



UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II



Università degli Studi di Napoli Federico II
Ph.D. Program in
Computational and Quantitative Biology
XXXVI Cycle

THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

**MODELLING AND CONTROL OF SPATIAL
BEHAVIOURS IN MULTI-AGENT SYSTEMS
WITH APPLICATIONS TO BIOLOGY AND ROBOTICS**

by
ANDREA GIUSTI

Advisor: Prof. Mario di Bernardo

Co-advisors: Prof. Diego di Bernardo and prof. Marco Faella



SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA ELETTRICA E DELLE TECNOLOGIE
DELL'INFORMAZIONE

**MODELLING AND CONTROL OF SPATIAL
BEHAVIOURS IN MULTI-AGENT SYSTEMS**
WITH APPLICATIONS TO BIOLOGY AND ROBOTICS

Ph.D. Thesis presented
for the fulfillment of the Degree of Doctor of Philosophy
in Computational and Quantitative Biology
by

ANDREA GIUSTI

December 2023



Approved as to style and content by

Prof. Mario di Bernardo, Advisor

Università degli Studi di Napoli Federico II
Ph.D. Program in Computational and Quantitative Biology
XXXVI cycle - Chairman: Prof. Michele Ceccarelli

Candidate's declaration

I hereby declare that this thesis submitted to obtain the academic degree of Philosophiæ Doctor (Ph.D.) in Computational and Quantitative Biology is my own unaided work, that I have not used other than the sources indicated, and that all direct and indirect sources are acknowledged as references.

Parts of this dissertation have been published in international journals and conference articles (see list of the author's publications at the end of the thesis).

Naples, February 23, 2024

A handwritten signature in black ink that reads "Andrea Giusti". The signature is written in a cursive style and is positioned above a horizontal line.

Andrea Giusti

*Agli amici, lontani e vicini, perché mi ricordano che
le cose importanti non sono nei libri.*

*Ai colleghi, con cui ho condiviso lunghe giornate ed
innumerevoli pause caffè.*

*A chi mi ha insegnato a capire i dettagli, e chi invece
mi ha insegnato a guardare oltre.*

*A chi ha condiviso la mia passione nel progettare e
costruire qualcosa.*

A Franci, che c'era dall'inizio.

Ed alla mia famiglia, che in effetti, c'era da prima.

Abstract

Large-Scale Multi-Agent Systems (LS-MAS) consist of several autonomous components, interacting in a non-trivial way, so that the emerging behaviour of the ensemble depends on the individual dynamics of the components and their reciprocal interactions. These models can describe a rich variety of natural systems, as well as artificial ones, characterised by unparalleled scalability, robustness, and flexibility. Indeed, a crucial objective is devising efficient strategies to model and control the spatial behaviours of LS-MAS to achieve specific goals. However, the inherent complexity of these systems and the wide spectrum of their emerging behaviours pose significant challenges. The overarching goal of this thesis is, therefore, to advance methods for modelling, analyzing and controlling the spatial behaviours of LS-MAS, with applications to cellular populations and swarm robotics. The thesis begins with an overview of the existing Literature, and is then organized into two distinct parts. In the context of swarm robotics, Part I deals with distributed control algorithms to spatially organize agents on geometric patterns. The contribution is twofold, encompassing both the development of original control algorithms, and providing a novel formal analysis, which allows to guarantee the emergence of specific geometric patterns. In Part II, looking at the spatial behaviours of biological agents, experiments are carried out to study the movement of microorganisms and their response to light stimuli. This allows the derivation and parametrization of mathematical models that capture these behaviours, and pave the way for the development of innovative approaches for the spatial control of microorganisms. The results presented in the thesis were developed by leveraging formal analytical tools, simulations, and experiments, using innovative platforms and original computational frameworks.

Keywords: emerging behaviours, control theory, multi-agent systems, biology, swarm robotics.

Sintesi in lingua italiana

I Sistemi Multi-Agente su Larga Scala (LS-MAS) sono costituiti da molteplici componenti autonome, che interagiscono tra di loro, così che il comportamento emergente dell'insieme dipenda dalla dinamica delle singole componenti e dalle loro reciproche interazioni. Questi modelli possono descrivere una ricca varietà di sistemi, sia naturali che artificiali, caratterizzati da scalabilità, robustezza e flessibilità. Infatti, lo sviluppo di strategie per modellare e controllare i comportamenti spaziali dei LS-MAS, rappresenta un problema cruciale. Tuttavia, l'intrinseca complessità di questi sistemi e l'ampio spettro dei loro comportamenti, pongono grosse sfide. L'obiettivo di questa tesi è, quindi, quello di contribuire allo sviluppo di metodi per modellare, analizzare e controllare i comportamenti spaziali dei LS-MAS, con applicazioni alle popolazioni cellulari ed agli sciami robotici. La tesi inizia con una panoramica della Letteratura esistente, per poi essere organizzata in due parti distinte. Nel contesto della robotica degli sciami, la Parte I tratta gli algoritmi di controllo distribuito per organizzare spazialmente gli agenti su pattern geometrici. Il duplice contributo, comprende sia lo sviluppo di nuovi algoritmi di controllo, sia l'introduzione di un'analisi formale, che consente di garantire l'emergere di specifici pattern. Nella Parte II, al fine di esaminare i comportamenti spaziali di agenti biologici, vengono condotti esperimenti per studiare il movimento dei microrganismi e la loro risposta agli stimoli luminosi. Ciò consente la derivazione e la parametrizzazione di modelli matematici che catturano questi comportamenti, ed aprono la strada allo sviluppo di approcci per il controllo spaziale dei microrganismi. I risultati presentati nella tesi sono stati sviluppati sfruttando strumenti analitici, simulazioni ed esperimenti, utilizzando piattaforme innovative e strumenti computazionali originali.

Parole chiave: comportamenti emergenti, teoria del controllo, sistemi multi-agente, biologia, sciami robotici.

Acknowledgements

Part of the research presented in this dissertation was developed during a nine months period at the BioCompute Lab (University of Bristol), under the supervision of Dr. Thomas Goroehowski.

Moreover, it has been partially carried out in the framework of the projects PRIN 2017 “Advanced Network Control of Future Smart Grids” and PON INSIST 2019 “Sistema di monitoraggio intelligente per la sicurezza delle infrastrutture”.

Contents

Abstract	i
Sintesi in lingua italiana	ii
Acknowledgements	iii
List of Acronyms	viii
1 Introduction	1
1.1 Motivation	1
1.2 Open challenges	2
1.3 Main contributions	3
1.4 Thesis outline	4
2 Background	5
2.1 Emergent behaviours	6
2.2 Modelling & Analysis	8
2.3 Control algorithms	10
2.3.1 Hierarchic classification of control algorithms	11
2.3.2 Distributed and centralized implementations	13
2.4 Discussion	14

I	Geometric pattern formation	15
3	Introduction to geometric pattern formation	16
3.1	State of the art on geometric pattern formation	17
3.1.1	Position-based approaches	18
3.1.2	Displacement-based approaches	19
3.1.3	Distance-based approaches	20
3.1.4	Open problems	21
3.2	Mathematical preliminaries	21
3.2.1	Dynamical systems	21
3.2.2	Frameworks and rigidity	22
3.2.3	Swarms of mobile agents	23
3.3	Discussion	25
4	Distributed control for geometric pattern formation	26
4.1	Control goal and performance metrics	26
4.2	Design of the control law	29
4.2.1	Problem formulation	29
4.2.2	Distributed control law	30
4.3	Numerical validation	32
4.3.1	Simulation setup	32
4.3.2	Tuning of the control gains	35
4.3.3	Robustness analysis	36
4.3.4	Comparison with other established algorithm	41
4.4	Adaptive tuning of control gains	44
4.4.1	Robustness analysis	45
4.5	Experimental validation	47
4.6	Discussion	49
5	Convergence toward geometric patterns	51
5.1	Problem formulation	52

5.2	Convergence to a rigid lattice configuration	54
5.3	Numerical validation	59
5.3.1	Simulation setup	59
5.3.2	Numerical results	60
5.4	Further results	63
5.5	Discussion	64
II Spatial behaviours of microorganisms		65
6	Introduction to spatiotemporal control of microorganisms	66
6.1	Movement of microorganisms	67
6.1.1	Run and tumble dynamics	68
6.1.2	Persistent Turning Walkers	69
6.2	Light induced behaviours	71
6.3	Control applications & experimental platforms	73
6.4	Discussion	74
7	Experimental setup and data processing	76
7.1	DOME: the experimental platform	76
7.1.1	Hardware	77
7.1.2	Software	79
7.2	Experiments with microorganisms	79
7.2.1	Microorganisms	79
7.2.2	Experimental protocol	83
7.3	Data extraction and analysis	85
7.3.1	The behaviour in a dark environment	87
7.3.2	The influence of light	89
7.4	Discussion	95

8	Data-driven modelling	96
8.1	Stochastic model of motion	96
8.2	Model parametrization	98
8.3	Numerical validation	100
8.4	Towards spatial control	104
8.5	Discussion	105
9	Conclusions	108
A	SwarmSim: a framework for Multi-Agent Systems simulation	111
A.1	Core functionalities and practical utility	111
A.2	Discussion	113
B	Tracking software	114
B.1	Background modelling	114
B.2	Objects detection	115
B.3	Objects tracking	117
B.4	Discussion	118
	Bibliography	122
	Author's publications	138
	External resources	139

List of Acronyms

The following acronyms are used throughout the thesis.

DLP	digital light projector
DOME	Dynamic Optical Micro-Environment
LS-MAS	Large-Scale Multi-Agent Systems
OLS	Ordinary Least Squares
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
PTW	Persistent Turing Walker
SDE	Stochastic Differential Equation
UAV	Unmanned Aerial Vehicle

Introduction

1.1 Motivation

The behaviour of a rich variety of systems, in Nature and Technology, depends on the dynamics of multiple individual components and their reciprocal interactions. This is the case for both natural systems, such as gene networks in cells, flocks of birds, or global climate; and artificial ones, among others robotic swarms and internet [46, 109]. When the number of agents becomes extremely large, the role of their interconnections more intricate, and the interactions with the environment predominant, we talk of *Large-Scale Multi-Agent Systems* (LS-MAS), or *swarms* [13].

The distinctive trait of such systems lies in the emergence of complex behaviours, stemming from the interactions among the individual agents. Examples of such *emerging behaviours* are the synchronized flashing of fireflies, the flocking of birds [5], or the formation of patterns by swarming bacteria [29]. Furthermore, artificial multi-robot systems can be engineered to replicate these behaviours, and to address complex tasks, such as manipulation or patrolling, in a cooperative manner. The coexistence and the interplay, between the *microscopic* dynamics of the individual agents, and the emerging *macroscopic* behaviour of the swarm, define the characteristic *multi-scale* nature of these systems. Crucially, the distributed nature of LS-MAS guarantees *scalability*, i.e. the ability to cope with large and often fluctuating numbers of agents, *robustness* to agents' failure and various disturbances, and *flexibility* to adapt to different environments and tasks

[13]. These features catalysed interests around such systems, and fostered the development of new technologies and methodologies, with applications spanning from synthetic biology [46] to swarm robotics [13], from opinion dynamics [63] to infrastructures' resilience [48].

1.2 Open challenges

A pressing problem for the deployment of real world applications is devising efficient strategies to model **LS-MAS** and control their spatial behaviour to achieve specific goals, e.g. microbial pattern formation or swarming robots coordination. However, the inherent complexity of such systems, and the wide spectrum of their possible emerging behaviours pose significant challenges to the systematic study of their dynamics and the design of general methodologies to, either, model or control their behaviour. A first difficulty lies in the formal description and classification of the different behaviours that can emerge, which, in many cases, have not been clearly classified.

The lack of general tools for the study of emerging behaviours in **LS-MAS** constitutes a second challenge. Conventional approaches from control theory can fail in guaranteeing stability, scalability and robustness, primarily due to the high dimensionality and the often non-linear nature of **LS-MAS**; these difficulties worsening in the presence of a variable number of agents and uncertainties. In particular, the study of stability offers unparalleled insights into the nature of dynamical systems and has been the pillar of this field, nevertheless the complexity of **LS-MAS** often caused this formal analysis trudging behind. Indeed, many emerging behaviours have been studied and a variety of control algorithms has been validated by simulations or experiments, while the analytical study of their stability remains unaddressed.

