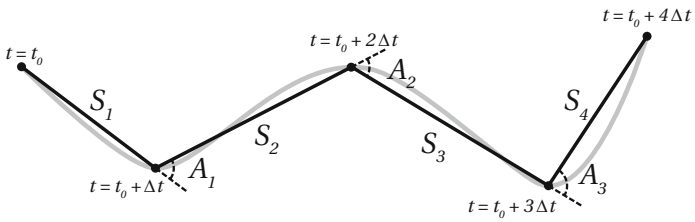
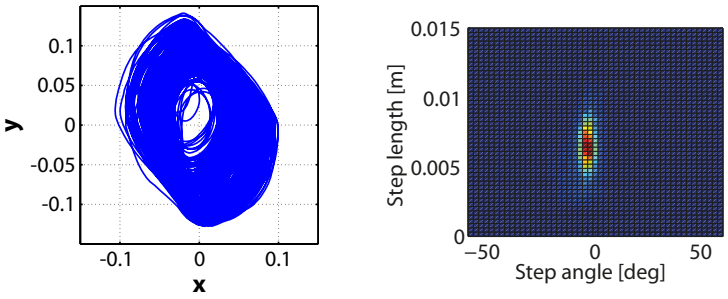


Fig. 7.1 A two-dimensional trajectory (in gray) sampled at regular times $t_0, t_0 + \Delta t, \ldots, t_0 + n\,\Delta t$ is represented as a sequence of step lengths S_i and angles A_i



(a) A sample trajectory

(b) Corresponding two-dimensional histogram

Fig. 7.2 (a) A sample trajectory. (b) Its corresponding two-dimensional distribution of step lengths and step angles.

In order to compare real and simulated trajectories represented as histograms of step lengths and step angles, we use the Kolmogorov-Smirnov (KS) statistic (or distance), which corresponds to the maximum distance between the corresponding cumulative distribution functions. The justification for the use of the KS distance is that the KS statistical test is non-parametric (i.e, no assumption is made about the underlying distributions), and it is sensitive to both the location and the shape of the empirical distribution functions. Therefore, our calibration method does not only try to match the mean linear and angular velocity of the original trajectory, but also the higher moments of the two-dimensional distribution.

When extending the KS test to two dimensions, the calculation of the cumulative distribution function is not as straightforward as in the one-dimensional case, as there are four distinct ways of cumulating data along the directions of the coordinate axes. In our implementation[1], we followed the Fasano and Franceschini variation presented in [211], which has been shown

[1] The calibration procedure was implemented by Ezequiel di Mario.

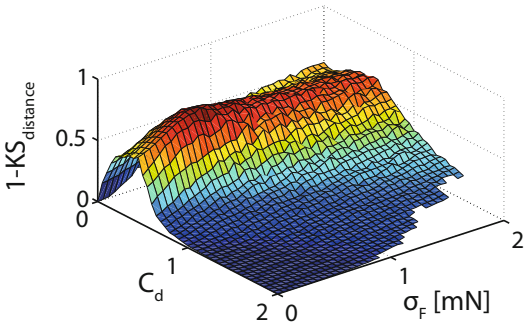
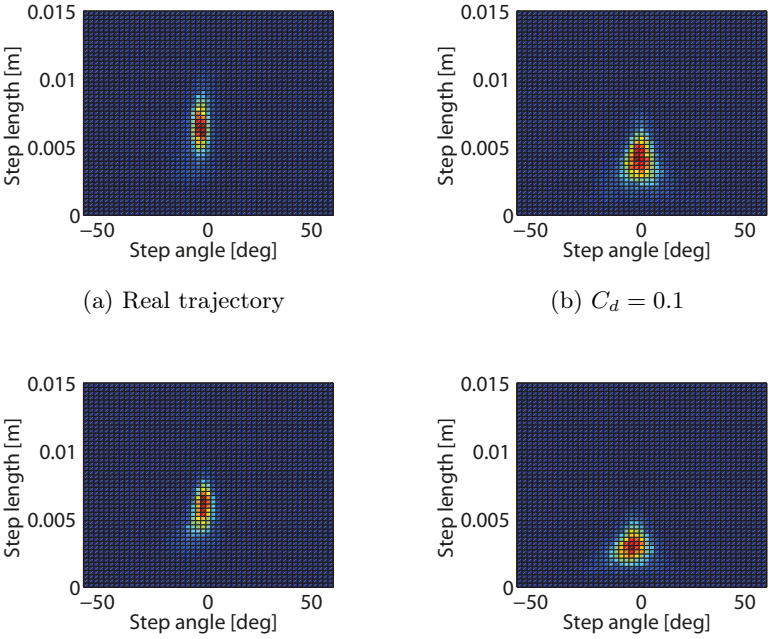


Fig. 7.3 Plot of $1 - \mathrm{KS}_{\mathrm{distance}}$ between real and simulated trajectories as a function of the drag coefficient C_d and the standard deviation σ_F of the added stochastic force. Missing data in the plot are due to numerical instabilities for too large values of C_d and F_{stoch}.



(a) Real trajectory

(b) $C_d = 0.1$

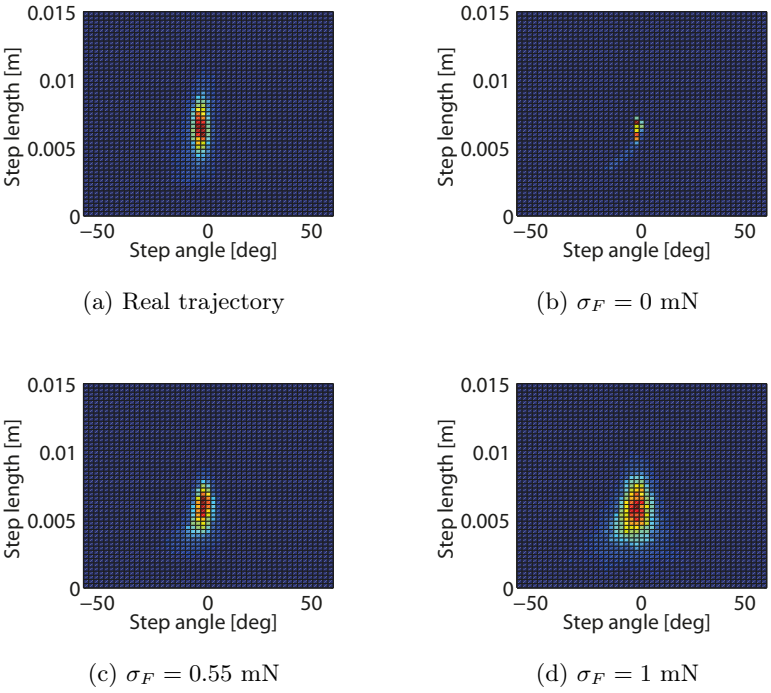
(c) $C_d = 0.45$

(d) $C_d = 0.8$

Fig. 7.4 Step length and angle distributions for different drag coefficients. (a) Real trajectory. (b) $C_d = 0.1$. (c) $C_d = 0.45$. (d) $C_d = 0.8$.

to reduce the computational complexity from $O(n^3)$ in Peacock's original version to $O(n^2)$ without sacrificing the test's power to distinguish dataset differences.

Finally, we systematically explore the parameter space to find the optimal modeling parameters, i.e., those that minimize the KS distance between the



(a) Real trajectory

(b) $\sigma_F = 0$ mN

(c) $\sigma_F = 0.55$ mN

(d) $\sigma_F = 1$ mN

Fig. 7.5 Step length and angle distributions for different standard deviations of the stochastic force. (a) Real trajectory. (b) $\sigma_F = 0$ mN. (c) $\sigma_F = 0.55$ mN. (d) $\sigma_F = 1$ mN.

simulated and real trajectories. Due to the small number of parameters in our case study, there is no need for more complex optimization techniques. However, more complex optimization methods can be used in principle, as long as they are noise-resistant [212].

More specifically, we vary the drag coefficient C_d between 0 and 2 with increments of 0.05, and the standard deviation of the stochastic force F_{stoch} from 0 mN to 2 mN with increments of 0.05 mN. For each set of parameters, we simulate a trajectory of 30 minutes, and compute the KS distance between the resulting simulated trajectories and a real, pre-recorded trajectory using all pumps at full power, with two pumps directly connected to opposite perpendicular inlets and two other connected to both one perpendicular and one tangential inlet each. All trajectories are sampled at the same rate (10 Hz). Figure 7.3 shows a plot of $1 - \mathrm{KS}_{\mathrm{distance}}$ for improved visualization of the optimum.

The KS distance is minimized for a drag coefficient $C_d = 0.45$, and a standard deviation of the stochastic force $\sigma_F = 0.55$ mN. One can observe a ridge along the drag coefficient axis, which indicates that the faithfulness of

the simulation rapidly decreases when leaving the [0.3 0.5] range for the drag coefficient. Values for the stochastic force and the drag coefficient that are too large lead to numerical instabilities.

The qualitative effects of different values for the drag coefficient are shown in Figure 7.4. If the drag coefficient is too low, the block does not follow the flow: the mean step length becomes shorter, and the mean angle is around zero, as opposed to the negative step angles of the real trajectory that show the prevailing turning direction of the circular flow. Conversely, high values of the drag coefficient cause the block to be dragged to the center of the tank, and eventually to rotate almost in place, thereby resulting in short step lengths and negative step angles due to the rotation. Regarding the stochastic force (Figure 7.5), low values of the standard deviation lead to regular trajectories, and the resulting distribution shows little variance. However, if the stochastic force is too high, the trajectories become very irregular and similar to those observed in Brownian motion, while the step length and angle distribution becomes wider.

7.2 Calibration of Probabilistic Models

As outlined in Section 5.3, the essence of probabilistic modeling is to encapsulate a multitude of non-modeled fluctuations into a set of coupled stochastic processes. These stochastic processes are in principle characterized by a hierarchy of PDFs $P_n(n = 1, 2, \ldots)$, thereby resulting in a large—possibly infinite—number of parameters. However, in most cases, one can assume that these processes are Markovian, and therefore reduce the number of parameters (Section 5.3.1). These parameters take the form of *transition probabilities* in discrete-time models or *transition rates*, written k, in continuous-time models. If the Markov property holds, one can relate the probability that a reaction R occur in the next finite time interval $[t, t + \Delta t)$ and the rate k_R of this reaction in a precise fashion:

$$P(\mathbf{X}(t + \Delta t) = \mathbf{x} + \boldsymbol{\nu}_R | \mathbf{X}(t) = \mathbf{x}) \simeq k_R \cdot \Delta t + o(\Delta t) \tag{7.1}$$

where $\boldsymbol{\nu}_r$ is the population change caused by the reaction, and $o(\Delta t)$ is an error term that goes to zero faster than Δt itself. We shall note that probabilities need to be in the interval $[0, 1]$ whereas rates may take any positive value in the interval $[0, \infty)$. As $\Delta t \to 0$, the probability that an event occurs during the short time interval $[t, t + \Delta t)$ becomes small, and the events can be assumed to be Poisson distributed with rate k_R. This formal relationship allows us to infer reaction rates from collision probabilities calculated on non-infinitesimal time intervals.

In the particular case of aggregation and SA, one is often interested in determining the probability p^{coll} for a given robot or building block to collide with another robot during a (non-infinitesimal) time interval $[t, t + \Delta t)$.

Assuming that the system is well-mixed and two-dimensional, one can use the following geometric approximation of the probability of collision (Figure 7.6):

$$p^{\text{coll}} \sim \frac{A_{\text{swept}}}{A_{\text{total}}} = \frac{v\,\Delta t\,w_d}{A_{\text{total}}} \tag{7.2}$$

where v is the average velocity of the robot, w_d the lateral collision distance, and A_{total} the total area of the arena. Interestingly, this expression, known as the *swept volume* or *area* approximation, has been independently derived for estimating the probability of collision among molecules [193] (in three dimensions) and mobile robots [48] (in two dimensions).

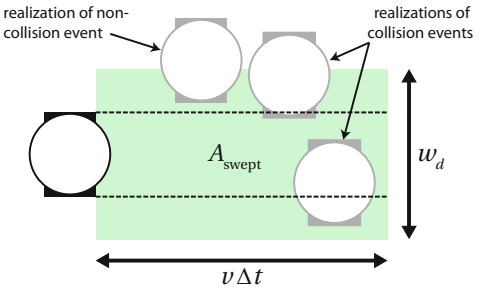


Fig. 7.6 A robot moving at an average speed v sweeps an area $A_{\text{swept}} = v\,\Delta T\,w_d$ during the time interval $[t, t + \Delta T)$. Assuming that the other robots are uniformly distributed throughout the arena, one can then approximate the probability of collision p^{coll} using Equation 7.2. A similar reasoning can be applied in three dimensions.

The assumption underlying Equation 7.2 is two-fold. First, the robots are randomly and uniformly re-distributed throughout the arena at each sampling time (well-mixed property). Second, the robots are identical in size and geometry, such that w_d is a good approximation of the lateral distance of collision. In more complicated scenarios, multiple encountering probabilities that depend on the size and the geometry of the different objects are used [48, 82].

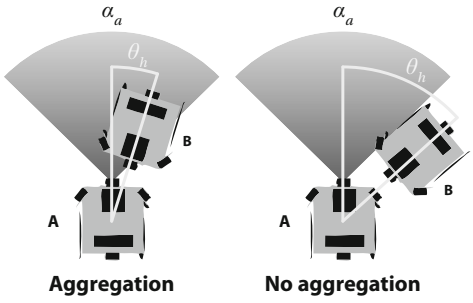


Fig. 7.7 Two colliding robots can assemble only if their relative heading θ_h is smaller than $\alpha_a/2$. The blue circular sector represents both the detection and the communication area of the robots.

7.2.1 Example from Case Study II

A collision does not necessarily lead to the formation of a bond due to the directionality characterizing the building blocks, be they robots or molecules. In Case Study II for instance, robots must be aligned to some extent in order to aggregate successfully. We approximate this constraint by stating that the absolute value of the heading θ_h must be smaller than $\alpha_a/2$, where α_a is the central angle of the detection and communication sector (see Figure 7.7). Because of the non-holonomic nature of the Alice robots, we assume that there is always at least one robot, which we denote **B**, that is aligned upon collision (**B** is the robot that runs into the other). Furthermore, we assume that the absolute value of the heading of **B** with respect to another robot **A** is uniformly distributed in $[0, \pi]$. As a result, the probability p_a that, upon collision, two robots are properly aligned is $p_a \sim \alpha_a/(2 \cdot \pi)$. Since each robot and each chain has two valid binding sites, the overall probability p_b that a bond is formed can be written

$$p_b = p_c \cdot 2\, p_a \cong \frac{\widehat{v}\, T\, w_d}{A_{\text{total}}} \cdot \frac{\alpha_a}{\pi}. \tag{7.3}$$

In first approximation, the probability that a robot leaves a chain does not depend on the geometry of the robots, but on the leaving probabilities encoded in the controller. We consider two cases: (1) a robot has a single neighbor and his leaving probability is $p_{leave,1}$, or (2) a robot has two neighbors and his leaving probability is $p_{leave,2}$. As we observed in real experiments, some robots may be unable to leave because they are physically trapped by their neighbors. This effect can be captured by dividing the leaving probability by a factor $1 - p_s$, where p_s corresponds to the probability that a robot remains stuck. Since there is no simple geometrical approximation for the probability p_s, we need to measure it using either realistic physical simulations or real experiments. In our case, we assume that p_s is negligibly small.

7.2.2 Example from Case Study V

In this section, we show how to determine the reaction rates of the CRNs given in Section 6.3.3. (Note that we restrict our discussion to variant A of Case Study V, but this approach is valid for variant B as well.) Some of these rates are determined using geometrical approximations; other are measured directly using orthogonal experiments, generally carried out in simulation. In the latter case, one needs to carefully design these experiments such as to minimize the influence of parameters that are not measured explicitly. Also, the obtained estimate should be invariant across the design space under consideration (e.g., number of robots, size and shape of the arena, mobility pattern).

In Case Study V, the *swept area* approximation needs to be modified to account for the fact that the cumulated area of the spots is non-negligible with respect to the total area of the arena. Therefore, one can write the rate r_i^e of encountering of a robot and spot i as:

$$r_i^e = \frac{1}{\Delta t} \cdot p_{c,spot} = \frac{1}{\Delta t} \cdot \frac{v \cdot \Delta t \cdot d_{spot}}{A_{\text{total}} - A_{\text{spots}}} = \frac{v \cdot d_{spot}}{A_{\text{total}} - A_{\text{spots}}} \tag{7.4}$$

where v is the average velocity of a robot, d_{spot} is the diameter of a spot, A_{total} is the area of the arena, and A_{spots} is the cumulated area of all other spots. Based on Equation 7.4, one can then derive the following reaction rates[2]:

$$\left.\begin{array}{l} r_{c,i}^e = p_{c,spot} \cdot (1 - p_{w,good})/\Delta t \\ r_{w,i}^e = p_{c,spot} \cdot p_{w,good}/\Delta t \end{array}\right\} \quad \text{if } i \text{ is a good spot,} \tag{7.5}$$

$$\left.\begin{array}{l} r_{c,i}^e = p_{c,spot} \cdot (1 - p_{w,bad})/\Delta t \\ r_{w,i}^e = p_{c,spot} \cdot p_{w,bad}/\Delta t \end{array}\right\} \quad \text{if } i \text{ is a bad spot.} \tag{7.6}$$

The rates r_i^a and r_i^t of aggregation of two robots within a spot, and of one robot and a pair of aggregated robots, respectively, can be written:

$$r_i^a = \sigma_a \frac{v \cdot d_{\text{robot}}}{A_{\text{spot},i}} \qquad r_i^t = \sigma_t \frac{v \cdot 2\, d_{\text{robot}}}{A_{\text{spot},i}} \tag{7.7}$$

where v is the average velocity of a robot, d_{robot} is the lateral collision distance between two robots, $A_{\text{spot},i}$ is the area of the spot i, and σ_a and σ_t are two parameters that account for the partial IR coverage of the Alice robot, which may sometimes prevent aggregation. We set hereafter $\sigma_a = 0.6$ and $\sigma_t = 0.5$; these values were measured using orthogonal "experiments" implemented at the submicroscopic level using Webots (Figure 7.8).

The identification of the rates $r_{w,i}^m$, $r_{c,i}^m$, $r_{w,i}^l$, $r_{c,i}^l$, $r_{w,i}^d$, and $r_{c,i}^d$ is more difficult because they depend on the probability p^{leave} that a robot encounters the border of the spot during the next time interval $[t, t + \Delta t)$. Using again a geometric approximation, we can write

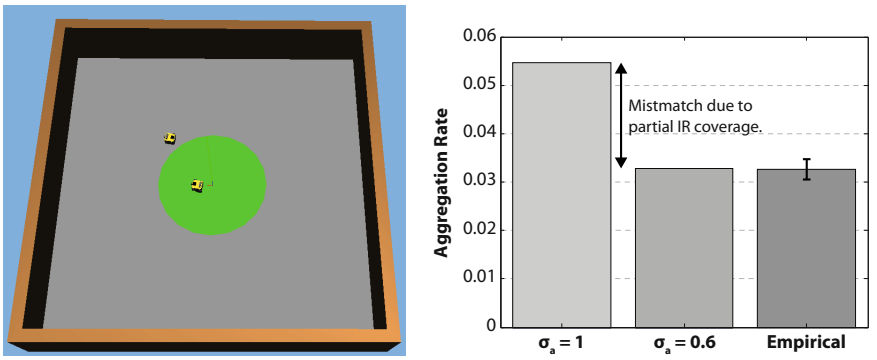
$$p^{\text{leave}} = \frac{\widehat{v} \cdot \Delta t}{d_{spot}/2} = \frac{\Delta t}{T_t} \tag{7.8}$$

where T_t is the average time taken by a robot to traverse a spot of diameter d_{spot}. Thus, if i denotes a good spot, we can write

$$\begin{array}{ll} r_{w,i}^m = p^{\text{leave}} \cdot (1 - p_{\text{good}}^{\text{leave}}) \cdot p_{w,good}/\Delta t & r_{c,i}^m = p^{\text{leave}} \cdot (1 - p_{\text{bad}}^{\text{leave}}) \cdot (1 - p_{w,good})/\Delta t \\ r_{w,i}^l = p^{\text{leave}} \cdot p_{\text{bad}}^{\text{leave}}/\Delta t & r_{c,i}^l = p^{\text{leave}} \cdot p_{\text{good}}^{\text{leave}}/\Delta t \\ r_{w,i}^d = p^{\text{leave}} \cdot (1 - p_{\text{bad}}^{\text{leave}})/\Delta t & r_{c,i}^d = p^{\text{leave}} \cdot (1 - p_{\text{good}}^{\text{leave}})/\Delta t. \end{array} \tag{7.9}$$

If i denotes a bad spot, $p_{\text{good}}^{\text{leave}}$, $p_{\text{bad}}^{\text{leave}}$, and $p_{w,good}$ must be changed to $p_{\text{bad}}^{\text{leave}}$, $p_{\text{good}}^{\text{leave}}$, and $p_{w,bad}$, respectively.

[2] Note that, according to these definitions, the equality $r_i^e = r_{c,i}^e + r_{w,i}^e$ holds, as expected.



(a) Orthogonal experiment

(b) Geometrical approximations vs empirical rate

Fig. 7.8 An example of orthogonal experiment implemented in Webots used to determine the parameter σ_i that accounts for the partial IR coverage of the Alice robot (Equation 7.7). (a) Initially, the two robots have random positions within an arena whose size allow for minimizing the overall experimental time while reproducing the robot-spot encountering dynamics found in real experiments. As soon as both robots are exploring the spot, the supervisor starts recording the time until they aggregate. (b) The empirical rate is obtained by fitting an exponential distribution to the aggregation times obtained from 1000 runs. The parameter σ_a of the geometrical approximation given by Equation 7.7 is then adjusted accordingly. Error bars indicate the 95% confidence interval.

Summary and Conclusion

This chapter discusses the main challenges posed by the calibration of models of SMPs. On the one hand, submicroscopic models are generally characterized by a large number of parameters directly anchored to a particular physical phenomenon. On the other hand, probabilistic models often involve parameters that result from the lumping of many different phenomena, thereby making their calibration particularly difficult. To overcome this problem, the MLMM proposes a series of useful tools.

First, in some cases, it is possible to estimate the transition probabilities of a given model by using geometrical approximations. The underlying assumption is often that the system is well-mixed. Second, one may use both the real physical system and models at lower abstraction level to construct orthogonal "experiments" designed specifically for estimating one or more parameters of the model. Third, appropriate numerical methods may allow for extracting the model's parameters directly from real or simulated trajectories of the system. The latter approach was illustrated in this chapter with a specific focus on trajectories in two-dimensional, physical space. Chapter 9 introduces a more general and systematic method that can deal with n-dimensional trajectories in state space.

8

Model Validation and Analysis

Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful.
— George Edward Pelham Box (1919)
in *Empirical Model-Building* (1987)

Even with a great deal of care in constructing and calibrating a model, a validation of its underlying assumptions is necessary. One cannot prove the *correctness* of a model in the mathematical acceptation of the term; one can merely test it until it fails, and then modify it accordingly. An immediate corollary of this statement is that a model can only be proved *wrong*; however, a contradiction often provides deep insights into the system and its model. This observation is one of the strongest justification of multi-level approaches to the modeling of distributed robotic systems. Indeed, the different models of the same hierarchy generally disagree with each other to some extent, thereby pinpointing the assumptions that are not fulfilled and allowing the modeler to explore the various trade-offs between accuracy, scalability, and computational cost. Hereafter, we illustrate this claim by discussing in detail the validity of the models presented in the previous chapters; in particular, we analyze the impact of various assumptions on the models' predictive accuracy.

8.1 Embodiment

The main reason why ABMs are two orders of magnitude faster than their submicroscopic counterpart (see Section 8.4) is because they neglect most physical effects (e.g., they assume that the motion of robots can be described using simple kinematics), and they use simple collision detection routines (e.g., the robots can be represented as spherical objects with perfect omnidirectional sensing coverage). In other words, they do not account for the specific physical and geometrical features of the robots or the environment, which we call *embodiment* hereafter. While detailed physics-based simulations can be useful (and even necessary) in some cases, more abstract models are often sufficient to yield accurate predictions depending on the performance metric of interest and the properties of the target system. For

G. Mermoud, *Stochastic Reactive Distributed Robotic Systems*,
Springer Tracts in Advanced Robotics 93,
DOI: 10.1007/978-3-319-02609-1_8,

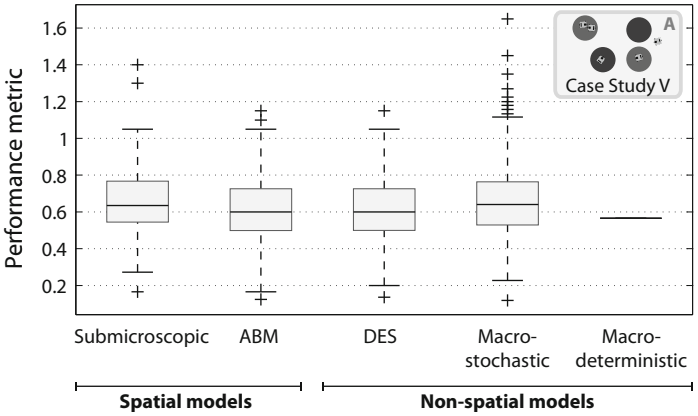
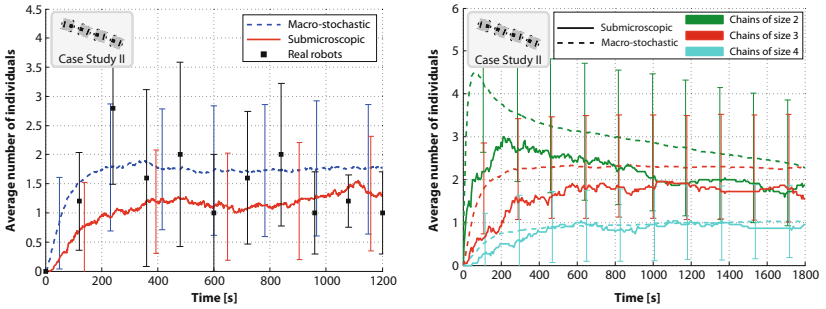


Fig. 8.1 Box plots of the system performance predicted by the different models of Case Study V for the default parameter set (see Table 4.2). On each box, the central mark is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually. Note the excellent agreement of all models, in spite of their very different levels of abstraction.

instance, in Case Study V, the ABMs has the same prediction accuracy as submicroscopic models (Figure 8.1). Importantly, not only the median or the mean performance is correctly predicted, but also the general shape of the distribution.

In Case Study II, however, some important divergences can be observed. Figure 8.2b compares the prediction of macro-stochastic and submicroscopic models in the case of the probabilistic controller with $p_{leave,1} = 10^{-4}$ and $p_{leave,2} = 10^{-9}$. While one can observe an excellent fit for chains of size four, smaller chains tend to grow faster in macro-stochastic simulations than in submicroscopic simulations because of spatiality and embodiment. Indeed, both of these characteristics are abstracted at the macroscopic level. For instance, embodiment generates geometric obstructions that make bonds less likely to form; often this is due to large chains obstructing the paths of other robots, or aggregates forming near walls where further aggregation is not possible.

In some cases, even submicroscopic simulations are not sufficient to account properly for embodiment. We implemented the two controllers of Case Study II using a group of 19 Alice robots, and we performed five real experiments of 20 minutes. A four minute excerpt from one such experiment is provided in Figure 8.3. Again, we observed some discrepancies due to embodiment between our submicroscopic model and real experiments (Figure 8.4). First, due to the Alice's weak motor system, occasionally a robot may become physically trapped by others in the same aggregate, making



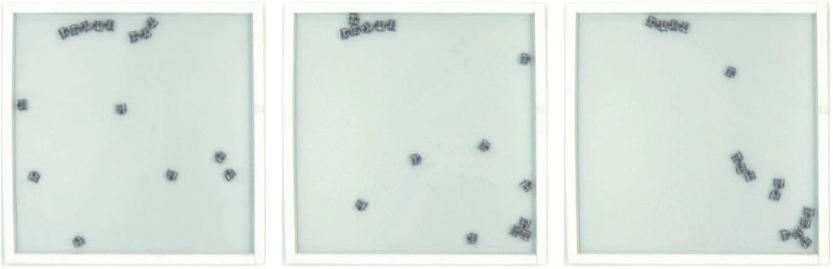
(a) Experimental validation (b) Inter-level validation

Fig. 8.2 Validation of Webots and CRN models of Case Study II for the probabilistic controller with $N_0 = 19$. (a) Experimental validation (black square are experimental data averaged over 5 runs) of submicroscopic and macro-stochastic predictions of the time evolution of the average number of chains of size three with $p_{leave,1} = p_{leave,2} = 10^{-3}$. (b) Comparison of the time evolution of the average number of chains of size two (green), three (red) and four (cyan) predicted by macro-stochastic models (dashed lines) and submicroscopic models (continuous lines) with $p_{leave,1} = 10^{-4}$ and $p_{leave,2} = 10^{-9}$.

disaggregation impossible. These effects are difficult to capture accurately at the submicroscopic level as they result from subtle physical interactions (e.g., protruding details of the robot's body that interlock or rub against each other). Second, we observe that some robots may aggregate with the side of a chain due to IR reflections that are not modeled in Webots, leading in some cases to unstructured aggregates, a phenomenon also observed in previous experiments of object aggregation [81]. The probability of occurrence of this type of aggregation is a function of the chain size (the more robots in a chain, the more in-chain binding sites) as well as the shape and sensor arrangement of the robots (which exhibit a strong heterogeneity, both in sensor and actuator response and in sensor arrangement). Furthermore, irregular shapes tend to trigger additional incorrect attachments, which may lead to non-linear, positive feedbacks. This effect is difficult to capture accurately at the submicroscopic level. An accurate modeling of IR reflections would be computationally expensive, and group heterogeneity would require the manual collection of statistically significant data for each sensor of each robot. As such, this second discrepancy is ignored by our models.

In Case Study V, embodiment explains the divergence observed between submicroscopic and macro-stochastic models for $N_{\text{spots}} = 10$ when the robots have an exact estimate of the spots' type and the spots outnumber the robots (Figure 8.5). In this case, since there is no "intrinsic" randomness due to the noise, a robot with $p_{\text{bad}}^{\text{leave}} = 0$ may explore indefinitely a spot, waiting for a mate. If there are more spots than robots, deadlocks may arise—each robot

(a) Initial distribution (b) After 30 sec (c) After 1 min

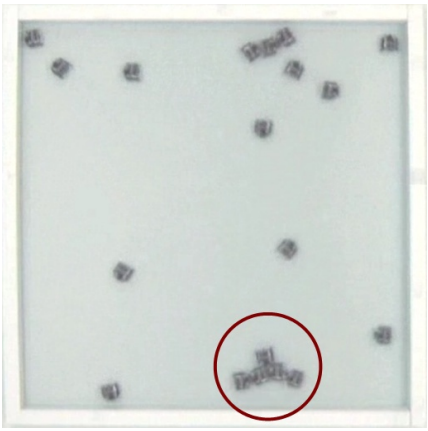


(d) After 2 min (e) After 3 min (f) After 4 min

Fig. 8.3 Snapshots of a four minute excerpt from a real experiment (Case Study II) with $N_0 = 15$ Alice robots and $p_{leave,1} = p_{leave,2} = 10^{-3}$. (a) At $t = 0$, the system is in an arbitrary, homogeneous state. (b) After 30 seconds, some aggregates have already begun to form. (c), (d), and (e) The system continues to evolve from 1 to 3 minutes; images are captured at uniform intervals of one minute. (f) Finally, after 4 minutes, most Alices have bonded with others to form an aggregate (only a single robot remains disaggregated).

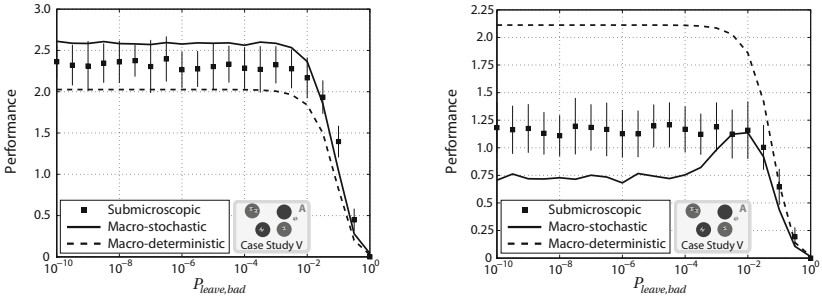
waiting for another one in a different spot. As a result, the optimum shifts towards non-zero leaving probabilities as the ratio of spots to robots increases. This result is similar to that obtained in the stick-pulling experiment [48], which involved however a deterministic timeout for controlling the waiting time rather than a leaving probability.

In reality, the combined effects of embodiment, wheel slip, and actuation inaccuracies sometimes cause the robots to involuntarily leave the spot they are exploring. This phenomenon artificially maintains the leaving probabilities above an artificial threshold, regardless of the real value of $p_{\text{good}}^{\text{leave}}$ and $p_{\text{bad}}^{\text{leave}}$. The drop in performance predicted by macro-stochastic simulations for small leaving probabilities is therefore not observed in submicroscopic simulations that accurately capture these effects. The impact of embodiment on the predictions of macro-deterministic models is discussed in Section 8.3.



(a) Physically trapped robot (b) Malformed chain due to IR reflections

Fig. 8.4 Two types of discrepancies observed in Case Study II between submicroscopic models and real experiments. (a) In simulation, robots can leave from the middle of a chain; in reality, a robot (encircled in red) may remain physically trapped by its neighbors. (b) A robot may aggregate on the side of a chain due to IR reflections that are not modeled in Webots.



(a) 4 spots (b) 10 spots

Fig. 8.5 Predicted performance of the group with $k = 2$ for (a) $N_{\text{spots}} = 4$ and (b) $N_{\text{spots}} = 10$, and a constant number of robots $N_0 = 5$ that have an exact estimate of the spots' type in Case Study V. Macro-stochastic and macro-deterministic models are consistent with the submicroscopic simulations as long as there are less spots than robots, but their predictions diverge for $N_s = 10$.

8.2 Spatiality

Spatiality is another feature of multi-robot systems that is difficult to capture at high abstraction levels [190]. When the position of the individual robots is not captured explicitly, most models assume that the system is *well-mixed* (Section 5.3.5), even though some sophisticated approaches based on the coupling of PDEs and rate equations are capable of handling non-well-mixed systems as long as they remain relatively well-behaved (i.e., advection-diffusion systems) [24, 189, 190, 213].