Moreover, we still lack effective methodologies to build mathematical models, able of describing the intricate behaviours of real-world **LS-MAS**, such as the coordinated motion of fishes and birds, or the cooperation in insects colonies. The nature of such systems poses great difficulties to the bottom-up development of models from first principles. Therefore, we have to rely on top-down approaches, such as data-driven modelling [14, 122].

Lastly, the design of distributed control algorithms to achieve desired

emerging behaviours has been addressed primarily in specific cases, with a more comprehensive design framework still missing. This is certainly due to intrinsic difficulties in developing control strategies able to close a loop between the microscopic and the macroscopic scales, but also the lack of models for real-world [LS-MAS](#) represents a significant obstacle.

1.3 Main contributions

In this context, this thesis aims to contribute to the development of novel and effective methods for modelling, analyzing and controlling [LS-MAS](#) of mobile agents, with a focus on studying their spatial behaviours for applications to cellular populations and swarm robotics. To enhance the understanding of these emerging behaviours, and of the existing modelling and control approaches, we provide a concise review of the existing Literature. From this, we move to tackle the design and the formal analysis of distributed control algorithms for pattern formation. Specifically, we develop an interaction law that allows a swarm to self organize onto geometric patterns, such as triangular and square lattices. While this approach is validated via simulations and experiments, we apply formal tools to a similar, and very popular control algorithm, proving its local stability. Furthermore, we address the problem of constructing mathematical models able to describe the spatial behaviour of real world [LS-MAS](#). In particular, starting from experimental data we acquired, we characterize and then model, the movement and the light response of some microorganisms, in order to provide the first step towards the development of new methodologies to control their spatial distribution.

To develop our results we employed formal tools, such as graph theory and Lyapunov stability; and developed a novel simulation platform, called [SwarmSim](#), to carry out agent-based simulations of [LS-MAS](#). Also, we performed experiments, using two innovative platforms, the [Robotarium](#) for swarm robotics applications and the [DOME](#) for biological agents. Overall, this thesis addresses general open problems in the field of [LS-MAS](#) and their application to robotic and biological agents, providing both novel methodological results and new numerical tools.

1.4 Thesis outline

The thesis is organized as follows; Chapter 2 provides a brief overview of the literature on [LS-MAS](#) of mobile agents, presenting the most relevant aspects of modelling, classifying, analyzing, and controlling their emerging behaviours. In particular we focus on those behaviours that influence the spatial organization of these systems. The rest of the thesis is then organized in two parts. Part I concerns geometric pattern formation, and specifically the development of novel swarming strategies and their proof of stability, in the context of robotics applications. Specifically, Chapter 3 introduces the problem of geometric pattern formation, reviews the existing Literature and the necessary preliminary mathematical formalism. Chapter 4 presents our own solution to achieve this behaviour with a novel distributed approach, while the formal study on the stability of specific geometric configurations is discussed in Chapter 5.

Part II, deals with a biological application, specifically the movement of microorganisms (protozoa and microscopic algae), and their response to light stimuli. The problem is first introduced and described in Chapter 6, while Chapter 7 presents our experimental methodology and results. Finally, Chapter 8 discusses how to mathematically model such behaviours, and how these can be leveraged to control the spatial distribution of the microorganisms. Conclusions are drawn in Chapter 9.

Two software tools we developed in the process of our research are presented in the Appendices. Specifically, Appendix A presents Swarm-Sim, an agent-based simulator we developed for the study of multi-agent systems, while Appendix B presents our computer vision software for the automatic detection and tracking of moving objects in the [DOME](#).

The results reported in this thesis appeared in the articles listed in the [Author's publications](#) section.

Background

As introduced in Chapter 1, *Large-Scale Multi-Agent Systems (LS-MAS)* consist of several autonomous components, or *agents*, interacting in a non-trivial way [113], so that the emerging behaviour of the ensemble depends on the individual dynamics of the components and their reciprocal interactions, and vice versa [36]. A strictly related concept is that of *swarm*, which is mostly used in robotics to describe a *LS-MAS* of mobile robots moving together in some coordinated manner and emphasizes the idea of physically embodied agents [13]. In the rest of the thesis we will use these two terms interchangeably.

LS-MAS can describe a rich variety of natural systems, as well as artificial ones. Moreover, such systems show desirable properties, namely scalability, robustness and flexibility. These aspects together motivate the interest in understanding, modelling and controlling their behaviour. In this Chapter, we will discuss what characterizes the emergent behaviours of *LS-MAS* and how these can be classified, focusing on those behaviours that influence the spatial distribution of the swarm. Then, we will introduce the possible modelling approaches for *LS-MAS*, and how these can be used to understand the emerging properties of the system. Finally, an overview of the existing control approaches is presented.

2.1 Emergent behaviours

A distinctive characteristic of **LS-MAS** is the wide range of complex emergent behaviors they can show. Unlike ordinary systems, whose steady-state behavior can generally be described in terms of equilibrium points, limit cycles, quasi-periodic or chaotic attractors [123], in **LS-MAS** the relative behaviors of agents can determine the origin of complex emerging properties. The simpler of emerging behaviours are consensus and synchronization [105]. Other examples include aggregation, flocking, area coverage, morphogenesis and more [13, 82].

Various classifications of these behaviours have been proposed, but a general consensus is still lacking. Overall, one of the most influential was proposed by Brambilla et al. [13], where emergent behaviours are classified in *spatial organization*, *navigation*, *collective decision making* and *others*.

Here, we do not aim to propose a new general classification, but try to adjust the existing ones to define the class of *spatial behaviours* of our interest. Specifically, we will say spatial behaviours are those characterized by a steady state configuration that satisfies some geometric constraints on the states of the agents. These will include consensus, aggregation, density regulation and more (see below). Contrarily, we will not include behaviours explicitly involving persistent movement¹ (e.g. flocking and synchronization), or more complex ones requiring interactions with the external environment, such as object assembling (that instead is included within *spatial organizing behaviours* in [13]), cooperative transportation, mapping, herding, etc.

Moreover, some of the spatial behaviors, despite being treated in a vast existing Literature, have been mostly defined heuristically and ambiguously, with different authors adopting different definitions, e.g. [13, 82]. Therefore, in the following, we list the spatial behaviours we identified, together with the proposed definitions and some possible applications. We will loosely use $\mathbf{x}_i \in \mathbb{R}^d$ to represent the state of agent i , and $\rho(\mathbf{x}, t) : \mathbb{R}^d \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$ for the density distribution of the agents in a state space, of dimension d . Moreover, we will implicitly refer to the steady state behaviour, and therefore omit the dependence on time:

¹This does not require the agents to necessarily become static, as long as their motion does not influence the resulting steady state behaviour.

- *Consensus* describes the convergence of all the agents to the same equilibrium state, formally $\mathbf{x}_i = \mathbf{x}_j \forall i, j$ with $\dot{\mathbf{x}}_i = 0$. Possible examples encompass rendezvous [62], the alignment of magnetosomes in magnetotactic bacteria or of schooling fishes [24]. Moreover, if the state of the agents represents their opinion, consensus can represent unanimous decisions in social networks [132].
 - *Aggregation* is a weaker form of consensus, requiring all the agents to converge towards a bounded set of their state space, formally guaranteeing that eventually $\|\mathbf{x}_i - \mathbf{x}_j\| < M \forall i, j$. A classic example is spatial aggregation of mobile agents [42, 73], but it can also describe the emergence of bounded consensus in opinion dynamics models [76].
 - *Dispersion* is the opposite behaviour to aggregation, requiring agents to spread and diverge from one another, formally $\|\mathbf{x}_i - \mathbf{x}_j\| \rightarrow \infty \forall i, j$. This simple behaviour [11] can have important applications in exploration [31, 32], or search and localization [146].
 - *Morphogenesis* is the formation of organic-like shapes, a behaviour that eludes a formal definition. It was famously introduced by Alan Turing in [129], and recently inspired algorithms for the spontaneous organization of robots [18, 91]. A similar behaviour, the natural formation of patterns by swarming bacteria, has also been used to encode digital information [29].
 - *Pattern formation*, in a very general formulation, requires each agent to converge to a different point p of a specified set $A \subset \mathbb{R}^d$, formally $\forall i \exists \mathbf{p} \in A : \mathbf{x}_i = \mathbf{p}$, with $\mathbf{x}_i \neq \mathbf{x}_j \forall i, j$. Depending on the actual definition of the set A , it can describe a quite various range of behaviours, such as geometric pattern formation [45, 147], 1D shape formation [59, 85] and 2D shape formation [103].
 - *Density regulation* requires the swarm to distribute accordingly to some density distribution of interest ρ_d , formally $\rho(\mathbf{x}, t) \propto \rho_d(\mathbf{x})$. It has been implemented in swarming robots [32, 35], with applications to surveillance [118], in mammalian cells for tissue engineering [43], and in microorganisms [69, 84].
-

In this thesis we will mainly focus on two of these behaviours, specifically Part I will focus on *pattern formation*, while Part II will deal with the motion of microorganisms and how this can induce *density regulation*.

2.2 Modelling & Analysis

Modelling the dynamics of LS-MAS is a crucial step, as it allows to perform simulations, carry out mathematical analysis of their emergent behaviours and, finally, design model-based control strategies. Mathematical models of dynamical systems typically consists of a set of first-order Ordinary Differential Equations (ODEs), each one describing the time evolution of one of the *state variables* of the system. Such equations can be either *linear* or *non-linear*. In the first case, they can be easily written in matrix form and analysed with the tools of linear algebra [65, Section 4.3]. On the contrary, non-linear systems do not admit such simple analysis. While their local behaviour around a specific state can, often, be studied through linearization, a more general analysis requires different tools, such as Lyapunov analysis [65, Section 4.1]. Moreover, some systems, for example biological ones, have inherently stochastic behaviours. In such cases ODEs are replaced by Stochastic Differential Equations (SDEs), which can include randomness, usually in the form of additive white noise [33].

Clearly, most LS-MAS of interest involve non-linear dynamics and, of course, an extremely large and often variable, number of state variables, therefore ad-hoc modelling approaches are necessary. Specifically, models of LS-MAS can be divided into two main groups, which reflect the multi-scale nature of their dynamics:

- *Microscopic models*, also known as agent-based models, describe each agent (cell, robot or component) individually, and the topology of their interactions is represented by a (di-)graph [16]. This formulation is, usually, intuitive and easy to build, moreover it allows the straightforward integration of a distributed control action into the agents' dynamics. Nevertheless, inferring the emerging properties of the ensemble and their robustness to changes in the network topology remains challenging.
 - *Macroscopic models* describe the whole ensemble (colony, consor-
-

tium, swarm) at aggregate level as a continuum. Therefore, individual agents are disregarded and instead the average behavior is studied, capturing at once the time and spatial evolutions by using Partial Differential Equations (PDEs). Specifically, the resulting equation usually takes the form of a Fokker-Planck equation, also known as forward Kolmogorov equation [102]. These models are not always feasible, especially in the case of non trivial and heterogeneous interaction topologies, and require analysis tools different from those based on ODEs. On the other hand, they are particularly suited for stochastic systems and certain applications, such as density control. For a deeper discussion on this approach see [33].

Given the different characteristics of these modeling approaches, tools to obtain one model from the other are necessary. Specifically, moving from a discrete to a macroscopic description, a process called *continuification* [81], requires specific tools such as the mean-field limit [33], *graphons* [78], or the continuous conversion of Partial Difference Equations [89]. Moreover, this process, by approximating the individual states of the agents with a continuous distribution, implies some loss of information [70]. Also the reverse process of *discretization* comes with its own challenges, as it requires to infer the specific state of agents and their interaction topology. Indeed, this process is rarely considered, but has a crucial role in obtaining a distributed control law (ODEs), to be implemented on the agents, out of a continuous control law (PDEs) designed on a macroscopic model, as done, for example, in [81].

The development of the model itself represents another crucial aspect. Indeed, bottom-up approaches, that consist in deriving the dynamics of the system starting from first principles, are hardly applicable to LS-MAS, due to their complexity. Instead, top-down approaches (e.g. data-driven modelling), leverage the observation of specific dynamics and data to capture the aspects of interest in the dynamics of the system [14]. These second approaches, while still presenting numerous challenges, can be more suited to deal with LS-MAS. An example is the modelling approach described in Chapter 8.

Once a LS-MAS model has been obtained, in one of the possible representations, it needs to be analyzed to infer the emerging properties of the system, such as the existence and the stability of equilibria, or more

complex steady state behaviors as those discussed in Section 2.1. We will also be interested in studying robustness to variations of the number of agents, noise and disturbance. Due to the variety of emerging behaviors **LS-MAS** can show, the analysis of such systems requires ad-hoc methods. Two fundamental tools in the study of such behaviors are Lyapunov's and graph theory [86]. In particular, Lyapunov theory has been extended in multiple directions, for example vector Lyapunov functions allow to study the robustness to agents removal [112], while graph theory allows to model and analyze the network of interactions between the agents. In Chapter 5 we use these tools to prove the stability of specific geometric pattern formations. Other useful tools are *passivity* [65, Chapter 6] and *contraction* [15] theories, that have been extended to assess emerging properties of the ensemble from the study of the individual agents' dynamics [2, 105].

2.3 Control algorithms

LS-MAS can be engineered to achieve desired goals and leverage cooperation to solve complex problems, such as surveillance [77], exploration [64], herding [1] or transportation [6, 40]. Unfortunately the classic paradigm of control theory, does not apply as is to these systems. Here, there is no longer a single process and a controller, that observes the output of the process and computes the appropriate inputs to steer the process towards the desired state. Instead, to control **LS-MAS**, one needs to close a feedback loop across multiple scales, understanding how to drive the individual agents (microscopic level), to obtain the desired effect at a macroscopic level (see Figure 2.1). For example, in density regulation problems, one might be able to measure the (macroscopic) spatial density of the swarm, while the control action would be exerted by controlling the movement of the single agents. Currently, despite the existence of many specific solutions and a vast body of research, no general method has been developed to translate macroscopic specifications into instructions for the individual agents to follow [52].

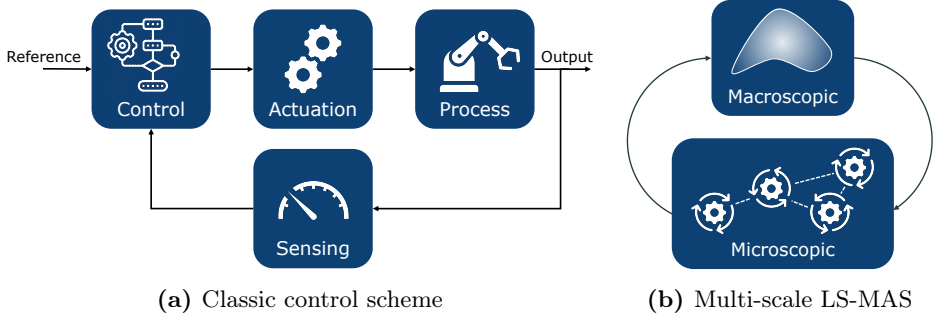


Figure 2.1: Schematic comparison between classic systems and LS-MAS.

2.3.1 Hierarchic classification of control algorithms

To better frame the open challenges and to understand how to achieve control of LS-MAS, it is useful to define a hierarchic structure of the control process [82]. It is a standard approach in many engineering disciplines, according to the motto *divide et impera*, but has not received the deserved attention in the context of LS-MAS. Majid et al. in [82], divide the tasks commonly found in swarm robotics into two layers, *low-level tasks*, such as aggregation or flocking, and *high-level tasks*, including collective mapping or transportation, concerning the collective interaction of the swarm with the environment, but the connection between the two levels is fuzzy and not addressed directly. A more refined hierarchy, focuses on the control of Unmanned Aerial Vehicles (UAVs) and encompasses five layers [148]. The highest level, *decision-making*, performs planning and task allocation, then the *path planning* generates the path to reach the given goal. The *control* layer is in charge of the multi-agent coordination, including formation control and obstacle avoidance, and can rely on the services provided by the *communication* layer. Finally, the *application* layer represents the lowest level and deals with the execution of the application-specific task. Here the analysis is narrowed down to the specific case of groups of UAVs, and decentralization is considered only for the lower levels.

Here, we propose a simple yet general hierarchical classification, distinguishing three control layers, each of which is in charge of tracking the reference generated by the upper level, as depicted in Figure 2.2. In general the higher the control level the greater the abstraction of the system

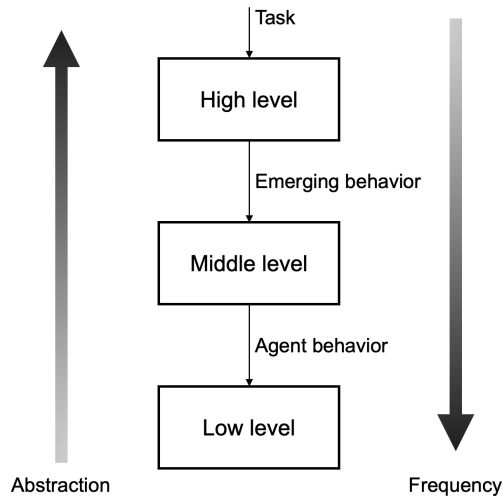


Figure 2.2: Hierarchy of control algorithms for LS-MAS. The higher the layer the more abstract is the system representation and the lower the required execution frequency.

and the slower the dynamics to be controlled, and therefore the execution frequency.

High level control layer. Starting from the high level (macroscopic) specification of the task to be solved (e.g. patrol a certain region), eventually provided by the human operator, high level control algorithms are in charge of defining the appropriate emerging behavior of the swarm that is required to solve the given task (e.g. flock towards north, then form a geometric pattern). The specific properties of the swarm, such as the number agents, and of the agents themselves are completely abstracted. The execution is most likely centralized and relatively slow, due to the slow evolution of the high level dynamics of interests. Possibly, the high level control might also be executed off-line.

Middle level control layer. Given the desired emerging behavior of the swarm, computed by the high level control algorithm, the middle level algorithm computes the behaviour of each individual agent (e.g. agent i

moves north with a certain speed, while agent j moves towards it). Simple models of the agents' dynamics (e.g. first order integrator) and their interactions (e.g. a graph) are used, using any of the representations discussed in Section 2.2. The execution might be either centralized or, preferably, distributed.

Low level control layer. Finally, the lower level implements the “servo” control of the single agent (e.g. turn the left wheel forward). The implementation is necessarily platform dependent and completely distributed, therefore an accurate model of the agent and its physical characteristics is necessary. Most likely, a high execution frequency will be required, in order to provide the fast response necessary to the functioning of physical sensors and actuators.

The high and low levels are less relevant in this thesis, as they do not directly deal with the multi-agent aspect of the problem, and can, in most cases, be addressed with tools from other areas of engineering. Therefore, we will mostly focus on middle layer control strategies and particularly, on spatiotemporal control, aiming to regulate some quantity of interest, in both space and time, to obtain the desired emerging behaviour.

2.3.2 Distributed and centralized implementations

As said above, given the model of a [LS-MAS](#) of interest and the desired emerging behavior, the middle level control is a feedback control algorithm that makes the target behavior emerge. Such control law may be either centralized or distributed. In the first case a single entity observes the whole system and defines a control action that is imposed on the system. It can be implemented either by communicating the control action to each agent (*centralized internal control*), or by applying some modification to the environment the system is evolving in (*centralized external control*) [111]. On the other hand, a *distributed control* law prescribes that each agent autonomously computes its own action. Moreover, a distributed control algorithm should be local, that is, it only uses information that can be directly measured or estimated by the agents. Distributed and local control approaches intrinsically provide *scalability* with the size of the system and *robustness* to failure of agents [13]. Indeed, most of the

literature on the control of **LS-MAS** focuses on these approaches, with applications to virtually any task. This is also the approach used for pattern formation and discussed in Part I.

Conversely, a centralized controller needs an infeasible increase of sensing and computation capabilities, as the number of agents increases, and represents a single point of failure. One possible solution to achieve scalability in centralized architectures, is represented by macroscopic models (see Section 2.2). Indeed, such models provide a representation of the system that is independent of the number of agents and their individual state. A possible application of this approach is the spatial control of microorganisms, discussed in Part II.

2.4 Discussion

This Chapter provided a brief overview of modelling and controlling emergent behaviours of **LS-MAS**. Firstly, the spatial behaviours were introduced as those influencing the steady-state distribution of the agents in the space. Among these, pattern formation and density regulation will be discussed in the rest of the thesis. To study the properties of these behaviours, one needs to properly describe the dynamics of the system of interest. Hence, we discussed the microscopic and macroscopic modelling approaches, the respective features and the most relevant analysis tools. Finally, we focused on the algorithms to control the emergent behaviours, and how these can deal with the multi-scale nature of **LS-MAS**. In particular, we discussed of a hierarchic control architecture, to address the different problems in the control of such systems, and the importance of decentralization, to improve scalability and robustness of the system. The next Chapters will present our contributions on the control of geometric pattern formation (Chapters 3 - 5 in Part I), and the study of spatial behaviours in microorganisms (Chapters 6 - 8 in Part II).

Part I

Geometric pattern formation

Introduction to geometric pattern formation

As introduced in Chapter 1 spatial organisation is a crucial feature for both natural and artificial multi-agent systems. Here we will focus on *geometric pattern formation* (see Section 2.1), where the agents self-organize their relative positions into some repeating geometric structure or *pattern*, e.g., arranging themselves on a lattice consisting of repeating adjacent triangles (see Figure 3.1). It is crucial in many tasks involving large-scale multi-agent systems and especially in *swarm robotics* [92]. Examples include sensor networks deployment [67, 147], cooperative transportation and construction [40, 87, 104], 2D or 3D exploration and mapping [64] or area coverage [137]. Moreover, the formation of patterns is common in many biological systems where agents, such as cells or microorganisms, form organized geometric structures, e.g., [124].

To achieve pattern formation we have to overcome the two main difficulties commonly found in swarm robotics, as discussed in Section 2.3. Firstly, as there are no leader agents, the pattern must emerge by exploiting a control strategy that is the same for all agents, *distributed* and *local* (i.e., each agent can only use information about “nearby” agents). Secondly, the number of agents is large and may change over time; therefore, the control strategy must also be *scalable* to varying sizes of the swarm and *robust* to uncertainties due to its possible variations. This sets the problem of achieving pattern formation apart from the more classical *for-*

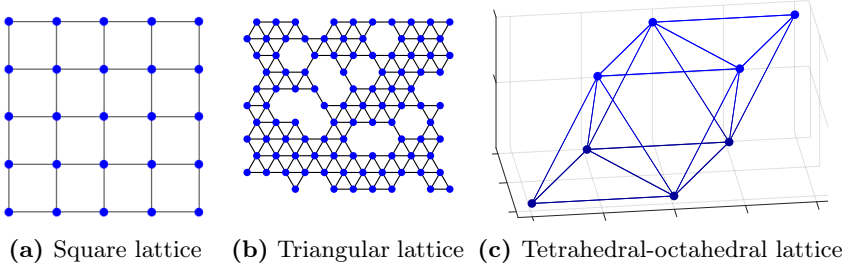


Figure 3.1: Examples of lattices in 2D (a-b) and 3D (c) spaces.

mation control problems [93] where agents are typically fewer and have pre-assigned roles within the formation. Moreover, geometric formations can also emerge as a by-product of *flocking* algorithms as those described in [95, 136], but, in such cases the focus of the control strategy is to achieve coordinated motion, rather than desired regular formations to emerge.

The next pages will give an overview of the existing Literature on geometric pattern formation, with a particular attention to control strategies based on the use of virtual forces, and then introduce some useful mathematical tools.