In Case Study V, the average inter-collision time between robots and spots is 13 s; given an average speed of about 2.2 cm/s, a robot travels about 29 cm between two collisions, which is in the order of the arena size. This observation indicates in principle that our system is well-mixed and that the geometric approximations discussed in Section 7.2.2 are applicable. This claim is corroborated by Figure 8.1, which exhibits an excellent agreement between spatial and non-spatial models.

In contrast, non-spatial models of Case Study III exhibits a faster convergence than spatial models (Figure 8.6), even for large number of building blocks ($N_0 = 1000$). This discrepancy is caused by stable, non-reactive aggregates, which surround the remaining free building blocks, thereby hindering their mobility. This suboptimal mixing tends to slow down the aggregation process; this phenomenon is not captured by non-spatial models.

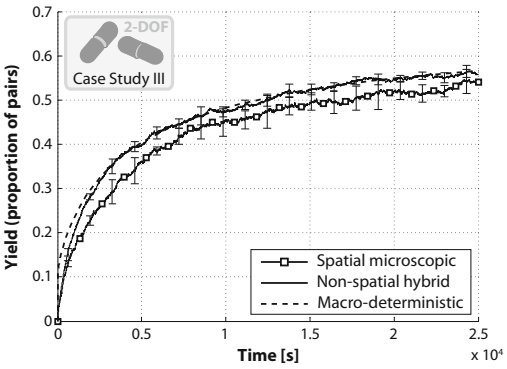


Fig. 8.6 Inter-level validation of a suite of models of Case Study III (2-DOF) for a large number of building blocks ($N_0 = 1000$). We compare the system's dynamics as predicted by the macro-deterministic model with a discretization factor $K = 3000$ (Section 6.3.2), the non-spatial hybrid MCM (Section 6.2.2), and the spatial ABM described in [206].

8.3 The Macro-deterministic Approximation

As discussed in Section 5.3.4, one sufficient condition for macro-deterministic models to be accurate (in first order) is that the system involves a large (ideally infinite) number of robots. However, for some systems, this condition is not *necessary*. For instance, macro-deterministic models of Case Study II

exhibit a small first-order error for $N_0 = 19$, well within the standard deviation of the stochastic simulations (Figure 8.7). Of course, the second-order error is still large given the high stochasticity of the system.

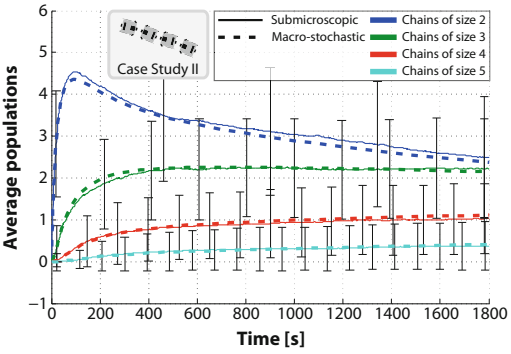
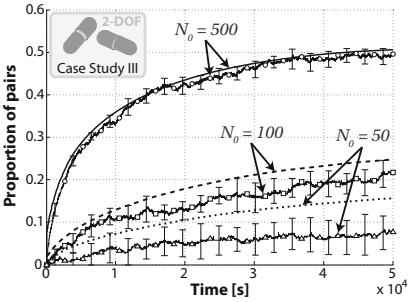


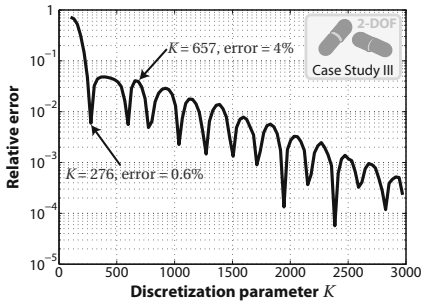
Fig. 8.7 Comparison of the prediction of macro-stochastic (continuous lines) and macro-deterministic (dashed lines) models for the time evolution of the system with $N_0 = 19$, $p_{leave,1} = 10^{-4}$ and $p_{leave,2} = 10^{-9}$. These results show that, for Case Study II, the macro-deterministic approximation remains valid, in the first order, even for small N_0.

For other systems, small copy numbers lead to quantization errors that may be amplified by non-linearities due to positive feedbacks, large differences in time scales (stiffness), and more generally rare reactions with large effect. The impact of these factors on the first-order error depends strongly on the model structure and its parameters. As a rule of thumb, simple, linear models such as the CRN for $k = 1$ (Equation 6.18) are typically less sensitive to such structural effects than more complex—and often non-linear—models such as the CRNs for $k > 3$ (equations 6.19 and 6.20).

Case Study III (2-DOF) offers an insightful illustration of this dependence on the model's structure. As one may expect, the first-order accuracy of the macro-deterministic model exhibits a graceful decrease as N_0 decreases (Figure 8.8a). For $N_0 = 50$, the macro-deterministic model actually predicts a faster growth of the pair ratio than its lower abstraction counterpart, whereas an almost perfect match is observed for $N_0 = 500$. However, the validity of the macro-deterministic approximation does not only depend on the copy number N_0; the structure of the network and the number of interactions also plays a key role. Figure 8.8b shows the relative error of the macro-deterministic model for different levels of discretization K of the bond energy (see sections 4.3 and 6.3.1) with respect to a baseline prediction for $K = 3000$. The relative error depends on K in a strongly non-linear fashion; multiple local minima can be observed where the macro-deterministic model performs well in spite of a small value of K. We hypothesize that these non-linearities arise from the opposite effects of state discretization. On the one hand, it improves the description of the system's dynamics. On the other hand, it weakens the macro-deterministic approximation by lowering individual reaction rates while keeping the effect of firing events on the performance metric identical. The computational cost of the model being proportional to K, one



(a) Effect of copy numbers



(b) Effect of discretization

Fig. 8.8 Illustration of the effect of varying copy numbers and discretization on the macro-deterministic model's accuracy in Case Study III (2-DOF). (a) Comparison of the long run prediction (50 000 s) of the hybrid Monte Carlo model ($N_0 = 50$, triangles; $N_0 = 100$, squares; $N_0 = 500$, circles) and the macro-deterministic model ($N_0 = 50$, dotted line; $N_0 = 100$, dashed line; $N_0 = 500$, continuous line) for different total number of building blocks N_0. (b) Relative error of the macro-deterministic model as a function of the level of discretization of the bond energy $K \in [100, 3000]$. The error is computed with respect to a baseline prediction with $K = 3000$. Interestingly, in the close neighborhood of $K = 276$, the error is small (approximately 0.6%) and it attains 4% for $K = 657$.

could use a non-uniform discretization of the state space such that the error is minimized. Nevertheless, even small K (e.g., around 276, see Figure 8.8b) are appropriate for qualitatively probing the dynamics of the system. Note however that these "good" values of K may vary as a function of the control and design parameters of the system. For $K \geq 2000$, we consistently observe excellent quantitative agreement with models at lower abstraction level.

Case Study V offers another illustration of how the model's structure influences the validity of the macro-deterministic approximation. The compartmentation caused by spots acts like vesicles in cell biology, that is, they form weakly coupled subsystems characterized by even smaller copy numbers, and reactions whose rate usually is small (e.g., two robots aggregating in a spot), and effect large (e.g., destruction of a spot). This phenomenon is also found in biology as a strong limitation of the use of macro-deterministic models [214]. In particular, the strong irregularity in the landscape of spot destructions around $p^{\text{leave}}_{\text{good}} = p^{\text{leave}}_{\text{bad}} = 0$ for $k = 3$ is not captured at all by the macro-deterministic model (Figure 8.9), because it does not account for the situation where all robots are either aggregated or exploring different spots, thus leading to deadlocks similar to those discussed in Section 8.1.

Figure 8.5 offers a better illustration of this effect by assuming that the robots have an exact estimate of the spots' type. Macro-deterministic models do not account for any drop in performance when the spots outnumber the robots (Figure 8.5b); even worse, they predict an increase of the performance

as the number of spots increase. This finding emphasizes once more the importance of using models at multiple abstraction levels; abstractions that seem correct *a priori* may actually be error-prone in some specific scenarios. In our particular case, neglecting the effect of robots that "lose" track of the spot they are exploring leads to inaccuracies at the macro-deterministic level when leaving probabilities become small. More importantly, the small number of robots and spots in the system as well as the compartmentation effect caused by spots render the macro-deterministic approximation invalid, and predictions of such models inaccurate, in particular when the ratio of spots to robots increases.

8.3.1 Fixed Points and Stability

When the macro-deterministic approximation is valid, one may perform an analytic study of the system's behavior. For instance, we may look for the fixed points of the model of Case Study III described in Section 6.3.2. In particular, by setting $x_i(k+1) = x_i(k)$ in Equation 6.15, we obtain:

$$x_i(k) = \frac{f(i) \cdot p^{\text{join}}}{2 \cdot p^{\text{leave}}(s_i)} \cdot x_s(k)^2 = \alpha_i \cdot x_s(k)^2 \tag{8.1}$$

with α_i a simplification variable. By conservation of the number of building blocks, one can also write:

$$x_s(k) + 2 \cdot \sum_{i=1}^{K} x_i(k) = 1, \tag{8.2}$$

and replacing $x_i(k)$ according to Equation 8.1, we obtain the following expression:

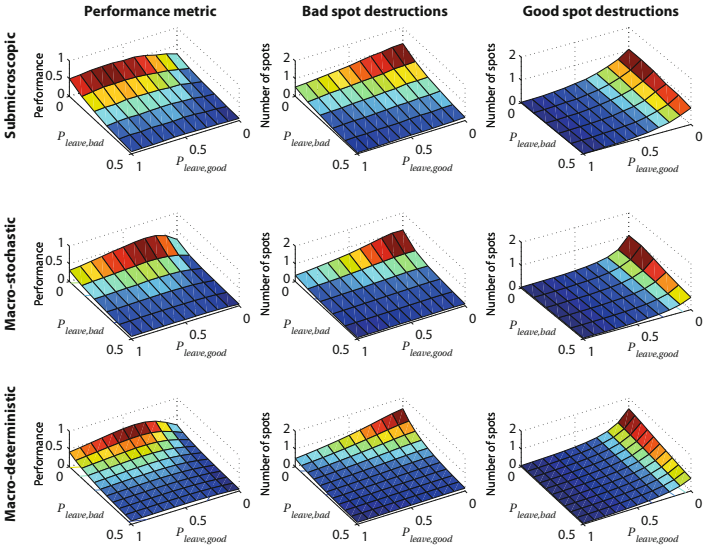
$$x_s(k) + 2 \sum_{i=1}^{K} \alpha_i \cdot x_i(k)^2 - 1 = 0. \tag{8.3}$$

Solving for $x_s(k)$ yields the first coordinate of the fixed point:

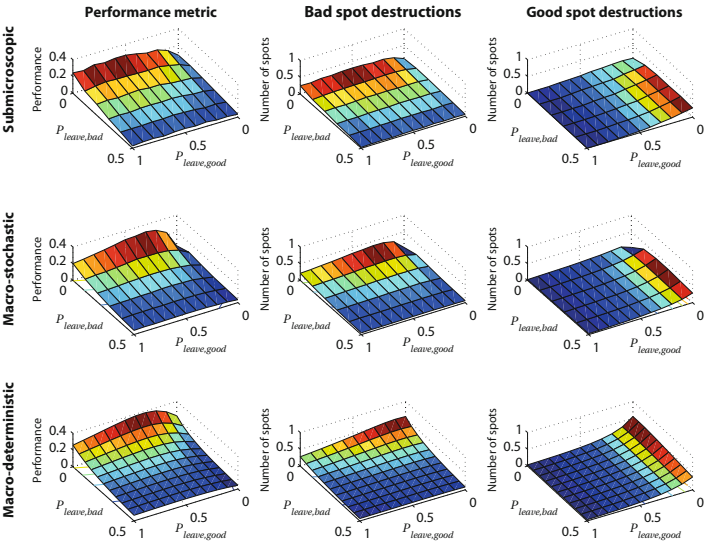
$$\bar{x}_s = \frac{-1 + \sqrt{1 + 4 \sum_{i=1}^{K} \alpha_i}}{2 \sum_{i=1}^{K} \alpha_i}. \tag{8.4}$$

The other coordinates can be easily derived from the relation given by Equation 8.1.

Now, if we are interested in the stability of this fixed point, we shall compute the eigenvalues of its associated Jacobian matrix $\mathbf{J}$ of the vector-valued function $\mathbf{F}(\bar{\mathbf{x}}) : \mathbb{R}^{K+1} \to \mathbb{R}^{K+1}$, defined as follows:



(a) Scenario $k = 2$



(b) Scenario $k = 3$

Fig. 8.9 Comparison of the predictions of submicroscopic, macro-stochastic, and macro-deterministic models of Case Study V for (a) $k = 2$ and (b) $k = 3$

$$\mathbf{x}(k+1) = \mathbf{F}\big(\mathbf{x}(k)\big) \qquad \text{with } \mathbf{x}(k) = \begin{pmatrix} x_s(k) \\ x_1(k) \\ \vdots \\ x_K(k) \end{pmatrix}. \tag{8.5}$$

For the sake of simplicity, let's consider the case where $K = 1$. Then, the relation $x_1(k) = (1 - x_s(k))/2$ (obtained from Equation 8.2) reduces the system to a single difference equation:

$$x_s(k+1) = f(x_s(k)) = -p^{\text{join}} x_s(k)^2 + (1 - p^{\text{leave}})\, x_s(k) + p^{\text{leave}}. \tag{8.6}$$

The derivative of $f(x)$ evaluated at the fixed point $\bar{x}_s$ indicates the stability of the fixed point. Namely, the fixed point is stable if and only if

$$f'(\bar{x}_s) = \left|1 - \sqrt{p^{\text{leave}}}\sqrt{4\,p^{\text{join}} + p^{\text{leave}}}\right| \leq 1. \tag{8.7}$$

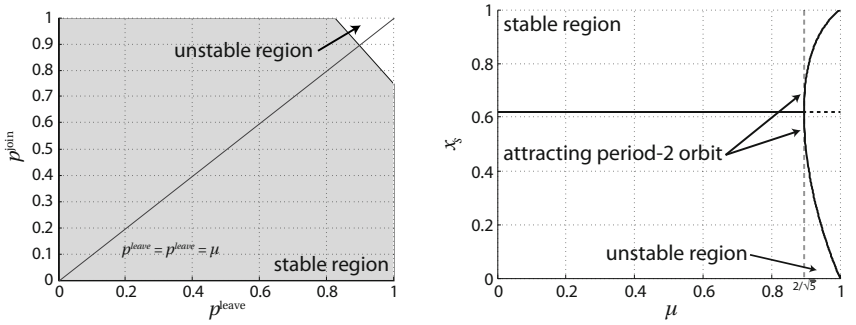
The stability of the fixed point $\bar{x}_s$ as a function of the parameters p^{leave} and p^{join} is illustrated in Figure 8.10a. In the region of stability, all trajectories converge to $\bar{x}_s$ in finite time. As one approaches the region of instability, trajectories oscillate around the fixed point before convergence. In the region of instability, an attracting period-2 orbit appears, which is given by the following equation:

$$f(f(\bar{x}_{s,2})) = \bar{x}_{s,2}. \tag{8.8}$$

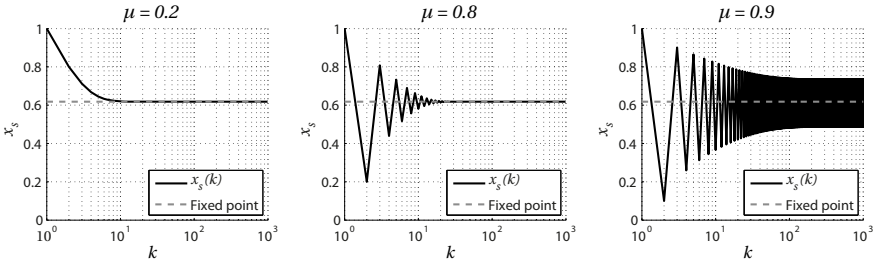
This behavior is called a *period doubling bifurcation*. Figure 8.10b depicts the bifurcation diagram in case where $p^{\text{leave}} = p^{\text{join}} = \mu$. The fixed point $\bar{x}_s = \frac{1}{2}\left(-1 + \sqrt{5}\right)$ becomes unstable at $\mu = 2/\sqrt{5}$, and gives rise to a period-2 orbit whose amplitude A is proportional to μ:

$$A(\mu) = -\frac{\sqrt{-4 + 5\,\mu^2}}{\mu}. \tag{8.9}$$

Figure 8.10c illustrates the qualitative behavior the system as one approaches and enters the region of instability. In reality, oscillations are never observed in Case Study III because p^{leave} and p^{join} need to be small for the model to be valid (see Section 7.2 for a more complete discussion). The oscillations observed in simulation are merely an artifact of invalid modeling assumptions (in particular, the time-discrete nature of the model coupled with large transition probabilities). Therefore, one can safely conclude that the system will indeed converge to a unique fixed point, which is given by Equation 8.1.



(a) Stability region (b) Bifurcation diagram



(c) Sample trajectories

Fig. 8.10 (a) Stability of the fixed point $\bar{x}_s$ in parameter space, and (b) the bifurcation diagram for $p^{\text{leave}} = p^{\text{join}} = \mu$. Bold lines denote stable fixed points or attracting periodic orbits whereas dashed lines denote unstable fixed points. (c) Sample trajectories for $\mu = \{0.2, 0.8, 0.9\}$ illustrate the period doubling bifurcation at $\mu = 2/\sqrt{5} = 0.8944$.

8.4 Scalability

The main motivation for building up models at higher abstraction level is computational efficiency. Table 8.1 provides the typical speed-up factor (with respect to real time) of the different models of Case Study V. Note the significant impact of spatiality on computational complexity: non-spatial models are up to four orders of magnitude faster than spatial models. Of course, to be completely fair, one shall also account for the differences in implementation; however, such a thorough analysis is beyond the scope of this work, and we provide these benchmarking data as an illustration of the large diversity of computational costs observed in such hierarchies of models.

Also, these speed-up factors are estimated from simulations for a given set of parameter, and they may vary as a function of those parameters. For instance, all microscopic models have a speed-up factors inversely proportional to the number of robots, since they keep track of the state of each of them

Table 8.1 Typical speed-up (with respect to real time) of the different models of Case Study V for $k = 2, 6$ robots, and 4 spots. The last column denotes whether the model's outcome varies at each run or not. Note that, while macro-deterministic models exhibit a lower speed-up than macro-stochastic models, the latter typically require a large number of runs to yield statistically significant predictions.

Model	**Spatiality**	**Implementation**	**× real time**	**Stochastic**
Submicroscopic	yes	Webots	10^1	yes
ABM	yes	Netlogo	10^3	yes
DES	no	Custom (C++)	10^7	yes
Macro-stochastic	no	StochKit	10^7	yes
Macro-deterministic	no	MATLAB™	10^5	no

individually. Another important cause for decreased computational performance in time-discrete models such as ABMs is the presence of different time scales (e.g., *rare* events such as spot destructions, and *frequent* events such as robot collisions) that prevent the time step from being too large. While increasing the time step speeds up the simulation, it also diminishes the accuracy of the simulation, ultimately leading to missed events. Similarly, very frequent events can significantly slow down discrete-event and stochastic simulations, as they explicitly capture each individual event. Very sophisticated optimized variants of the Gillespie's SSA have been developed to overcome this problem [176, 177, 187, 215].

Even though macroscopic models are in principle not directly affected by the number of robots involved in the system, the complexity of the underlying CRN usually depends on the complexity of the environment. In our case, for instance, experiments showed that simulation times scale in $O\left(N_{\text{spots}}^3\right)$ for macro-deterministic models, and in $O\left(e^{N_{\text{spots}}}\right)$ for macro-stochastic models, with N_{spots} the number of spots in the system. In terms of memory usage, macro-deterministic models also scale up much better than stochastic simulations. Stochastic simulations used 8 GB of RAM for $N_{\text{spots}} = 2000$ and $N_0 = 4000$ whereas macro-deterministic models only required 1 GB for the same set of parameters. These findings, along with the fact that macro-deterministic models become more accurate as N_{spots} and N_0 increase (see Section 8.3), motivate the use of macro-deterministic models for the study of large distributed systems, or the use of hybrid approaches that preserve the stochastic treatment of rare reactions with large effect, yet enable scalability by solving deterministically the rest of the CRN [216]. We shall however outline the fact that the implementation plays a key role in the scalability of these models.

8.5 Parameter Sensitivity

Fallacious assumptions are not the only cause for divergent predictions; inaccurate parameterization is another common source of error. Sensitivity analysis answers the question:"If the model's parameters are inaccurate, what will the effect be on the model's predictions?" More formally, it consists in estimating the relationship between parameters' variation and predictions' variations. Let $M(\mathbf{p}, t, \mathbf{x})$ denote the prediction a model M parameterized by a vector $\mathbf{p}$ at time t for an initial state $\mathbf{x}$. Then, one is interested in estimating the function:

$$S_M(\Delta\mathbf{p}) = \mathrm{dist}\big(M(\mathbf{p}, \mathbf{x}, t), M(\mathbf{p} + \Delta\mathbf{p}, \mathbf{x}, t)\big) \tag{8.10}$$

where $\mathrm{dist}(\cdot, \cdot)$ denotes some distance function and $\Delta\mathbf{p}$ is a vector of parameter variations. If $S_M(\Delta\mathbf{p})$ remains small even for large deviations $\Delta\mathbf{p}$, then even rough approximations of the model's parameters are sufficient to obtain a reasonable prediction. Instead, if $S_M(\Delta\mathbf{p})$ grows quickly with $\Delta\mathbf{p}$, then the model requires an accurate estimate its underlying parameters.

Hereafter, we perform a sensitivity analysis of the submicroscopic model of Case Study IV described in Section 6.1.2. In particular, we compare in simulation the effect of perturbing the calibrated parameters (i.e., the drag coefficient C_d and the standard deviation σ_F of the stochastic force) on the predicted distributions of self-assembled structures. Our approach consists in calculating the distance between a reference distribution (obtained from simulated SA experiments with the calibrated parameters) and the distributions corresponding to perturbed parameters. For each set of parameters, we perform 1000 simulation runs (5 minutes of simulated time each), and classify the resulting self-assembled structures into geometrical classes. As mentioned in Section 4.4, we use only four blocks in this study, which leads to the ten possible outcomes depicted at the bottom of Figure 8.11. The reference distribution (Figure 8.12) is obtained by aggregating 10 realizations of 1000 runs using the calibrated parameters found in Section 7.1.2.

We then compare the resulting distributions of self-assembled geometries using the $l1$-norm:

$$l1\text{-norm} = \sum_{i=1}^{10} |x_i^a - x_i^b| \tag{8.11}$$

where x_i^a, and x_i^b represent the proportion of geometries of class i for the set of parameters a and b, respectively. Figure 8.11 depicts the distance (as defined by Equation 8.11) between the reference distribution (see Figure 8.12) and distributions yielded by parameters ranging between 0 and 1.5 for C_d and 0 mN and 1.5 mN for σ_F.

Our results show a significant sensitivity of the submicroscopic model to variations of both the drag coefficient and the random force. First, one can observe important qualitative variations of the obtained distributions throughout the parameter space. The magnitude of these variations is much

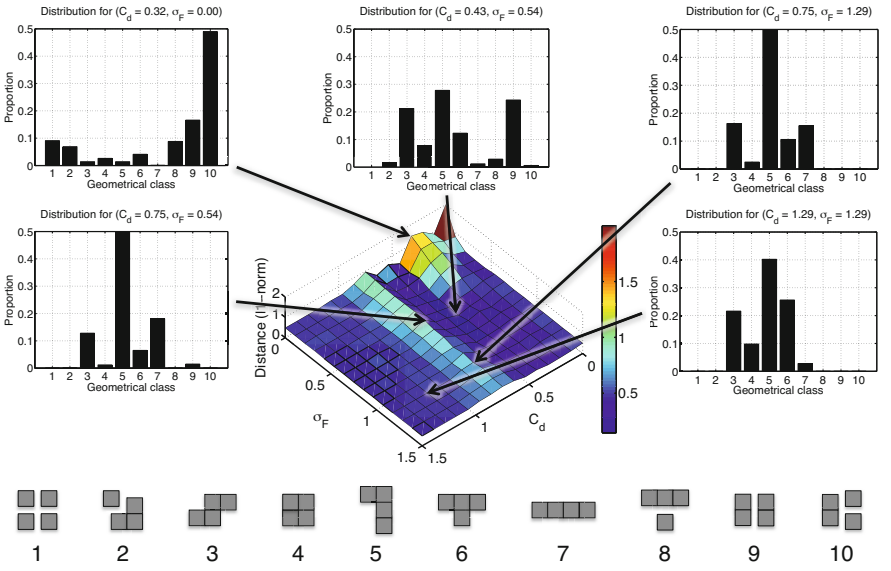


Fig. 8.11 *Top*: Plot of the distance (as defined in Equation 8.11) between the reference distribution (i.e., the distribution yielded by $C_d = 0.45$ and $\sigma_F = 0.55$ mN) and distributions obtained with parameters ranging between 0 and 1.5 for the drag coefficient C_d, and 0 mN and 1.5 mN for the standard deviation σ_F of the stochastic force. *Bottom*: Sketch of the 10 classes of self-assembled geometries discovered during the simulation.

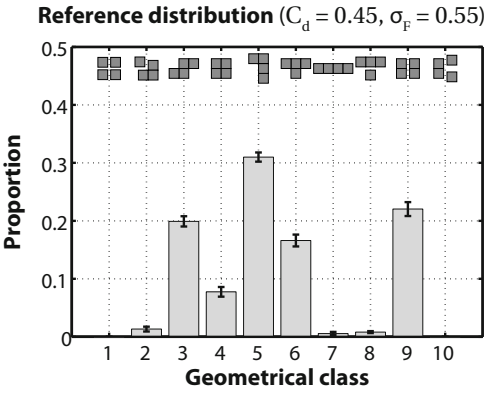


Fig. 8.12 Average distribution of self-assembled geometries for $C_d = 0.45$ and $\sigma_F = 0.55$ mN (i.e., the optimal parameters reported in Section 7.1.2) averaged over 10 realizations of 1000 runs. The error bars denote the standard deviation, and the average distance (as defined in Equation 8.11) between realizations is 0.065.

larger than the fluctuations observed in different realizations of the reference distribution (see Figure 8.12). Second, a more quantitative analysis shows that the most important divergences occur for small values of either parameters. This result is important, as it shows that even small variations of the physical parameters can lead to dramatic changes in the outcome of the simulation. It also further outlines the importance of a proper parameter

calibration. Interestingly, the distance landscape contains local minima; in particular, a ridge crosses the entire parameter set around $C_d = 0.7$, thereby giving rise to a large region (corresponding to large stochastic forces) that achieve small, yet suboptimal, distances. This finding reflects the difficulty of "reverse-engineering" SA processes, that is, inferring a model of their dynamics based on samples of the formed structures.

The variations in the dataset shown in Figure 8.11 can be better visualized using a Principal Component Analysis (PCA). Figure 8.13 depicts both the dataset and the original variables (i.e., the 10 geometrical classes) projected on the two-dimensional plane spanned by the two principal components. Interestingly, one can directly observe that principal component 1 is aligned with the physically disconnected geometries (classes 1, 2, 8, 9, and 10), which are favored by small values of C_d and σ_F. Indeed, while disconnected geometries are never observed in most scenarios (in particular when the mobility of the blocks with respect to each other is sufficiently high), they can become prominent in others (e.g., see the distribution for $C_d = 0.43$ and $\sigma_F = 0.54$ in Figure 8.11). On the other hand, the large elongated cluster aligned with the second component corresponds to scenarios that yield a unique aggregate; interestingly, there is no region of the space that contains most of the points. This observation further confirms that SA is sensitive to variations of the physical parameters of the model.

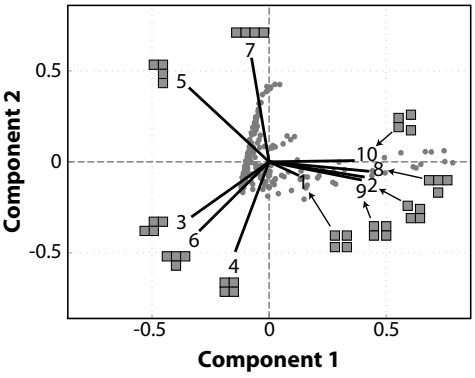


Fig. 8.13 Biplot of the dataset (small dots) shown in Figure 8.11 projected on the two-dimensional plane spanned by the its principal components. Projections of the original geometrical classes are depicted by solid lines.

Summary and Conclusion

This chapter describes the conditions of validity of the models described in earlier chapters, in particular in relation with some specific properties of the underlying system. Embodiment proves to be particularly difficult to capture, and it may lead to important inaccuracies in certain cases, even at

the submicroscopic level. More abstract models generally lump these effects into one or more parameters that are generally quite difficult to calibrate because they require experimental data obtained with real robots. Spatiality turns out to be considerably more tractable, especially when the system is well-mixed. When this is not the case, one may either model the system at the microscopic level or, if possible, use a macroscopic formulation capable of handling non-well-mixed systems.

One fundamental property of most robotic systems is that they operate far from the thermodynamic limit where the number of robots approaches infinity. As a result, the validity of the macro-deterministic approximation needs to be systematically re-examined. Our results show that the effect of this approximation varies greatly as a function of the structure of the model, and does not only depend on the copy numbers. As a rule of thumb, the presence of rare reactions with large effect is the essential indication that the macro-deterministic approximation is invalid. In spite of these pitfalls, macro-deterministic models are important tools of the MLMM as they allow one to investigate analytically the system's behavior (e.g., existence and stability of fixed points, chaotic behavior, etc.).

This chapter also provides some insights into the computational cost and the scalability of the different models described in earlier chapters. Our observations are two-fold. First, non-spatial models are up to four orders of magnitude faster than spatial models, thereby outlining the drastic computational cost of spatiality. Second, macro-stochastic models are generally less scalable and more expensive than their macro-deterministic counterpart. The latter assertion must be qualified by the fact that the implementation plays a key role in the scalability of the models.

Finally, we discuss the important notion of parameter sensitivity, albeit superficially. Indeed, we focus our analysis on the submicroscopic model of Case Study IV described in Section 6.1.2. Our results not only stress the importance of proper calibration of the model's parameters, but they also preclude the possibility of inferring a model of a given SA process based only on a set of final outcomes. While this issue is not crucial at the centimeter-scale, where real-time and accurate tracking of each building block is feasible, it poses important challenges for studying SA at the microscale, where such detailed analysis of the system's dynamics is difficult.

9

Automated Multi-level Modeling

No more than (these) machines need the mathematician know what he does.
— Henri Poincaré (1854-1912)
Quoted in *The World of Mathematics* (New York 1956), by JR Newman.

Nearly all steps of a typical engineering process can be reliably automated. Software tools such as Mathematica™, Coq, and BLAST can solve almost arbitrarily intricate equations, demonstrate complex theorems, and prove the correctness of algorithms. Machine tools and 3D printers can construct simple or complex objects based on a CAD model, with virtually no human intervention. Whenever a process can be broken into a set of relatively systematic and formal recipes, there is room for automation. Following this line of reasoning, and assuming that the MLMM provides such recipes, this chapter explores the opportunity of automating the generation of models at multiple abstraction levels. In particular, we refine and extend the MLMM by introducing a methodological and computational framework, called the M^3 framework[1], which provides a consistent set of theoretical and computational tools targeted to the automated construction of models of SMPs at multiple abstraction levels.

Our approach builds upon the considerations of the previous chapters, and particularly on the central and essential idea that distributed reactive systems can be described as sets of interacting agents with two essential features: *hybridity* (i.e., they have both a continuous, physical state, and a discrete, logical state) and *stochasticity* (i.e., their dynamics can be described essentially as a stochastic process). Crucially, we assume that the *discrete* component of the particles will serve as a blueprint for constructing models at increasing level of abstraction, similarly to the MLMM. More formally, we define our system as a set of coupled hybrid automata [217], which we refer to as *particles* hereafter, whose states have two components: (i) a *continuous* component that typically denotes the physical state of the particle (e.g., its position and velocity in physical space, its temperature, its battery level,

[1] The name of the M^3 framework originates from both the association of its creators' names (Mermoud, Mastrangeli, and Martinoli) and the contraction of the acronym MLMM.

G. Mermoud, *Stochastic Reactive Distributed Robotic Systems*,
Springer Tracts in Advanced Robotics 93,
DOI: 10.1007/978-3-319-02609-1_9,

etc.), and (ii) a *discrete* component that denotes the logical state, called *control mode*, of the particle. The discrete dynamics depend on the continuous dynamics, and vice versa (Figure 9.1).

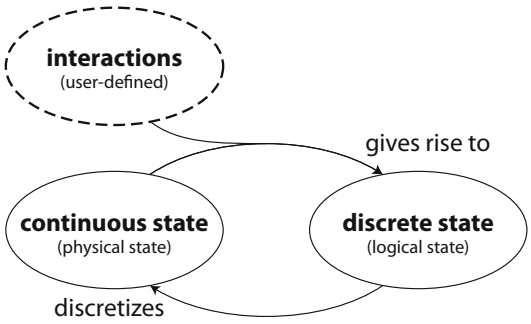


Fig. 9.1 The M^3 framework assumes any SMP can be described as a hybrid entity. The continuous, physical state of each particle, in conjunction with a user-defined list of interactions, gives rise to the discrete, logical components of the state space, which serves as discretization blueprint of the continuous state space.

Our methodology makes the fundamental assumption that the particles are strictly reactive, that is, **any change in control mode can be interpreted as the result of an interaction with other particles or with the environment**. A corollary of this definition is that one can associate each control mode of the particle to a given interaction configuration. In robotics, when designing a robot's controller, one naturally arranges the different interaction configurations that the robot can be in into groups or classes indexed by a set of behaviors. For instance, the designer will group all situations in which the robot is close to an obstacle (e.g., another robot or an arena wall), devise an appropriate control law for such situations, and associate the resulting class to a behavior *obstacle avoidance*.

Following this methodology, the controller of each robot naturally reflects the most important states of the robot. As a result, one can use the robot's controller as a blueprint to construct a meaningful partition of the continuous phase space, and thereby deriving models at higher abstraction level. In the MLMM, the control modes are defined *a priori*, upon designing the system; as a result, they merely offer a robot-centric (or particle-centric, in our case) representation of the world. As a result, it is often necessary to refine the resulting models *a posteriori*, in order to account for hidden degrees of freedom (see Chapter 6). In a pure bottom-up approach, this problem is solved by starting with a detailed model that accounts for all interactions in the system, including those occurring between particles and the environment.