3.1 State of the art on geometric pattern formation

Most of the existing distributed control algorithms for geometric pattern formation rely on the use of *virtual forces* (or *virtual potentials*) [19, 86, 95, 108, 120, 125, 147]. Within this framework, first introduced for obstacle avoidance [66], agents move under the effect of virtual forces generated by the presence of their neighboring agents and the environment, causing attraction, repulsion, alignment, etc. In the most common case, where agents are homogeneous and exert radial forces depending on their relative distance, the resulting virtual force \mathbf{u}_i acting on agent i can be written as

$$\mathbf{u}_i(t) := \sum_{j \in \mathcal{L}_i(t)} f(\|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|) \frac{\mathbf{x}_i(t) - \mathbf{x}_j(t)}{\|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|}, \quad (3.1)$$

where $\mathcal{I}_i(t)$ is the set of agents i can interact with (usually depending on the relative distance), \mathbf{x}_i represent the position of the agent and $f : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ is the *virtual interaction function*, describing the influence of interacting agents. The most common interaction embeds short range repulsion, to avoid collisions, and long range attraction, to keep the swarm cohesive. This simple control law can then be modified or extended to obtain different behaviours (e.g. alignment, pattern formation, flocking, etc.). The main advantage of this approach is that it is inherently local and distributed, two key features of swarm robotics algorithms.

To classify existing solutions to pattern formation, we employ the same taxonomy used in [93], and later extended in [108], which is based on the type of information available to the agents. Namely, existing strategies can be classified as being (i) *position-based* when it is assumed agents know their position and orientation and those of their neighbours, in a global reference frame; (ii) *displacement-based* when agents can only sense their own orientation with respect to a global reference direction (e.g., North) and the relative positions of their neighbours; (iii) *distance-based* when agents can measure the relative positions of their neighbours with respect to their local reference frame. In terms of sensor requirements, position-based solutions are the most demanding, requiring global positioning sensors, typically GPS, and communication devices, such as WiFi or LoRa. Differently, displacement-based methods require only a distance sensor (e.g., LiDAR) and a compass, although the latter can be replaced by a coordinated initialisation procedure of all local reference frames [23]. Finally, distance-based algorithms are the least demanding, needing only the availability of some distance sensors.

3.1.1 Position-based approaches

In [99], a position-based algorithm was proposed to achieve 2D triangular lattices in a constellation of satellites in a 3D space. This strategy combines global attraction towards a reference point with local interaction among the agents to control both the global shape and the internal lattice structure of the swarm. In [19], a position-based approach was presented that combines the common radial virtual force (also used in [55, 120, 125]) with a normal force. In this way, a network of connections is built such that each agent has at least two neighbours. Importantly, this approach

requires the acquisition of positions from two-hop neighbours. In [147], a position-based strategy is presented to achieve triangular and square patterns, as well as lines and circles, both in 2D and 3D; the control strategy features global attraction towards a reference point and re-scaling of distances between neighbours, with the virtual forces changing according to the goal pattern. Therein, a qualitative comparison is also provided with the distance-based strategy from [120], showing more precise configurations and a shorter convergence time, due to the position-based nature of the solution. Finally, a simple position-based algorithm for triangular patterns, based on virtual forces and requiring communication between the agents, is proposed in [127] to have unmanned aerial vehicles perform area coverage.

3.1.2 Displacement-based approaches

In [75], a displacement-based approach is presented based on the use of a geometric control law similar to the one proposed in [71]. The aim is to obtain triangular lattices but small persisting oscillations of the agents are present at steady state, as the robots are assumed to have a constant non-zero speed. In [3, 4], an approach is discussed inspired by covalent bonds in crystals, where each agent has multiple attachment points for its neighbours. Only starting conditions close to the desired pattern are tested, as the focus is on navigation in environments with obstacles. In [116] the desired lattice is encoded by a graph, where the vertices denote possible *roles* the agents may play in the lattice and edges denote rigid transformations between the local frames or reference of pairs of neighbours. All agents communicate with each other and are assigned a label (or identification number) through which they are organised hierarchically to form triangular, square, hexagonal or octagon-square patterns. Formation control is similarly addressed in [22]. The algorithm proposed therein is made of a higher level policy to assign positions in a square lattice to the agents, and a lower level control, based on virtual forces, to have the agents reach these positions. The algorithm can be readily applied to the formation of square geometric patterns, but not to triangular ones. Notably, the reported convergence time is relatively long and increases with the number of agents. Finally, a solution to progressively deploy a swarm on a predetermined set of points is presented in [74]. The algorithm can

be used to perform both formation control and geometric lattice formation, even though the orientation of the formation cannot be controlled. Moreover, this strategy requires local communication between the agents and the knowledge of a common graph associated to the formation.

3.1.3 Distance-based approaches

A popular distance-based approach for the formation of triangular and square lattices, based on *gravitational* virtual forces, was proposed in [119] and later further investigated in [55, 120]. In these studies, triangular lattices are achieved with long-range attraction and short-range repulsion virtual forces only, while square lattices are obtained through a selective rescaling of the distances between some of the agents (see Section 4.3.4 for further details). The main drawback of the gravitational strategy [55, 106, 119, 120] is that it can produce the formation of multiple aggregations of agents, each respecting the desired pattern, but with different orientations. Another problem, described in [120], is that, for some values of the parameters, multiple agents can converge towards the same position and collide.

Similar approaches are also used to obtain triangular lattices in flocking algorithms [95, 136, 138], but such solutions mostly focus on the coordination of the motion. An extension of the gravitational strategy to achieve the formation of hexagonal lattices was proposed in [106], but with the requirement of an ad-hoc correction procedure to prevent agents from remaining stuck in the centre of a hexagon. In [125], an approach exploiting Lennard-Jones-like virtual forces is numerically optimised to locally stabilise a hexagonal lattice. This control law relies on stability conditions provided by *harmonic approximation* [57], but such conditions are only necessary, and not sufficient to prove the stability of the lattice configuration. Moreover, this approach requires time-varying control gains and synchronous clocks among the agents. A different strategy, based on geometric arguments, was proposed in [71]. It allows to build triangular lattices and, notably, its convergence was proved exploiting the Lyapunov method. A 3D extension was later presented in [72].

3.1.4 Open problems

From this analysis emerges that many existing strategies are limited to 2D domains and can only achieve triangular patterns (see Figure 3.1b) [99, 127]. More flexible solutions either show good performance but require expensive sensors and communication devices [74, 116, 147], or have lesser requirements but result in more poor performance [4, 22, 120]. Moreover, validation is mainly achieved numerically or experimentally [3, 4, 19, 75, 116, 120, 147], with only few strategies supported by formal proofs of convergence. Therefore, two pressing open challenges in geometric pattern formation are (i) the design of local and distributed control strategies that can combine low sensor requirements with flexibility and consistently high performance, and (ii) the formal proof of convergence for such algorithms.

We tackled both this challenges. Chapter 4 will present the distributed control algorithm we developed to address the formation of both triangular and square lattices, while our formal study on the stability of specific lattice configurations is presented in Chapter 5.

3.2 Mathematical preliminaries

Here we introduce some notation and mathematical tools that will be useful in Chapters 4 and 5.

Notation Given a vector $\mathbf{v} \in \mathbb{R}^d$, $[\mathbf{v}]_i$ is its i -th element, $\|\mathbf{v}\|$ its Euclidean norm, and $\hat{\mathbf{v}} := \frac{\mathbf{v}}{\|\mathbf{v}\|}$ its direction. $\mathbf{0}$ denotes a column vector of appropriate dimension with all elements equal to 0. Given a matrix \mathbf{A} , $[\mathbf{A}]_{ij}$ is its (i, j) -th element. Given a set \mathcal{B} , its cardinality is denoted by $|\mathcal{B}|$. Finally, we refer to \mathbb{R}^2 as the *plane*.

3.2.1 Dynamical systems

We start by recalling the concepts of equilibrium set and stability. Given a continuous-time, autonomous dynamical system

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)), \quad \mathbf{x}(0) = \mathbf{x}_0, \quad (3.2)$$

with state vector $\mathbf{x}(t) \in \mathbb{R}^d$, and $\mathbf{x}_0 \in \mathbb{R}^d$, we term as $\phi(t, \mathbf{x}_0)$ its trajectory starting from $\mathbf{x}(0) = \mathbf{x}_0$.

Definition 3.1 (Equilibrium set). *A set $\Xi \subset \mathbb{R}^d$ is an equilibrium set for system (3.2) if $\mathbf{f}(\mathbf{x}) = \mathbf{0} \forall \mathbf{x} \in \Xi$.*

Definition 3.2 (Local asymptotic stability [68, Definition 1.8]). *An equilibrium set Ξ for system (3.2) is locally asymptotically stable if $\forall \epsilon > 0, \exists \delta > 0$ such that if $\min_{\mathbf{y} \in \Xi} \|\mathbf{x}_0 - \mathbf{y}\| < \delta$, then*

1. $\min_{\mathbf{y} \in \Xi} \|\phi(t, \mathbf{x}_0) - \mathbf{y}\| < \epsilon, \forall t > 0$, and
2. $\lim_{t \rightarrow +\infty} \phi(t, \mathbf{x}_0) \in \Xi$.

3.2.2 Frameworks and rigidity

Next let us introduce the concept of *framework*, useful to add a spatial dimension to the mathematical concept of *graph*.

Definition 3.3 (Incidence matrix). *Given a digraph with N vertices and m edges, its incidence matrix $\mathbf{B} \in \mathbb{R}^{N \times m}$ has elements defined as*

$$[\mathbf{B}]_{ij} := \begin{cases} +1, & \text{if edge } j \text{ starts from vertex } i, \\ -1, & \text{if edge } j \text{ ends in vertex } i, \\ 0, & \text{otherwise.} \end{cases}$$

Definition 3.4 (Framework [86, p. 120]). *Consider a (di-)graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with N vertices, and a set of positions $\mathbf{p}_1, \dots, \mathbf{p}_n \in \mathbb{R}^d$ associated to its vertices, with $\mathbf{p}_i \neq \mathbf{p}_j \forall i, j \in \{1, \dots, N\}$. A d -dimensional framework is the pair $(\mathcal{G}, \bar{\mathbf{p}})$, where $\bar{\mathbf{p}} := [\mathbf{p}_1^\top \cdots \mathbf{p}_n^\top]^\top \in \mathbb{R}^{dN}$. Moreover, the length of an edge, say $(i, j) \in \mathcal{E}$, is $\|\mathbf{p}_i - \mathbf{p}_j\|$.*

Definition 3.5 (Congruent frameworks [61, p. 3]). *Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and two frameworks $(\mathcal{G}, \bar{\mathbf{p}})$ and $(\mathcal{G}, \bar{\mathbf{q}})$, these are congruent if $\|\mathbf{p}_i - \mathbf{p}_j\| = \|\mathbf{q}_i - \mathbf{q}_j\| \forall i, j \in \mathcal{V}$.*

Definition 3.6 (Rigidity matrix [61, p. 5]). *Given a d -dimensional framework with $N \geq 2$ vertices and m edges, its rigidity matrix $\mathbf{M} \in \mathbb{R}^{m \times dN}$*

has elements defined as

$$[\mathbf{M}]_{e,(jd-d+k)} := \begin{cases} [\mathbf{p}_j - \mathbf{p}_i]_k, & \text{if edge } e \text{ goes from vertex } i \text{ to vertex } j, \\ [\mathbf{p}_i - \mathbf{p}_j]_k, & \text{if edge } e \text{ goes from vertex } j \text{ to vertex } i, \\ 0, & \text{otherwise.} \end{cases}$$

with $k \in \{1, \dots, d\}$.

Definition 3.7 (Infinitesimal rigidity [86, p. 122]). *A framework with rigidity matrix \mathbf{M} is infinitesimally rigid if, for any infinitesimal motion, say \mathbf{u} ,¹ of its vertices, such that the length of the edges is preserved, it holds that $\mathbf{M}\mathbf{u} = \mathbf{0}$.*

To give a geometrical intuition of the concept of infinitesimal rigidity, we note that an infinitesimally rigid framework is also rigid [86, p. 122], according to the definition below.²

Definition 3.8 (Rigidity [61, p. 3]). *A framework is rigid if every continuous motion of the vertices, that preserves the length of the edges, also preserves the distances between all pairs of vertices.*

Consequently, in a rigid framework, a continuous motion that does *not* preserve the distance between any two vertices also does *not* preserve the length of at least one edge.

To more easily assess the infinitesimal rigidity of a framework, it is possible to use the following result.

Theorem 3.1 ([53, Theorem 2.2]). *A d -dimensional framework with $N \geq d$ vertices and rigidity matrix \mathbf{M} is infinitesimally rigid if and only if $\text{rank}(\mathbf{M}) = dN - d(d+1)/2$.*

3.2.3 Swarms of mobile agents

Now, let us formally define a *swarm* as a set of $N \in \mathbb{N}_{>0}$ identical agents, say $\mathcal{S} := \{1, 2, \dots, N\}$, that can move in \mathbb{R}^d and interact with

¹ \mathbf{u} can be interpreted as either a velocity or a small displacement.

²Rarely, a rigid framework is not infinitesimally rigid; e.g. [61, p. 7].

their neighbors to generate emergent behavior [13]. For each agent $i \in \mathcal{S}$, $\mathbf{x}_i(t) \in \mathbb{R}^d$ denotes its position at time $t \in \mathbb{R}_{\geq 0}$. Moreover, we call $\bar{\mathbf{x}}(t) := [\mathbf{x}_1^\top(t) \cdots \mathbf{x}_n^\top(t)]^\top \in \mathbb{R}^{dN}$ the *configuration* of the swarm, define $\mathbf{x}_c(t) := \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i(t) \in \mathbb{R}^d$ as its *center*, and denote by $\mathbf{r}_{ij}(t) := \mathbf{x}_i(t) - \mathbf{x}_j(t) \in \mathbb{R}^d$ the relative position of agent i with respect to agent j . Finally, for planar swarms ($d = 2$) we define $\theta_{ij}(t) \in [0, 2\pi]$ as the angle between \mathbf{r}_{ij} and the horizontal axis, and given any two pairs of agents, (i, j) and (h, k) , we denote with $\theta_{ij}^{hk}(t) \in [0, 2\pi]$ the absolute value of the angle between the vectors \mathbf{r}_{ij} and \mathbf{r}_{hk} .

Definition 3.9 (Interaction set). *Given a swarm \mathcal{S} and a sensing radius $R_s \in \mathbb{R}_{>0}$, the interaction set of agent i at time t is $\mathcal{I}_i(t) := \{j \in \mathcal{S} \setminus \{i\} : \|\mathbf{r}_{ij}(t)\| \leq R_s\}$.*

Definition 3.10 (Adjacency set). *Given a swarm \mathcal{S} and some finite $R_{\min}, R_{\max} \in \mathbb{R}_{>0}$, with $R_{\min} \leq R_{\max}$, the adjacency set of agent i at time t is (see Figure 4.1)*

$$\mathcal{A}_i(t) := \{j \in \mathcal{S} \setminus \{i\} : R_{\min} \leq \|\mathbf{r}_{ij}(t)\| \leq R_{\max}\}. \quad (3.3)$$

Definition 3.11 (Links). *A link is a pair $(i, j) \in \mathcal{S} \times \mathcal{S}$ such that $j \in \mathcal{A}_i(t)$, and $\|\mathbf{r}_{ij}(t)\|$ is its length. The set of all links existing in a certain configuration $\bar{\mathbf{x}}$ is denoted by $\mathcal{E}(\bar{\mathbf{x}})$.*

We will say that two agents are connected if and only if $(i, j) \in \mathcal{E}(\bar{\mathbf{x}})$, that also implies $(j, i) \in \mathcal{E}(\bar{\mathbf{x}})$. Moreover, notice that if $R_{\max} \leq R_s$ then $\mathcal{A}_i \subseteq \mathcal{I}_i$.

Definition 3.12 (Swarm graph and framework). *The swarm graph is the digraph $\mathcal{G}(\bar{\mathbf{x}}) := (\mathcal{S}, \mathcal{E}(\bar{\mathbf{x}}))$, whose vertices correspond to the agents in the swarm and whose edges correspond to the links³. The swarm framework is $\mathcal{F}(\bar{\mathbf{x}}) := (\mathcal{G}(\bar{\mathbf{x}}), \bar{\mathbf{x}})$.*

Definition 3.13 (Congruent configurations). *Given a configuration $\bar{\mathbf{x}}^\diamond$, we define the set of its congruent configurations $\Gamma(\bar{\mathbf{x}}^\diamond)$ as the set of configurations with congruent associated frameworks (see Definition 3.5), that is $\Gamma(\bar{\mathbf{x}}^\diamond) := \{\bar{\mathbf{x}} \in \mathbb{R}^{dN} : \|\mathbf{x}_i - \mathbf{x}_j\| = \|\mathbf{x}_i^\diamond - \mathbf{x}_j^\diamond\|, \forall i, j \in \mathcal{S}\}$.*

³Formally, $\mathcal{G}(\bar{\mathbf{x}})$ is a directed graph, even though $\mathcal{E}(\bar{\mathbf{x}})$ is such that the existence of link (i, j) implies the existence of link (j, i) .

These configurations are obtained by translations and rotations of the framework $\mathcal{F}(\bar{\mathbf{x}}^\diamond)$; thus, it is immediate to verify that $\Gamma(\bar{\mathbf{x}}^\diamond)$ is connected and unbounded for any $\bar{\mathbf{x}}^\diamond$.

3.3 Discussion

In this Chapter we introduced the problem of geometric pattern formation, discussing the possible applications and the main challenges. We reviewed the existing solutions, classified according to the sensory requirements, and the open challenges, mainly the design of flexible control algorithms and their analytical validation. Finally, we introduced some notation and the mathematical tools, used in Chapters 4 and 5 to address, respectively, the problems of designing and analysing solutions for the formation of geometric patterns.

Distributed control for geometric pattern formation

This Chapter follows the contents and the structure of our work [45], where we introduced a distributed displacement-based control law that allows large groups of agents to achieve triangular and square lattices in 2D, with low sensor requirements and without needing communication between the agents. Also, a simple, yet powerful, adaptation law is proposed to automatically tune the control gains in order to reduce the design effort, while improving robustness and flexibility. We show the validity and robustness of our approach via numerical simulations and experiments, comparing it, where possible, with other approaches from the existing literature.

4.1 Control goal and performance metrics

To describe the planar lattices studied in this Chapter let us introduce the following definition.

Definition 4.1 ((L, R) -Lattice). *Given some $L \in \{4, 6\}$ and $R \in \mathbb{R}_{>0}$, a (L, R) -lattice is a set of points in the plane that coincide with the vertices of an associated regular tiling [34]; R is the distance between adjacent vertices and L is the number of adjacent vertices each point has.*

In Definition 4.1, $L = 4$, and $L = 6$ correspond to square and triangular

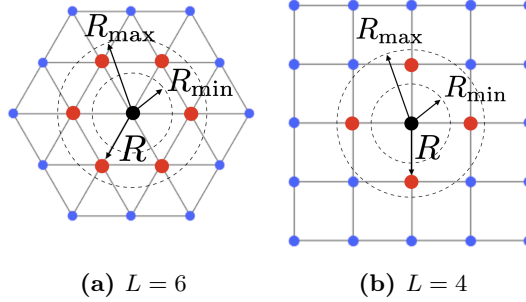


Figure 4.1: (L, R) -lattice formations: triangular ($L = 6$) (a) and square ($L = 4$) (b). Red dots are agents in the adjacency set (\mathcal{A}_i) of the generic agent i depicted as a black dot.

lattices,¹ respectively, as portrayed in Figure 4.1. Let us recall the concept of *swarm* as introduced in Section 3.2.3, we will say it *self-organises into a (L, R) -lattice* if (i) each agent has at most L links (see Definition 3.11), and (ii) given any two links (i, j) and (h, k) in \mathcal{E} it holds that θ_{ij}^{hk} (i.e. the angle between the two as defined in Section 3.2.3) is some multiple of $2\pi/L$. Therefore, to assess whether a swarm self-organises into some desired (L, R) -lattice, we introduce the following two metrics.

Definition 4.2 (Regularity metric). *Given a swarm and a desired (L, R) -lattice, the regularity metric $e_\theta(t) \in [0, 1]$ is*

$$e_\theta(t) := \frac{L}{\pi} \cdot \theta_{\text{err}}(t), \quad (4.1)$$

where, omitting the dependence on time,

$$\theta_{\text{err}} := \frac{1}{|\mathcal{E}|^2 - 2|\mathcal{E}|} \sum_{(i,j) \in \mathcal{E}} \sum_{(h,k) \in \mathcal{E}} \min_{q \in \mathbb{Z}} \left| \theta_{ij}^{hk} - q \frac{2\pi}{L} \right|. \quad (4.2)$$

¹ Regular tilings exist only for $L \in \{3, 4, 6\}$. The case $L = 2$ corresponds more trivially to a line, rather than a planar structure. The case $L = 3$, corresponding to hexagonal tilings, where vertices appear in two different spatial configurations (one with edges at angles $\pi/2, 7/6\pi, 11/6\pi$, and one with edges at angles $\pi/6, 5/6\pi, 3/2\pi$ —plus an optional offset). Hence, the control strategy we propose here would need to be extended to select one or the other of the possible hexagonal configurations.

The regularity metric e_θ , derived from [120], quantifies the incoherence in the orientation of the links in the swarm. In particular, $e_\theta = 0$ when all the pairs of links form angles that are multiples of $2\pi/L$ (which is desirable to achieve the (L, R) -lattice), while $e_\theta = 1$ when all pairs of links have the maximum possible orientation error, equal to π/L . ($e_\theta \approx 0.5$ generally corresponds to the agents being arranged randomly.)

Definition 4.3 (Compactness metric). *Given a swarm of N agents and a desired (L, R) -lattice, the compactness metric $e_L(t) \in [0, (N - 1 - L)/L]$ is*

$$e_L(t) := \frac{1}{N} \sum_{i=1}^N \frac{||\mathcal{A}_i(t)| - L|}{L}, \quad (4.3)$$

where $\mathcal{A}_i(t)$ is the adjacency set of agent i , as given in Definition 3.10.

The compactness metric e_L measures the average difference between the number of neighbours each agent has and the one they are ought to have if they were arranged in a (L, R) -lattice. According to this definition, e_L reaches its maximum value, $e_{L,\max} = (N - 1 - L)/L$, when all agents are concentrated in a small region, and links exist between all pairs of agents, while $e_L = 1$ when all the agents are scattered loosely in the plane, and no links exist between them, finally, $e_L = 0$ when all the agents have L links (typically we will require that e_L is below some acceptable threshold, see Section 4.3.1). It is important to remark that, if the number N of agents is finite, e_L can never be equal to zero, because the agents on the boundary of the group will always have less than L links (see Figure 4.1), but this effect gets less relevant as N increases. Note that a similar metric was also independently defined in [116]. We remark that the compactness metric inherently penalizes the presence of holes in the configuration and the emergence of detached swarms, as those scenarios are characterized by larger boundaries.

For the sake of brevity, in what follows we will omit dependence on time when that is clear from the context.

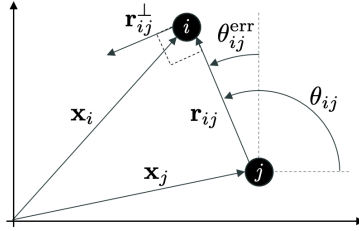


Figure 4.2: Schematic diagram of two agents, i and j , showing the key variables describing the agents' position and their geometrical relationship.

4.2 Design of the control law

4.2.1 Problem formulation

Consider a planar swarm \mathcal{S} whose agents' dynamics is described by the first order model

$$\dot{\mathbf{x}}_i(t) = \mathbf{u}_i(t), \quad \forall i \in \mathcal{S}, \quad (4.4)$$

where $\mathbf{x}_i(t) \in \mathbb{R}^2$ represent the position of agent i , as described in Section 3.2.3 and depicted in Figure 4.2, and $\mathbf{u}_i(t) \in \mathbb{R}^2$ is the input signal determining its velocity.