Therefore, the M^3 framework relies on a detailed microscopic representation of the system, which we call Canonical Microscopic Model (CMM), which is a *formal* description of a distributed system that can be *algorithmically* constructed. Importantly, the CMM is a mathematical construct, rather than a computational model; it provides in particular a baseline formalism for our methodology based on the abstraction of detailed models into

coarser models. More generally, it is not intended for simulation purposes, as it is often difficult to specify completely (see Example 9.15 on p. 132). The CMM exhibits a few key properties: (i) it describes a given distributed system as a set of coupled hybrid automata, thereby allowing for a natural coupling between the continuous and discrete components of the state space; (ii) the underlying assumptions of the CMM allow the algorithmic construction of the control space $\mathcal{V}$ of its constitutive particles solely based on their trajectories in the *continuous state space* $\mathbb{X}$; (iii) because the control modes in $\mathcal{V}$ are mapped to a unique interaction configuration, they form a partition of the *continuous phase space* $\mathbb{S}$, that is, the continuous space of the entire system; (iv) ultimately, by a proper aggregation of those control modes, one can obtain a more tractable and meaningful set of *meta*states, which we denote $q_1, \ldots, q_r$. Importantly, this process of aggregation is precisely the mental process carried out by the designer of a robotic system. The latter *meta*states are the basis for an algorithmic conversion of any CMM into an equivalent macroscopic representation based on the CRN formalism.

As we shall see, the underlying assumptions of the M^3 framework make the intermediate models considerably large and complicated; however, they allow for the automated generation, and subsequent reduction, of models at different abstraction level. The framework constructs the models in a bottom-up fashion: based on observed trajectories of the system and a description of the possible interactions among its constituent components, the framework generates models at increasing levels of abstraction. Figure 9.2 illustrates the governing principle of our approach: for each newly observed interaction configuration, a new control mode is created, and added to the control graph of the corresponding particle in the CMM (Section 9.1). This control graph is then used as blueprint for generating a macroscopic model of the system (Section 9.3). By recording relevant information about the state of the system at each transition, one can also obtain a precise estimate of the parameters of this model (Section 9.3.1). Owing to this systematic approach, the only piece of information that the user needs to provide is the list of interactions that are relevant to the system's dynamics according to a given criterion or performance metric.

9.1 Canonical Microscopic Model (CMM)

Consider a system of m interacting particles $\mathcal{P} = \{P_1, \ldots, P_m\}$ embedded in a n-dimensional continuous state space $\mathbb{X} \subseteq \mathbb{R}^n$. These particles may interact with each other through a set of interactions $\mathcal{I} = \{I_1, \ldots, I_t\}$ (see Definition 9.6 on p. 131). At the microscopic level, each particle P is an individual entity that is described as a hybrid automaton composed of the following components:

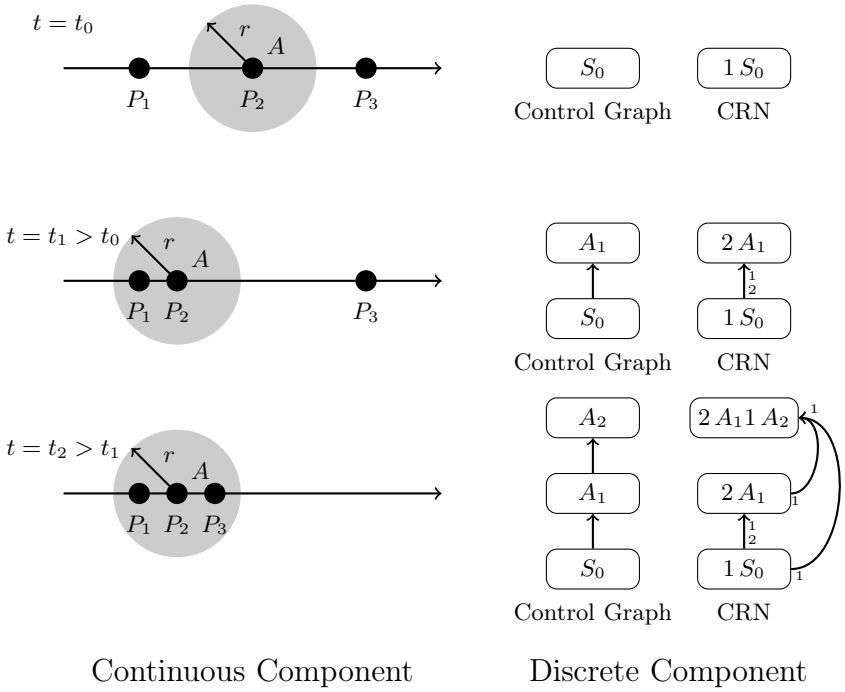


Fig. 9.2 Illustration of the model construction for a 1-dimensional system of 3 particles $\{P_1, P_2, P_3\}$ belonging to the same class and one interaction A, which is active whenever two particles are closer than a given distance r. At $t = t_0$, no interaction has occurred yet, and both the control graph of the particles contains only one mode S_0, which denotes the absence of interaction. Incidentally, the CRN has only one species, and no reaction. Upon the encountering of P_1 and P_2 at $t = t_1$, a new control mode A_1 is appended to the control graph, and the reaction $2 \cdot 1\,S_0 \rightharpoonup 2\,A_1$ is added to the CRN. The control mode A_i denotes that the particle has i partners through interaction A. Finally, at $t = t_2$, P_3 joins the aggregate formed by P_1 and P_2, thereby making P_2 switch to a new control mode A_2, and creating a new reaction $1\,S_0 + 2\,A_1 \rightharpoonup 2\,A_1 1\,A_2$.

Definition 9.1 (Variables). *The real-valued state vector* $\mathbf{x} = [x_1, \ldots, x_n]^T \in \mathbb{X}$ *denotes the position of the particle* P *in the continuous state space* $\mathbb{X}$*. We denote* $\mathcal{X}_i = \{x_{1,i}, \ldots, x_{n,i}\}$ *the set of state variables of the particle* P_i*.*

Definition 9.2 (Control graph). *The finite directed multigraph* $G = (\mathcal{V}, \mathcal{E})$ *is called the* control graph *of* P*. The vertices* ξ_j *in* $\mathcal{V}$ *are called* control modes*, and they form the* control space $\mathcal{V} = \{\xi_1, \ldots, \xi_k\}$*. The edges* e_i *in* $\mathcal{E}$ *are called* control switches*. Each control mode* ξ_j *is labeled with a unique indicator function* $\phi_j : \mathcal{I} \to \{0, 1\}$*, called an* interaction configuration*, defined as*

$$\phi_j(I) = \begin{cases} 1 & \text{if } I \text{ is active in mode } \xi_j, \\ 0 & \text{otherwise.} \end{cases} \tag{9.1}$$

Therefore, each control mode is associated to a unique interaction configuration, and vice versa, such that there exists a one-to-one map $\Phi : \mathcal{V} \to \{\phi_1, \ldots, \phi_k\}$.

Definition 9.3 (Flow condition). *We define the vertex-labeling function* $flow$ *that assigns to each control mode* $\xi \in \mathcal{V}$ *a collection of stochastic processes* $Y_i^{\xi}(t)\,(i = 1, \ldots, n)$ *whose variables are from* $\bigcup_{P_i \in \mathcal{P}} \mathcal{X}_i$.

In other words, the trajectory $x_i(t)$ *along the* i*-th dimension of state space is a realization of some stochastic process* $Y_i^{\xi}(t)$. *More formally, we have:*

$$\mathbf{x}(t) \triangleq \left[Y_1^{\xi_i}(t) \ldots Y_n^{\xi_i}(t)\right]. \tag{9.2}$$

Definition 9.4 (Jump conditions). *The edge-labeling function* $jump(e)$ *assigns to each interaction switch* $e \in \mathcal{E}$ *a predicate that is defined by the interaction associated to the switch* e *(see Definition 9.6).*

Definition 9.5 (Events). *A finite set* $\Sigma = \{\sigma_1, \ldots, \sigma_{2\,t}\}$ *of events and an edge-labeling function event* $: \mathcal{E} \to \Sigma$ *that assigns to each control switch an event. An event* σ_i *denotes either the* creation *or the* destruction *of one interaction* I.

Particles interact via joint events. Let us consider two particles P_1 and P_2: If event σ is both in Σ_1 and Σ_2, then P_1 and P_2 must synchronize on σ-transitions. In other words, P_1 and P_2 must synchronize on transitions labeled by the same interaction.

Definition 9.6 (Interaction). *An interaction* I *is defined as a quintuplet* $(P_i, P_j, cond, \mathcal{D}_1, \mathcal{D}_2)$ *where* P_i *and* P_j *are the particles that may interact through* I*; the predicate* $cond$ *describes the conditions in which* I *is active, and its free variables are in* $\mathcal{X}_i \cup \mathcal{X}_j$*; the sets* $\mathcal{D}_1, \mathcal{D}_2 \subseteq \mathcal{I}$ *contain the interactions in which* P_i *and* P_j *must be involved, respectively, for* I *to be active.*

Remark 9.7 (Monotonicity). Interactions may depend on each other, yet in a restricted manner: an interaction may only depend on the *presence* of another interaction, and not on its absence. This property is important, as it ensures the *monotonicity* of dependency chains, that is, the creation of an interaction may only lead to other interactions being created, but never to a destruction (and vice versa). As a result, if there is a finite number of interactions, dependency chains are finite as well. In the worst case, all interactions are either created (upon creation of an interaction) or destroyed (upon destruction of an interaction).

Remark 9.8. Importantly, the CMM makes no assumption regarding the homogeneity of the system; in other words, all particles are considered to be completely different entities. As a result, a different interaction needs to be defined for each pair of control modes of each particle in the system.

Remark 9.9. Because interactions may depend on each other, one can cast previous works based on grammatical approaches such as [46] into the CMM. Indeed, any graph grammar consists of a set of rules that may be applied or not as a function of the label of the reactants. In our case, the labels correspond to the control modes of each particle, and the rules correspond to feasible interactions, which both modify and depend on these labels.

Definition 9.10 (Phase space). *The phase space* $\mathbb{S}$ *of the system is the product of all individual state spaces, that is,* $\mathbb{S} = \mathbb{X}^m \subseteq \mathbb{R}^{n \cdot m}$*, and we refer to its elements* $\mathbf{x} = [\mathbf{x}_1^T, \ldots, \mathbf{x}_m^T]^T \in \mathbb{S}$ *as* extended states *of the system.*

Definition 9.11 (Extended control space). *The extended control space* $\mathcal{V}^{ext}$ *of the system is the product of all individual control space, that is,* $\mathcal{V}^{ext} = \mathcal{V}_1 \times \ldots \times \mathcal{V}_m$ *and we refer to its elements* $\boldsymbol{\xi} = [\xi^1 \ldots \xi^m]^T \in \mathcal{V}^{ext}$ *as the* extended modes *of the system.*

Theorem 9.12 (Continuous-discrete phase mapping). *Given a finite set of interactions* $\mathcal{I}$*, there exists a function* $\Omega : \mathbb{S} \to \mathcal{V}^{ext}$*, which maps each extended state* $\mathbf{x} \in \mathbb{S}$ *to a corresponding extended mode* $\boldsymbol{\xi} \in \mathcal{V}^{ext}$*, such that* $\boldsymbol{\xi}$ *is the active control mode whenever the system is in state* $\mathbf{x}$*.*

A proof of Theorem 9.12 is provided in Appendix B.

Lemma 9.13 (Phase space partitioning). *There exists a partition* $Q = \{Q_1, \ldots, Q_q\}$ *of the phase space* $\mathbb{S}$ *such that* $Q_i = \{\mathbf{x} \in \mathbb{S} \mid \Omega(\mathbf{x}) = \boldsymbol{\xi}_i\}$ *denotes the set of continuous extended states that correspond to the same extended mode of the system.*

Definition 9.14 (Performance Metric). *A performance metric* Γ *is a real-valued function of trajectories in phase space* $\Gamma : \mathbb{S} \times T \to \mathbb{R}$*. Alternatively, a performance metric can be defined on trajectories (or* traces*) in extended control space* $\Gamma : \mathcal{V}^{ext} \times T \to \mathbb{R}$*.*

Example 9.15 (Brownian Magnetic Particles). Consider the example of m ferromagnetic particles floating in a glass of water (Section 5.1). We restrict the interactions of interest to physical contacts among the particles. As a result, there exists one interaction for each pair of particles (i.e., $n \cdot (n-1)/2$ interactions overall). In these settings, particles have a continuous state $\mathbf{x} = (x, y) \in \mathbb{X} = \mathbb{R}^2$, which corresponds to their position in two-dimensional space. Their control space[2] is $\xi \in \mathcal{V} = \{0, 1, \ldots, 2^{m-1}\}$. A possible (but

[2] In this particular case, all particles have an identical control space, but it is by no means a requirement of the model.

Table 9.1 Summary of the common notation used in this chapter

Symbol	Description
n	Dimensionality of the continuous state space
m	Number of particles in the system
$\mathcal{P} = \{P_1, \ldots, P_m\}$	Set of particles in the system
$\mathbf{x}_i \in \mathbb{X} \subseteq \mathbb{R}^n$	State vector of particle P_i
$G_i = (\mathcal{V}_i, \mathcal{E}_i)$	Control graph of particle P_i
$\mathcal{V}_i = \{\xi_1^i, \ldots, \xi_k^i\}$	Control space of particle P_i
$\mathbb{S} = \mathbb{X}^m$	Phase space of the system
$\mathcal{V}^{\text{ext}} = \mathcal{V}_1 \times \ldots \times \mathcal{V}_m$	Extended control space
$\mathcal{I} = \{I_1, \ldots, I_t\}$	Set of interactions
$\lvert \cdot \rvert$	Cardinality of a given set

non-unique and non-minimal) encoding of the interaction mode $\xi = k$ is that the non-zero bits of k denote the indices of the interacting particles (assuming a bit numbering starting at 1). For instance, $\xi = 9$ is the interaction mode that corresponds to being in contact with particles P_4 and P_1, since $9 = 1001_2$. A single, non-interacting particle is therefore characterized by an interaction mode $\xi = 0$.

Given the above encoding of states, the condition for jumping along edge $e = (k, l)$ is:

$$jump(e) : \begin{cases} 1 & \text{if } \exists P_j \in \mathcal{P} \text{ s.t. } ||\mathbf{x} - \mathbf{x}_j|| \leq 2 \cdot R \text{ and } l = k + 2^{j-1} \\ 0 & \text{otherwise} \end{cases} \tag{9.3}$$

where R is the radius of the particles, thereby describing the radius of particles P and P_j.

The particles perform a random walk throughout the environment, that is, their continuous dynamics are governed by a collection of Wiener processes, which are independent for $\xi = 0$. When $\xi > 0$, the particles share a process with their interacting mates, i.e., a single realization governs the dynamics of an aggregate of particles, such that their relative pose is conserved.

9.2 Constructing the CMM

The CMM constitutes the most detailed microscopic description of a distributed system. However, it is clear from the above example that the derivation (and the implementation) of such a model is difficult for most practical purposes. This observation raises the question of how ABMs tackle this intrinsic limitations of microscopic modeling [218]. Hereafter, we list a series of simplifications that exploit the natural symmetries of distributed systems while preserving the expressiveness of the model. Even though these simplifications are described in abstract terms, they are typically those carried out naturally by the designer of an ABM.

First, it is generally impossible to define *a priori* the control space $\mathcal{V}$ of a particle. The main reason for this is that the number of interactions to be defined grows in principle in $O(a \cdot m^2)$, and the size of the control space of each particle grows in $O(2^{a \cdot (m-1)})$, where a is a constant that depends on the richness[3] of the underlying process. These unfavorable scaling laws quantify the real cost of modeling completely heterogeneous and asymmetric systems at the microscopic level. Fortunately, most (if not all) distributed systems exhibit some invariants that we can exploit to simplify the description of the CMM (and, consequently, of the models derived from it).

First, when the particles cannot or need not be distinguished from each other, they can be arranged into classes. For instance, when dealing with multi-robot systems, it is common to assume that robots with identical specifications are actually copies of each other[4]. Formally, the set $\mathcal{P}$ may be partitioned into an arbitrary set of *classes* of particle $\mathcal{C}_i$ (such that $\bigcup_i \mathcal{C}_i = \mathcal{P}$). Particle of the same class have similar control graphs. As a direct consequence, interactions no longer need to be defined for each pair of particles, but only for each pair of classes, thereby simplifying dramatically the description of the CMM. One can write an interaction $I = (\mathcal{C}_i, \mathcal{C}_j, \mathit{cond}, \mathcal{D}_1, \mathcal{D}_2)$, where $\mathcal{C}_i$ and $\mathcal{C}_j$ now denote two classes of particles, *cond* remains unchanged, and the sets $\mathcal{D}_1, \mathcal{D}_2 \subset \mathcal{I} \times \mathbb{N}_+^\star$ contain pairs of the form (I', k). For I to be active, particles P_1 and P_2, respectively, need to be involved in *at least* k interactions of type I'.

To further simplify the description of the CMM, ABMs generally rely on another important symmetry of distributed systems: most particles do not necessarily behave differently in different control modes; rather, they exhibit behavioral patterns that may span multiple control modes. Therefore, instead of providing an explicit and *a priori* description of each control mode[5], one can define a set of predicates $Q = \{q_1, \ldots, q_{k'}\}$ that are assigned truth values by the modes of $\mathcal{V}$. Given a predicate q of Q, we write $[\![q]\!]$ for the region of $\mathcal{V}$ that satisfy q. Then, it is sufficient to provide a description of the regions defined by Q, and the proper behavior is assigned to each particle in a dynamic manner as the exploration of the control space progresses.

As mentioned earlier, it is prohibitively expensive to define the control space of each individual particle *a priori*. Fortunately, the control space can be built iteratively as the simulation progresses (see Algorithm 2 on p. 135): starting from an initial control space $\mathcal{V} = \{0\}$, which contains only a single,

[3] The *richness* of a process is roughly proportional to a, that is, the average number of *different* interactions that may occur between each pair of particles in the system. A very rich process might involve several types of interactions, such as elastic and non-elastic collisions, long-range attraction, communication, etc.

[4] Note that this particular assumption can become a notable source of error in some specific multi-robot scenarios.

[5] The description of a control mode ξ is provided by the different vertex-labeling functions *init*, *inv*, and flow as well as the map Φ.

Algorithm 2. Iterative construction of the control space $\mathcal{V}$.

Require: $\mathcal{V}_i = \{0\}, \forall i \in \{1, \ldots, m\}$
  while $t < t_{end}$ **do**
    for each $P_i \in \mathcal{P}$ **do**
      Update $\mathbf{x}_i(t+\Delta t)$ according to $flow(\xi_i(t))$ and/or observations of the system

      if an interaction has occurred/ended **then**
        Compute new interaction configuration ϕ'
        if $\neg\exists\xi' \in \mathcal{V}_i$ s.t. $\Phi_i(\xi') = \phi'$ **then**
          Create ξ' and updates Φ_i s.t. $\Phi_i(\xi') = \phi'$
          Append ξ' to $\mathcal{V}_i$ and $e = (\xi_i, \xi')$ to $\mathcal{E}_i$
        end if
        $\xi_i(t + \Delta t) \leftarrow \xi'$
      end if
    end for
    $t \leftarrow t + \Delta t$
  end while

non-interacting mode, each newly observed control mode $\xi > 0$ is appended to $\mathcal{V}$.

In many cases, particles may interact with entities that are not particles *per se*, such as obstacles or specific environmental features, because the designer is not interested in the state of these objects. Therefore, we treat such entities as *virtual particles*, that is, particles whose control space is ignored, but that can interact with regular particles, thereby affecting their control space . Importantly, this mechanism allows the M^3 framework to capture spatial heterogeneity of the environment (e.g., non-uniform agitation or mobility of the agents, presence of obstacles, environmental templates). By defining virtual particles that represent specific regions of the space, one naturally includes spatial information in the control modes of the particles. While this feature is readily available in the current version of the M^3 framework, we have yet to demonstrate its applicability on relevant case studies.

9.3 From the CMM to Macroscopic Models

In this section, we describe an algorithmic approach to construct models of distributed systems, based on a minimal *a priori* description of the system. This method relies on the theoretical foundations laid by the earlier sections; in particular, it uses the control graph of the particles of the system as a blueprint of the model structure. For the sake of scalability, it makes the same general assumptions as those described in Section 9.2; in particular, the algorithm does not require the user to define interactions between pairs of particle, but rather between pair of classes.

A direct consequence of this simplification is that the set of the control modes of each particle in the system no longer provides a complete specification of how these particles interact with each other at the global level. Therefore, one needs to keep track of these global arrangements of interaction by introducing the notion of *interaction graph* and *aggregate*.

Definition 9.16 (Interaction Graph). *The interaction graph G^{int} is a graph whose vertex set $\mathcal{V}^{int}$ is indexed by $\mathcal{P}$, the set of individual particles in the system, and whose edge set $\mathcal{E}^{int}$ is indexed by the set of interactions (more particularly, there exists an edge indexed by interaction I between two vertices indexed by particle P_1 and P_2 iff these particles may interact through I).*

Definition 9.17 (Aggregate). *We note $\mathcal{A} = \{A_1, \ldots, A_a\}$ the set of all possible* aggregates*, that is, connected subgraphs of the interaction graph G^{int}. We also define the* composition *of two aggregates $A_1 \|^I A_2$, which denotes the aggregate resulting from the connection of aggregates A_1 and A_2 through interaction I. The set $\mathcal{A}$ is closed under the operation of composition.*

In order to build a model of the system, one needs to sort the different observed aggregates into categories that will later serve as "species" in the population model. Therefore, we need to define an *equivalence relation* on the set of aggregates.

Definition 9.18 (Equivalence relation). *An* equivalence relation $\sim_{\mathcal{S}}$ *on a given space $\mathcal{S}$ divides $\mathcal{S}$ into a set of disjoint subsets called* equivalence classes*. The set of equivalence classes is called the* quotient space *and denoted $\mathcal{S}/\sim_{\mathcal{S}}$. Any quotient space $\mathcal{S}/\sim_{\mathcal{S}}$ is a* partition *of $\mathcal{S}$. The map $\pi : \mathcal{S} \to \mathcal{S}/\sim_{\mathcal{S}}$ that maps each element of $\mathcal{S}$ to its equivalence class is called the* quotient map*.*

The equivalence relation $\sim_{\mathcal{A}}$ also defines the *level of detail* of the model. One may use the graph isomorphism as equivalence relation (recall that, formally, aggregates are merely subgraphs of G^{int}), which would account for each and every difference in interaction topology. Alternatively, one may rely on coarser criteria (e.g., cardinality, maximum or average vertex degree, etc.) to discriminate between aggregates, thereby leading to simpler models.

The formalism of hybrid automata is useful from a theoretical point of view (e.g., reachability, liveness, etc.), but it is not practical numerically. First, the fact that it does not specify any particular structure for the stochastic processes that govern the continuous dynamics of the particles makes both the specification and the implementation of the CMM difficult[6]. Second, the continuous component of the CMM is typically the most difficult to handle at the macroscopic level.

Hereafter, we overcome these two limitations by exploiting the CRN formalism (see Section 5.3.3); in particular, we show how one can abstract

[6] Nevertheless, this particular feature of the CMM is precisely what makes it relevant theoretically, i.e., it formalizes in general terms the very notion of microscopic model.

the CMM to a corresponding CRN. Indeed, one can write the CMM as a CRN $\mathcal{N} = (\mathcal{R}, \mathcal{S})$ whose species are the control modes in $\mathcal{S} = \bigcup_{P_i \in \mathcal{P}} \mathcal{V}_i$ and the reactions are represented by the interactions in $\mathcal{I}$. Therefore, the reaction

$$\xi_k^i + \xi_l^j \xrightarrow{I} \xi_{k'}^i + \xi_{l'}^j \tag{9.4}$$

exists iff $I = (P_i, P_j, cond, \mathcal{D}_1, \mathcal{D}_2) \in \mathcal{I}$ and every interaction in $\mathcal{D}_1$ and $\mathcal{D}_2$ are active in mode ξ_k^i and ξ_l^j, respectively.

The only remaining component of the CRN to be defined are the propensity functions. At the level of the CMM, propensity functions can be seen as depending on the continuous dynamics of the particles:

$$\mathbf{a}_R(\mathbf{s}, [\mathbf{x}_1^\mathsf{T} \dots \mathbf{x}_m^\mathsf{T}])\, dt = \begin{cases} 1 & \text{if } I \text{ is active,} \\ 0 & \text{otherwise.} \end{cases} \tag{9.5}$$

where $\mathbf{s}$ is a population vector that denotes the state of the CRN (Section 5.3.3); $[\mathbf{x}_1^\mathsf{T} \dots \mathbf{x}_m^\mathsf{T}] \in \mathbb{S}$ is the phase space of the CMM; and I is the interaction associated to R. In other words, the continuous dynamics of the CMM determine which reactions fire in the next time interval $[t, t + dt)$.

When moving up along the abstraction axis (Figure 5.2), one wishes to model the system as a continuous-time Markov process. As a result, propensity functions must no longer depend on the continuous dynamics of the CMM, but solely on the state $\mathbf{s}$ of the CRN (which will be denoted $\mathbf{x}$ from here on, consistently with Section 5.3.3).

9.3.1 Rate Estimation

This section describes the exact form of these propensity functions, and more particularly the estimation of the underlying reaction rates. In the MLMM, one generally performs orthogonal experiments to empirically calibrate the model (Chapter 7). In the M^3 framework, the analysis of the process dynamics provides a precise estimate of the reaction rates, and, to some extent, a measure of the validity of this estimate.

Assuming the Markov property, the time t until the next firing of reaction R is an exponential random variable with mean $1/a_R(\mathbf{x})$, that is, its PDF is given by

$$f(t) = a_R(\mathbf{x}) \cdot e^{-a_R(\mathbf{x}) \cdot t} \tag{9.6}$$

where $\mathbf{x}$ is the state of the CRN (i.e., a population vector), and $a_R(\mathbf{x})$ is the propensity function which we wish to estimate. The function $a_R(\mathbf{x})$ has a unique free variable k_R, called the *rate constant* of reaction R. The form of $a_R(\mathbf{x})$ depends on the type of the reaction R, as prescribed by the law of mass-action. Therefore, if R is the unimolecular reaction $S_1 \to$ product(s), it has the form

$$a_R(\mathbf{x}) = k_R \cdot x_1 \tag{9.7}$$

where x_1 is the population of species S_1, and assuming that at least one copy of S_1 is required for the reaction R to occur. If R is a bimolecular reaction $S_1 + S_2 \to$ product(s), the propensity function $a_R(\mathbf{x})$ becomes

$$a_R(\mathbf{x}) = k_R \cdot x_1 \cdot x_2. \tag{9.8}$$

If two copies of the same reactant are involved, that is, $R : S_1 + S_1 \to$ product(s), then we have

$$a_R(\mathbf{x}) = k_R \cdot \frac{1}{2} \cdot x_1 \cdot (x_1 - 1). \tag{9.9}$$

For the sake of simplicity, we shall summarize these three forms using the following notation:

$$a_R(\mathbf{x}) = k_R \cdot \widetilde{a}_R(\mathbf{x}) \tag{9.10}$$

where $\tilde{a}_R(\mathbf{x})$ takes the appropriate form according to the stoichiometry of R, and does not depend on k_R.

Therefore, the problem we intend to solve hereafter is the following: Given a sequence of events $(e_1, \ldots, e_n)$, with $e_i = (R_i, t_i, \mathbf{x}_i)$, what are the most likely rate vector $\hat{\mathbf{k}} = \left[\hat{k}_1, \ldots, \hat{k}_N\right]$ of the underlying CRN? More formally, we want to solve the following problem:

$$\hat{\mathbf{k}} = \underset{\mathbf{k}}{\operatorname{argmax}}\ \mathcal{L}(\mathbf{k}|e_1, \ldots, e_n) = \underset{\mathbf{k}}{\operatorname{argmax}}\, f(e_1, \ldots, e_n|\mathbf{k}) \tag{9.11}$$

where $\mathcal{L}(\mathbf{k}|e_1, \ldots, e_n)$ is the likelihood of the rate vector $\mathbf{k}$ given the sequence of observed events $(e_1, \ldots, e_n)$. The solution to this problem has a *closed-form* expression (see Appendix A for the complete derivation):

$$\widehat{k}_{R_j} = k_{R_j} = \frac{\sum_{i=1}^{n} \mathbf{1}_{\{R_i = R_j\}}}{\sum_{i=1}^{n} \left(t_i \cdot \widetilde{a}_{R_j}(\mathbf{x}_i)\right)}. \tag{9.12}$$

9.4 Model Reduction and Refinement

Now that we know how to construct and calibrate a CRN from an underlying description based on the CMM, we shall discuss how the *structure* of this CRN can be adjusted to obtain the desired level of detail. Both *reduction* (i.e., a decrease of the level of detail) and *refinement* (i.e., an increase of the level of detail) of CRNs rely on the notion of *partitioning*, which is induced by an *equivalence relation* (see Definition 9.18).

Definition 9.19 (Reduction and Refinement of a CRN). *A CRN* $\widetilde{\mathcal{N}} = (\widetilde{\mathcal{R}}, \widetilde{\mathcal{S}})$ *is said to be a* reduction *of* $\mathcal{N} = (\mathcal{R}, \mathcal{S})$ *(and, conversely,* $\mathcal{N}$ *is said to be a* refinement *of* $\widetilde{\mathcal{N}}$*) iff (1) there exists an equivalence relation* $\sim_{\mathcal{S}}$ *on* $\mathcal{S}$ *such that* $\widetilde{\mathcal{S}} = \mathcal{S}/\sim_{\mathcal{S}}$ *and (2) there exists a surjective map such that each reaction* $R \in \mathcal{R}$ *has a corresponding abstracted reaction* $\widetilde{R} \in \widetilde{\mathcal{R}}$ *with*

$$r_{\widetilde{R},j} = \sum_{i=1}^{M} \mathbf{1}_{S_i \in \widetilde{S}_j} \cdot r_{R,i} \tag{9.13}$$

$$p_{\widetilde{R},j} = \sum_{i=1}^{M} \mathbf{1}_{S_i \in \widetilde{S}_j} \cdot p_{R,i} \qquad j = 1, \ldots, \widetilde{M} \tag{9.14}$$

where $\mathbf{1}_{S_i \in \widetilde{S}_j}$ *is an indicator function, that is, it equals* 1 *iff* $S_i \in \widetilde{S}_j$ *and* 0 *otherwise. The propensity function of each reaction* $\widetilde{R} \in \widetilde{\mathcal{R}}$ *is the sum of the propensity functions of all reactions in the preimage of* $\widetilde{R}$*.*

9.4.1 Model Reduction

Model reduction is completely specified by Definition 9.19, and there is no particular caveat concerning its algorithmic implementation: reactions and species are lumped according to an underlying equivalence relation. However, one may consider many different levels of detail, which correspond to different equivalence relations, of course, but also different rationales. There exists in particular a trivial partition that allows one to form an exact model of any system: a unique region that spans the entire phase space, in which the system is in with probability one at all times. Without further guidance, any partitioning algorithm would lead to this trivial partition; therefore, one needs an objective criterion to guide the algorithm.

In our approach, this criterion is given by the performance metric Γ, such that it does not vary "too much" under the process of partitioning. Given a CRN $\mathcal{N}$, one can compute an estimator $\hat{\Gamma}$ of the performance metric of the underlying system. The error $e = \left|\hat{\Gamma} - \Gamma\right|$ generally increases as the abstraction of the CRN increases. The reason for this is two-fold: (i) the predictions of the model become more inexact, (ii) the variability of Γ for a given state $\mathbf{x}$ of the CRN increases. The challenge is therefore to simplify the model as much as possible while maintaining this value e under a certain threshold. Section 9.5.1 illustrates this trade-off by comparing models of Case Study I at different levels of detail.

9.4.2 Model Refinement

In the general case, model refinement is an inference problem in its own right, and Definition 9.19 provides no algorithmic specification of how a given refinement $\mathcal{N}$ of an existing CRN $\widetilde{\mathcal{N}}$ is constructed and calibrated. Napp and Klavins [219] solved a particular instance of this problem in an attempt to deal with the suboptimal mixing of their system. In particular, they considered a refined CRN whose species are "augmented" with the types of previous interaction partners. This approach is motivated by the fact that two particles are more likely to interact next if they interacted recently (see Figure 5.5 on p. 75). As a result, one should account for this fact by

augmenting any state with the previous interaction partner: for instance, the augmented state A_X denotes the particles that are in state A, but recently were in state X. As a result, the reaction $A_X \rightharpoonup X_A$ is more likely to fire in the next $[t, dt)$ than, say, the reaction $A_X \rightharpoonup Y_A$, even though both $A \rightharpoonup X$ and $A \rightharpoonup Y$ have the same rate in the original CRN.

In the M^3 framework, we adopt a similar approach, but in more systematic terms. Indeed, it is relatively easy to reproduce the approach of Napp and Klavins by accounting for history in the state construction: rather than mapping control modes to interaction configurations in the CMM, we map them to *interaction traces*, that is, a sequence of triplets (I_1, d_1, t_1), where $d_i \in \{+, -\}$ indicates whether the interaction $I_i \in \mathcal{I}$ has been created or destroyed at time t_i, respectively. Incidentally, any interaction configuration can be re-constructed based on an interaction trace. For instance, the trace

$$\big((A, +, 0.25), (B, +, 5), (A, -, 6.5), (B, +, 8)\big) \tag{9.15}$$

yields the interaction configuration B_2. Of course, tracking the complete interaction history of each particle in the CMM leads to an explosion of the number of state. One can therefore adopt a mixed approach where only a limited number of previous interaction events are explicitly recorded, and the others are summarized as an interaction configuration.

However, for most practical purposes, this approach based on interaction history is not efficient, because there are only a few states that actually need to be augmented, whereas the others can be safely left unchanged. Therefore, we propose a systematic method for augmenting, or *refining*, a macroscopic model *a posteriori*, that is, after its generation. More specifically, given a CRN $\mathcal{N} = (\mathcal{R}, \mathcal{S})$ and an associated sequence of events $(e_1, \ldots, e_n)$, we aim at constructing a refinement $\widetilde{\mathcal{N}} = (\widetilde{\mathcal{R}}, \widetilde{\mathcal{S}})$ of $\mathcal{N}$ that is such that the likelihood $\mathcal{L}(\widetilde{\mathbf{k}}|e_1, \ldots, e_n) > \mathcal{L}(\mathbf{k}|e_1, \ldots, e_n)$, i.e., the sequence of events is better explained by the refined model than the original one.