Remark 4.1. *First order models like (4.4) are often used in the literature [19, 71, 72, 147]. In some other works [106, 120] a second order model is used, given by $m\ddot{\mathbf{x}}_i + \mu\dot{\mathbf{x}}_i = \mathbf{u}_i$, where \mathbf{u}_i is a force, m is a mass and μ is a viscous friction coefficient. Under the simplifying assumptions of small inertia ($m\|\dot{\mathbf{v}}_i\| \ll \mu\|\mathbf{v}_i\|$) and $\mu = 1$, the two models coincide.*

We want to design a *distributed* feedback control law $\mathbf{u}_i = \mathbf{g}(\{\mathbf{r}_{ij}\}_{j \in \mathcal{I}_i}, L, R)$ to let the swarm self-organise into a desired triangular or square lattice, starting from any set of initial positions in some disk of radius r , while guaranteeing the control strategy to be:

1. *robust* to failures of agents and to noise;
2. *flexible*, allowing dynamic reorganisation of the agents into different patterns;

3. *scalable*, allowing the number of agents N to change dynamically.

We will assess the effectiveness of the proposed strategy by using the performance metrics e_θ and e_L introduced above (see Definitions 4.2 and 4.3).

4.2.2 Distributed control law

To solve this problem we propose a distributed displacement-based control law of the form

$$\mathbf{u}_i(t) = \mathbf{u}_{r,i}(t) + \mathbf{u}_{n,i}(t), \quad (4.5)$$

where $\mathbf{u}_{r,i}$ and $\mathbf{u}_{n,i}$ are the *radial* and *normal* control inputs, respectively. The two inputs have different purposes and each comprises several *virtual forces*. The radial input $\mathbf{u}_{r,i}$ is the sum of attracting/repelling actions between the agents, with the purpose of aggregating them into a compact swarm, while avoiding collisions and keeping the desired inter-agent distance. The normal input $\mathbf{u}_{n,i}$ is also the sum of multiple actions, used to adjust the angles of the relative positions of the agents.

Note that the control strategy in (4.5) is *displacement-based* because it only requires each agent i (i) to be able to measure the relative positions of the agents close to it (in the sets \mathcal{I}_i and \mathcal{A}_i , given in Definitions 3.9 and 3.10), and (ii) to possess knowledge of a common reference direction. Next, we describe in detail each of the two control actions in (4.5).

Radial Interaction

The radial control input $\mathbf{u}_{r,i}$ in (4.5) is defined as the sum of several virtual forces, one for each agent in \mathcal{I}_i (neighbours of i), each force being attractive (if the neighbour is far) or repulsive (if the neighbour is close). Specifically, we set

$$\mathbf{u}_{r,i} = G_{r,i} \sum_{j \in \mathcal{I}_i} f_r(\|\mathbf{r}_{ij}\|) \frac{\mathbf{r}_{ij}}{\|\mathbf{r}_{ij}\|}, \quad (4.6)$$

where $G_{r,i} \in \mathbb{R}_{\geq 0}$ is the radial control gain. Note that $\mathbf{u}_{r,i}$ is termed as *radial* input because in (4.6) the attraction/repulsion forces are parallel

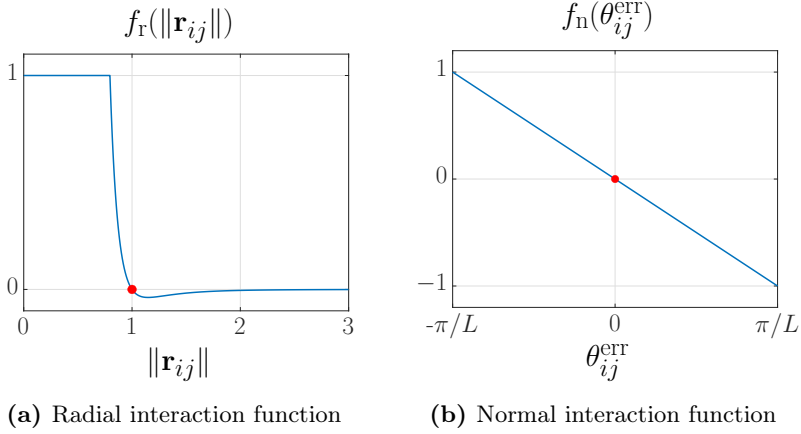


Figure 4.3: Interaction functions. Red dots highlight zeros of the functions. Parameters are taken from Tab 4.2.

to the vectors \mathbf{r}_{ij} (see Figure 4.2). The magnitude and sign of each of these forces depend on the distance, $\|\mathbf{r}_{ij}\|$, between the agents, according to the *radial interaction function* $f_r : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$. Here, we select f_r as the Physics-inspired Lennard-Jones function [13], given by

$$f_r(\|\mathbf{r}_{ij}\|) = \min \left\{ \left(\frac{a}{\|\mathbf{r}_{ij}\|^{2c}} - \frac{b}{\|\mathbf{r}_{ij}\|^c} \right), 1 \right\}, \quad (4.7)$$

where $a, b \in \mathbb{R}_{>0}$ and $c \in \mathbb{N}$ are design parameters. In (4.7), f_r is saturated to 1 to avoid divergence for $\|\mathbf{r}_{ij}\| \rightarrow 0$. f_r is portrayed in Figure 4.3a

Normal Interaction

For any link (i, j) , we define the *angular error* $\theta_{ij}^{\text{err}} \in]-\frac{\pi}{L}, \frac{\pi}{L}]$ as the difference between θ_{ij} , introduced in Section 3.2.3, and the closest multiple of $2\pi/L$ (see Figure 4.2), that is,

$$\theta_{ij}^{\text{err}} := \theta_{ij} - \frac{2\pi}{L} \arg \min_{q \in \mathbb{Z}} \left\{ \left| \theta_{ij} - q \frac{2\pi}{L} \right| \right\}, \quad (4.8)$$

Then, the normal control input $\mathbf{u}_{n,i}$ in (4.5) is chosen as

$$\mathbf{u}_{n,i} = G_{n,i} \sum_{j \in \mathcal{A}_i} f_n(\theta_{ij}^{\text{err}}) \frac{\mathbf{r}_{ij}^\perp}{\|\mathbf{r}_{ij}\|}, \quad (4.9)$$

where $G_{n,i} \in \mathbb{R}_{\geq 0}$ is the normal control gain. Note that each of the normal vector forces is applied in the direction of \mathbf{r}_{ij}^\perp , that is the vector normal to \mathbf{r}_{ij} , obtained by applying a $\pi/2$ counterclockwise rotation (see Figure 4.2). The magnitude and sign of these forces are determined by the *normal interaction function* $f_n :]-\frac{\pi}{L}, \frac{\pi}{L}] \rightarrow [-1, 1[$, given by

$$f_n(\theta_{ij}^{\text{err}}) = -\frac{L}{\pi} \theta_{ij}^{\text{err}}. \quad (4.10)$$

f_n is portrayed in Figure 4.3b.

4.3 Numerical validation

In this section, we assess the performance and the robustness of our proposed control algorithm (4.5) through an extensive simulation campaign. The experimental validation of the strategy is later reported in Section 4.5. First in Section 4.3.2, using a numerical optimisation procedure, we tune the control gains $G_{r,i}$ and $G_{n,i}$ in (4.6) and (4.9), as the performance of the controlled swarm strongly depends on these values. Then in Section 4.3.3, we assess the robustness of the control law with respect to (i) agents' failure, (ii) noise, (iii) flexibility to pattern changes, and (iv) scalability. Finally in Section 4.3.4, we present a comparative analysis of our distributed control strategy and other approaches previously presented in the literature. The simulations and experiments performed in this and the next Sections are summarised in Table 4.1

4.3.1 Simulation setup

We consider a swarm consisting of $N = 100$ agents (unless specified differently). To represent the fact that the agents are deployed from a unique source (as typically done in the literature, see e.g., [120]), their initial positions are drawn randomly with uniform distribution from a disk

Scenario	Section	Figure
<i>Control law (4.5),(4.6),(4.9)</i>		
Tuning	4.3.2	4.4
Validation	4.3.2	4.5
Robustness to faults	4.3.3	4.6
Robustness to noise	4.3.3	4.7
Flexibility	4.3.3	4.8
Scalability	4.3.3	4.9
Comparison with established algorithm	4.3.4	4.10
<i>Adaptive gain tuning (4.23)</i>		
Validation	4.4	4.11
Robustness to faults	4.4.1	4.12
Flexibility	4.4.1	4.13
Scalability	4.4.1	4.14
Robotarium experiment	4.5	4.15

Table 4.1: List of simulations and experiments with indication of the corresponding section and figures.

of radius $r = 2$ centred at the origin.^{2, 3}

Initially, for the sake of simplicity and to avoid the possibility of some agents becoming disconnected from the group, we assume that the sensing radius R_s in Definition 3.9 is large enough so that

$$\forall i \in \mathcal{S}, \forall t \in \mathbb{R}_{\geq 0}, \quad \mathcal{I}_i(t) = \mathcal{S} \setminus i; \quad (4.11)$$

i.e., any agent can sense the relative position of all others. Later, in Section 4.3.3, we will drop this assumption and show the validity of our control strategy also for smaller values of R_s . All simulation trials are con-

²That is, denoting with $U([a, b])$ the uniform distribution on the interval $[a, b]$, the initial position of each agent in polar coordinates $\mathbf{x}_i(0) := (d_i, \phi_i)$ is obtained by independently sampling $\phi_i \sim U([0, 2\pi])$ and d_i is chosen according to the probability density function $p_l(\xi) : [0, r] \mapsto \mathbb{R}_{\geq 0}$ defined as $p_l(\xi) = 2\xi/r^2$.

³We also considered different deployment strategies (e.g., agents starting uniformly distributed from a larger disk or several disjoint disks) and verified that the results are qualitatively similar.

Parameter	Description	Value
R	Desired link length	1 m
R_{\min}	Minimum link length	0.6 m
R_{\max}	Maximum link length	1.1 m
V_{\max}	Maximum speed	5 m/s
t_{\max}	Maximum simulation time	200 s
Δt	Integration step	0.01 s
T_w	Time window	10 s
a	Radial interaction function $f_r(\cdot)$	0.15
b	"	0.15
c	"	5

Table 4.2: Simulation parameters

ducted in MATLAB using the agent based simulator we developed Swarm-Sim V1 (for more information see Appendix A or visit <https://github.com/diBernardoGroup/SwarmSimPublic/tree/SwarmSimV1>). The agents' speed is limited to $V_{\max} > 0$ and their dynamics is integrated using the forward Euler method with a fixed time step $\Delta t > 0$, The values of the parameters used in the simulations are reported in Table 4.2.

Performance evaluation

To assess the performance of the controlled swarm, we exploit the metrics e_θ and e_L given in Definitions 4.2 and 4.3. Namely, we select empirically the thresholds $e_\theta^* = 0.2$ and $e_L^* = 0.3$, which are associated to satisfactory compactness and regularity of the swarm. Then, letting $T_w > 0$ be the length of a time window, we say that e_θ is at *steady-state* from time $t' = k\Delta t$ (for $k \in \mathbb{Z}$) if

$$|e_\theta(t') - e_\theta(t' - j\Delta t)| \leq 0.1 e_\theta^*, \quad \forall j \in \left\{1, 2, \dots, \left\lfloor \frac{T_w}{\Delta t} \right\rfloor\right\}. \quad (4.12)$$

We give an analogous definition for the steady state of e_L (using e_L^* rather than e_θ^*). Then, we say that in a trial the swarm *achieved steady-state* at time t_{ss} if there exists a time instant such that both e_θ and e_L are at steady

state, and t_{ss} is the smallest of such time instants. Moreover, we deem the trial *successful* if $e_\theta(t_{\text{ss}}) < e_\theta^*$ and $e_L(t_{\text{ss}}) < e_L^*$. If in a trial steady-state is not reached in the time interval $[0, t_{\text{max}}]$, the trial is stopped (and deemed unsuccessful). We define

$$e_\theta^{\text{ss}} := \begin{cases} e_\theta(t_{\text{ss}}), & \text{if steady state is achieved,} \\ e_\theta(t_{\text{max}}), & \text{otherwise.} \end{cases} \quad (4.13)$$

$$e_L^{\text{ss}} := \begin{cases} e_L(t_{\text{ss}}), & \text{if steady state is achieved,} \\ e_L(t_{\text{max}}), & \text{otherwise.} \end{cases} \quad (4.14)$$

Finally, to assess how quickly the pattern is formed, we define

$$T_\theta := \min\{t' \in \mathbb{R}_{\geq 0} : e_\theta(t') \leq e_\theta^*, \forall t \geq t'\}, \quad (4.15)$$

$$T_L := \min\{t'' \in \mathbb{R}_{\geq 0} : e_L(t'') \leq e_L^*, \forall t \geq t''\}, \quad (4.16)$$

$$T := \max\{T_\theta, T_L\}. \quad (4.17)$$

4.3.2 Tuning of the control gains

For the sake of simplicity, in this section we assume that $G_{r,i} = G_r$ and $G_{n,i} = G_n$, for all $i \in \mathcal{S}$; later, in Section 4.4, we will present an adaptive control strategy allowing each agent to independently vary online its own control gains. To select the values of G_r and G_n giving the best performance in terms of regularity and compactness, we conducted an extensive simulation campaign and evaluated. Specifically, for each pair $(G_r, G_n) \in \{0, 1, \dots, 30\} \times \{0, 1, \dots, 30\}$, we executed 30 trials, starting from random initial conditions, and averaged the following cost function:

$$C(e_\theta^{\text{ss}}, e_L^{\text{ss}}) := \left(\frac{e_\theta^{\text{ss}}}{e_\theta^*}\right)^2 + \left(\frac{e_L^{\text{ss}}}{e_L^*}\right)^2. \quad (4.18)$$

The results are reported in Figure 4.4 for the triangular ($L = 6$) and the square ($L = 4$) lattices; in the former case, the pair $(G_r^*, G_n^*)_{L=4}$ minimising C is $(22, 1)$, whereas in the latter case it is $(G_r^*, G_n^*)_{L=6} = (15, 8)$. Both pairs achieve $C \leq 1$, implying $e_\theta^{\text{ss}} \leq e_\theta^*$ and $e_L^{\text{ss}} \leq e_L^*$.