From a structural point of view, the refinement of a CRN is rather simple: we select a species of interest, say S_0, and we split it into two subspecies S_0^a and S_0^b. We then duplicate all reactions that involve S_0, and update their reactants and products accordingly. For instance, the reaction

$$R_j: \quad S_0 + 2\,A_1 \rightharpoonup 3\,A_2 \tag{9.16}$$

is duplicated as follows:

$$R_j^a: \quad S_{0,a} + 2\,A_1 \rightharpoonup 3\,A_2 \tag{9.17}$$

$$R_j^b: \quad S_{0,b} + 2\,A_1 \rightharpoonup 3\,A_2. \tag{9.18}$$

Contrary to the approach proposed by Napp and Klavins [219], the resulting subspecies $S_{0,a}$ and $S_{0,b}$ need not be associated to a particular state property, such as the type of a previous interaction partner; they are mere duplicates of the original species. We know that both $S_{0,a}$ and $S_{0,b}$ correspond to the

same control mode as S_0 in the original CRN, but their specific "meaning" is implicitly determined by the framework upon inferring the rates of the reactions they are involved in.

However, the estimation procedure described in Section 9.3.1 cannot be readily applied because the sequence of events $(e_1, \ldots, e_n)$ describes the firing of the original reactions, but it contains no information regarding the newly created reactions. More specifically, if an event $e_i = (R_i, t_i, \mathbf{x}_i)$ is such that R_i is duplicated, one needs to re-assign this event to either $R_i = R_i^a$ or $R_i = R_i^b$, and update accordingly the population vectors $\mathbf{x}_{j>i}$ of the upcoming events.

To solve this problem, we use an expectation-maximization (EM) algorithm, that is, an iterative method for finding maximum likelihood estimates of parameters in statistical models that depend on unobserved latent variables. An EM algorithm consists of an expectation (E) step, which computes the expectation of the log-likelihood evaluated using the current estimate of the parameters $\mathbf{k}$, and a maximization (M) step, which computes new values of these parameters by maximizing the log-likelihood function. More formally, given a CRN parameterized by an unknown rate vector $\mathbf{k}$, the E-step can be written

$$Q(\mathbf{k}|\mathbf{k}^{(t)}) = E_{\mathbf{Z}|\mathbf{X},\mathbf{k}^{(t)}}\big[\ln \mathcal{L}(\mathbf{k}|\mathbf{X},\mathbf{Z})\big], \tag{9.19}$$

that is, the expected value of the log-likelihood function with respect to the conditional distribution of the set $\mathbf{Z}$ of unobserved latent variables (the reactions and the population vectors associated the events to be re-assigned), given a set $\mathbf{X}$ of observed data (the timestamp of all events, and the reactions and the population vectors associated to the valid events), under the current estimate of $\mathbf{k}^{(t)}$. Then, the M-step consists in finding the rate vector maximizing this quantity:

$$\mathbf{k}^{(t+1)} = \arg\max_{\mathbf{k}} Q(\mathbf{k}|\mathbf{k}^{(t)}). \tag{9.20}$$

The E-step requires to compute the log-likelihood function over all possible values of $\mathbf{Z}$. More specifically, since $\mathbf{Z}$ consists of the reactions and population vectors associated the events to be re-assigned, and that each assignment depends on the assignment of earlier events, one need to keep track of the log-likelihood of all possible trajectories of the system, which boils down to finding a constrained solution of the CME. In the general case, solving the CME numerically is computationally expensive or even infeasible as the number of reachable states can be very large or infinite [185]. However, in our case, we can dramatically simplify the problem at hand by exploiting the pieces of information available from the observed data $\mathbf{X}$. In particular, we know *when* a reaction is supposed to fire, therefore the only question is *which* one does. In other words, we know that the PDF of the state distribution remains constant between two firing events, thereby making the problem quasi-discrete, and suitable for being solved by dynamic programming (see Algorithm 3 on p. 142), using an approach similar to the Viterbi algorithm [220].

Finally, the M-step consists in selecting the trajectory with the largest log-likelihood, and applying the procedure described in Section 9.3.1 to find the rate vector $\mathbf{k}^{(t+1)}$ that corresponds to this most likely trajectory.

Algorithm 3. E-step of the EM algorithm used for CRN refinement.

Require: Set $X = \{(\mathbf{x}_0, L_0)\}$ with $L_0 = 0$
  for each event $e_i = (R_i, t_i, \mathbf{x}_i)$ **do**
    for each $(\mathbf{x}, L) \in X$ **do**
      if R_i is refined into R_i^a and R_i^b **then**
        if R_i^a is feasible given $\mathbf{x}$ **then**
          Add $\left(\mathbf{x} + \boldsymbol{\nu}_{R_i^a}, L + \ln\left(a_{R_i^a}(\mathbf{x}) \cdot e^{-a_0(\mathbf{x})\cdot t_i}\right)\right)$ to X
        end if
        if R_i^b is feasible given $\mathbf{x}$ **then**
          Add $\left(\mathbf{x} + \boldsymbol{\nu}_{R_i^b}, L + \ln\left(a_{R_i^b}(\mathbf{x}) \cdot e^{-a_0(\mathbf{x})\cdot t_i}\right)\right)$ to X
        end if
      else
        if R_i is feasible given $\mathbf{x}$ **then**
          Add $\left(\mathbf{x} + \boldsymbol{\nu}_{R_i}, L + \ln\left(a_{R_i}(\mathbf{x}) \cdot e^{-a_0(\mathbf{x})\cdot t_i}\right)\right)$ to X
        end if
      end if
      Remove $(\mathbf{x}, L)$ from X
    end for
  end for

9.5 Results and Discussion

In this section, we present the results of a series of quantitative case studies that demonstrate the validity of the framework, and also outline some of its limitations. An important feature of the M^3 framework is its platform-independence; indeed, the very same computational framework is applicable regardless of the exact nature of the underlying system (see Figure 9.3). To illustrate this claim, we demonstrate hereafter the use of the M^3 framework on case studies I, II, III, and IV. All graphs of this section use the following standardization: bold lines indicate the mean trajectory over 1000 runs, and dashed lines depict the $\pm 1.96\,\sigma$ interval (corresponding to the 95% confidence interval if the data are normal). If not indicated otherwise, the ground truth is always shown in black, and corresponds to original data collected from real or simulated experiments for constructing the models; the predictions of these models are shown in green or in red. The score of the models is a measure of the quality of their prediction (the lower, the better), which is given by the average KS distance between ground truth trajectories and predicted trajectories.

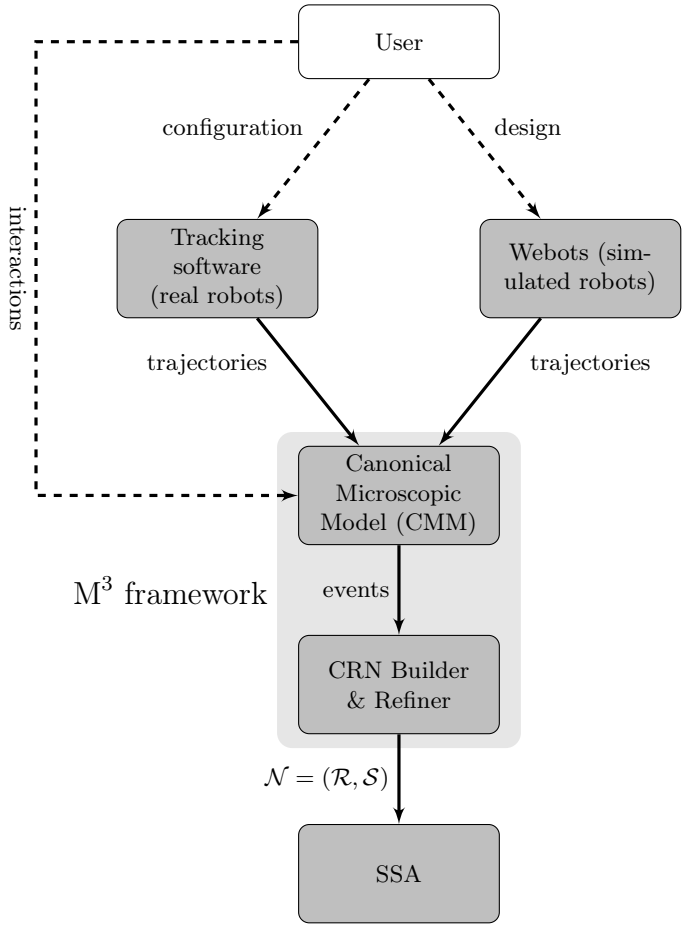


Fig. 9.3 Overview of the software deployment around the M^3 framework in this study, and the different types of information flowing between its constitutive modules. Dashed arrows denote flows that are not automated, but need to be performed only once prior to the experiment.

First, we shall provide a validation of the models generated by the M^3 framework; in particular, we aim at illustrating the excellent accuracy of the rate estimation described in Section 9.3.1. Figure 9.4 provides a comparison of the model's predictions and the ground truth from Case Study I (Section 4.1). An important feature of the generated model is that it is detailed and large (3965 species and 10948 reactions), which makes it difficult to solve and calibrate. Note however that even such large models are still tractable using efficient solvers such as StochKit; as a reference, the

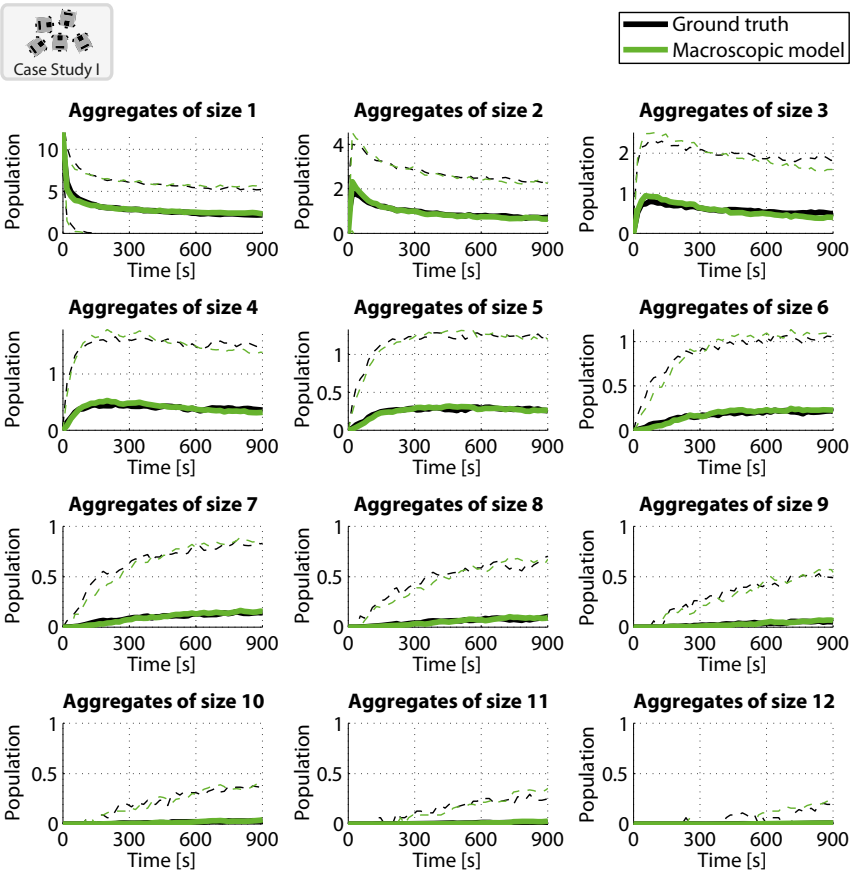


Fig. 9.4 Graphical representation of the most detailed model (3965 species, 10948 reactions) generated by the M^3 framework on Case Study I. All species contributing to the same aggregate are lumped into one trajectory. The model's predictions agree almost perfectly with the original data, both in terms of the mean and the variance.

trajectories shown in Figure 9.4 were obtained in about 30 seconds on a quad core 2.66 GHz desktop workstation, and their post-processing lasted a few minutes.

9.5.1 Model Reduction

As advocated by several previous works [221, 222], models can often be simplified dramatically without sacrificing much of their accuracy. Most previous approaches to model reduction use *a posteriori* strategies that consist in lumping and/or eliminating insignificant reactions and species on the basis

Table 9.2 Properties and score of macroscopic models at different level of detail for Case Study I

Level of Detail	Species	Reactions	Score
Cardinality	12	60	$4.16 \cdot 10^{-2}$
Min Vertex Degree	62	406	$1.81 \cdot 10^{-2}$
Mean Vertex Degree	166	1072	$1.78 \cdot 10^{-2}$
Isomorphism	3965	10948	$2.04 \cdot 10^{-2}$

of the reaction kinetics. Alternatively, advanced simulation schemes such as the slow-scale variant of SSA constitute robust and seamless methods for reducing models on the fly as they are being solved [223].

In the context of the M^3 framework, our approach to model reduction is different; it consists in tuning the level of detail of the CMM (or, more formally, the equivalence relation on the set $\mathcal{A}$). For example, Figure 9.5 depicts the predictions of the model (Case Study I) produced by the coarsest non-trivial equivalence relation on $\mathcal{A}$, that is, two aggregates A_i and A_j are equivalent if and only if they have the same cardinality. Figure 9.6 provides two graphical representations of this model.

Intermediate levels of detail are also considered: in a second level of detail, we also include the minimum vertex degree as a criterion for discriminating between aggregates. The third level of detail uses, on top of the cardinality and the minimum vertex degree, the mean vertex degree. Finally, the highest level of detail recourses to full graph isomorphism as equivalence relation, thereby resulting in the enormous model described earlier (3965 species and 10948 reactions).

Table 9.2 lists the size and the score of each model. As expected, more detailed models tend to perform better than the coarsest one. However, interestingly, the most detailed model performs worse than both intermediate models, in spite of their much smaller size. This result outlines two critical facts. First, large models require large datasets in order to achieve an accurate calibration; more generally, because they involve numerous species and reactions, they are difficult to analyze and inspect. Second, owing to the power of automated modeling, one can explore such trade-offs between size and accuracy in models of SMPs. In our particular case, for instance, our results indicate clearly that an accurate model needs to account for the connectivity of the aggregate, to some extent[7]. However, it seems that it is sufficient to track the minimum vertex degree rather than, say, the mean or the complete vertex degree distribution. Such pieces of information about the relevance of given system's features are crucial to model design, and can be obtained efficiently only through automated modeling, as the manual construction of such detailed models is prohibitively difficult and time-consuming.

[7] Note that the model presented in [92] does not account for any such feature.

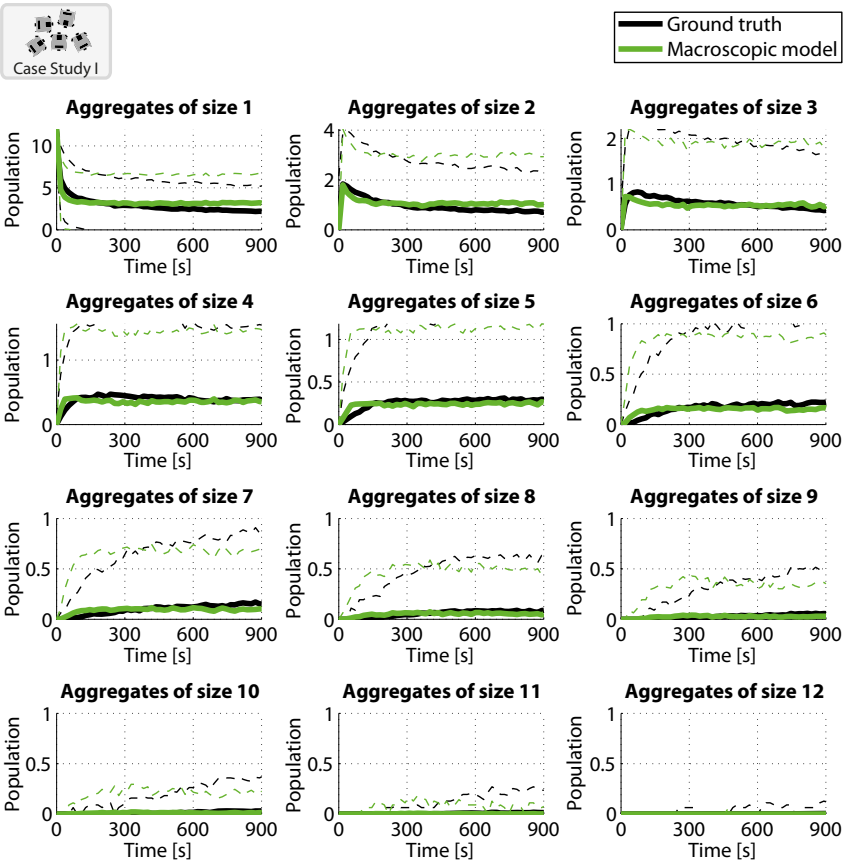
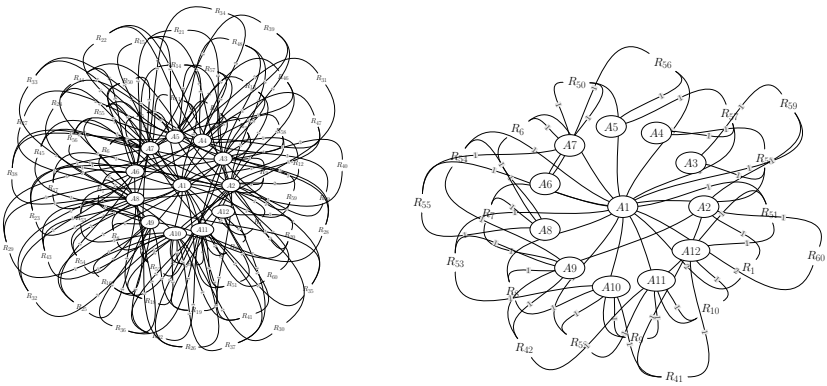


Fig. 9.5 Validation of the reduced model (12 species, 60 reactions) generated by the M^3 framework on Case Study I. One can observe that the model's predictions still agree relatively well with the original data, at least from a qualitative standpoint, in spite of the dramatic reduction in size (from 3965 to 12 species, and from 10948 to 60 reactions).

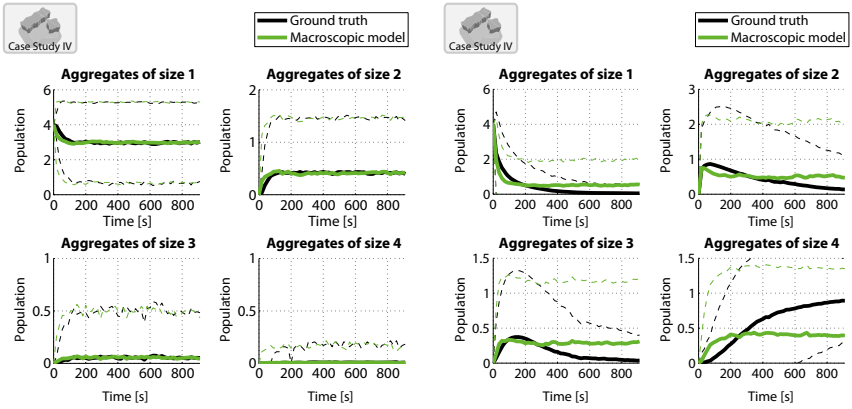
9.5.2 Model Refinement

One fundamental assumption of the methodology described in this chapter is that the underlying system can be modeled using a continuous-time Markov process. As discussed in Section 5.3.5, if the system is not well-mixed, the models generated by the M^3 framework might not provide accurate predictions. However, for most practical purposes, even non-Markovian dynamics can be accurately approximated by a continuous-time Markov process, and the quality of this approximation dramatically depends on the model's structure.



(a) Complete representation (b) Simplified representation

Fig. 9.6 Graphical representations of the coarsest model generated by the M^3 framework on Case Study I. (a) Complete representation of the associated CRN (12 species, 60 reactions). (b) Simplified representation in which reactions with a rate lower than 0.01 have been pruned. The inspection of automatically generated models is generally hindered by the lack of meaningful organization, even when they remain compact; however, in this simplified representation, most reactions can be easily associated to their counterpart in the model presented in [92].



(a) Well-mixed system (b) Non-well-mixed system

Fig. 9.7 Predictions of models generated by the M^3 framework on two simulated variations of Case Study IV: (a) a well-mixed system in which long range forces are negligible (score $= 1.8 \cdot 10^{-2}$), and (b) the same system rendered non-well-mixed by an increase of the magnets' strength (score $= 19.8 \cdot 10^{-2}$).

To illustrate this statement, we consider two variations of Case Study IV in simulation. In the first case, the system is relatively well-mixed and the long range forces induced by the magnets are almost negligible. In the second case, we increase the strength of the magnets such that the influence of spatiality becomes more important. The predictive accuracy of the generate models in either case is different, as illustrated by Figure 9.7. Indeed, it is clear, both qualitatively and quantitatively, that the accuracy of the predictions degrades as the influence of spatiality increases.

A common approach to mitigating the adverse effects of spatiality is to keep track of the interaction history. As discussed in Section 9.4.2, the M^3 framework can rather easily track the interaction history of the particles in the CMM. Furthermore, one can tune the number of previous interaction events being accounted for during the state construction. Figure 9.8 compares the predictions of a regular model without history and a model constructed using a 1-step history. While this approach allows for a clear improvement of the predictive accuracy, we shall outline that the use of history leads to a severe increase of the model size.

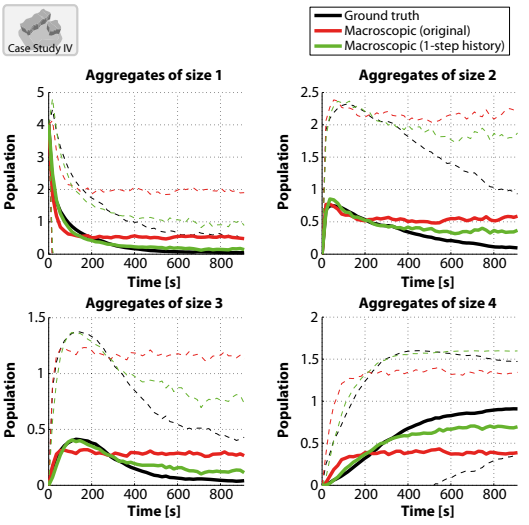


Fig. 9.8 Comparison of the predictions of macroscopic models with and without a 1-step history in the non-well-mixed variation of Case Study IV. The model size increases from 6 species and 12 reactions without history (score $= 20.5 \cdot 10^{-2}$) to 22 species and 220 reactions with history (score $= 11.7 \cdot 10^{-2}$).

Alternatively, one can selectively refine portions of the models using the procedure described in Section 9.4.2, and thereby accounting for some of the spatial features of the underlying system. Figure 9.9 depicts the evolution of both the score and the negative likelihood of the refined models. As expected from the properties of the EM algorithm, the negative log-likelihood monotonically decreases as the level of refinement increases, which is not the case of the score. The consequences of this observation are two-fold: (i) a model's likelihood is not exactly correlated with its predictive accuracy, and

(ii) as observed earlier in Section 9.5.1, a subtle trade-off between compactness and expressivity must be found in order to produce accurate accurate models. In other words, in spite of the optimality of our calibration procedure (in terms of model's likelihood given an original dataset), the choice of the model structure is crucial for achieving both accuracy and compactness.

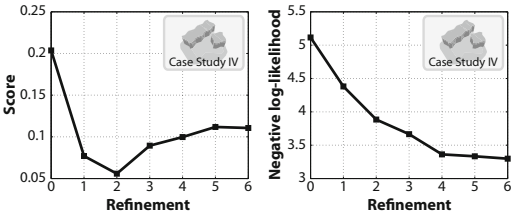


Fig. 9.9 Evolution of the score (left) and the negative log-likelihood (right) of different model refinements of Case Study IV. The 0-th refinement indicates the original model; the other models are obtained by refining iteratively the following species (see Figure 4.6): (1) A, (2) B, (3) C, (4) F, (5) E, and (6) D.

Figure 9.10 illustrates the benefits of refinement in a more qualitative fashion by comparing the predictions of the original model, and its best refinement. The clearest improvement concerns the trajectory of single building blocks (top left), which is inaccurately predicted by the original model both in transient and steady state, but becomes a near-perfect fit in the refined model. Other trajectories are also significantly enhanced, even though some discrepancies are still present, in particular in the long run.

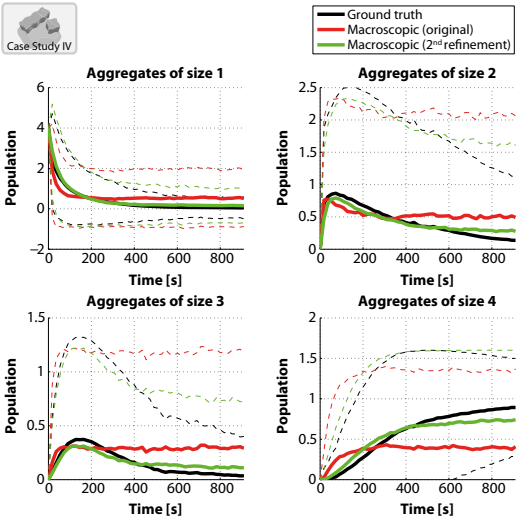


Fig. 9.10 Comparison of the predictions of the original model (score $= 20.5 \cdot 10^{-2}$) and its second refinement (score $= 5.5 \cdot 10^{-2}$) in the non-well-mixed variation of Case Study IV.

9.5.3 Scalability

To investigate the scalability of the M^3 framework to large-scale distributed systems, we apply it to Case Study III, which can be easily scaled up to many building blocks. Figures 9.11 and 9.12 illustrate the performance of the generated models using only 10 and 100 building blocks, respectively. In both cases, the model's prediction agrees almost perfectly with the original data. The reason for the high accuracy of the generated models is threefold: (i) the underlying system is well-mixed, (ii) the irreversibility of the bonds make spatial effects described earlier less prominent, and (iii) the model structure is simple, with only two species and two reactions.

Interestingly, in spite of the large number of building blocks, the model remains compact, and therefore the computational cost of the model generation is low. As far as scalability is concerned, the only decisive criterion is the size of the generated model, as well as, to some extent, the amount of data needed for the rate calibration. As a matter of fact, the size of the model (and, incidentally, the computational cost of its generation) is generally weakly correlated with the size of the system. A rather obvious illustration of this fact is that the largest model presented in this section contains 3965 species and 10948 reactions (Figure 9.4), and is based on a system that involves only 12 robots, whereas the model of Figure 9.12, which is based on a system that involves 100 agents, contains only two species and two reactions.

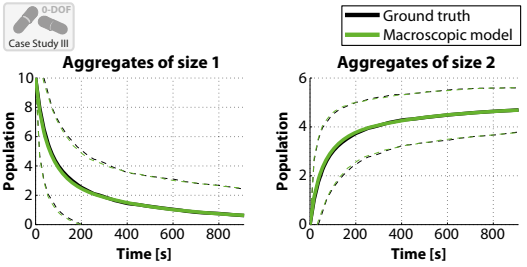


Fig. 9.11 Validation of the macroscopic model generated by the M^3 framework on Case Study III (10 building blocks).

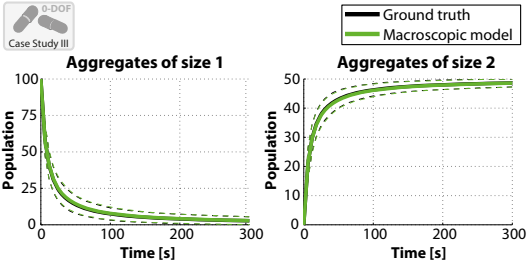


Fig. 9.12 Validation of the macroscopic model generated by the M^3 framework on Case Study III (100 building blocks).

9.5.4 Embodiment

As outlined in previous chapter, the notion of embodiment is crucial to the proper modeling of distributed robotic systems. We demonstrate the ability of the M^3 framework to capture such features by modeling two systems in which embodiment plays a key role: (i) the pairwise SA of microscale building blocks (Case Study III) with misalignment, and (ii) the stochastic SA of Alice robots into chains (Case Study II).

In the first case, we compare four models with different levels of discretization. In the M^3 framework, discretization is implemented by adding interactions that are active only when the building blocks have a given alignment. We implement four distinct models with 1, 2, 5, and 10 different categories of interaction, with respective score of $3.57 \cdot 10^{-2}$, $2.76 \cdot 10^{-2}$, $2.26 \cdot 10^{-2}$, and $2.24 \cdot 10^{-2}$. Figure 9.13 depicts the original trajectories along with the predictions of the two extreme models (with 1 and 10 interactions, respectively). As expected, the most detailed model is also the most accurate, but the difference is significant only in the transient regime; at the steady state, all models yield the same state distribution. Unexpectedly, the model also indicates the formation of trimers, and even tetramers, of low stability. While these species are marginal, their presence in the model illustrates one fundamental benefit of automated modeling: the impact of the designer's preconceptions about the underlying system is minimized. Indeed, in Section 6.3.2, the equivalent model designed by hand does not account for the formation of assemblies larger than dimers. Of course, one may argue that the reason for this is that trimers and tetramers are unstable, and can therefore be safely neglected; yet, the most likely reason is that the modeler forgot to include them in the model because it seemed *a priori* than only dimers could form.

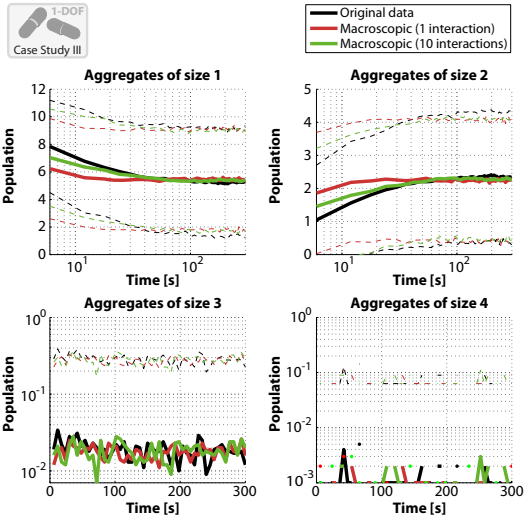


Fig. 9.13 Comparison of two models of Case Study III with misalignment: (i) a simple model with a single interaction capturing all bonds, and (ii) a more detailed model with 10 different interactions accounting for different degrees of misalignment. The score of each model is $3.57 \cdot 10^{-2}$ and $2.24 \cdot 10^{-2}$, respectively. Note that that the top plots are log-lin whereas the bottom plots are lin-log.

In the second case, we evaluate the ability of the M^3 framework to deal with multiple difficulties at once: embodiment, spatiality, scalability, and small original dataset. Indeed, based on only 10 runs with 20 real Alice robots, we construct a macroscopic model of Case Study II that yields quite accurate predictions (Figure 9.14) in spite of the small number of runs and the large size of the resulting model (452 species and 1474 reactions). As it turns out, some strong fluctuations of the average original trajectory are observed; yet, its variance is in the same order as that exhibited by the model, thereby indicating a strong intrinsic stochasticity of the underlying system.

In terms of embodiment, the framework allows one to probe some insightful characteristics of the system, such as the proportion of well-formed chains[8] of a given size. Figure 9.15 shows that the proportion of well-formed chains decreases with their size: while almost 90% of the chains of size 3 are well-formed, the ratio drops to 60% for chains of size 5, and less than 20% for chains of size 6. This type of statistics would have been difficult to obtain without an appropriate analysis framework, and even more difficult to model, given the large number of non-well-formed conformations.

9.5.5 Limitations

At its current level of development, the M^3 framework still exhibits several important limitations, which will constitute important lines of research in the future:

- Our calibration method based on Maximum Likelihood Estimation (MLE) requires that the system is fully observable. In particular, the framework does not include any inference mechanism to deal with measurement time-gaps, unobservable variables, or inaccurate observations. However, this limitation might be overcome using more sophisticated calibration and inference techniques (used for instance for the inference of Hidden Markov Models [224]) such as particle filtering [225], Markov Chain Monte Carlo methods [226], or evolutionary algorithms [165].
- The models generated by the M^3 framework are characterized by parameters (i.e., reaction rates) that have no direct anchoring to the control and design parameters of the underlying system. Instead, they result from the lumping of several low-level parameters and phenomena. As a result, these models are difficult to leverage for design and control purposes[9]. However, owing to its solid methodological foundations, the M^3 framework has the potential to overcome this limitation in an elegant way.

[8] A chain of size n is *well-formed* if it involves two robots with a single neighbor (one at each end), and $n-2$ robots have two neighbors. Non well-formed chains may form due to IR reflections, as shown in Figure 8.4.

[9] Not impossible, though, as demonstrated by Chapter 12, which introduces an extension of the M^3 framework for the model-based, real-time control of stochastic self-assembling systems.

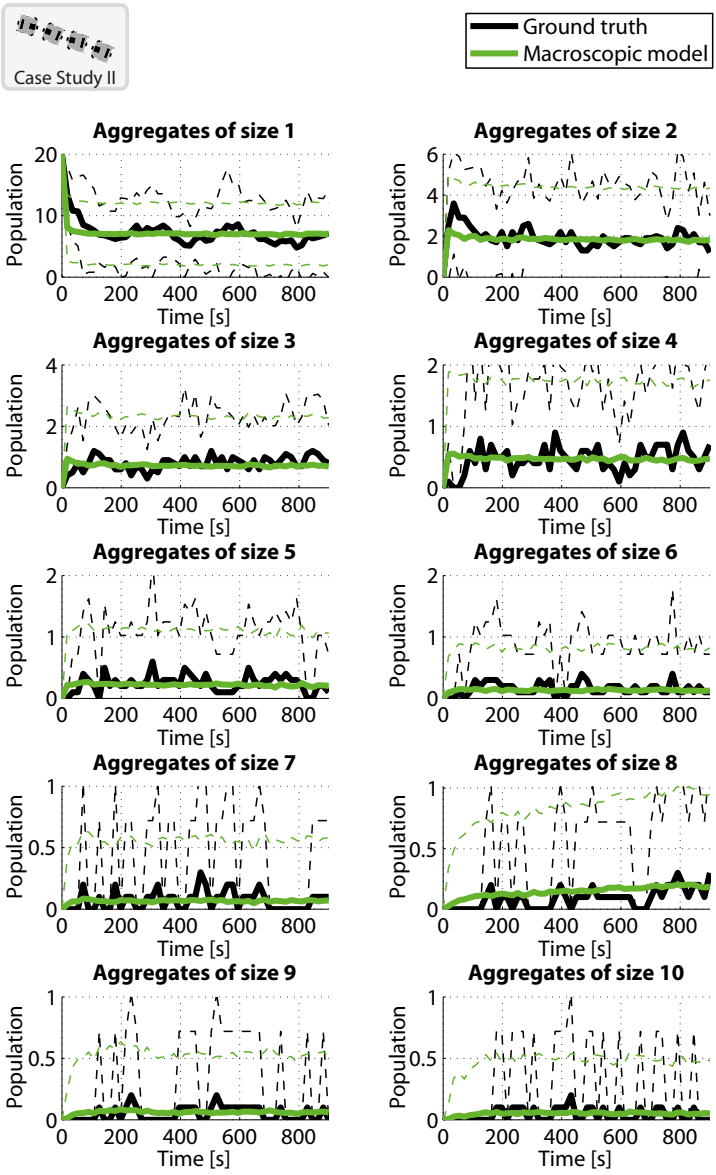


Fig. 9.14 Validation of the macroscopic model generated by the M^3 framework on Case Study II (1000 runs, 20 robots). Ground truth is provided by real experiments with 20 Alice robots (10 runs).

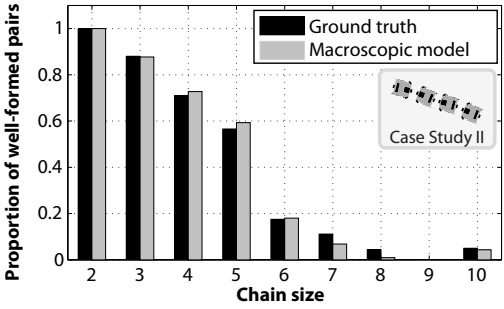


Fig. 9.15 Proportion of well-formed chains for each size category as observed in real experiments (ground truth) and as predicted by the macroscopic model generated by the M^3 framework.

Indeed, since each reaction is associated to a unique interaction, which is in turn characterized by a set of low-level parameters of the system, a careful inspection of the different reaction rates might allow one to infer analytical expressions for the reaction rates. This task require that the framework, on top of exploiting observations from prior experiments, collects additional experimental data for selected control and design parameters (ideally in an automatic and dynamic manner). For instance, the framework could leverage principles of experimental design [227] and evolutionary strategies [162, 228] to construct accurate and transparent models while minimizing the number of experiments required to generate these models.

- Even though the M^3 framework provides formal methods for reducing and increasing the level of detail of a given model (see Section 9.4), it lacks automated strategies for exploring the trade-offs between model size and model accuracy. Again, the framework is intended as a solid formal and methodological foundation, upon which one can build more advanced strategies.
- There is no experimental evidence that models constructed on the basis of observations of 10 units can predict the performance of, say, 1000 units. We hypothesize that the models generated by the M^3 framework will scale well as long as the individual properties of the units and their density remain the same, but we have yet to demonstrate this claim experimentally. Another related question is, what is the minimal number n_{min} of units required to obtain an accurate model (within a certain error margin) for a specific number $n > n_{min}$ of units? This question is very important in the context of SMPs, as one is often interested in extrapolating the performance of a large-scale group based on preliminary experiments based on a few prototypes. Owing to its solid methodological anchoring, we believe that the M^3 framework might offer an excellent basis for the development of mathematical and computational tools capable of answering this important question.

Summary and Conclusion

This chapter introduces the M^3 framework, which provides a consistent set of theoretical and computational tools targeted to the automated construction of models of SMPs at high abstraction level. The essential assumption underlying our approach is that all behavioral changes can be interpreted as the result of an interaction with another robot or the environment. Based on this axiom, we can systematically construct a hybrid, microscopic representation of the system, which relies on the theoretical concept of CMM, that is, a mathematical construct upon which all microscopic models are copied or emulated. By updating the CMM as new observations of the system are collected, one obtains an accurate, microscopic representation of the said system, which can be automatically converted into an equivalent macroscopic model.

Our approach offers the ability to tune the degree of detail of the generated models by automatically lumping the species that correspond to "similar" graph structures. Conversely, the models generated by the M^3 framework can be refined in order to improve their accuracy when the underlying system is not well-mixed. We shall stress the subtle balance between size and accuracy of the generated models. On the one hand, models need to be sufficiently detailed to accurately capture the dynamics of interest of the system. On the other hand, large models are more difficult to analyze and to calibrate. Beyond their practical benefits, these features allow us to gain some deep insights into the system under investigation. If a model that captures a particular property of the system is not more accurate than a less detailed model, then this property is probably not important to the system's dynamics.

Our calibration method based on MLE presents the advantage of being very robust. Even for systems that involve multiple difficulties such as embodiment, spatiality, and scalability, our method yields accurate predictions by lumping all relevant system's properties into the available reaction rates. However, this approach makes the analysis of these rates particularly laborious since one cannot establish an one-to-one mapping between each rate and a given system's property or parameter. This limitation, along with the strict requirement of perfect observability of the underlying system, is currently an obstacle to the applicability of the M^3 framework in some circumstances (e.g., microscale SA). However, we believe that future theoretical developments and the use of more advanced machine learning methods shall allow for relaxing both of these requirements.

Part III

Design and Optimization

10

Model-Based Design

He knew how to take what could be, and make it what is.
– Wynton Marsalis (1961)
describing Louis Armstong's musical improvisation ability
in *Ken Burn's Jazz* documentary on PBS.

Once accurate and computationally efficient models are available, one can use their predictions for designing and optimizing the underlying system. In this work, we distinguish the notions of *design* and *optimization*, even though they are generally considered as parts of the same process in other engineering disciplines. On the one hand, design is concerned with the definition of the *qualitative, structural* specification of the underlying system, e.g., deciding between different control strategies or robots' geometries. On the other hand, optimization consists in fine-tuning the *quantitative, numerical* parameters of the said specification.

In this chapter, we investigate two important aspects of the model-based design of distributed robotic systems. First, we discuss in detail the trade-offs that exist between top-down and bottom-up approaches to model-based design, with a specific emphasis on resource-constrained, miniature robots. Second, we illustrate the use of models at multiple abstraction levels for assessing the costs and benefits of deterministic and probabilistic controllers (Case Study II). In both cases, the question at hand requires deep insights into the system's dynamics, which we believe can only be obtained by a proper combination of real experiments and models at multiple abstraction levels.

10.1 Top-Down vs Bottom-Up Model-Based Design

The design and control methodology prescribed by the MLMM is essentially bottom-up; that is, we start with a feasible specification of the real system[1], and then build up a series of increasingly abstract models, carefully

[1] This specification may range from a complete implementation to realistic simulations, but, in any case, all the necessary components for building the system must be available.

G. Mermoud, *Stochastic Reactive Distributed Robotic Systems*,
Springer Tracts in Advanced Robotics 93,
DOI: 10.1007/978-3-319-02609-1_10,

validating each against those at lower abstraction levels. One important benefit of this approach is its direct anchoring to reality, allowing one to predict and optimize the performance of the real system. However, bottom-up approaches generally yield macroscopic models that are difficult to analyze mathematically (e.g., non-linear, time-delayed systems of differential or difference equations, sometimes partial). More critically, while they allow for a precise design of the microscopic behavior based on simple, robust techniques (e.g., behavior-based control), they also require a good deal of intuition for achieving the desired coordinated behavior at the macroscopic level (i.e., they are non-constructive).

In top-down design methodologies, this limitation is overcome by shaping up the robots and their controllers such that they fulfill the requirements of coordination algorithm. However, these approaches generally make quite strong assumptions (e.g., perfect localization, discrete environment, absence of noise, etc.), possibly leading to degraded performance when implemented on the target system. Furthermore, top-down approaches usually neither predict real system performance nor provide bounds for performance loss due to these invalid assumptions; complete collapse of the collective dynamics is possible, especially when dealing with stochastic systems, as it is often the case with distributed systems of miniature robots.

The goal of this section is two-fold. First, it aims to illustrate how top-down approaches can be applied to Case Study V, which was specifically designed as a benchmark of control methodologies for resource-constrained distributed robotic systems [181]. In particular, we propose a top-down solution based on graph theory to the problem presented in Section 4.5; this approach yields a *semi-centralized* control, in which robots are essentially autonomous, but exploit information broadcasted by a central planner. Second, we compare the performance of this top-down design to a bottom-up solution that relies solely on local information, and yields a fully distributed controller, which is optimized for a wide range of scenarios based on models at multiple abstraction levels.

As discussed in Section 4.5, the collaborative spot-destruction problem is a typical dynamical allocation problem that requires the presence of κ_i robots at spot i, and the robots must be re-distributed as quickly as possible after each spot destruction. The solution to this problem is two-fold: (1) one needs to determine the optimal distribution of robots $\mathbf{x}^d$ over the set of spots such that the system performance metric given by Equation 4.12 is maximized, and (2) one needs a strategy to distribute the robots according to $\mathbf{x}^d$ in a *scalable* and *efficient* manner.

10.1.1 Optimal Desired Distribution

First, we provide a concise and formal solution to the first problem, which depends only on the number of robots N_0, the number of spots N_{spots}, and their respective characteristic thresholds κ. We then describe and compare two distinct model-based strategies (*top-down* and *bottom-up*, respectively) to distribute the robots according to the desired distribution $\mathbf{x}^d$. The relevance and the performance of these strategies depend not only on the number of robots, but also on their capabilities (e.g., computational power, communication and sensing capabilities, localization, etc.) as well as the amount of information available to the central planner, if any.

Determining the desired distribution $\mathbf{x}^d$ that maximizes the system performance (Equation 4.12) is non-trivial, especially if $N_0 < \sum_{i \in \mathcal{S}} \kappa_i$ where $\mathcal{S}$ denotes the set of spots. Fortunately, one can construct the desired distribution $\mathbf{x}^d$ such that the performance of the system is optimal[2] by solving the following optimization problem:

$$\begin{aligned}
\mathbf{x} &= \arg\max_{\mathbf{x}} \; E[\text{Perf.}|\hat{\boldsymbol{\kappa}}] = \arg\max_{\mathbf{x}} \; \sum_{\kappa=1}^{N_\kappa} E[\kappa \cdot \Lambda_\kappa|\hat{\boldsymbol{\kappa}}] \\
&= \arg\max_{\mathbf{x}} \; \sum_{\kappa=1}^{N_\kappa} \kappa \cdot E[\Lambda_\kappa|\hat{\boldsymbol{\kappa}}] \qquad (10.1) \\
E[\Lambda_\kappa] &= \rho \sum_{i}^{N_{\text{spots}}} Pr\{x_i \geq \kappa_i \cap \kappa_i = \kappa|\hat{\kappa}_i\} \\
&= \rho \sum_{i}^{N_{\text{spots}}} Pr\{x_i \geq \kappa_i\} \frac{\chi(\kappa, \hat{\kappa}_i)}{\sum_{\lambda=1}^{N_\kappa} \chi(\lambda, \hat{\kappa}_i)} \qquad (10.2)
\end{aligned}$$

where, assuming that x_i is a sequence of i.i.d. binomial random variables,

$$Pr\{x_i \geq j\} = \sum_{l=\kappa}^{N_0} \binom{N_0}{l} x_i^l \, (1 - x_i)^{N_0 - l}$$

and $\hat{\kappa}_i$ is the estimate of the threshold of spot i.

Because of the use of binomial random variables, the problem is non-convex. However, one can solve this problem using non-linear programming (we used the MATLAB™ function `fmincon`), with initial conditions given by a simple heuristic that distributes the robots on the spots with large κ_i (but smaller than the total number of robots) first.

[2] Note that we assume in this subsection that the system is optimal *after* convergence, and we do not optimize for the speed of convergence.

10.1.2 Top-Down Strategy

We use a novel technique developed by Berman et al. [27] that allows one, under certain assumptions, to distribute unlabeled robots over a set of spots in an arena in a scalable and efficient manner. To achieve that, an omniscient central planner[3] needs to (i) construct a graph $\mathcal{G} = (\mathcal{S}, \xi)$ whose vertices are the spots, and whose edges are feasible paths connecting the spots, (ii) compute the optimal transition rate matrix $\mathbf{K} = (k_{ij})$ that allows for the fastest convergence, and (iii) broadcast both $\mathcal{G}$ and $\mathbf{K}$ to the robots.

Interestingly, the spot-destruction problem can be thought as an allocation problem, i.e., it requires the presence of κ_i robots at spot i. Our solution is therefore to steer the swarm of robots to a given desired distribution $\mathbf{x}^d$ using the technique described above; upon the destruction of a spot, the central planner computes new $\mathcal{G}$ and $\mathbf{K}$ and broadcasts them to the robots. However, beyond this intuitive similarity, many details are left to be worked out. In particular, the many assumptions made in [27] need to be relaxed throughout the process of implementation. For instance, we shall optimize the transition rates while retaining their feasibility on a real robotic system; also, we shall determine the desired distribution $\mathbf{x}^d$ for finite number of robots, as well as a method for constructing both $\mathcal{G}$ and $\mathbf{K}$ such that they can be broadcasted to resource-constrained robots such as the Alice robot. Hereafter, we discuss each of these assumptions by proposing either an objective criterion for their validity, or a metric of their impact on the system performance. Figure 10.1 illustrates each step of the design process visually.

Optimizing Transition Rates

The optimal transition rate matrix $\mathbf{K}$ is computed by the central planner, which solves a Semi-Definite Program (SDP) [229] that finds the *transition rates* from one spot to another that ensure the fastest convergence to the desired distribution. Moreover, under the constraint $\mathbf{K}\mathbf{x}^d = 0$, which ensures that the robots will indeed converge to the desired distribution $\mathbf{x}^d$, the transition rates can be made such that, after attaining convergence, the system makes as *few* transitions as possible (so that they are feasible in the context of a real robotic system). The objective function used to limit the transitions after convergence can take two distinct forms:

$$f(\mathbf{K}) = \sum_{(i,j)\in\xi} k_{ij}\, x_i^d \tag{10.3}$$

$$\text{or,} \quad f(\mathbf{K}) = \max_{(i,j)\in\xi} k_{ij}\, x_i^d. \tag{10.4}$$

[3] For the sake of scalability, the central planner does not know about the position of each robot. Also, in [27], the central planner assumes that there is an infinite number of robots.

Equation 10.3 represents the total number of transitions per time unit after attaining equilibrium, while Equation 10.4 represents the maximum number of transitions per time unit. To solve the SDP problem, we use `CVX`, a MATLAB™ package for specifying and solving convex programs [230].

Graph Structure

One general assumption made by Berman et al. [27] is that the underlying graph $\mathcal{G}$ is strongly connected (i.e., a directed path exists between any pair of distinct vertices). As discussed below (Section 10.1.2), our method of broadcasting the graph structure to the robots imposes that edges must be unidirectional; indeed, light gradients used for navigation must be wide enough for robots to follow them while being non-overlapping.

In this work, we construct a maximal planar graph using Delaunay triangulation (in $O(n \log n)$ time), and then an arbitrary triangle is turned to a cycle, and is chosen as the starting graph. The other spots are added one by one, keeping the graph strongly connected. Compared to other graph structures such as cycles, Delaunay graphs have maximum connectivity under our constraints and therefore result in the fastest convergence.

Broadcast and Navigation

Another fundamental assumption made by Berman et al. [27] is that the central entity can broadcast the graph structure $\mathcal{G}$ and the matrix $\mathbf{K}$ to the robots. In turn, the robots shall be able to navigate along the edges of the graph. However, resource-constrained robots such as the Alice are generally endowed with limited and unreliable sensing, communication, and navigation capabilities, thus making these assumptions unrealistic. Furthermore, computation, energy, and memory limitations impose severe restrictions on the use of advanced map-based navigation algorithms.

We propose an original solution based on augmented reality to solve this problem using a simple, nearly reactive, algorithm. Light gradients depicted on the arena by a video projector allow the central planner to locally tune the robots' behavior such that it accounts for the plan optimized at the macroscopic level.

More specifically, the central planner draws gradients between spots; intensity at the darker end of the gradient originating from spot i to spot j is related to the net rate of transition from spot i (exit rate $\sum_j k_{ij}$), and the width of the edge is proportional to $k_{ij} / \sum_l k_{il}$. The robot wanders about randomly in the spot until it encounters a gradient at its border. After getting an estimate of $\sum_j k_{ij}$, the robot draws T_{exp} from an exponential distribution with rate $\sum_j k_{ij}$ (or with mean $1/\sum_j k_{ij}$).

If the robot has already spent more than T_{exp} units of time in the spot, it makes the transition. Otherwise, the robot waits until it has spent more

than T_{exp} units of time in the spot, and then makes the next transition. This ensures that the rate of exit of robots from the spot is the closest that we can get to $\sum_j k_{ij}$, as required. Also, the probability of choosing an edge to spot j is directly proportional to the relative widths of the edges with respect to one another, which are given by the ratio $k_{ij}/\sum_l k_{il}$. If these choices are made independent of the activity of the robot in the spot, then by thinning of Poisson processes [231], we know that the transitions from spot i to spot j is dictated by a Poisson process with the parameter k_{ij}, as required.

The range of intensities that are available to represent the gradients is limited by the operating range of the light sensor as well as the intensity of the projector. Hence, instead of using constraint 10.3 or 10.4, a different objective function needs to be *minimized*:

$$f(\mathbf{K}) = \max \sum_j k_{ij}. \tag{10.5}$$

After obtaining the transition matrix $\mathbf{K}$, we can scale it such that the maximum value of $\sum_j k_{ij}$ can be represented by the available intensity range. We denote the scaled transition matrix $\mathbf{K}_{opt}$.

Robot Controller

Finally, one needs to translate the solution described in the previous sections into an actual robot controller. There is no automated way of constructing such a controller; however, in a top-down approach, the designer merely encodes the various requirements prescribed at higher abstraction levels in the robot controller while accounting for the technological limitations of the robotic platform. In our case, robots are programmed with a simple behavior-based controller composed of five states: *search*, *climb*, *in spot*, *probe*, and *u-turn* (Figure 10.1). Other auxiliary states are used, but they are not mentioned here for the sake of clarity.

In the state *search*, the robot performs a simple random walk, i.e., it alternates between forward motion and tumbling in a random direction. If the robot detects a gradient, it transitions to the state *climb*; in this state, the robot moves up the gradient using a simple reactive scheme similar to Braitenberg vehicles. When no change in intensity is detected, it means either that the robot is in a spot if the intensity is high (transition to the state *in spot*), or that the robot is lost if the intensity is low (transition to the state *search*). In the state *in spot*, the robot moves forward; upon detecting a drop in intensity (i.e., the robot reaches the border of the spot), it transitions to the state *probe* and stops. In the state *probe*, it samples its light sensors for a certain amount of time. Depending on the average of these measurements, the robot may transition to either of two states: (i) if it encountered an outgoing gradient and it has already spent more than T_{exp} units of time in

the spot[4], then it transitions to the state *climb*; (ii) if it encountered an incoming gradient or it has spent less than T_{exp} units of time in the spot, then it transitions to the state *u-turn* in order to remain within the spot.

10.1.3 Bottom-Up Strategy

In the bottom-up strategy, we start with a simple behavior-based controller (Figure 10.1), which we optimize using the suite of models constructed in Section 6.3.3. One of the critical feature of the bottom-up approach is that it deals from the beginning with the intrinsic limitations of the robotic nodes rather than setting requirements that are not necessarily feasible. For instance, in our particular case study, we do not assume that the robots receive broadcasted navigational hints from a central planner, thereby leading to a fully distributed, on-board, more robust control solution. The robots exploit local information (in our particular case, the intensity of the spots) to tune their behavior in a way that is optimal for a wide range of scenarios (i.e., for various combinations of spot thresholds κ_i for $i = 1, \ldots, N_{\text{spots}}$). More specifically, no gradient is projected on the arena, and the robots move from spot to spot using a random walk instead of a directed gradient-based movement as that used in the top-down approach. Upon detecting a change in light intensity, a robot enters into the spot, and starts exploring it. Each time the robot reaches the spot's border, it will leave with probability $p^{\text{leave}}(\hat{\kappa})$, which depends on its estimate $\hat{\kappa}$ of the threshold of the spot.

One important difficulty with this approach is that we need to define the leaving probabilities $p^{opt}_{leave}(\hat{\kappa})$ such that the system performs the best according to Equation 4.11. We solve this problem in Section 11.2.

10.1.4 Results and Discussion

One of the main findings of our experimental study is that the top-down approach does not perform significantly better (or worse) than the bottom-up approach with the leaving probabilities optimized in Section 11.2 (i.e., $p^{\text{leave}}_{\text{opt}} = \left[p^{\text{leave}}(\hat{\kappa} = 1), \ldots, p^{\text{leave}}(\hat{\kappa} = N_\kappa = 3)\right] = \left[0.335, 0.0002, 0.005\right]$). However, one also needs to check that these optimized controllers do not actually perform the same as any other non-trivial controller. We rule out this hypothesis by performing a third type of experiments that use a so-called *baseline* controller, which is essentially the bottom-up controller with all leaving probabilities set to zero. A non-parametric statistical test shows that both the top-down and the bottom-up controllers perform *significantly* better (with

[4] If the robot encounters an outgoing gradient for the first time since entering the spot, it first draws T_{exp} from an exponential distribution whose mean is inversely proportional to the average intensity of the gradient. Furthermore, because of the limited computational power available on our robotic platform, we use a lookup table to generate these random numbers.

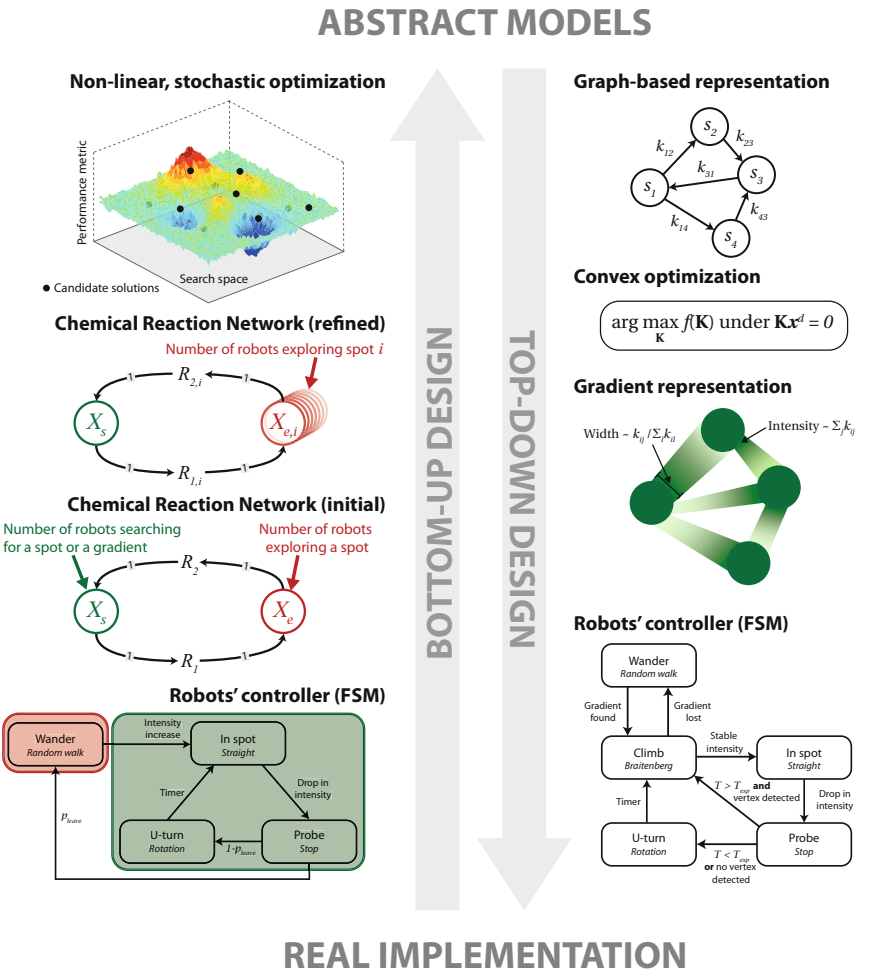
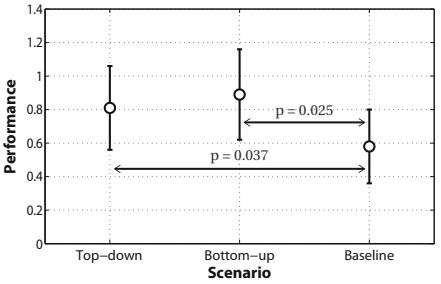
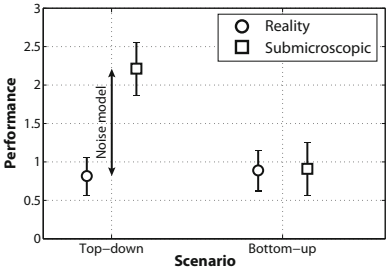


Fig. 10.1 Schematic comparison of bottom-up and top-down approaches to model-based design. Top-down strategies start with an abstract model (e.g., a graph) of the system, which is then iteratively refined into a robot controller. Instead, bottom-up strategies assume the prior existence of the robots' controller, which is then abstracted to a high-level, macroscopic CRN.



(a) Experimental results



(b) Comparison reality vs submicroscopic

Fig. 10.2 (a) Graphical illustration of the experimental results reported in Table 10.1, and (b) their comparison with a submicroscopic model implemented in Webots (50 runs). We hypothesize that the discrepancy observed in the prediction of the top-down performance is due to the naive noise model used in the submicroscopic model.

Table 10.1 Summarized results of three experiments (top-down, bottom-up, and baseline, 10 runs of 20 minutes each) using 6 real Alice robots and 4 spots (with $\kappa \in [1, 2, 3]$). The system performance is computed using Equation 4.12. The distance between the actual distribution $\mathbf{x}(t)$ and the desired distribution $\mathbf{x}^d$ of robots is given by Equation C.1, and averaged over all sampled data points.

Metric	**Moment**	**Case studies**		
		Top-down	*Bottom-up*	*Baseline*
Performance	*mean*	0.81	0.89	0.58
	median	0.85	0.95	0.57
	std dev	0.25	0.27	0.22
Distance $\|\mathbf{x}(t) - \mathbf{x}^d\|^2$	*mean*	0.36	0.37	0.25
	median	0.39	0.36	0.25
	std dev	0.11	0.06	0.02

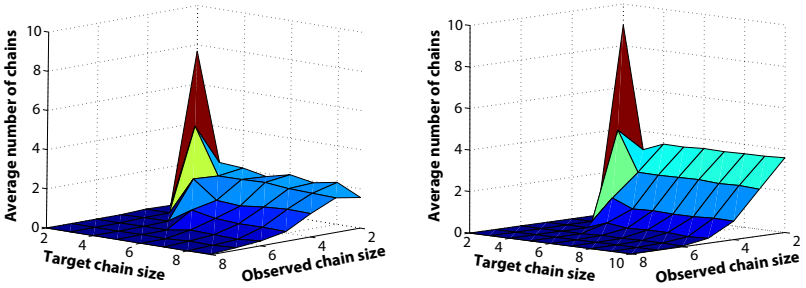
respective p-values of 0.037 and 0.025 using Mann-Whitney test) than the baseline controller. Table 10.1 summarizes the results of these experiments. In particular, we shall outline that the large variability of both the system performance and the average distance between the actual distribution and the desired distribution of robots does not allow one to conclusively determine which of the optimized controllers perform best in the context of study. Our results show that a sophisticated approach such as the one proposed by Berman et al. [27] does not bring any significant performance increase in the context of our case study, in spite of the supplementary infrastructure it uses.

There are multiple reasons why the extra information provided to the robots in the top-down approach does not significantly improve the system's

performance. First, when the density of spots is high enough, a simple search behavior such as the one used in the bottom-up approach does not perform significantly worse than gradient ascent, because robots can compensate for their lack of information with higher velocities. Second, the interaction between robots in a spot dramatically affects the transition rates; for instance, when two robots collide and avoid each other near the border of a spot, one may get lost during the maneuver. Also, when a spot is explored by many robots, the time they spend avoiding each other becomes non-negligible, thus modifying the effective exit rate. Last, the assumptions underlying our formulation of the expected system performance described earlier are not necessarily met in reality. In particular, dispatching more than κ_i robots to a given spot i may actually be beneficial because, again, it makes the system more robust. This fact also explains the discrepancy between performance and distance to the desired distribution observed in our experimental results. Finally, and most importantly, the presence of noise in sensor measurements and the heterogeneity of the projected picture dramatically affect the effective transition rates.

As a result, robust approaches are favored over complex strategies. In our bottom-up approach, leaving probabilities are not specifically optimized for a given configuration, but rather for a wide range of scenarios; therefore, the performance loss observed during the process of implementation is less than in the case of the system developed using a top-down scheme. To verify this assertion, we implemented a submicroscopic model using Webots, and performed 50 simulation runs. On the one hand, the top-down approach performed significantly better in simulation (mean = 2.21, median = 2.21, std dev = 0.35) than in reality (mean = 0.81, median = 0.85, std dev = 0.25, see Table 10.1). However, the performance of the bottom-up approach in simulation remained stable (mean = 0.91, median = 0.95, std dev = 0.35). The main difference between these simulations and reality lies in the modeling of noise; simulations assume that the noise on the light sensors is Gaussian, but the heterogeneity of the projected picture actually yields time-dependent and space-dependent multimodal noise distributions. Also, most of the technological limitations of the platform (e.g., memory, resolution of analog/digital conversion, limited floating point handling) are not captured in the simulations.

These findings confirm the better robustness of bottom-up approaches to noise and unpredictability. Nevertheless, top-down approaches offer generally a much better theoretical tractability at the macroscopic level; in particular, they allow for the use of efficient optimization methods such as linear and nonlinear convex optimization. Also, top-down approaches have the potential to perform more like deterministic strategies, especially when uncertainty and noise are low.



(a) Submicroscopic model (b) Macro-stochastic model

Fig. 10.3 Purely deterministic control does not yield desirable results when attempting to form long chains. As we increase the target chain size from two to eight with $N_0 = 19$ robots, both (a) submicroscopic simulations (averaged over 100 runs) and (b) stochastic simulations of the CRN (averaged over 500 runs) show that a lack of exploration causes the system to quickly converge to the trivial distribution (Figure 10.4), in which short chains are largely favored.

10.2 Deterministic vs Probabilistic Controllers

This section aims to illustrate concretely how modeling may enlighten the designer when it comes to choose between different design options that are difficult to evaluate experimentally. In our particular case, we consider the choice offered by Case Study II between a deterministic controller that relies on local communication and a probabilistic controller with no such requirement (see Section 4.2 for further details about both approaches).

We study the deterministic controller using submicroscopic and macro-stochastic models with varying maximal chain sizes from two to eight with $N_0 = 19$ robots (Figure 10.3). These results demonstrate that the controller successfully limits the size of the chain, but, as the target chain size increases, the distribution of chain size approaches the one yielded by the baseline aggregation-only controller (see Section 4.2), which is depicted in Figure 10.4 for $N_0 = 19$ robots .

Similarly to the deterministic controller, we model the probabilistic controller at the submicroscopic and macro-stochastic levels. Using these models, we explore the entire parameter space $(p_{leave,1}, p_{leave,2}) \in [10^{-9}, 10^{-1}]^2$ (19 Alice robots and 30 minutes of simulated time). At the submicroscopic level, the parameter space is discretized into 9×9 logarithmic grid, each point being averaged over 100 runs. The macro-stochastic model allows us to achieve a finer discretization of 16×32 averaged over 500 runs.

As discussed previously, one might think *a priori* that the best strategy to achieve long chains is to choose small leaving probabilities. Perhaps

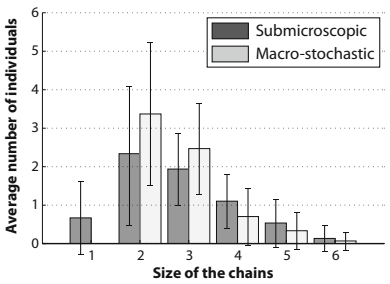


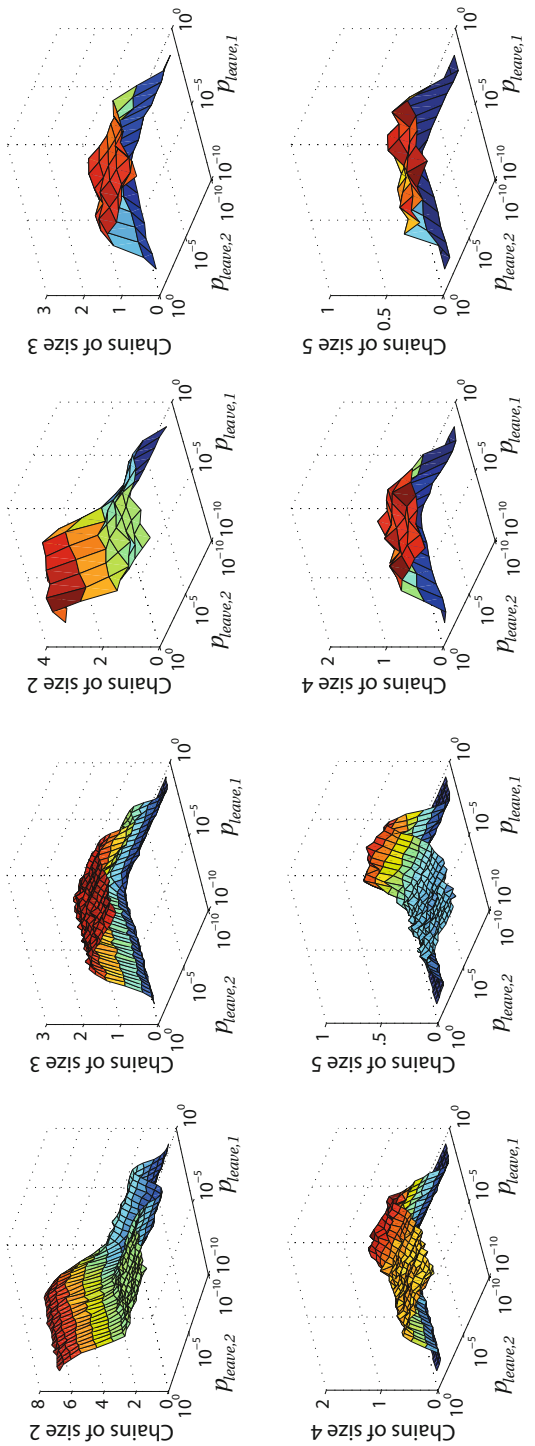
Fig. 10.4 Comparison of the trivial distribution for $N_0 = 19$ robots predicted by submicroscopic and macro-deterministic models. The average number of single robots predicted by the macro-deterministic model is zero. It is clear that setting leaving probabilities to zero does not favor the formation of long chains due to the lack of exploration.

not intuitively, this strategy proves to be suboptimal because the system converges to an undesirable configuration that tends towards the trivial distribution as the leaving probabilities tend to zero. However, the results of these searches, depicted in Figure 10.5, confirm the existence of an optimal region for $p_{leave,1} \in [10^{-4}, 10^{-3}]$ and $p_{leave,2} < 10^{-4}$ where the ratio between the number of pairs and the number of longer chains decreases because exploration (disaggregation) and exploitation (aggregation) are well balanced, thus leading to a non-trivial chain size distribution.