In Figure 4.5, we report four snapshots, at different time instants, of two representative simulations, together with the metrics $e_\theta(t)$ and $e_L(t)$,

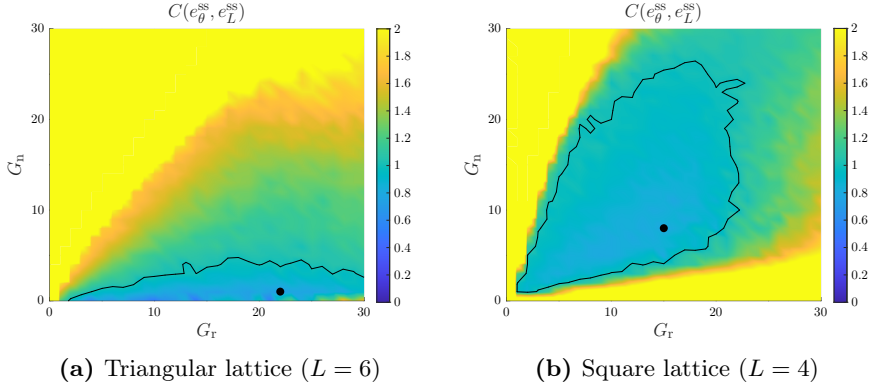


Figure 4.4: Tuning of the control gains G_r and G_n . The black dots correspond to $(G_r^*, G_n^*)_{L=6}$ and $(G_r^*, G_n^*)_{L=4}$, minimising the metric cost C defined in (4.18). The black curves delimit the regions where $C \leq 1$.

for the cases of a triangular and a square lattice, respectively. The control gains were set to the optimal values $(G_r^*, G_n^*)_{L=6}$ and $(G_r^*, G_n^*)_{L=4}$. In both cases, the metrics quickly converge below their prescribed thresholds, as $T < 2.75$ s. Moreover, note that $e_L(t)$ decreases faster than $e_\theta(t)$, meaning that the swarm tends to first reach the desired level of compactness and then agents' positions are rearranged to achieve the desired pattern. Finally, we note, and it is immediate to verify, that it is possible to control the orientation of the resulting lattice simply by applying a uniform offset to the agents' compasses.

4.3.3 Robustness analysis

In this section, we investigate numerically the properties that we required in Section 4.2.1, that is robustness to faults and noise, flexibility, and scalability.

Robustness to faults

To analyse the robustness of the controlled swarm to agents' faults, we ran a series of simulations in which we removed a percentage of the agents at a certain time instant, and assessed the capability of the swarm to recover the desired pattern. For the sake of brevity, we report only one

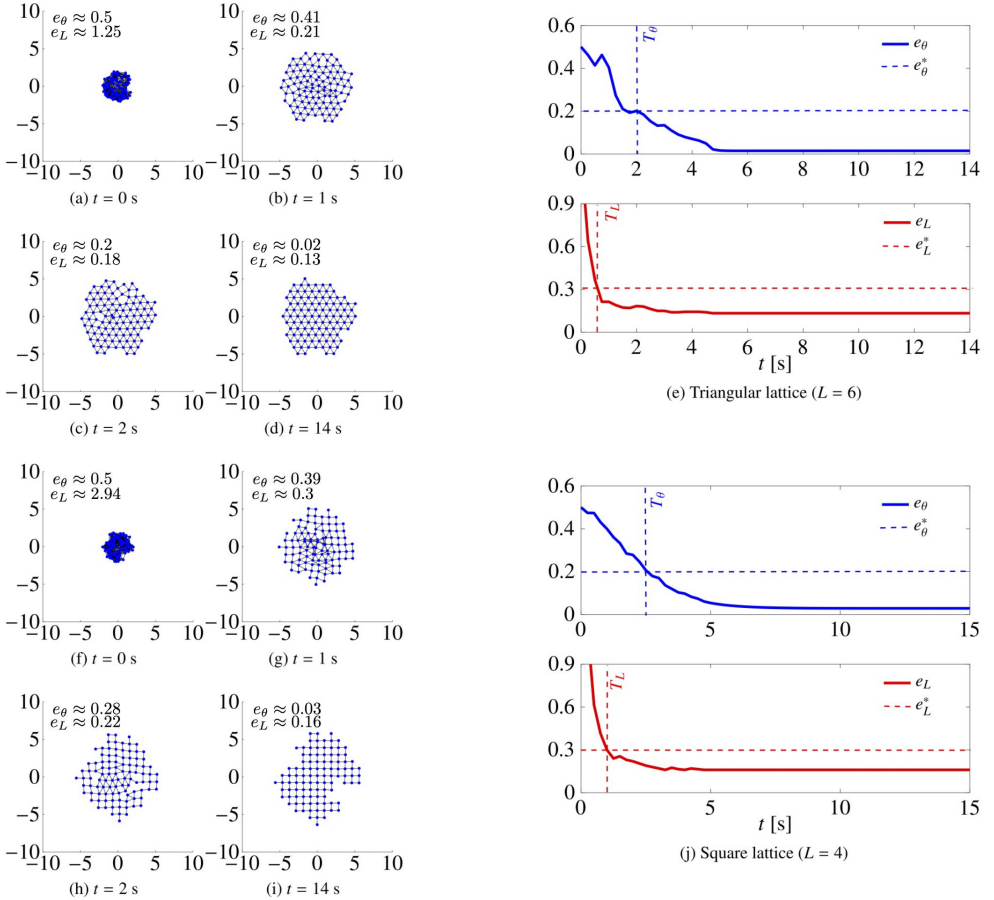


Figure 4.5: Snapshots at different time instants of a swarm of $N = 100$ agents being controlled to form a triangular lattice (a-d) and a square lattice (f-i). For each snapshot, we also report the values of e_θ and e_L . (e) and (j) show the time evolution of the metrics e_θ and e_L for $L = 6$ and $L = 4$, respectively. When $L = 6$, we set $(G_r, G_n) = (G_r^*, G_n^*)_{L=6}$; when $L = 4$, we set $(G_r, G_n) = (G_r^*, G_n^*)_{L=4}$. (See 4.3.2 for details on how the gains were tuned.)

of them as a representative example in Figure 4.6, where, with $L = 4$, 30% of the agents were removed at random at time $t = 30$ s. We notice that, as the agents are removed, $e_L(t)$ and $e_\theta(t)$ suddenly increase, but, after a short time, they converge again to values below the thresholds,

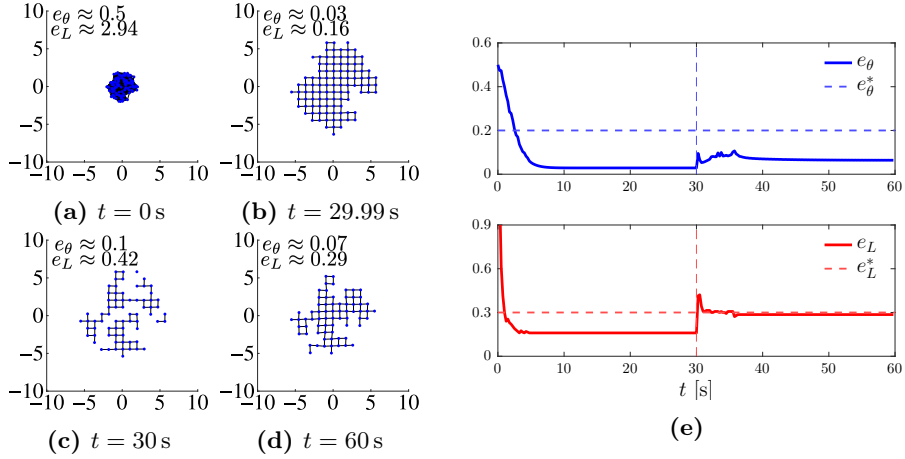


Figure 4.6: Robustness to agents' removal. (a-d) Snapshots at different time instants of a swarm achieving square lattice. Initially, there are $N = 100$ agents with 30 agents being removed at $t = 30$ s. (e) Time evolution of the metrics; dashed vertical lines denote the time instant when agents are removed. Here $L = 4$, and $(G_r, G_n) = (G_r^*, G_n^*)_{L=4}$.

recovering the desired pattern, despite the formation of small holes in the pattern at steady-state that increase e_L^{ss} . Finally, we also considered the case where the faulty agents stay still in their positions after the fault, with other agents having to form the lattice around them. We observed that when the fault takes place after a satisfying structure is formed, the metrics are not affected by the event (the numerical results are omitted here as redundant).

Robustness to noise

We assessed the robustness to noise both on actuation and on sensing, in two separate tests. In the first case, we assumed that the dynamics (4.4) of each agent is affected by additive white Gaussian noise with standard deviation σ_a . In the second case, we assumed that, for each agent, both the distance measurements $\|\mathbf{r}_{ij}\|$ in (4.6) and the angular measurements θ_{ij}^{err} in (4.9) are affected by additive white Gaussian noise (i.i.d. for each i and j) with standard deviation σ_m and $\sigma_m \frac{\pi}{L}$, respectively.

In particular, we set $L = 4$ and varied either σ_a or σ_m in intervals of

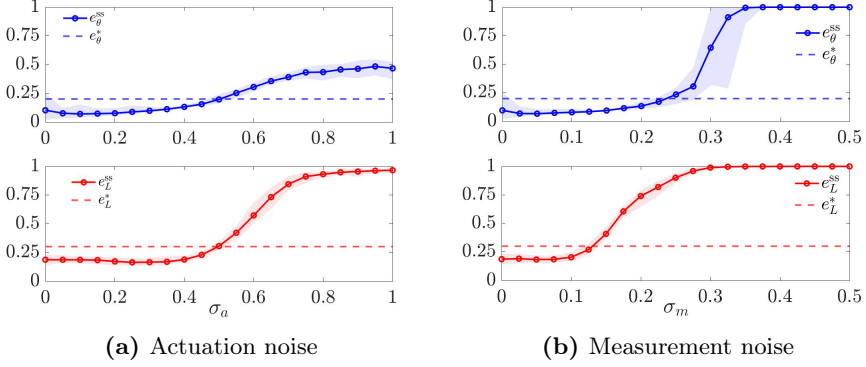


Figure 4.7: Robustness to noise. Value of the metrics e_{θ}^{ss} and e_L^{ss} , averaged over $M = 30$ trials, when (a) the intensity σ_a of the actuation noise is varied and (b) the intensity σ_m of the measurement noise is varied. The shaded areas represent the maximum and minimum values obtained over the M trials. Here $L = 4$, and $(G_r, G_n) = (G_r^*, G_n^*)_{L=4}$.

interest with small increments. For each value of σ_a and σ_m , we ran $M = 30$ trials, starting from random initial conditions, and reported the average values of e_{θ}^{ss} and e_L^{ss} in Figure 4.7. We observe that, while in the ranges $\sigma_a \in [0, 0.45]$ and $\sigma_m \in [0, 0.125]$ the strategy guarantees robustness, for large enough noise ($\sigma_a \geq 0.45$ or $\sigma_m \geq 0.125$) performance is increasingly worsened with trials eventually becoming unsuccessful (the swarm never achieving the desired lattice configuration). Interestingly, we find that for smaller noise ($0 < \sigma_a \leq 0.2$ or $0 < \sigma_m \leq 0.1$) performance is actually improved, as small random inputs can prevent the agents from getting stuck in undesired configurations, including those containing holes.