The long-term evolution of the distribution of chain size for leaving probabilities that lie within the optimal region ($p_{leave,1} = 10^{-4}$ and $p_{leave,2} = 10^{-9}$) demonstrates clearly that one can achieve a basic control over the distribution of chain size by setting appropriate leaving probabilities only. More precisely, it is possible to skew the steady state distribution towards longer chains. Higher, non-optimal leaving probabilities enable faster convergence, but they do not favor the formation of long chains.

10.2.1 Results and Discussion

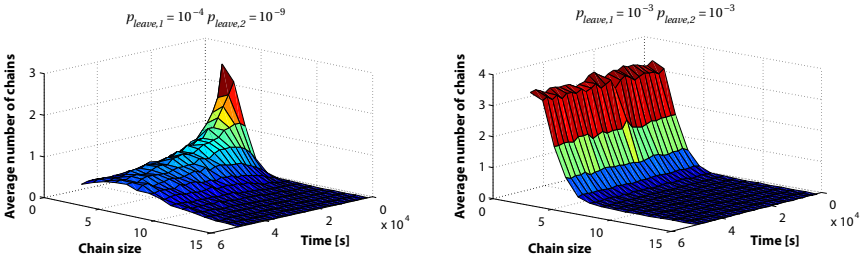
Our results make clear that one cannot favor the formation of chains of a specific size by limiting the size of the formed chains (deterministic controller) or by setting appropriate leaving probabilities (probabilistic controller). Indeed, the landscapes depicted in Figure 10.3 and Figure 10.5 do not exhibit a systematic variation in the ratio of different chain size populations that would enable a precise selection of the most favored type of chain. The deterministic controller, by using communication among the robots, can achieve a proper non-linear feedback that depends on the size of the chain, thereby preventing the formation of chains longer than a certain target size, whereas the probabilistic controller is purely linear, i.e., the robots have the exact same behavior regardless of the size of the chain. However, the deterministic controller is unable to achieve long chains because of its intrinsic lack of exploration whereas the probabilistic controller enables, with proper leaving probabilities, a balance of exploration and exploitation. One could however envision combining both controllers into a single, hybrid controller. Indeed,



(a) Macro-stochastic model

(b) Submicroscopic model

Fig. 10.5 Systematic exploration of the parameter space spanned by $p_{leave,1}$ and $p_{leave,2}$ (logarithmic scale) for $t = 30$ min. (a) The existence of an optimal region for $p_{leave,1} \in [10^{-4}, 10^{-3}]$ and $p_{leave,2} < 10^{-4}$ where exploration (disaggregation) and exploitation (aggregation) are well balanced is confirmed by the macro-stochastic model. (b) The lower resolution and the larger variance of the submicroscopic model's predictions make this optimal region fade out, but the general landscape is qualitatively the same.



(a) Small probabilities (b) Large probabilities

Fig. 10.6 Long-term prediction (up to 14 hours) of the chain size distribution yielded by the macro-stochastic model for two different parameter sets. (a) For small probabilities ($p_{leave,1} = 10^{-4}$ and $p_{leave,2} = 10^{-9}$), it is clear that the convergence is slow and the time required to reach the steady state is at least one order of magnitude higher than the duration of our experiments. (a) For larger probabilities ($p_{leave,1} = 10^{-3}$ and $p_{leave,2} = 10^{-3}$), the convergence is much faster and on the same time scale as the duration of our experiments (about 30 minutes).

by optimizing the leaving probabilities for the formation of arbitrarily long chains while limiting explicitly their size to a given upper bound N_{max}, one would end up favoring the formation of chains of size N_{max}. However, this hybrid approach would be less scalable than the purely probabilistic controller presented here. Indeed, implementing a controller based on local communication is already challenging at this scale, as depending on lighting conditions and robot density, the robots experience extreme packet corruption and loss. In our algorithm, this significantly increases the amount of time a chain's size takes to propagate to its outer nodes, possibly allowing undesirable aggregations to occur in the interim. We believe this problem will become overwhelming as target systems become smaller.

Another important insight provided by our results is that the SA process takes a long amount of time to stabilize, in particular when leaving probabilities are low, as depicted by Figure 10.6. These results demonstrate clearly that, for small leaving probabilities, the time scale of the SA process is much larger than that of the experiments presented in this section. As a result, the influence of the trivial distribution is important in the beginning, as it is essentially the transient, short-term distribution of any experiment. This apparent flaw of the approach is actually an opportunity: by choosing a proper duration of the process, one can actually control how close to the trivial

distribution the system will be. Of course, this approach is possible only if the leaving probabilities depend on the time (in a basic fashion, since they will be set to zero after a time t_f)[5].

Summary and Conclusion

In this chapter, we investigate two orthogonal approaches (top-down vs bottom-up) to the design of distributed controllers. The main claim of this chapter is that conventional top-down design of multi-robot systems is generally not amenable to efficient implementations when dealing with resource-constrained robots; faithful and computationally efficient models built incrementally from the bottom up are an essential ingredient to the design and the control of such systems. Ultimately, we believe that these two approaches should be combined into a more powerful model-based control design methodology that has the potential to achieve higher, more tunable coordination at the macroscopic level while incorporating all a priori known technological limitations at the microscopic level. Of course, our study does not expose all the features that make either approach more or less suitable to a given system; this endeavor is by itself a whole body of future research.

Another key contribution of this chapter is the comparison of (i) a deterministic, non-linear controller where robots communicate with each other in order to determine the size of the chain and adapt their behavior accordingly, and (ii) a probabilistic, linear controller where the distribution of chain size is controlled by the leaving probabilities of the robots. Using a combination of submicroscopic and macroscopic models, we perform systematic searches of the parameter space, which shed light on (i) the inability of the deterministic controller to achieve long chains because of its lack of exploration, and (ii) the existence of optimal regions within the parameter space of the probabilistic controller where exploration and exploitation are well balanced, thus favoring the formation of larger chains. However, the linearity of the probabilistic controller prevents any positive or negative feedback that would favor specifically a given target chain size. Based on the insights gained from the models, we suggest a solution based on a hybrid controller using optimized leaving probabilities for the formation of arbitrarily long chains while explicitly limiting their size to a given target size N_{max}.

[5] Note that this requirement does not jeopardize the scalability of our approach since one could imagine to use an external observer to broadcast a predetermined message or to modify an environmental parameter of the system (e.g., illumination, temperature, pH) at t_f in order to signal the termination of the SA process.

11

Model-Based Optimization

To the optimist, the glass is half full.
To the pessimist, the glass is half empty.
To the engineer, the glass is twice as big as it needs to be.
– Unknown

While intuition and educated guesses can be sufficient for determining the qualitative, structural specification of a distributed robotic system, modeling is often necessary when it comes to finding the optimal parameters of the said specification. However, most classical optimization schemes are unable to deal with the combined effects of non-convexity, discontinuity, and stochasticity found in models of SMPs at low abstraction level. Even macro-deterministic models generated in a bottom-up fashion are in principle non-convex, and may exhibit numerous local minima that are difficult to deal with. In such cases, one needs to recourse either to optimization meta-heuristics such as Genetic Algorithm (GA) or Particle Swarm Optimization (PSO) (and, more specifically, their noise-resistant variants [212]) or to systematic searches of the parameter space. Both approaches require underlying models that exhibit an excellent balance between computation cost and accuracy, as they involve numerous evaluations of candidate solutions. Optimization meta-heuristics can deal with parameter spaces of high dimensionality, but they are often used as black box methods. Instead, systematic searches become difficult to use with more than three parameters, but they offer more insights into the global, qualitative behavior of the system, which is very important from a design perspective.

11.1 Example from Case Study III

In this section, we optimize three parameters of the 2-DOF variant of Case Study III. Namely, we consider two design parameters, i.e., the maximal bond energy E_{bond} and the misalignment tolerance σ_θ, and one control parameter, i.e., the agitation of the system ν_s. In reality, the design parameters would be encoded into the physico-chemical properties of the self-assembling building

G. Mermoud, *Stochastic Reactive Distributed Robotic Systems*,
Springer Tracts in Advanced Robotics 93,
DOI: 10.1007/978-3-319-02609-1_11,

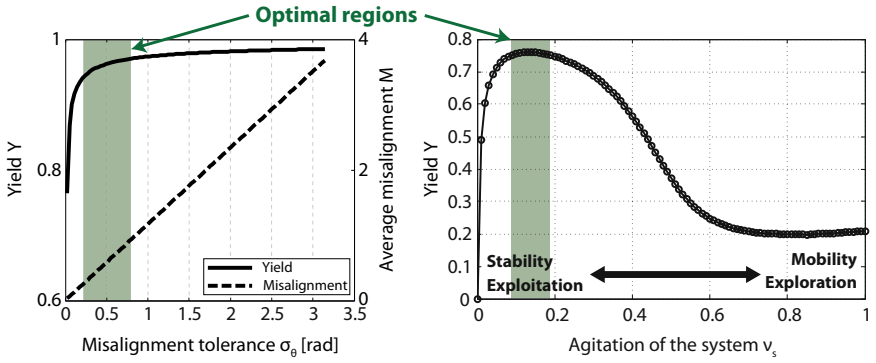


Fig. 11.1 Illustration of different design and control trade-offs in Case Study III. *Left:* Influence of the misalignment tolerance σ_θ (in radians) on both the yield Y and the average misalignment of the aggregates M (in radians) after 25000 timesteps. Note the linear relationship between the average misalignment and the misalignment tolerance. *Right:* The yield Y is a non-linear function of the agitation of the system ν_s, and it has an optimum around $\nu_s^\star = 0.15$. Generally speaking, high yields (higher than 70%) are achieved by moderate agitations (between 0.05 and 0.3), which exhibit a good balance between exploitation and exploration.

blocks such as their shape, their material, or their surface chemistry, which would in turn characterize the interaction forces mediating the SA process. For instance, capillary forces (occurring at liquid-fluid interfaces) tend to be longer range and much stronger than purely hydrophobic interaction (occurring in the bulk of the liquid), thereby leading to a higher misalignment tolerance as well as a larger maximal bond energy.

As stated before, the yield is not the unique performance metrics of the system, and one may want to optimize also the average misalignment M. In this regard, the misalignment tolerance σ_θ of the building blocks plays a key role. Indeed, a large misalignment tolerance both increases the yield and worsen the average alignment of the aggregates (Figure 11.1, left). Note that the relationship between σ_θ and M is linear whereas the one between σ_θ and Y is not. Therefore, one can find an optimal trade-off between high yields and moderate misalignment by choosing a low misalignement tolerance (around 0.5 rad). This type of information is crucial when designing a new system, and it would be time consuming to achieve such systematic exploration of the parameter space using realistic simulations and prohibitively difficult using real hardware.

Our macroscopic models can also be used for optimizing the control of the system, yet in an *offline* fashion. In particular, we investigate how the yield of the system varies as a function of the agitation ν_s. To this end, we systematically vary the parameter ν_s in the interval $[0, 1]$ with a granularity of 0.01 (Figure 11.1, right). Interestingly, the relation between the yield and the agitation is strongly non-linear and exhibits an optimum around $\nu_s^\star = 0.15$.

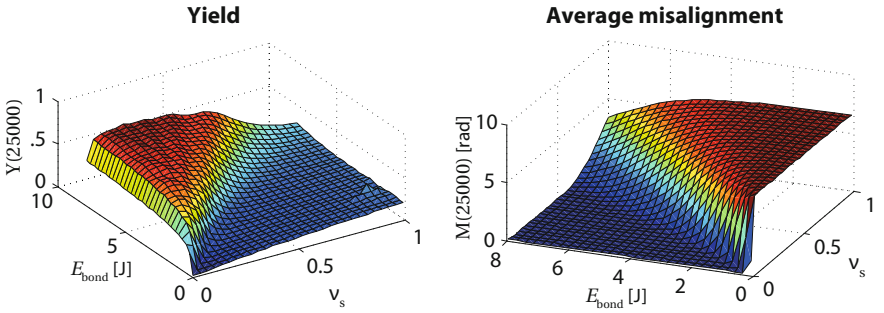


Fig. 11.2 Yield Y and the average misalignment M (in radians) at $k = 25000$ for two systematically varying parameters: the agitation of the system $\nu_s \in [0, 1]$, and the maximal bond energy $E_{\text{bond}} \in [0, 8]$ (macro-deterministic model with $N_0 = 1000$, $K = 1500$). Both high yields and small misalignments are achieved with moderate agitation and high bond energies.

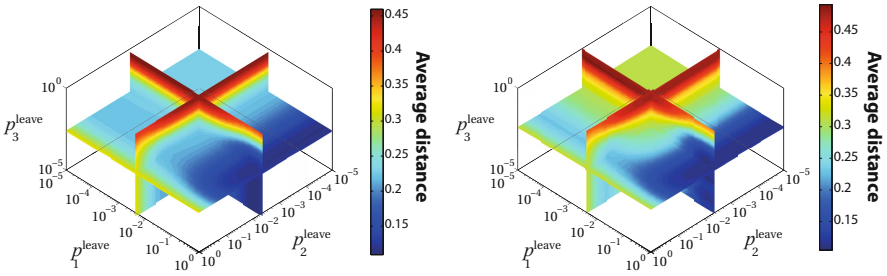
We expect the same type of complex behavior to appear in real systems since the role of agitation is always two-fold. On one hand, it favors aggregation by increasing the average velocity of the particles (and therefore the number of collisions). On the other hand, it increases the shear experienced by the aggregates, thereby decreasing their stability. Therefore, one key design question arises: In terms of yield and misalignment of the formed aggregates, is a system with strong agitation and stable bonds preferable to a system with low agitation and unstable bonds? Figure 11.2 shows how each performance metric is influenced by these parameters. As expected, highest yields are achieved within the zone of moderate agitation and high bond energies; this zone also corresponds to the lowest average misalignments[1]. These results emphasize the crucial role of agitation for optimizing, and more generally, controlling SA processes.

11.2 Example from Case Study V

This section discusses the optimization of the parameters of the bottom-up controller described in Section 10.1.3. We assume that the assumptions made in Section 10.1.1 hold, and therefore we aim to find the leaving probabilities $p_{\text{opt}}^{\text{leave}}(\hat{\kappa})$ for $\hat{\kappa} = 1, \ldots, N_\kappa$ that minimize the *average* distance between the desired distribution $\mathbf{x}^d$ and the distribution $\mathbf{x}(t)$ predicted by macroscopic models.

More formally, the distance function to be minimized is the square of the l^2-norm (Equation C.1) between the desired distribution $\mathbf{x}^d$ and the actual distribution $\mathbf{x}(t)$ of robots at time t yielded by the set of leaving

[1] Admittedly, these two metrics depend implicitly on each other since the stability of the aggregates (and therefore the yield) is a function their misalignment.



(a) Macro-deterministic model (b) Macro-stochastic model

Fig. 11.3 Systematic exploration of the three-dimensional parameter space spanned by the leaving probabilities $p^{\text{leave}}(\hat{\kappa} = 1), \ldots, p^{\text{leave}}(\hat{\kappa} = N_\kappa = 3)$ using (a) a macro-deterministic model and (b) a macro-stochastic model (averaged over 10000 runs). The color indicates the distance (square of the l^2-norm) between the desired distribution $\mathbf{x}^d$ and the actual distribution $\mathbf{x}(t)$ at $t = 100$ s averaged over a test set of 100 randomly generated scenarios. Each graph is generated by linearly interpolating a set of $20 \times 20 \times 20$ data points logarithmically distributed in the interval $[10^{-5}, 1]$ along each dimension.

probabilities $\mathbf{p}^{\text{leave}}$ at $t = 100$ s. Note that we do not optimize the system for fastest convergence; rather, we target an optimal, though transient, performance at $t = 100$ s, which is compatible with the dynamics observed in the real system. The optimization is carried out offline on a test set of 100 randomly generated scenarios.

A coarse exploration of the three-dimensional parameter space spanned by the leaving probabilities (Figure 11.3) reveals that the distance function is relatively well-behaved, and conventional non-linear programming algorithms are sufficient for solving this problem. Also, we observe that both models agree relatively well with each other; we can therefore use the macro-deterministic model for the optimization, thereby reducing the computational effort by two orders of magnitude[2].

Summary and Conclusion

This chapter introduces two examples of model-based optimization of SMPs. We show in particular how computationally inexpensive models relying on the macro-deterministic assumption allow for an efficient optimization of design and control parameters. Not only standard optimization algorithms require deterministic, continuous objective functions, but even noise-resistant

[2] Here we assume that the macro-stochastic model requires in the order of 10000 runs to yield a statistically significant prediction.

optimization meta-heuristics such as GA or PSO benefit from highly abstracted models, since they involve numerous evaluations of candidate solutions.

This chapter also outlines the importance of systematic searches in the context of SMPs. While optimization algorithms are often used as black box methods, systematic searches allow the designer to get a qualitative picture of the global behavior of the system, which is a crucial piece of information in order to make reasoned design choices. In particular, investigating the various trade-offs exhibited by the system (e.g., exploration vs exploitation, performance vs robustness) is of utmost importance in real applications, where the costs and benefits of each design choice need to be very carefully examined.

12

Model-Based Real-Time Control

The first rule [...] is that automation applied to an efficient operation will magnify the efficiency. The second is that automation applied to an inefficient operation will magnify the inefficiency.
— Bill Gates (1955)

Automated modeling paves the way for more advanced uses of models built in a bottom-up fashion. In this chapter, we describe an extension of the M^3 framework (Chapter 9) that leverages the concepts of CRN and Markov Decision Process (MDP) to achieve the controlled formation of target assemblies of Lily modules (Case Study IV). Figure 12.1 depicts the global structure of the control framework. The system is monitored by an overhead camera and analyzed using SwisTrack (Section 3.2.3). The resulting trajectories are then used by the M^3 framework to build the CMM, and the equivalent CRN, in real time. Finally, the optimal mode of agitation is determined using the optimization scheme described in Section 12.1, and transmitted to the pumps at regular time intervals. The control loop is closed by incorporating the state changes resulting from this choice in the model.

12.1 Optimization

As stated in Section 4.4, we can assume that the probability of a given structure to be formed can be tuned by adjusting relevant parameters of the system, including those which can be controlled in real time such as the agitation of the system. More formally, the research question that we address in this chapter is the following: given a stochastic distributed system with a finite set of agitation modes $\mathcal{M} = \{m_0, \ldots, m_n\}$, what is the mode m_i to be selected at time t such that the time to form a given target structure T is minimal? We show how this problem is equivalent to another well-known problem, that is, the solving of MDPs. Indeed, forming the structure T is equivalent to attaining a target population $\mathbf{x}_T = (x_{T,1}, \ldots, x_{T,M})$ such that

$$x_{T,i} = \begin{cases} 1 & \text{if } S_i = T, \\ 0 & \text{otherwise.} \end{cases} \tag{12.1}$$

G. Mermoud, *Stochastic Reactive Distributed Robotic Systems*,
Springer Tracts in Advanced Robotics 93,
DOI: 10.1007/978-3-319-02609-1_12,

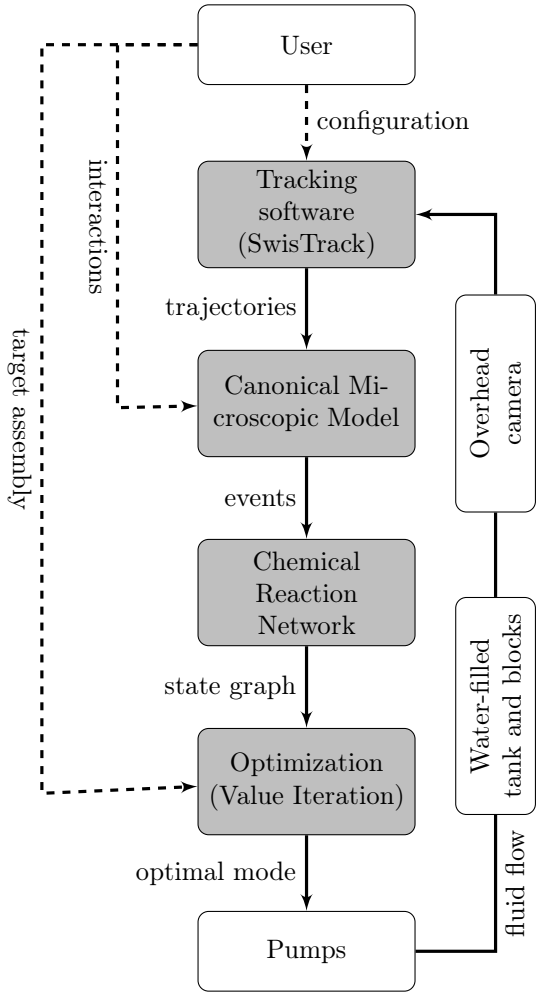


Fig. 12.1 Overview of the M^3 framework as deployed in this study, and the different types of information flowing between its constitutive modules. Gray-shaded nodes are computational entities whereas other nodes are physical entities. Dashed arrows denote flows that are not automated, but need to be performed only once prior to the experiment. Note that the closed-loop control is completely automated.

Therefore, our problem consists in determining the mode $m_s \in \mathcal{M}$ to be selected at time t given a current population $\mathbf{x}(t)$ such that the expected time to reach $\mathbf{x}_T$ is minimized.

For each mode m, we have an estimate of the propensity function $a_R^{(m)}(\mathbf{x})$ for each reaction $R \in \mathcal{R}$. Denote $k_{ij}^{(m)} = a_R^{(m)}(\mathbf{x}_i)$ the rate of the reaction R, whose associated population change is $\boldsymbol{\nu}_R = \mathbf{x}_j - \mathbf{x}_i$, if mode m is selected, and define for any $m \in \mathcal{M}$:

$$\lambda_i^{(m)} = \sum_j k_{ij}^{(m)}, \qquad p_{ij}^{(m)} = \frac{k_{ij}^{(m)}}{\lambda_i^{(m)}}. \tag{12.2}$$

Note that each state has only one optimal choice for the mode which minimizes the expected time it takes to reach the target population $\mathbf{x}_T$ and this choice is independent of the past states or how much time has been spent in the present state.

Denote by T_{ij} the expected time it takes the system to attain the population $\mathbf{x}_j$ for the first time if it starts with population $\mathbf{x}_i$ and makes optimal choice for the mode at each subsequent state. Hence, T_{ij} is the optimal *first-passage time* from population $\mathbf{x}_i$ to $\mathbf{x}_j$. We consider the target population $\mathbf{x}_T$ to be an absorbing state, which is reasonable if the experiment stops as soon as the desired state is attained.

Now for T_{ij} to be optimal, it is easy to show that they must satisfy:

$$T_{it} = \min_m \left\{ \sum_{j \neq i,t} p_{ij}^{(m)} \cdot T_{jt} + \frac{1}{\lambda_i^{(m)}} \right\}. \tag{12.3}$$

This equation reiterates the Markov property of the system, that is, the expected time to reach population $\mathbf{x}_T$ is the sum of expected time to reach the state via any of its neighbor (except $\mathbf{x}_T$ itself) and the expected time to exit the present state. $N-1$ such equations can be written for different T_{it} with $i \neq t$.

Equation 12.3 is a Bellman equation corresponding to our MDP [232], and can be solved to obtain the expected times and the optimal modes for each population state. We used the Policy Iteration method to solve the equations in our case. The optimization is performed upon each aggregation or disaggregation event observed in the system, and every 10 seconds otherwise. Previous solutions are kept in memory, and used for initializing the subsequent iterations to speed up the optimization process.

12.2 Methods

To demonstrate the effectiveness of our automatic model building framework and the relevance of our optimization algorithm, we performed four distinct experiments using the assembly E depicted in Figure 4.6 as target structure and with different control algorithms: (I) mode m_0 only, (II) mode m_1 only, (III) randomized control, where the two modes alternate randomly with an average switching period of 15 s, and (IV) optimized control, in which the optimizer selects the most appropriate mode of agitation as a function of the current state of the system and of the current state of the model. The performance of the system is given by the time of the first-passage time of the assembly E, and bounded by the maximal duration of the run.

Each experiment consists of a series of 40 runs of 30 minutes each. Each run starts with all blocks being isolated and at random locations. In experiment IV, the optimization relies on an initial model based on the observations made during two series (one per mode) of 10 runs of 5 minutes each.

However, as explained earlier, the model is constantly enhanced, both qualitatively (e.g., if a new type of aggregate is discovered) and quantitatively (i.e., the reaction rates are adjusted) as the experiment progresses.

The underlying models are constructed based on a single type interaction, which is active when two blocks are both close to each other and appropriately aligned. As a result, several assemblies that are actually distinct from each other cannot be discriminated by the model, as illustrated by Figure 4.6.

The choice of E as target structure was made because it can be univocally mapped to a unique species of the CRN, and it can be formed out of both C_1 and C_2. Indeed, the assembly D can also be univocally mapped to a unique species of the CRN, but cannot be formed out of C_2. As a result, the optimizer cannot effectively decide which mode of agitation should be applied when a trimer (i.e., C_1 or C_2) is present since these two assemblies are undistinguishable from a topological perspective. Note however that this is by no means an intrinsic limitation of our methodology, but rather a consequence of the simplicity of the underlying model.

12.3 Results

First, our results support the intuitive argument that SA, as any self-organized process, requires a subtle interplay between exploitation and exploration—as expressed by the low-agitation m_0 and the high-agitation m_1, respectively. Indeed, both experiments I and II exhibit poor performance even as compared to the naive strategy that alternates between the two modes of agitation randomly. More importantly, our results show that one can significantly improve the performance of the system by optimizing the mode of agitation as a function of the system's state. Indeed, Figure 12.2 exhibits a 40% and 66% decrease of the average and median first-passage time, respectively, under optimized control. The mean/median first-passage time of the optimized experiment (IV) is 524/205 seconds versus 930/612 seconds for the randomized experiment (III). A Mann-Whitney test rejects the null hypothesis that these two distributions of first-passage times are from the same distribution with equal medians with a p-value of $5.8 \cdot 10^{-3}$.

Figure 12.3 shows that the strategy adopted by the optimizer is quite intuitive; the mode m_0 (low agitation) is active as long as assemblies that may lead to E (i.e., assemblies A, B, C_1, and C_2) are present, and switches to the mode m_1 (strong agitation) as soon as some incorrect tetramer is formed. However, the optimization also exhibits some interesting and less intuitive behaviors. First, when only single blocks are present in the system, it sets the mode m_1 so as to favor mutual collisions. Upon the formation of a dimer B, the system may switch to mode m_0 in order to preserve it; however, while most reactions have clearly different rates for m_0 and m_1 (typically, one order of magnitude or more), the reaction $\mathsf{A} + \mathsf{B} \to \mathsf{C}_x$ exhibits relatively similar rates in either mode, thereby allowing for a dynamic switching

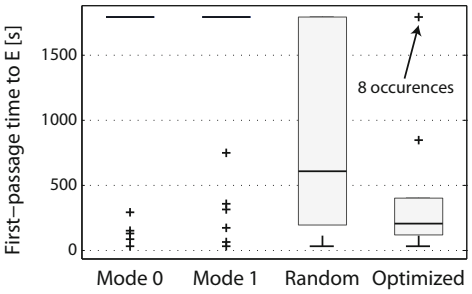


Fig. 12.2 Box plot of the first-passage time to the target structure E obtained over 40 runs of 30 minutes each for experiments I to IV. On each box, the central mark is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually.

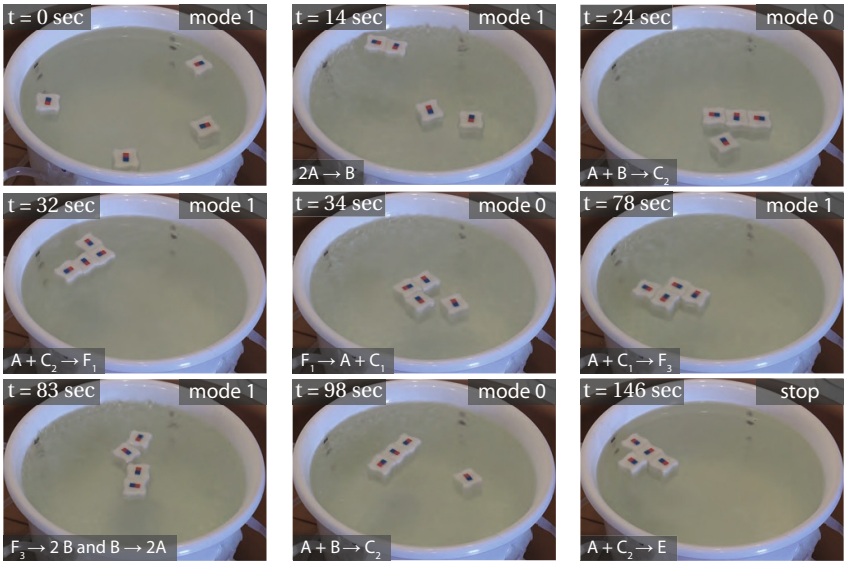


Fig. 12.3 Assembly sequence during a run of experiment IV (optimized control, see Section 12.3). The snapshots show the state of the system immediately after a reaction event. The reaction that fired is shown in the bottom left corner and the current time in the top left corner. The mode of agitation chosen by the controller is shown in the top right corner.

between two behaviors, as a function of the time spent in each. For instance, the optimizer may select mode m_0 in order to conserve the formed dimer, but as the process progresses, the rate of the reaction leading to the trimer formation in mode m_0 decreases, until it becomes smaller than the rate associated to mode m_1, thereby leading to the selection of the latter. This type of adaptive behavior is a *built-in* feature of our automated modeling approach, which is usually obtained using *ad hoc* learning strategies (e.g., reinforcement learning) elsewhere.

Summary and Conclusion

In this chapter, we demonstrate the practical relevance of the M^3 framework by applying it for modeling and controlling, automatically and in real time, the stochastic SA of four Lily modules. We describe how the resulting models can be used to optimize a bang-bang controller, and our results show a significant improvement of the system performance with respect to strategies based on single modes of agitation or a random switching between the modes. We thus demonstrate that the M^3 framework can be used for controlling a highly stochastic, distributed system without recourse to *ad hoc* control algorithms. Indeed, both the M^3 framework and the optimization technique we propose are in principle completely platform-independent.

This chapter does not investigate the use of more complex models (by adopting a 8-neighbors topology, for instance), and how they may enhance the overall performance of the system. Preliminary results that are not reported in this manuscript pinpoint an important challenge when dealing with larger models, that is, finding a proper, dynamic balance between an *explorative* behavior where the system tries to gather data for constructing and calibrating the model, and an *exploitative* behavior where the system uses the model to optimizing its functioning. Solving this problem is a critical line of future research; we believe that more advanced machine learning techniques are necessary, in particular those developed in the field of graph and network analysis.

13

Conclusion

But what is it good for?
— An engineer at the Advanced Computing Systems Division of IBM, commenting on the microchip in 1968

This dissertation proposed a unified methodological framework for the model-based, optimal design of SMPs. The notion of SMPs encompass all systems whose dynamics result from the multiple, stochastic interactions of a set of constitutive components endowed with a reactive behavior and minimal sensing, computing, and actuation capabilities, such as sub-centimeter-sized robots, insects, cells, MEMS devices, or molecules. As a result, the spectrum of potential applications of this work ranges from biomedical engineering (where large groups of ultra-small robots are envisioned as both diagnostic and therapeutic tools) to environmental engineering (e.g., spill detection, pollution monitoring, water purfication). The experimental portion of this work is *not* intended as a proof-of-concept of these anticipated applications; rather, it supports our theoretical and methodological findings, which constitute the main body of our research. In particular, the claims of this dissertation can be articulated as follows.

First, based on a series of thorough experimental and theoretical studies, this work sheds light on the critical benefits and limitations of the MLMM, which, in spite of intensive research efforts in the last decade [48, 81, 82, 190, 195, 206, 233], was still lacking a more critical and in-depth evaluation:

- We established a systematic classification of the models that can be used to describe SMPs, and we enlightened the role of stochastic processes as simplification and abstraction mechanisms in this hierarchy.
- We discussed in detail the requirement of the MLMM that individual robots must be endowed with a controller that can be represented as a FSM. Because many SMPs do not fulfill this requirement, the applicability of the MLMM is limited when dealing with non-robotic components.
- We showed that the MLMM lacks some critical features for allowing a more algorithmic approach to the modeling of SMPs. The MLMM is a collection of recipes that require a great deal of expertise and *ad hoc* adaptations as a function of the underlying system.

G. Mermoud, *Stochastic Reactive Distributed Robotic Systems*,
Springer Tracts in Advanced Robotics 93,
DOI: 10.1007/978-3-319-02609-1_13,

Second, these limitations were addressed by introducing a more systematic and generic approach to the modeling of SMPs.

- The cornerstone of this methodological effort is the M^3 framework, which provides a consistent set of theoretical and computational tools targeted to a complete automation of the modeling process. Building up in particulier on the seminal works of Schweitzer [159] and Milutinovic [51], the M^3 framework constitutes a milestone in the on-going research efforts towards generic and unified approaches to the modeling of multi-robot systems.
- The M^3 framework was thoroughly validated based on a series of experimental studies carried out in simulation, with real passive devices, and with real robots. These studies outlined the robustness, platform-independence, and scalability of the framework.

Third, this dissertation discussed in detail the role of multi-level modeling, and the M^3 framework in particular, when designing, optimizing, and controlling distributed robotic systems:

- We showed in particular how conventional top-down approaches [27, 50] may not be suitable as the influence of noise and uncertainty increases. Our findings question in particular the use of such strategies when dealing with very resource-constrained robots.
- We discussed the role of systematic searches based on highly efficient computational models for gaining insights into the global behavior of a system.
- We demonstrated experimentally the real-time optimized control of a highly stochastic self-assembling system based on the M^3 framework, without recourse to *ad hoc* control algorithms [107, 130] or active building blocks [46].

13.1 Outlook

The M^3 framework enables a simpler and broader use of multi-level modeling for designing distributed systems at large. In the near future, we have two primary objectives. First, we aim at demonstrating the relevance of the framework for designing, modeling, and optimizing the SA of real MEMS devices. Second, even though virtual particles are an important feature of the framework for capturing spatial heterogeneities, we have yet to demonstrate its applicability on relevant case studies.

From a more long-term perspective, we hope that the M^3 framework could become a standard item of the distributed systems engineer's toolbox. This ambitious objective defines several important axes for future research:

- The assumption of perfect observability of the underlying system must be relaxed, that is, the framework must be robust enough to deal with lacunary trajectories, both in state and time.

- The generated models must be directly anchored to the control and design parameters of the underlying system, such that the framework enables the optimization of the continuous values of these parameters—by opposition to the discrete approach reported in this work.
- The framework must be extended with mechanism for guiding the process of model refinement and reduction, as well as the exploration of the parameter space, such that the amount of experimental data required to construct and calibrate the models is minimized while preserving their accuracy.
- The ability of the generated models to predict the performance of systems larger (in terms of number of agents) than those on which they have been calibrated must be demonstrated experimentally.

From a more practical standpoint, the M^3 framework as it was used in this thesis presents itself as a pure research tool that offers little accessibility to the average user. One important task would be therefore to perform a complete overhaul of the framework in the perspective of an open-source, public release. A graphical user interface, pre-defined input interfaces for existing tracking tools or simulators, and post-processing features for visualizing and analyzing the resulting models are typical features to be implemented down the road.

As a final, and more personal, note, in spite of the theoretical and methodological flavor of my work, its ultimate intent is to bring more robust and cost-efficient engineered systems to life by exploiting the critical benefits of distributed strategies. Nowadays, I believe that very few industrial applications impose the use of large, coordinated groups of cheap, resource-constrained robots rather than a handful of expensive, but sophisticated robots that ignore each other. Yet, many current technological trends pinpoint a future where distributed systems will be the norm rather than the exception. I hope that this work is one solid brick of this formidable edifice under construction.

A

Maximum Likelihood Estimation of Reaction Rates

Hereafter, we solve the following optimization problem:

$$\hat{\mathbf{k}} = \underset{\mathbf{k}}{\operatorname{argmax}}\ \mathcal{L}(\mathbf{k}|e_1, \ldots, e_n) = \underset{\mathbf{k}}{\operatorname{argmax}}\, f(e_1, \ldots, e_n|\mathbf{k}). \tag{A.1}$$

where $\mathcal{L}(\mathbf{k}|e_1, \ldots, e_n)$ is the likelihood of the rate vector $\mathbf{k}$ given the sequence of observed events $(e_1, \ldots, e_n)$.

We can write the probability $f(e_i|\mathbf{k})$ of a single event e_i as follows:

$$\begin{aligned} f(e_i|\mathbf{k}) &= a_{R_i}(\mathbf{x}_i) \cdot e^{-a_{R_i}(\mathbf{x}_i)\cdot t_i} \cdot \prod_{R_j \neq R_i} \int_{t_i}^{\infty} a_{R_j}(\mathbf{x}_i) \cdot e^{-a_{R_j}(\mathbf{x}_i)\cdot t} \cdot dt \\ &= a_{R_i}(\mathbf{x}_i) \cdot e^{-a_{R_i}(\mathbf{x}_i)\cdot t_i} \cdot \prod_{R_j \neq R_i} -e^{-a_{R_j}(\mathbf{x}_i)\cdot t} \Big|_{t=t_i}^{t=\infty} \\ &= a_{R_i}(\mathbf{x}_i) \cdot e^{-a_{R_i}(\mathbf{x}_i)\cdot t_i} \cdot \prod_{R_j \neq R_i} e^{-a_{R_j}(\mathbf{x}_i)\cdot t_i} \\ &= a_{R_i}(\mathbf{x}_i) \cdot \prod_{R_j} e^{-a_{R_j}(\mathbf{x}_i)\cdot t_i} = a_{R_i}(\mathbf{x}_i) \cdot e^{-a_0(\mathbf{x}_i)\cdot t_i} \end{aligned} \tag{A.2}$$

where

$$a_0(\mathbf{x}) \triangleq \sum_{R_j \in \mathcal{R}} a_{R_j}(\mathbf{x}). \tag{A.3}$$

Since we assume independence of events (Markovian property), we can write:

$$\begin{aligned} \mathcal{L}(\mathbf{k}|e_1, \ldots, e_n) &= f(e_1, \ldots, e_n|\mathbf{k}) \\ &= f(e_1|\mathbf{k}) \cdot \ldots \cdot f(e_n|\mathbf{k}) \\ &= \prod_{i=1}^{n} a_{R_i}(\mathbf{x}_i) \cdot e^{-a_0(\mathbf{x}_i)\cdot t_i}. \end{aligned} \tag{A.4}$$

For the sake of simplicity, we will omit the arguments of $\mathcal{L}$ in the sequel.

Now, we can try to solve the optimization problem formulated by Equation A.1. To make our problem simpler (both from an analytical and a numerical standpoint), we work with the natural logarithm of the likelihood function:

$$\ln \mathcal{L} = \ln \left(\prod_{i=1}^{n} a_{R_i}(\mathbf{x}_i) \cdot e^{-a_0(\mathbf{x}_i)\cdot t_i} \right) = \sum_{i=1}^{n} \Big(\ln a_{R_i}(\mathbf{x}_i) - a_0(\mathbf{x}_i) \cdot t_i \Big). \quad \text{(A.5)}$$

First, we need to compute the gradient of the log-likelihood function $\ln \mathcal{L}$:

$$\nabla \ln \mathcal{L} = \left(\frac{\partial \ln \mathcal{L}}{\partial k_{R_1}}, \ldots, \frac{\partial \ln \mathcal{L}}{\partial k_{R_N}} \right) \quad \text{(A.6)}$$

with

$$\begin{aligned} \frac{\partial \ln \mathcal{L}}{\partial k_{R_j}} &= \sum_{i=1}^{n} \left(\frac{\partial \ln a_{R_i}(\mathbf{x}_i)}{\partial k_{R_j}} - \frac{\partial a_0(\mathbf{x}_i) \cdot t_i}{\partial k_{R_j}} \right) \\ &= \sum_{i=1}^{n} \left(\frac{1}{a_{R_i}(\mathbf{x}_i)} \frac{\partial a_{R_i}(\mathbf{x}_i)}{\partial k_{R_j}} - \frac{\partial a_0(\mathbf{x}_i)}{\partial k_{R_j}} \cdot t_i \right) \end{aligned} \quad \text{(A.7)}$$

where

$$\frac{\partial a_{R_i}(\mathbf{x}_i)}{\partial k_{R_j}} = \frac{\partial k_{R_i} \cdot \widetilde{a}_{R_i}(\mathbf{x}_i)}{\partial k_{R_j}} = \begin{cases} \widetilde{a}_{R_i}(\mathbf{x}_i) & \text{if } R_j = R_i \\ 0 & \text{otherwise} \end{cases} \quad \text{(A.8)}$$

and

$$\frac{\partial a_0(\mathbf{x}_i)}{\partial k_{R_j}} \cdot t_i = \frac{\partial a_{R_j}(\mathbf{x}_i)}{\partial k_{R_j}} \cdot t_i = t_i \cdot \widetilde{a}_{R_j}(\mathbf{x}_i). \quad \text{(A.9)}$$

Replacing these terms into Equation A.7, we obtain

$$\frac{\partial \ln \mathcal{L}}{\partial k_{R_j}} = \sum_{i=1}^{n} \left(\frac{\mathbf{1}_{R_i=R_j}}{k_{R_i}} - t_i \cdot \widetilde{a}_{R_j}(\mathbf{x}_i) \right) \quad \text{(A.10)}$$

where $\mathbf{1}_{R_i=R_j}$ is the indicator function. A local extremum of the function $\ln \mathcal{L}$ corresponds to a zero of the gradient

$$\nabla \ln \mathcal{L} = \big(0, \ldots, 0\big) \quad \text{(A.11)}$$

which is equivalent to writing

$$\begin{aligned} \sum_{i=1}^{n} \frac{\mathbf{1}_{R_i=R_j}}{k_{R_i}} &= \sum_{i=1}^{n} \Big(t_i \cdot \widetilde{a}_{R_j}(\mathbf{x}_i) \Big) \\ \frac{1}{k_{R_j}} \cdot \sum_{i=1}^{n} \mathbf{1}_{R_i=R_j} &= \sum_{i=1}^{n} \Big(t_i \cdot \widetilde{a}_{R_j}(\mathbf{x}_i) \Big) \\ \widehat{k}_{R_j} = k_{R_j} &= \frac{\sum_{i=1}^{n} \mathbf{1}_{\{R_i=R_j\}}}{\sum_{i=1}^{n} \big(t_i \cdot \widetilde{a}_{R_j}(\mathbf{x}_i) \big)} \end{aligned} \quad \text{(A.12)}$$

for $j = 1, \ldots, N$. Importantly, the rate of the reaction $R = R_j$ also depends on events that do not involve R.

For this point to be a maximum of $\ln \mathcal{L}$, we need the Hessian matrix

$$H(\ln \mathcal{L}) = \begin{pmatrix} \frac{\partial^2 \ln \mathcal{L}}{\partial k_{R_1}^2} & \frac{\partial^2 \ln \mathcal{L}}{\partial k_{R_1} \partial k_{R_2}} & \cdots & \frac{\partial^2 \ln \mathcal{L}}{\partial k_{R_1} \partial k_{R_N}} \\ \frac{\partial^2 \ln \mathcal{L}}{\partial k_{R_2} \partial k_{R_1}} & \frac{\partial^2 \ln \mathcal{L}}{\partial k_{R_2}^2} & \cdots & \frac{\partial^2 \ln \mathcal{L}}{\partial k_{R_2} \partial k_{R_N}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \ln \mathcal{L}}{\partial k_{R_N} \partial k_{R_1}} & \frac{\partial^2 \ln \mathcal{L}}{\partial k_{R_N} \partial k_{R_2}} & \cdots & \frac{\partial^2 \ln \mathcal{L}}{\partial k_{R_N}^2} \end{pmatrix} = \begin{pmatrix} H_1 & 0 & \dots & 0 \\ 0 & H_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & H_N \end{pmatrix} \tag{A.13}$$

to be negative-definite, which is clearly the case for all $\mathbf{k} \in \mathbb{R}_+^N$ since we have that

$$H_j = -\sum_{i=1}^{n} \frac{\mathbf{1}_{R_i=R_j}}{k_{R_j}^2}. \tag{A.14}$$

Importantly, the Hessian matrix can be used to compute the variance-covariance matrix of the estimated parameters, which is defined as the inverse of the Fisher information matrix $\mathcal{I}$, which is itself the negative of the expected value of the Hessian matrix:

$$\mathrm{var}(\mathbf{k}) = \big[\mathcal{I}(\mathbf{k})\big]^{-1} = \Big(- E\big[H(\ln \mathcal{L})\big]\Big)^{-1}. \tag{A.15}$$

In our case, since $H(\ln \mathcal{L})$ is diagonal, we have:

$$\mathrm{var}(\mathbf{k}) = \begin{pmatrix} -\big(E[H_1]\big)^{-1} & 0 & \dots & 0 \\ 0 & -\big(E[H_2]\big)^{-1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & -\big(E[H_N]\big)^{-1} \end{pmatrix}, \tag{A.16}$$

which translates into

$$\mathrm{var}(k_{R_j}) = -\Big(E[H_j]\Big)^{-1} = \left(E\left[\sum_{i=1}^{n} \frac{\mathbf{1}_{R_i=R_j}}{k_{R_j}^2} \right] \right)^{-1} = \frac{E[k_{R_j}^2]}{\sum_{i=1}^{n} \mathbf{1}_{R_i=R_j}}. \tag{A.17}$$

To avoid to compute the expectation of the squared reaction rate, one can use the following approximation for large sample sizes:

$$\mathrm{var}(k_{R_j}) \simeq -\left(\frac{\partial^2 \mathcal{L}(k_{R_j} | e_1, \dots, e_n)}{\partial k_{R_j}^2}\bigg|_{k_{R_j}=\widehat{k}_{R_j}} \right)^{-1} = \frac{\widehat{k}_{R_j}^2}{\sum_{i=1}^{n} \mathbf{1}_{R_i=R_j}}. \tag{A.18}$$

These terms are of course similar to the diagonal terms of the Hessian matrix.

B

Proof of Theorem 9.12

Theorem 9.12 (Continuous-discrete phase mapping). *Given a finite set of interactions $\mathcal{I}$, there exists a function $\Omega : \mathbb{S} \to \mathcal{V}^{ext}$, which maps each extended state $\mathbf{x} \in \mathbb{S}$ to a corresponding extended mode $\boldsymbol{\xi} \in \mathcal{V}^{ext}$, such that $\boldsymbol{\xi}$ is the active control mode whenever the system is in state $\mathbf{x}$.*

Proof. First, we show the *existence* of an image $\boldsymbol{\xi} \in \mathcal{V}^{\text{ext}}$ for each $\mathbf{x} \in \mathbb{S}$. By construction, each extended state $\mathbf{x}$ corresponds to a unique arrangement of the particles in $\mathbb{X}$. Given this arrangement, the control mode of each particle P_i is completely and uniquely determined by the status of the interactions in which P_i is involved. Whether an interaction $I \in \mathcal{I}$ is active depends solely on the predicate *cond*, which is completely and uniquely determined by $\mathbf{x}$, and the functions $\mathcal{D}_1$ and $\mathcal{D}_2$, which may depend on the status of other interactions. The property of monotonicity (Remark 9.7) and the finiteness of the set $\mathcal{I}$ ensures that any dependency chain is finite (in particular, if any cyclic dependency exists, all of its constitutive interactions will always remain inactive). As a result, given the position of each particle in $\mathbb{X}$, one can determine the status of each interaction, which in turn determines the interaction configuration of each particle, and the extended mode $\boldsymbol{\xi} \in \mathcal{V}^{\text{ext}}$.

Second, we show the *uniqueness* of $\boldsymbol{\xi}$. Let the extended state $\mathbf{x}$ have two images $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$. If these two extended modes are different, then the status of at least one interaction, say I_1, must be different in either modes. Since the predicate *cond* of I_1 is completely determined by $\mathbf{x}$, this difference must originate from the dependency of I_1 (given by either $\mathcal{D}_1$ or $\mathcal{D}_2$) on another interaction, say I_2, whose status is also different $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$. A similar argument can then be applied to I_2, which in turn depends on I_3, and so on, until the end of the dependency chain. In presence of cyclic dependencies, no interaction could have become active in the first place, in which case $\boldsymbol{\xi}_1$ equals $\boldsymbol{\xi}_2$. In absence of cyclic dependencies, the only cause for $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ being different is that the predicate *cond* of some interaction I_i is different in either modes, which is impossible since it is fully determined by $\mathbf{x}$. Therefore, we have $\boldsymbol{\xi}_1 = \boldsymbol{\xi}_2$.

C

Expectation of the Second-Order Error

We derive the expectation of the second-order error $\mathcal{E}_2$ between the steady state predictions of macro-stochastic and macro-deterministic models. Indeed, the finiteness of the population introduces a steady state error that grows as the number of agents decreases; an infinite number of agents makes both approaches equivalent. However, since we have a finite number N of agents, we expect to see some difference between the desired and actual distribution of agents, even after convergence. This steady state error has to be characterized in order to determine whether the given distribution has converged or not. In our case, we define the second-order error $\mathcal{E}_2$ as the square of the l^2-norm between the actual distribution and the desired distribution of agents:

$$\mathcal{E}_2 = \|\mathbf{x}^{N_0} - \mathbf{x}^{\infty}\|^2 = \sum_{i=1}^{N_S} \left(x_i^{N_0} - x_i^{\infty} \right)^2 . \tag{C.1}$$

Assume that the system is at equilibrium, and $x_i = \sum_{j=1}^{N_0} \mathbf{1}_{j,i}/N_0$ with

$$\mathbf{1}_{j,i} = \begin{cases} 1 \text{ if agent } j \text{ is in state } i \text{ at steady state,} \\ 0 \text{ otherwise.} \end{cases} \tag{C.2}$$

In the thermodynamic limit, each agent is in state i with probability x_i^{∞}. Also, each agent moves independently of the others. Hence, variables $\mathbf{1}_{j,i}$ are independent for each agent[1]. Therefore, we have (assuming $t \to \infty$):

$$\begin{aligned} E[\mathcal{E}_2] &= E\left[\sum_{i=1}^{N_S} (x_i^{N_0} - x_i^{\infty})^2 \right] = \sum_{i=1}^{N_S} E\left[(x_i^{N_0} - x_i^{\infty})^2 \right] \\ &= \sum_{i=1}^{N_S} E\left[(x_i^{N_0})^2 - 2\, x_i^{N_0}\, x_i^{\infty} + (x_i^{\infty})^2 \right] \\ E[x_i] &= E\left[\frac{\sum_{j=1}^{N_0} \mathbf{1}_{j,i}}{N_0} \right] = x_i^{\infty} \end{aligned} \tag{C.3}$$

[1] Actually, they have to follow the additional constraint $\sum_{i=1}^{N_S} \sum_{j=1}^{N_0} \mathbf{1}_{j,i} = N_0$ to conserve the number of agents. However, they are independent for each agent, which is the property used.

$$E[x_i^2] = E\left[\frac{\sum_{l,k} \mathbf{1}_{l,i} \cdot \mathbf{1}_{k,i}}{N_0^2}\right] = \frac{1}{N_0}\left(\sum_j E\left[\frac{\mathbf{1}_{j,i}^2}{N_0}\right] + \sum_{k \neq l} E\left[\frac{\mathbf{1}_{l,i} \cdot \mathbf{1}_{k,i}}{N_0}\right]\right)$$

$$= \frac{1}{N_0}\left(x_i^\infty + (N_0 - 1) \cdot (x_i^\infty)^2\right). \tag{C.4}$$

$$\text{Hence,} \quad E[\mathcal{E}_2] = \frac{1}{N_0}\sum_{i=1}^{N}(x_i^\infty - (x_i^\infty)^2) = \frac{1 - \sum_{i=1}^{N}(x_i^\infty)^2}{N_0} \tag{C.5}$$

Equation C.5 can be computed for a any given state distribution and number of agents, but it accounts only for the contribution of small copy numbers to the overall error exhibited by macro-deterministic models.

Glossary

ABM	Agent-Based Model
CAD	Computer-Aided Design
CFD	Computational Fluid Dynamics
CME	Chemical Master Equation
CMM	Canonical Microscopic Model
CRN	Chemical Reaction Network
CRNT	Chemical Reaction Network Theory
CRW	Correlated Random Walk
DES	Discrete Event Simulation
DOF	Degree of Freedom
FSM	Finite State Machine
GA	Genetic Algorithm
IR	Infrared
KS	Kolmogorov-Smirnov
MCM	Monte-Carlo Method
MDP	Markov Decision Process
MEMS	Micro-Electro-Mechanical System
MLE	Maximum Likelihood Estimation
MLMM	Multi-Level Modeling Methodology
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
PDF	Probability Density Function
PSO	Particle Swarm Optimization
SA	Self-Assembly
SMP	Smart Minimal Particle
SSA	Stochastic Simulation Algorithm

References

[1] Sleigh, C.: Six Legs Better: A Cultural History of Myrmecology (Animals, History, Culture). The Johns Hopkins University Press (February 2007) (English) cit. on p. 1

[2] Stoneking, M., Soodyall, H.: Human evolution and the mitochondrial genome. Current Opinion in Genetics & Development 6(6), 731–736 (1996) cit. on p. 1

[3] Atkins, P., de Paula, J.: Physical Chemistry, 7th edn. W.H. Freeman (December 2001) (English) cit. on p. 3

[4] Mastrangeli, M., Mermoud, G., Martinoli, A.: Modeling Self-Assembly Across Scales: The Unifying Perspective of Smart Minimal Particles. Micromachines 2(2), 82–115 (2011) cit. on p. 4

[5] Mirtschin, S., Slabon-Turski, A., Scopelliti, R., Velders, A.H., Severin, K.: A Coordination Cage with an Adaptable Cavity Size. Journal of the American Chemical Society 132(40), 14004–14005 (2010) cit. on p. 5

[6] Klavins, E.: Proportional-integral control of stochastic gene regulatory networks. In: 49th IEEE Conference on Decision and Control (CDC 2010), pp. 2547–2553 (2010) cit. on p. 5

[7] Shklarsh, A., Ariel, G., Schneidman, E., Ben-Jacob, E.: Smart Swarms of Bacteria-Inspired Agents with Performance Adaptable Interactions. PLoS Computational Biology 7(9), e1002177 (2011), cit. on p. 5

[8] Garnier, S., Jost, C., Jeanson, R., Gautrais, J., Asadpour, M., Caprari, G., Theraulaz, G.: Aggregation behaviour as a source of collective decision in a group of cockroach-like-robots. In: Advances in Artificial Life, France, pp. 169–178 (2005) (English) cit. on p. 5

[9] Mastrangeli, M., Abbasi, S., Varel, C., van Hoof, C., Celis, J.-P., Boehringer, K.F.: Self-assembly from milli-to nanoscales: methods and applications. Journal of Micromechanics and Microengineering 19, 1–37 (2009) cit. on pp. 5, 9, 16, 23

[10] Kernbach, S. (ed.): Handbook of Collective Robotics: Fundamentals and Challenges, 1st edn. Pan Stanford Publishing (April 2012) (English) cit. on p. 5

[11] 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., Deneubourg, J.-L.: Social integration of robots into groups of cockroaches to control self-organized choices. Science 318(5853), 1155–1158 (2007) cit. on pp. 5, 20, 21

[12] Julius, A., Halász, A., Sakar, M.S., Rubin, H., Kumar, V., Pappas, G.: Stochastic Modeling and Control of Biological Systems: The Lactose Regulation System of Escherichia Coli. IEEE Transactions on Automatic Control 53, 51–65 (2008) cit. on pp. 26

[13] Martel, S., Tremblay, C.C., Ngakeng, S., Langlois, G.: Controlled manipulation and actuation of micro-objects with magnetotactic bacteria. Applied Physics Letters 89(23), 233904–233904 (2006) cit. on p. 5

[14] Bogue, R.: The development of medical microrobots: a review of progress. Industrial Robot 35(4), 294–299 (2008) cit. on p. 9

[15] Bogue, R.: The fast-moving world of MEMS technology. Assembly Automation 29(4), 313–320 (2009) cit. on p.

[16] Lee, S.H., Chen, K.-N., Lu, J.J.-Q.: Wafer-to-Wafer Alignment for Three-Dimensional Integration: A Review. Journal of Microelectromechanical Systems 20(4), 885–898 (2011) cit. on p. 9

[17] Tolley, M., Kalontarov, M., Neubert, J., Erickson, D., Lipson, H.: Stochastic Modular Robotic Systems: A Study of Fluidic Assembly Strategies. IEEE Transactions on Robotics 26(3), 518–530 (2010) cit. on pp. 9, 23, 26

[18] Theraulaz, G., Bonabeau, E.: A brief history of stigmergy. Artificial Life 5(2), 97–116 (1999) cit. on pp. 9, 20

[19] Hsieh, M.A., Kumar, V., Chaimowicz, L.: Decentralized controllers for shape generation with robotic swarms. Robotica 26, 691–701 (2008) cit. on pp. 9, 14, 15, 100

[20] Turing, A.: On computable numbers, with an application to the Entscheidungsproblem. Proceedings of the London Mathematical Society 42, 230–265 (1937) cit. on p. 10

[21] Barrenetxea, G., Ingelrest, F., Schaefer, G., Vetterli, M., Couach, O., Parlange, M.: SensorScope: Out-of-the-Box Environmental Monitoring. In: Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN 2008), pp. 332–343. IEEE Computer Society (April 2008) cit. on p. 10

[22] Howard, A., Parker, L.E., Sukhatme, G.: Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection. International Journal of Robotics Research 25, 431–447 (2006) cit. on p. 10

[23] Correll, N., Martinoli, A.: Multirobot inspection of industrial machinery. IEEE Robotics & Automation Magazine 16(1), 103–112 (2009) cit. on pp. 10, 15, 30, 89

[24] Berman, S., Kumar, V., Nagpal, R.: Design of control policies for spatially inhomogeneous robot swarms with application to commercial pollination. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 378–385 (2011) cit. on pp. 10, 66, 114

[25] Amato, P., Masserini, M., Mauri, G., Cerofolini, G.: Early-stage diagnosis of endogenous diseases by swarms of nanobots: an applicative scenario. In: Dorigo, M., et al. (eds.) ANTS 2010. LNCS, vol. 6234, pp. 408–415. Springer, Heidelberg (2010) cit. on p. 10

[26] Hauert, S., Leven, S., Zufferey, J.-C., Floreano, D.: Communication-based Leashing of Real Flying Robots. In: 2010 IEEE International Conference on Robotics and Automation (ICRA 2010), pp. 15–20. IEEE (2010) (English) cit. on p. 10
[27] Berman, S., Halász, A., Hsieh, M.A., Kumar, V.: Optimized Stochastic Policies for Task Allocation in Swarms of Robots. IEEE Transactions on Robotics 25(4), 927–937 (2009) cit. on pp. 10, 14, 15, 162, 163, 167, 188
[28] Lochmatter, T.: Bio-Inspired and Probabilistic Algorithms for Distributed Odor Source Localization using Mobile Robots. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, Lausanne (2010) cit. on p. 10
[29] Pfeifer, R., Lungarella, M., Iida, F.: Self-organization, embodiment, and biologically inspired robotics. Science 318(5853), 1088–1093 (2007) cit. on pp. 11, 19
[30] Pfeifer, R., Bongard, J., Perry, D.: Designing Intelligence: Why Brains Aren't Enough, Anthology. GRIN Verlag (2011) cit. on p. 11
[31] Laibowitz, M., Paradiso, J.A.: Parasitic mobility for pervasive sensor networks. In: Gellersen, H.-W., Want, R., Schmidt, A. (eds.) PERVASIVE 2005. LNCS, vol. 3468, pp. 255–278. Springer, Heidelberg (2005) cit. on p. 11
[32] Chenciner, A.: Three body problem. Scholarpedia 2(10), 2111 cit. on p. 12
[33] Kellert, S.H.: In the wake of chaos. Unpredictable Order in Dynamical Systems. University of Chicago Press (1993) (English) cit. on p. 13
[34] Holland, J.H.: Emergence. From chaos to order. Oxford University Press (April 2000) (English) cit. on p. 13
[35] Wikipedia, Control theory (March 2012), `http://en.wikipedia.org/wiki/Control_theory` cit. on p. 13
[36] Strogatz, S.H.: Nonlinear dynamics and chaos. With applications to physics, biology, chemistry, and engineering. Westview Press (1994) (English) cit. on p. 13
[37] Thrun, S., Burgard, W., Fox, D.: Probabilistic robotics. The MIT Press (2005) (English) cit. on p. 14
[38] Arkin, R.C.: Behavior-based robotics. The MIT Press (1998) (English) cit. on p. 14
[39] Floreano, D., Mattiussi, C.: Bio-inspired artificial intelligence. Theories, Methods, and Technologies. The MIT Press (September 2008) (English) cit. on p. 14
[40] Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm intelligence. From natural to artificial systems. Oxford University Press, USA (1999) (English) cit. on p. 14
[41] Tabuada, P.: Verification and Control of Hybrid Systems. A Symbolic Approach. Springer-Verlag New York Inc. (2009) (English) cit. on p. 14
[42] Henzinger, T.A., Ho, P.H., Toi, H.W.: HYTECH: A model checker for hybrid systems. In: Grumberg, O. (ed.) CAV 1997. LNCS, vol. 1254, pp. 460–463. Springer, Heidelberg (1997) (English) cit. on p. 14
[43] Michael, N., Fink, J., Kumar, V.: Experimental Testbed for Large Multirobot Teams. IEEE Robotics & Automation Magazine 15(1), 53–61 (2008) cit. on p. 14
[44] Milutinovic, D., Lima, P.: Modeling and Optimal Centralized Control of a Large-Size Robotic Population. IEEE Transactions on Robotics 22(6), 1280–1285 (2006) cit. on pp. 14, 15

[45] Christensen, A.L., O'Grady, R., Dorigo, M.: Morphology control in a multirobot system - Distributed growth of specific structures using directional self-assembly. IEEE Robotics & Automation Magazine 14(4), 18–25 (2007) cit. on pp. 14, 22, 25

[46] Klavins, E.: Programmable Self-Assembly. IEEE Control Systems Magazine 27(4), 43–56 (2007) cit. on pp. 14, 21, 22, 25, 26, 132, 188

[47] Crespi, V., Galstyan, A., Lerman, K.: Top-down vs bottom-up methodologies in multi-agent system design. Autonomous Robots 24(3), 303–313 (2008) cit. on p. 14

[48] Martinoli, A., Easton, K., Agassounon, W.: Modeling swarm robotic systems: A case study in collaborative distributed manipulation. International Journal of Robotics Research 23(4-5), 415–436 (2004) cit. on pp. 14, 27, 48, 66, 75, 76, 99, 105, 112, 187

[49] Winfield, A.F.T., Liu, W., Nembrini, J., Martinoli, A.: Modelling a wireless connected swarm of mobile robots. Swarm Intelligence 2(2), 241–266 (2008) cit. on p. 14

[50] Matthey, L., Berman, S., Kumar, V.: Stochastic strategies for a swarm robotic assembly system. In: 2009 IEEE International Conference on Robotics and Automation (ICRA), pp. 1953–1958 (2009) cit. on pp. 14, 21, 188

[51] Milutinovic, D.L., Lima, P.U.: Cells and robots. Modeling and control of large-size agent populations. Springer (September 2007) (English) cit. on pp. 15, 26, 188

[52] Mather, T.W., Hsieh, M.A.: Macroscopic modeling of stochastic deployment policies with time delays for robot ensembles. International Journal of Robotics Research 30(5), 590–600 (2011) cit. on p. 15

[53] Nagy, Z., Oung, R., Abbott, J.J., Nelson, B.J.: Experimental investigation of magnetic self-assembly for swallowable modular robots. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1915–1920 (2008) cit. on pp. 15, 22, 23, 25

[54] Rentschler, M., Platt, S., Berg, K., Dumpert, J., Oleynikov, D., Farritor, S.: Miniature in vivo Robots for Remote and Harsh Environments. IEEE Transactions on Information Technology in Biomedicine 12(1), 66–75 (2008) cit. on p. 15

[55] Abbott, J.J., Nagy, Z., Beyeler, F., Nelson, B.J.: Robotics in the small - Part I: microrobotics. IEEE Robotics & Automation Magazine 14, 92–103 (2007) cit. on pp. 15, 16, 84

[56] Dong, L., Nelson, B.J.: Robotics in the small - Part II: nanorobotics. IEEE Robotics & Automation Magazine 14, 111–121 (2007) cit. on p. 15

[57] White, P., Kopanski, K., Lipson, H.: Stochastic self-reconfigurable cellular robotics. In: 2004 IEEE International Conference on Robotics and Automation (ICRA), pp. 2888–2893 (2004) cit. on p. 15

[58] Wood, R.J.: The First Takeoff of a Biologically Inspired At-Scale Robotic Insect. IEEE Transactions on Robotics 24(2), 341–347 (2008) cit. on p. 15

[59] Kernbach, S.: Jasmine: Swarm Robot Platform (November 2011), `http://www.swarmrobot.org` cit. on p. 15

[60] Rubenstein, M., Hoff, N., Nagpal, R.: Kilobot: A Low Cost Scalable Robot System for Collective Behaviors. Tech. Rep. (2011) cit. on p. 15

[61] Caprari, G., Siegwart, R.: Mobile micro-robots ready to use: Alice. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2005, pp. 3295–3300 (2005) cit. on pp. 15, 30

[62] Nagy, Z., Fluckiger, M., Oung, R., Kaliakatsos, I.K., Hawkes, E.W., Nelson, B.J., Harada, K., Susilo, E., Menciassi, A., Dario, P., Abbott, J.J.: Assembling reconfigurable endoluminal surgical systems: opportunities and challenges. International Journal of Biomechatronics and Biomedical Robotics 1(1), 3 (2009) cit. on p. 15

[63] Bergbreiter, S., Pister, K.: Design of an Autonomous Jumping Microrobot. In: 2007 IEEE International Conference on Robotics and Automation (ICRA), pp. 447–453 (2007) cit. on p. 15

[64] Woern, H., Szymanski, M., Seyfried, J.: The I-SWARM project. In: The 15th IEEE International Symposium on Robot and Human Interactive Communication (ROMAN 2006), pp. 492–496 (2006) cit. on p. 15

[65] Warneke, B., Last, M., Liebowitz, B., Pister, K.: Smart dust: Communicating with a cubic-millimeter computer. Computer 34(1), 44–51 (2001) cit. on p. 15

[66] Yesin, K., Vollmers, K., Nelson, B.J.: Modeling and control of untethered biomicrorobots in a fluidic environment using electromagnetic fields. International Journal of Robotics Research 25, 527–536 (2006) cit. on p. 15

[67] Donald, B.R., Levey, C.G., McGray, C.D., Paprotny, I., Rus, D.: An untethered, electrostatic, globally controllable MEMS micro-robot. Journal of Microelectromechanical Systems 15(1), 1–15 (2006) cit. on p. 15

[68] Bray, D.: Wetware: A Computer in Every Living Cell. Yale University Press (March 2011) (English) cit. on p. 18

[69] Marshall, W.F.: What Is It Like to Be a Cell? Science 325(5943), 948–948 (2009) cit. on p. 18

[70] Tamsir, A., Tabor, J.J., Voigt, C.A.: Robust multicellular computing using genetically encoded NOR gates and chemical 'wires'. Nature 469(7329), 212–215 (2011) cit. on p. 18

[71] Basu, S., Gerchman, Y., Collins, C.H., Arnold, F.H., Weiss, R.: A synthetic multicellular system for programmed pattern formation. Nature 434(7037), 1130–1134 (2005) cit. on p. 18

[72] Fiegna, F., Velicer, G.: Exploitative and hierarchical antagonism in a cooperative bacterium. PLoS Biology 3(11), 1980–1987 (2005) cit. on p. 18

[73] Alahmad, M.A., Hess, H.L.: Evaluation and analysis of a new solid-state rechargeable microscale lithium battery. IEEE Transactions on Industrial Electronics 55(9), 3391–3401 (2008) cit. on p. 19

[74] Nam, K.T., Wartena, R., Yoo, P.J., Liau, F.W., Lee, Y.J., Chiang, Y.-M., Hammond, P.T., Belcher, A.M.: Stamped microbattery electrodes based on self-assembled M13 viruses. Proceedings of the National Academy of Sciences of the United States of America 105(45), 17227–17231 (2008) cit. on p. 19

[75] Tominaka, S., Ohta, S., Obata, H., Momma, T., Osaka, T.: On-chip fuel cell: Micro direct methanol fuel cell of an air-breathing, membraneless, and monolithic design. Journal of the American Chemical Society 130(32), 10456 (2008) cit. on p. 19

[76] Nagpal, R., Zambonelli, F., Sirer, E., Chaouchi, H., Smirnov, M.: Interdisciplinary research: roles for self-organization. IEEE Intelligent Systems 21(2), 50–58 (2006) cit. on p. 19

[77] Baldassarre, G., Parisi, D., Nolfi, S.: Distributed coordination of simulated robots based on self-organization. Artificial Life 12(3), 289–311 (2006) cit. on p. 19

[78] Halley, J.D., Winkler, D.A.: Consistent Concepts of Self-organization and Self-assembly. Complexity 14(2), 10–17 (2008) cit. on pp. 19, 21

[79] Pine, A., Seymour, B., Roiser, J.P., Bossaerts, P., Friston, K.J., Curran, H.V., Dolan, R.J.: Encoding of Marginal Utility across Time in the Human Brain. Journal of Neuroscience 29(30), 9575–9581 (2009) cit. on p. 20

[80] Beckers, R., Holland, O., Deneubourg, J.-L.: From local actions to global tasks: Stigmergy and collective robotics. In: Artificial Life IV (January 1994) cit. on pp. 20, 21

[81] Martinoli, A., Ijspeert, A.J., Mondada, F.: Understanding collective aggregation mechanisms: From probabilistic modelling to experiments with real robots. Robotics and Autonomous Systems 29(1), 51–63 (1999) cit. on pp. 21, 111, 187

[82] Agassounon, W., Martinoli, A., Easton, K.: Macroscopic modeling of aggregation experiments using embodied agents in teams of constant and time-varying sizes. Autonomous Robots 17(2-3), 163–192 (2004) cit. on pp. 20, 105, 187

[83] Mamei, M., Zambonelli, F.: Pervasive Pheromone-Based Interaction with RFID Tags. ACM Transactions on Autonomous and Adaptive Systems 2(2), 4 (2007) cit. on p. 20

[84] Werfel, J., Nagpal R.: Extended stigmergy in collective construction. IEEE Intelligent Systems 21(2), 20–28 (2006) cit. on p. 20

[85] Zangwill, A.: Statistical physics - Advances in aggregation. Nature 411(6838), 651–652 (2001) cit. on p. 20

[86] Hong, L., Cacciuto, A., Luijten, E., Granick, S.: Clusters of amphiphilic colloidal spheres. Langmuir 24, 621–625 (2008) cit. on p. 20

[87] Parrish, J., Hamner, W.: Animal Groups in Three Dimensions: How Species Aggregate. Cambridge University Press (1997) cit. on p. 20

[88] Longair, M.S.: Galaxy Formation, 2nd edn. Astronomy and Astrophysics Library. Springer (January 2008) (English) cit. on p. 20

[89] Garnier, S., Jost, C., Gautrais, J., Asadpour, M., Caprari, G., Jeanson, R., Grimal, A., Theraulaz, G.: The embodiment of cockroach aggregation behavior in a group of micro-robots. Artificial Life 14(4), 387–408 (2008) cit. on pp. 20, 21

[90] Parrish, J., Edelstein-Keshet, L.: Complexity, pattern, and evolutionary trade-offs in animal aggregation. Science 284(5411), 99–101 (1999) cit. on p. 20

[91] Mermoud, G., Matthey, L., Evans, W.C., Martinoli, A.: Aggregation-mediated Collective Perception and Action in a Group of Miniature Robots. In: AAMAS 2010: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems, Toronto, Canada, pp. 599–606 (2010) cit. on pp. 21, 55

[92] Correll, N., Martinoli, A.: Modeling and designing self-organized aggregation in a swarm of miniature robots. International Journal of Robotics Research 30(5), 615–626 (2011) cit. on pp. 21, 39, 40, 41, 66, 72, 89, 145, 147

[93] Evans, W., Mermoud, G., Martinoli, A.: Comparing and modeling distributed control strategies for miniature self-assembling robots. In: 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 1438–1445 (2010) cit. on pp. 21, 22, 55
[94] Napp, N., Burden, S., Klavins, E.: Setpoint regulation for stochastically interacting robots. Autonomous Robots 30(1), 57–71 (2011) cit. on p. 21
[95] Whitesides, G.M., Grzybowski, B.A.: Self-assembly at all scales. Science 295(5564), 2418–2421 (2002) cit. on p. 21
[96] Boncheva, M., Bruzewicz, D., Whitesides, G.M.: Millimeter-scale self-assembly and its applications. Pure and Applied Chemistry 75(5), 621–630 (2003) cit. on p. 21
[97] Miyashita, S., Hadorn, M., Hotz, P.E.: Water Floating Self-assembling Agents. In: Nguyen, N.T., Grzech, A., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2007. LNCS (LNAI), vol. 4496, pp. 665–674. Springer, Heidelberg (2007) cit. on pp. 21, 22
[98] Gross, R., Dorigo, M.: Self-assembly at the macroscopic scale. Proceedings of the IEEE 96(9), 1490–1508 (2008) cit. on pp. 21, 25
[99] Rothemund, P.W.: Folding DNA to create nanoscale shapes and patterns. Nature 440(7082), 297–302 (2006) cit. on p. 22
[100] Manoharan, V., Elsesser, M., Pine, D.: Dense packing and symmetry in small clusters of microspheres. Science 301(5632), 483–487 (2003) cit. on p. 22
[101] Clark, T., Tien, J., Duffy, D., Paul, K., Whitesides, G.: Self-assembly of 10-μ m-sized objects into ordered three-dimensional arrays. Journal of the American Chemical Society 123(31), 7677–7682 (2001) cit. on p. 22
[102] Gracias, D.H., Kavthekar, V., Love, J.C., Paul, K.E., Whitesides, G.M.: Fabrication of micrometer-scale, patterned polyhedra by self-assembly. Advanced Materials 14(3), 235–238 (2002) cit. on p. 22
[103] Donald, B., Levey, C., Paprotny, I.: Planar Microassembly by Parallel Actuation of MEMS Microrobots. Journal of Microelectromechanical Systems 17(4), 789–808 (2008) cit. on pp. 22, 23
[104] Bowden, N., Choi, I., Grzybowski, B.A., Whitesides, G.M.: Mesoscale self-assembly of hexagonal plates using lateral capillary forces: Synthesis using the capillary bond. Journal of the American Chemical Society 121(23), 5373–5391 (1999) cit. on pp. 22, 23, 33
[105] Gracias, D.H., Tien, J., Breen, T., Hsu, C., Whitesides, G.M.: Forming electrical networks in three dimensions by self-assembly. Science 289(5482), 1170–1172 (2000) cit. on pp. 22, 23
[106] Tolley, M.T., Krishnan, M., Erickson, D., Lipson, H.: Dynamically programmable fluidic assembly. Applied Physics Letters 93(25), 254105 (2008) cit. on pp. 22, 23, 25
[107] Tolley, M.T., Lipson, H.: Programmable 3D Stochastic Fluidic Assembly of cm-scale modules. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4366–4371 (2011) cit. on pp. 22, 23, 25, 188
[108] Klavins, E., Ghrist, R., Lipsky, D.: A grammatical approach to self-organizing robotic systems. IEEE Transactions on Automatic Control 51(6), 949–962 (2006) cit. on pp. 22, 25
[109] Griffith, S.T., Goldwater, D., Jacobson, J.: Robotics - Self-replication from random parts. Nature 437(7059), 636–636 (2005) cit. on p. 22

[110] Zykov, V., Mytilinaios, E., Desnoyer, M., Lipson, H.: Evolved and Designed Self-Reproducing Modular Robotics. IEEE Transactions on Robotics 23(2), 308–319 (2007) cit. on p. 22

[111] Gilpin, K., Knaian, A., Rus, D.: Robot pebbles: One centimeter modules for programmable matter through self-disassembly. In: 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 2485–2492 (2010) cit. on pp. 22, 23

[112] Gilpin, K., Rus, D.: Modular robot systems. IEEE Robotics & Automation Magazine 17(3), 38–55 (2010) cit. on p. 22

[113] Marbach, D., Ijspeert, A.J.: Online optimization of modular robot locomotion. In: 2005 IEEE International Conference on Mechatronics and Automation (ICMA), pp. 248–253 (2005) cit. on p. 23

[114] Salemi, B., Moll, M., Shen, W.-M.: SUPERBOT: A Deployable, Multi-Functional, and Modular Self-Reconfigurable Robotic System. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3636–3641 (2006) cit. on p.

[115] Yim, M., Duff, D., Roufas, K.: PolyBot: a modular reconfigurable robot. In: 2000 IEEE International Conference on Robotics and Automation (ICRA), pp. 514–520 (2000) cit. on p.

[116] Murata, S., Yoshida, E., Kamimura, A., Kurokawa, H., Tomita, K., Kokaji, S.: M-TRAN: Self-reconfigurable modular robotic system. IEEE/ASME Transactions on Mechatronics 7(4), 431–441 (2002) cit. on p. 23

[117] Yim, M., Shen, W.-M., Salemi, B., Rus, D., Moll, M., Lipson, H., Klavins, E., Chirikjian, G.S.: Modular self-reconfigurable robot systems - Challenges and opportunities for the future. IEEE Robotics & Automation Magazine 14(1), 43–52 (2007) cit. on p. 23

[118] Kassner, M., Nemat-Nasser, S., Suo, Z., Bao, G., Barbour, J., Brinson, L., Espinosa, H., Gao, H., Granick, S., Gumbsch, P., Kim, K., Knauss, W., Kubin, L., Langer, J., Larson, B.C., Mahadevan, L., Majumdar, A., Torquato, S., van Swol, F.: New directions in mechanics. Mechanics of Materials 37(2-3), 231–259 (2005) cit. on p. 23

[119] Boncheva, M., Whitesides, G.M.: Making things by self-assembly. MRS Bulletin 30(10), 736–742 (2005) cit. on p. 23

[120] Jacobs, H.O., Tao, A., Schwartz, A., Gracias, D.H., Whitesides, G.M.: Fabrication of a cylindrical display by patterned assembly. Science 296(5566), 323–325 (2002) cit. on p. 23

[121] Stauth, S.A., Parviz, B.A.: Self-assembled single-crystal silicon circuits on plastic. Proceedings of the National Academy of Sciences of the United States of America 103, 13922–13927 (2006) cit. on p. 23

[122] Randhawa, J.S., Laflin, K.E., Seelam, N., Gracias, D.H.: Microchemomechanical Systems. Advanced Functional Materials 21(13), 2395–2410 (2011) cit. on p. 23

[123] Knuesel, R.J., Jacobs, H.O.: Self-assembly of microscopic chiplets at a liquid-liquid-solid interface forming a flexible segmented monocrystalline solar cell. Proceedings of the National Academy of Sciences of the United States of America 107(3), 993–998 (2010) cit. on p. 23

[124] Smith, J.S.: High density, low parasitic direct integration by fluidic self assembly (FSA). In: IEDM Technical Digest of the Electron Devices Meeting, pp. 201–204 (2000) cit. on p. 23

[125] Onoe, H., Matsumoto, K., Shimoyama, I.: Three-dimensional micro-self-assembly using hydrophobic interaction controlled by self-assembled monolayers. Journal of Microelectromechanical Systems 13(4), 603–611 (2004) cit. on p. 23

[126] Zheng, W., Chung, J., Jacobs, H.O.: Fluidic heterogeneous microsystems assembly and packaging. Journal of Microelectromechanical Systems 15(4), 864–870 (2006) cit. on p. 23

[127] Lee, S., Bashir, R.: Dielectrophoresis and chemically mediated directed self-assembly of micrometer-scale three-terminal metal oxide semiconductor field-effect transistors. Advanced Materials 17(22), 2671–2677 (2005) cit. on p. 23

[128] Shetye, S., Eskinazi, I., Arnold, D.: Magnetic Self-Assembly of Millimeter-Scale Components With Angular Orientation. Journal of Microelectromechanical Systems 19(3), 599–609 (2010) cit. on p. 23

[129] Lopez, G., Tanemura, T., Sato, R., Saeki, T., Hirai, Y., Sugano, K., Tsuchiya, T., Tabata, O., Fujita, M., Maeda, M.: DNA-grafted-polymer mediated self-assembly of micro components. In: 2010 5th IEEE International Conference on Nano/Micro Engineered and Molecular Systems (NEMS), pp. 245–249 (2010) cit. on p. 23

[130] Krishnan, M., Tolley, M.T., Lipson, H., Erickson, D.: Hydrodynamically Tunable Affinities for Fluidic Assembly. Langmuir 25(6), 3769–3774 (2009) cit. on pp. 23, 188

[131] Whitesides, G.M., Boncheva, M.: Beyond molecules: Self-assembly of mesoscopic and macroscopic components. Proceedings of the National Academy of Sciences of the United States of America 99(8), 4769–4774 (2002) cit. on p. 23

[132] Bishop, K.J., Wilmer, C.E., Soh, S., Grzybowski, B.A.: Nanoscale Forces and Their Uses in Self-Assembly. Small 5(14), 1600–1630 (2009) cit. on p. 23

[133] Torquato, S.: Inverse optimization techniques for targeted self-assembly. Soft Matter 5(6), 1157–1173 (2009) cit. on p. 24

[134] Adleman, L., Cheng, Q., Goel, A., Huang, M.-D., Kempe, D., de Espanés, P.M., Rothemund, P.W.K.: Combinatorial optimization problems in self-assembly. In: STOC 2002: Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing. ACM Request Permissions (May 2002) cit. on p. 24

[135] Cohn, H., Kumar, A.: Algorithmic design of self-assembling structures. Proceedings of the National Academy of Sciences of the United States of America 106(24), 9570–9575 (2009) cit. on p. 24

[136] Rechtsman, M.C., Stillinger, F.H., Torquato, S.: Self-assembly of the simple cubic lattice with an isotropic potential. Physical Review E 74(2), 021404 (2006) cit. on p. 24

[137] Torquato, S.: Optimal design of heterogeneous materials. Annual Review of Materials Research 40, 101–129 (2010) cit. on p. 24

[138] Jones, C., Mataric, M.J.: From local to global behavior in intelligent self-assembly. In: 2003 IEEE International Conference on Robotics and Automation (ICRA), pp. 721–726 (2003) cit. on p. 24

[139] Werfel, J., Nagpal, R.: Three-dimensional construction with mobile robots and modular blocks. International Journal of Robotics Research 27(3-4), 463–479 (2008) cit. on p. 24

[140] Arbuckle, D.J., Requicha, A.A.G.: Self-assembly and self-repair of arbitrary shapes by a swarm of reactive robots: algorithms and simulations. Autonomous Robots 28(2), 197–211 (2010) cit. on p. 24

[141] Miyashita, S., Goeldi, M., Pfeifer, R.: How reverse reactions influence the yield of self-assembly robots. International Journal of Robotics Research 30(5), 627–641 (2011) cit. on p. 24

[142] Sweeney, B., Zhang, T., Schwartz, R.: Exploring the parameter space of complex self-assembly through virus capsid models. Biophysical Journal 94(3), 772–783 (2008) cit. on p. 24

[143] Bohringer, K.F., Srinivasan, U., Howe, R.T.: Modeling of capillary forces and binding sites for fluidic self-assembly. In: The 14th International Conference on Micro Electro Mechanical Systems (IEEE MEMS 2001), pp. 369–374 (2001) cit. on p. 25

[144] Xiong, X., Liang, S.-H., Bohringer, K.F.: Geometric binding site design for surface-tension driven self-assembly. In: 2004 IEEE International Conference on Robotics and Automation (ICRA), pp. 1141–1148 (2004)
cit. on pp. 45

[145] Grzybowski, B.A., Bowden, N., Arias, F., Yang, H., Whitesides, G.M.: Modeling of menisci and capillary forces from the millimeter to the micrometer size range. Journal of Physical Chemistry B 105(2), 404–412 (2001) cit. on p. 25

[146] Kalontarov, M., Tolley, M.T., Lipson, H., Erickson, D.: Hydrodynamically driven docking of blocks for 3D fluidic assembly. Microfluidics and Nanofluidics 9, 551–558 (2010) cit. on p. 25

[147] Lienemann, J., Greiner, A., Korvink, J.G., Xiong, X., Hanein, Y., Bohringer, K.F.: Modeling, Simulation, and Experimentation of a Promising New Packaging Technology: Parallel Fluidic Self-Assembly of Microdevices. Sensors Update 13(1), 3–43 (2003) cit. on p. 25

[148] Lamber, P.: Capillary forces in microassembly. Modeling, simulation, experiments, and case study. Springer (October 2007) (English) cit. on p. 25

[149] Rivero, R.D., Shet, S., Booty, M.R., Fiory, A.T., Ravindra, N.M.: Modeling of magnetic-field-assisted assembly of semiconductor devices. Journal of Electronic Materials 37(4), 374–378 (2008) cit. on p. 25

[150] Hosokawa, K., Shimoyama, I., Hirofumi, M.: Dynamics of self-assembling systems: Analogy with chemical kinetics. Artificial Life 1, 413–427 (1995) cit. on pp. 25, 26

[151] James Wilkinson, D.: Stochastic Modelling for Systems Biology. Taylor & Francis (January 2006) cit. on p. 25

[152] Zheng, W., Jacobs, H.O.: Fabrication of Multicomponent Microsystems by Directed Three-Dimensional Self-Assembly. Advanced Functional Materials 15(5), 732–738 (2005) cit. on pp. 25, 72

[153] Mastrangeli, M., van Hoof, C., Baskaran, R., Celis, J.-P., Bohringer, K.F.: Agent-based modeling of mems fluidic self-assembly. In: The 23rd International Conference on Micro Electro Mechanical Systems (IEEE MEMS 2010), pp. 476–479. IEEE (2010) cit. on p. 26

[154] Gillespie, D.T.: Exact Stochastic Simulation of Coupled Chemical-Reactions. Journal of Physical Chemistry 81(25), 2340–2361 (1977) cit. on pp. 26, 36, 66, 71, 87

[155] Berman, S., Halász, A., Kumar, V., Pratt, S.: Bio-Inspired Group Behaviors for the Deployment of a Swarm of Robots to Multiple Destinations. In: 2007 IEEE International Conference on Robotics and Automation (ICRA), pp. 2318–2323 (2007) cit. on p. 26

[156] Tanner, H., Jadbabaie, A., Pappas, G.: Flocking in teams of nonholonomic agents. In: Cooperative Control. Univ. New Mexico, Dept. Mech. Engn, Albuquerque, NM 87131 USA, pp. 229–239 (2005) cit. on p. 26

[157] Jadbabaie, A., Lin, J., Morse, A.: Coordination of groups of mobile autonomous agents using nearest neighbor rules. IEEE Transactions on Automatic Control 48(6), 988–1001 (2003) cit. on p. 26

[158] Silva Pereira, G.A., Kumar, V., Montenegro Campos, M.F.: Closed loop motion planning of cooperating mobile robots using graph connectivity. Robotics and Autonomous Systems 56(4), 373–384 (2008) cit. on p. 26

[159] Schweitzer, F.: Brownian Agents and Active Particles: Collective Dynamics in the Natural and Social Sciences. Springer Series in Synergetics, vol. XVI. Springer (October 2003) cit. on pp. 26, 66, 188

[160] Goldberg, D., Mataric, M.J.: Detecting regime changes with a mobile robot using multiple models. In: 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 619–624 (2001) cit. on p. 26

[161] Goldberg, D., Mataric, M.J.: Coordinating mobile robot group behavior using a model of interaction dynamics. In: AGENTS 1999: Proceedings of the Third Annual Conference on Autonomous Agents. ACM (April 1999) cit. on p. 26

[162] Bongard, J., Lipson, H.: Automated reverse engineering of nonlinear dynamical systems. Proceedings of the National Academy of Sciences of the United States of America 104(24), 9943–9948 (2007) cit. on pp. 26, 154

[163] Schmidt, M.D., Lipson, H.: Distilling Free-Form Natural Laws from Experimental Data. Science 324(5923), 81–85 (2009) cit. on p. 26

[164] Schmidt, M.D., Vallabhajosyula, R.R., Jenkins, J.W., Hood, J.E., Soni, A.S., Wikswo, J.P., Lipson, H.: Automated refinement and inference of analytical models for metabolic networks. Physical Biology 8(5), 055011 (2011) cit. on p. 27

[165] Schmidt, M.D., Lipson, H.: Automated modeling of stochastic reactions with large measurement time-gaps. In: GECCO 2011: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, pp. 307–314 (July 2011) cit. on pp. 27, 152

[166] del Campo, A., Greiner, C.: SU-8: a photoresist for high-aspect-ratio and 3D submicron lithography. Journal of Micromechanics and Microengineering 17(6), 81–95 (2007) cit. on p. 32

[167] Hosokawa, K., Shimoyama, I., Miura, H.: Two-dimensional micro-self-assembly using the surface tension of water. Sensors and Actuators A 57(2), 117–125 (1996) cit. on p. 33

[168] Walther, F., Davydovskaya, P., Zuecher, S., Kaiser, M., Herberg, H., Gigler, A.M., Stark, R.W.: Stability of the hydrophilic behavior of oxygen plasma activated SU-8. Journal of Micromechanics and Microengineering 17(3), 524–531 (2007) cit. on p. 33

[169] Michel, O.: Webots: Professional Mobile Robot Simulation. International Journal of Advanced Robotics Systems 1(1), 39–42 (2004) cit. on p. 35

[170] Gerkey, B.P., Vaughan, R.T., Howard, A.: The Player/Stage project: Tools for multi-robot and distributed sensor systems. In: Proceedings of the 11th International Conference on Advanced Robotics (ICAR 2003), Coimbra, Portugal, pp. 317–323 (June 2003) cit. on p. 36

[171] Carpin, S., Lewis, M., Wang, J., Balakirsky, S., Scrapper, C.: USARSim: a robot simulator for research and education. In: 2007 IEEE International Conference on Robotics and Automation (ICRA), pp. 1400–1405 (2007) cit. on p. 36

[172] Freese, M., Singh, S., Ozaki, F., Matsuhira, N.: Virtual Robot Experimentation Platform V-REP: A Versatile 3D Robot Simulator. In: Ando, N., Balakirsky, S., Hemker, T., Reggiani, M., von Stryk, O. (eds.) SIMPAR 2010. LNCS (LNAI), vol. 6472, pp. 51–62. Springer, Heidelberg (2010) cit. on p. 36

[173] Sanft, K.R., Wu, S., Roh, M., Fu, J., Lim, R.K., Petzold, L.R.: StochKit2: software for discrete stochastic simulation of biochemical systems with events. Bioinformatics 27(17), 2457–2458 (2011) cit. on p. 36

[174] Cao, Y., Li, H., Petzold, L.R.: Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. Journal of Chemical Physics 121(9), 4059–4067 (2004) cit. on p. 36

[175] Slepoy, A., Thompson, A.P., Plimpton, S.J.: A constant-time kinetic Monte Carlo algorithm for simulation of large biochemical reaction networks. Journal of Chemical Physics 128(20), 205101 (2008) cit. on p. 36

[176] Cao, Y., Gillespie, D.T., Petzold, L.R.: Adaptive explicit-implicit tau-leaping method with automatic tau selection. The Journal of Chemical Physics 126(22), 224101 (2007) cit. on pp. 36, 121

[177] Gillespie, D.T.: Approximate accelerated stochastic simulation of chemically reacting systems. Journal of Chemical Physics 115(4), 1716–1733 (2001) cit. on pp. 36, 121

[178] Lochmatter, T., Roduit, P., Cianci, C., Correll, N., Jacot, J., Martinoli, A.: SwisTrack - a flexible open source tracking software for multi-agent systems. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4004–4010 (2008) cit. on p. 36

[179] Jeanson, R., Rivault, C., Deneubourg, J.-L., Blanco, S., Fournier, R., Jost, C., Theraulaz, G.: Self-organized aggregation in cockroaches. Animal Behaviour 69, 169–180 (2005) cit. on p. 40

[180] Sklar, E.: NetLogo, a Multi-agent Simulation Environment. Artificial Life 13(3), 303–311 (2007) cit. on pp. 43, 87

[181] Mermoud, G., Upadhyay, U., Evans, W.C., Martinoli, A.: Top-Down vs Bottom-Up Model-Based Methodologies for Distributed Control: A Comparative Experimental Study. In: Khatib, O., Kumar, V., Sukhatme, G. (eds.) 12th International Symposium on Experimental Robotics, ISER 2010 (December 2010) cit. on pp. 55, 160

[182] Wikipedia, Conceptual model (November 2011), `http://en.wikipedia.org/wiki/Conceptual_model` cit. on p. 59

[183] van Kampen, N.G.: Stochastic processes in physics and chemistry. North Holland (April 2007) cit. on pp. 66, 68

[184] Henzinger, T.A., Mikeev, L., Mateescu, M., Wolf, V.: Hybrid numerical solution of the chemical master equation. In: Proceedings of the 8th International Conference on Computational Methods in Systems Biology (CMSB 2010), pp. 55–65. ACM, Trento (2010) cit. on pp. 66, 69

[185] Wolf, V., Goel, R., Mateescu, M., Henzinger, T.A.: Solving the chemical master equation using sliding windows. BMC Systems Biology 4(1), 42 (2010) cit. on pp. 141

[186] Mateescu, M., Wolf, V., Didier, F., Henzinger, T.A.: Fast adaptive uniformisation of the chemical master equation. IET Systems Biology 4(6), 441–452 (2010) cit. on pp.

[187] Munsky, B., Khammash, M.: The finite state projection algorithm for the solution of the chemical master equation. Journal of Chemical Physics 124(4), 044104 (2006) cit. on pp. 66, 69, 121

[188] Gillespie, D.T.: Stochastic simulation of chemical kinetics. Annual Review of Physical Chemistry 58, 35–55 (2007) cit. on pp. 66, 67, 69, 70, 72

[189] Hamann, H., Worn, H., Crailsheim, K., Schmickl, T.: Spatial macroscopic models of a bio-inspired robotic swarm algorithm. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1415–1420 (2008) cit. on pp. 66, 114

[190] Prorok, A., Correll, N., Martinoli, A.: Multi-level spatial modeling for stochastic distributed robotic systems. International Journal of Robotics Research 30(5), 574–589 (2011) cit. on pp. 66, 114, 187

[191] Lerman, K., Galstyan, A., Martinoli, A., Ijspeert, A.J.: A macroscopic analytical model of collaboration in distributed robotic systems. Artificial Life 7(4), 375–393 (2001) cit. on p. 66

[192] Wikipedia, Markov property (March 2012), `http://en.wikipedia.org/wiki/Markov_property` cit. on p. 67

[193] Gillespie, D.T.: A Rigorous Derivation of the Chemical Master Equation. Physica A 188(1-3), 404–425 (1992) cit. on pp. 69, 105

[194] Cook, M., Soloveichik, D., Winfree, E., Bruck, J.: Programmability of Chemical Reaction Networks. In: Condon, A., Harel, D., Kok, J.N., Salomaa, A., Winfree, E. (eds.) Algorithmic Bioprocesses, pp. 543–584. Springer, Heidelberg (2009) cit. on p. 69

[195] Lerman, K., Martinoli, A., Galstyan, A.: A review of probabilistic macroscopic models for swarm robotic systems. In: Şahin, E., Spears, W.M. (eds.) Swarm Robotics WS 2004. LNCS, vol. 3342, pp. 143–152. Springer, Heidelberg (2005) cit. on pp. 72, 187

[196] Darling, R.W.R., Norris, J.R.: Differential equation approximations for Markov chains. Probability Surveys 5, 37–79 (2008) cit. on p. 72

[197] Gillespie, D.T.: Deterministic Limit of Stochastic Chemical Kinetics. Journal of Physical Chemistry B 113(6), 1640–1644 (2009) cit. on p. 72

[198] Feinberg, M.: Chemical reaction network structure and the stability of complex isothermal reactors. Chemical Engineering Science 42(10), 2229–2268 (1987) cit. on p. 73

[199] Feinberg, M.: Chemical reaction network structure and the stability of complex isothermal reactors. Chemical Engineering Science 43(1), 1–25 (1988) cit. on p. 73

[200] Feinberg, M.: The existence and uniqueness of steady states for a class of chemical reaction networks. Archive for Rational Mechanics and Analysis 132(4), 311–370 (1995) cit. on p. 74
[201] Martinez-Forero, I., Pelaez-Lopez, A., Villoslada, P.: Steady State Detection of Chemical Reaction Networks Using a Simplified Analytical Method. PLoS One 5(6), e10823 (2010) cit. on p. 74
[202] Joshi, B., Shiu, A.: Simplifying the Jacobian Criterion for precluding multistationarity in chemical reaction networks, arXiv.org, vol. math.DS (June 2011) cit. on p. 74
[203] Feliu, E., Wiuf, C.: Preclusion of switch behavior in reaction networks with mass-action kinetics, arXiv.org, vol. math.AG (September 2011) cit. on p. 74
[204] Pantea, C.: On the persistence and global stability of mass-action systems, arXiv.org, vol. math.DS (March 2011) cit. on p. 74
[205] Chen, S., Doolen, G.D.: Lattice Boltzmann Method for Fluid Flows. Annual Review of Fluid Mechanics 30(1), 329–364 (1998) cit. on p. 83
[206] Mermoud, G., Brugger, J., Martinoli, A.: Towards multi-level modeling of self-assembling intelligent micro-systems. In: AAMAS 2009: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems, pp. 89–96. International Foundation for Autonomous Agents and Multiagent Systems (May 2009) cit. on pp. 92, 114, 187
[207] Grinstead, C.M., Snell, J.L.: Introduction to Probability, 2nd revised edn. American Mathematical Society (1997) cit. on p. 93
[208] Correll, N.: Coordination Schemes for Distributed Boundary Coverage with a Swarm of Miniature Robots: Synthesis, Analysis and Experimental Validation. PhD thesis, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne (March 2007) cit. on p. 100
[209] Galstyan, A., Hogg, T., Lerman, K.: Modeling and mathematical analysis of swarms of microscopic robots. In: Proceedings of the 2005 IEEE Swarm Intelligence Symposium (SIS), pp. 201–208 (2005) cit. on p. 100
[210] Roduit, P.: Trajectory Analysis using Point Distribution Models: Algorithms, Performance Evaluation, and Experimental Validation using Mobile Robots. PhD thesis, École Polytechnique Fédérale de Lausanne (2009) cit. on p. 100
[211] Fasano, G., Franceschini, A.: A Multidimensional Version of the Kolmogorov-Smirnov Test. Monthly Notices of the Royal Astronomical Society 225(1), 155–170 (1987) cit. on p. 101
[212] Pugh, J., Martinoli, A., Zhang, Y.: Particle swarm optimization for unsupervised robotic learning. In: Proceedings of the 2005 IEEE Swarm Intelligence Symposium (SIS), pp. 92–99 (2005) cit. on pp. 103, 175
[213] Hamann, H., Worn, H.: A framework of space–time continuous models for algorithm design in swarm robotics. Swarm Intelligence 2(2-4), 209–239 (2008) cit. on p. 114
[214] Wilkinson, D.J.: Stochastic modelling for quantitative description of heterogeneous biological systems. Nature Reviews Genetics 10(2), 122–133 (2009) cit. on p. 116
[215] Cao, Y., Gillespie, D.T., Petzold, L.R.: Multiscale stochastic simulation algorithm with stochastic partial equilibrium assumption for chemically reacting systems. Journal of Computational Physics 206(2), 395–411 (2005) cit. on p. 121

[216] Kiehl, T., Mattheyses, R.M., Simmons, M.: Hybrid simulation of cellular behavior. Bioinformatics 20(3), 316–322 (2004) cit. on p. 121
[217] Henzinger, T.A.: The theory of hybrid automata. In: Eleventh Annual IEEE Symposium on Logic in Computer Science, pp. 278–292 (1996) cit. on p. 127
[218] Bonabeau, E.: Agent-based modeling: Methods and techniques for simulating human systems. Proceedings of the National Academy of Sciences 99(90003), 7280–7287 (2002) cit. on p. 133
[219] Napp, N., Thorsley, D., Klavins, E.: Hidden Markov Models for non-well-mixed reaction networks. In: American Control Conference, ACC 2009, pp. 737–744 (2009) cit. on pp. 139, 140
[220] Forney, G.D.: The Viterbi algorithm. Proceedings of the IEEE 61(3), 268–278 (1973) cit. on p. 141
[221] Okino, M.S., Mavrovouniotis, M.L.: Simplification of Mathematical Models of Chemical Reaction Systems. Chemical Reviews 98(2), 391–408 (1998) cit. on p. 144
[222] Munsky, B., Khammash, M.: The finite state projection approach for the analysis of stochastic noise in gene networks. IEEE Transactions on Automatic Control 53(special issue), 201–214 (2008) cit. on p. 144
[223] Gillespie, D.T., Cao, Y., Sanft, K.R., Petzold, L.R.: The subtle business of model reduction for stochastic chemical kinetics. Journal of Chemical Physics 130(6), 064103 (2009) cit. on p. 145
[224] Cappé, O., Moulines, E., Ryden, T.: Inference in Hidden Markov Models. Springer Series in Statistics. Springer (December 2010) (English) cit. on p. 152
[225] Kashiwaya, S.: Chemical reaction rate parameter estimation by MAP particle filter algorithm. In: IEEE Congress on Evolutionary Computation (CEC 2007), pp. 4489–4496 (2007) cit. on p. 152
[226] Golightly, A., Wilkinson, D.J.: Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo. Interface Focus 1(6), 807–820 (2011) cit. on p. 152
[227] Chaloner, K., Verdinelli, I.: Bayesian experimental design: A review. Statistical Science 10(3), 273–304 (1995) cit. on p. 154
[228] Bongard, J., Zykov, V., Lipson, H.: Resilient Machines Through Continuous Self-Modeling. Science 314(5802), 1118–1121 (2006) cit. on p. 154
[229] Fletcher, R.: Semi-Definite Matrix Constraints in Optimization. SIAM Journal on Control and Optimization 23(4), 493 (1985) cit. on p. 162
[230] Grant, M., Boyd, S.: CVX: Matlab Software for Disciplined Convex Programming (April 2011) cit. on p. 163
[231] Ross, S.M.: Introduction to Probability Models, 9th edn. Academic Press (December 2006) (English) cit. on p. 164
[232] Ronald, A.: Dynamic programming and Markov processes. MIT Press (1960) (English) cit. on p. 183
[233] Martinoli, A.: Swarm intelligence in autonomous collective robotics: from tools to the analysis and synthesis of distributed control strategies. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, Lausanne (1999) cit. on p. 187