We obtained qualitatively similar results when we assumed the presence of noise on the compass measurements of the agents (obtained by adding Gaussian noise on the variables θ_{ij}^{err} , with the noise value being the same for θ_{ij}^{err} and θ_{kl}^{err} when $i = k$).

Flexibility

In Figure 4.8, we report a simulation where L was initially set equal to 4 (square lattice), changed to 6 (triangular lattice) at time $t = 30$ s,

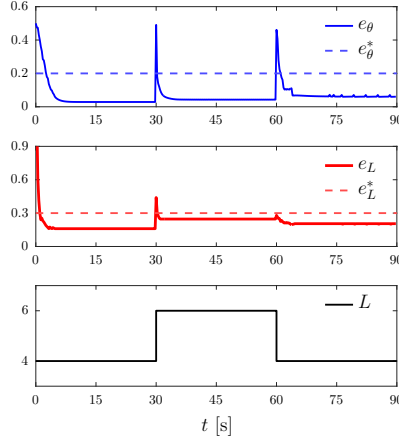


Figure 4.8: Flexibility to spatial reorganisation. Time evolution of the metrics e_θ^{ss} and e_L^{ss} as L changes as shown in the bottom panel. The gains are set as $(G_r, G_n) = (G_r^*, G_n^*)_{L=4}$.

and finally changed back to 4 at $t = 60$ s. The control gains are set to $(G_r^*, G_n^*)_{L=4}$ and kept constant during the entire the simulation. Clearly, as L is changed, both e_L and e_θ suddenly increase, but the swarm is quickly able to reorganise and reduce them below their prescribed thresholds in less than 5 s, thus achieving the desired patterns.

Scalability

We relaxed the assumption (4.11) and characterised e_L^{ss} as a function of the sensing radius R_s . The results are portrayed in Figure 4.9a, showing that the performance starts deteriorating for approximately $R_s < 6R$, until it becomes unacceptable for about $R_s < 1.1R$. Therefore, as a good trade-off between performance and feasibility, we set $R_s = 3R$. Then, to test for scalability, we varied the number N of agents (initially, $N = 100$), reporting the results in Figure 4.9b. We see that (i) the controlled swarm correctly achieves the desired pattern for at least four-fold changes in the size of the swarm, (ii) compactness (e_L^{ss}) improves as N increases, and (iii) the average convergence time T increases as N increases.

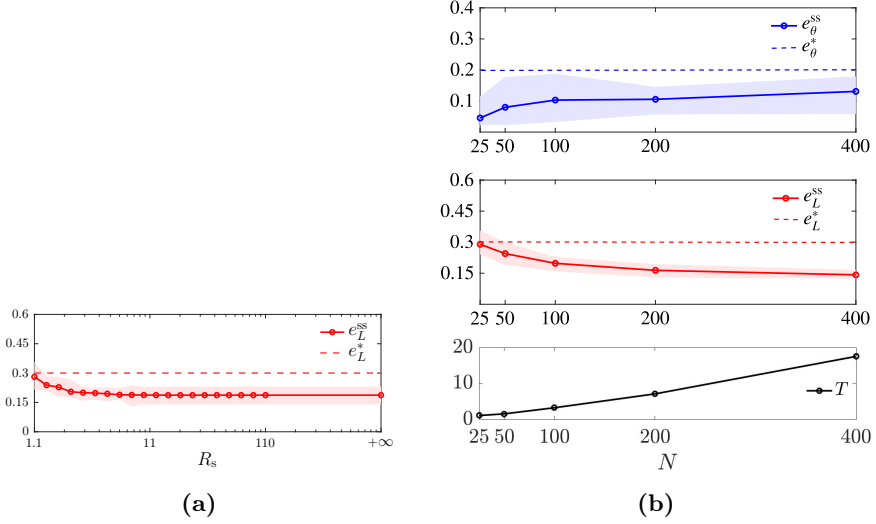


Figure 4.9: Scalability. (a) e_L^{ss} averaged over $M = 30$ trials for different values of the sensing radius R_s . (b) Metrics e_θ^{ss} and e_L^{ss} and the convergence time T averaged over the trials, with varying N , while $R_s = 3$ m, and agents' initial positions are drawn with uniform distribution from a disk with radius $r = \sqrt{N/25}$. The shaded areas represent the maximum and minimum values over the M trials. Here $L = 4$ and the gains are set as $(G_r, G_n) = (G_r^*, G_n^*)_{L=4}$.

4.3.4 Comparison with other established algorithm

As done in related literature [147] (yet for a position-based solution), we compared our control law (4.5) to the so-called “gravitational virtual forces” strategy first introduced in [119] and later extended in [120], that represents an established solution to geometric pattern formation problems.

In these works the agents' dynamics is described by

$$\begin{cases} \dot{\mathbf{x}}_i = \mathbf{v}_i, \\ \dot{\mathbf{v}}_i = \frac{1}{m}(\mathbf{u}_i - \mu\mathbf{v}_i), \end{cases} \quad \forall i \in \mathcal{S}, \quad (4.19)$$

where $\mathbf{u}_i \in \mathbb{R}^2$ is the control input, $m \in \mathbb{R}_{>0}$ is the mass of the agent and $\mu \in \mathbb{R}_{>0}$ is the friction damping factor. The control input \mathbf{u}_i is given by

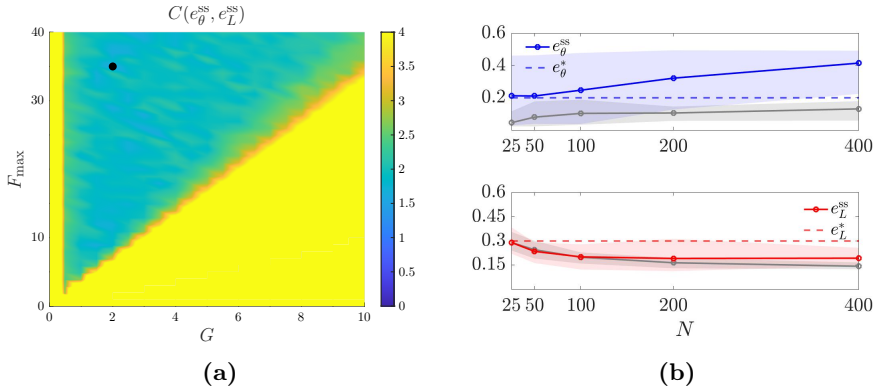


Figure 4.10: Testing the algorithm from [120]. (a) Tuning of parameters G and F_{max} for the square lattice ($L = 4$). The black dot denotes the optimal pair (G^*, F_{max}^*) . (b) Scalability test. The metrics e_L^{ss} and e_θ^{ss} are averaged over $M = 30$ trials, as N varies, and plotted against our results (in gray) in the same scenario (see Figure 4.9 for the sake of comparison). Agents' initial positions are drawn with uniform distribution from a disk of radius $r = \sqrt{N}/25$. The shaded area represents the maximum and minimum values over the trials. Here $L = 4$, and $(G, F_{\text{max}}) = (G^*, F_{\text{max}}^*)$.

the classic virtual forces control law, that is

$$\mathbf{u}_i = \sum_{j=1}^N f(\|\mathbf{r}_{ij}\|) \frac{\mathbf{r}_{ij}}{\|\mathbf{r}_{ij}\|}, \quad (4.20)$$

where f is a gravitational-like virtual force, namely

$$f(\|\mathbf{r}_{ij}\|) = \begin{cases} \left[G \frac{m^2}{\|\mathbf{r}_{ij}\|^2} \right]_0^{F_{\max}}, & \text{if } 0 \leq \|\mathbf{r}_{ij}\| \leq R, \\ - \left[G \frac{m^2}{\|\mathbf{r}_{ij}\|^2} \right]_0^{F_{\max}}, & \text{if } R < \|\mathbf{r}_{ij}\| \leq 1.5R, \\ 0, & \text{otherwise.} \end{cases} \quad (4.21)$$

Here G and $F_{\max} \in \mathbb{R}_{\geq 0}$ are tunable control gains, and $R \in \mathbb{R}_{> 0}$ is the desired link length, while the *saturation* function $[x(t)]_a^b : \mathbb{R} \rightarrow [a; b]$ is defined as

$$[x(t)]_a^b := \begin{cases} a, & \text{if } x(t) < a, \\ x(t), & \text{if } a \leq x(t) \leq b, \\ b, & \text{if } x(t) > b. \end{cases}$$

The control law given by (4.20) and (4.21) was showed to work for triangular lattices. To make it suitable for square patterns, a binary variable called *spin* is introduced for each agent, and the swarm is divided in two subsets, depending on the value of their spin. Then, agents with different spin aggregate at distance R , while agents with the same spin do so at distance $\sqrt{2}R$. The extension to the case of hexagonal lattice is discussed in [106] and requires communication among the agents. Notice that in (4.19), a second order damped dynamics is considered for the agents. Nevertheless, for the sake of comparison and by recalling Remark 4.1 we reduced it to the first order model in (4.4), by assuming that the viscous friction force is significantly larger than the inertial one. Then, to select the gravitational gain G and the saturation value F_{\max} in the control law from [120], we applied the same tuning procedure described in Section 4.3.2. In particular, we considered $(G, F_{\max}) \in \{0, 0.5, \dots, 10\} \times \{0, 1, \dots, 40\}$, and performed 30 trials for each pair of parameters, obtaining as optimal pair for the square lattice $(G^*, F_{\max}^*) = (35, 2)$ (see Figure 4.10a). All other parameters were kept to the default values in Table 4.2.

Then, we performed the same scalability test in Section 4.3.3 and report the results in Figure 4.10b. Remarkably, by comparing these results with ours, we see that our proposed control strategy performs better, obtaining much smaller values of e_θ^{ss} , regardless of the size N of the swarm. In particular, the control law from [120] only rarely achieves $e_\theta^{\text{ss}} \leq e_\theta^*$, implying a low success rate.

4.4 Adaptive tuning of control gains

Tuning the control gains (here $G_{r,i}$ and $G_{n,i}$) can in general be a tedious and time-consuming procedure. Therefore, to avoid it, we propose the use of a simple, yet effective adaptive control law, that might also improve the robustness and flexibility of the swarm. Specifically, for the sake of simplicity, $G_{r,i}$ is set to a constant value G_r for all the swarm, while each agent computes its gain $G_{n,i}$ independently, using only local information. Letting $e_{\theta,i} \in [0, 1]$ be the *average angular error* for agent i , given by

$$e_{\theta,i} := \frac{L}{\pi} \frac{1}{|\mathcal{A}_i|} \sum_{j \in \mathcal{A}_i} |\theta_{ij}^{\text{err}}|, \quad (4.22)$$

$G_{n,i}$ is varied according to the law

$$\frac{d}{dt} G_{n,i}(t) = \begin{cases} \alpha (e_{\theta,i}(t) - e_\theta^*), & \text{if } e_{\theta,i}(t) > e_\theta^*, \\ 0, & \text{otherwise.} \end{cases} \quad (4.23a)$$

$$G_{n,i}(0) = 0, \quad (4.23b)$$

where $\alpha > 0$ is an adaptation gain and e_θ^* (introduced in Section 4.3.1) is used to determine the amplitude of the dead-zone. Here, we empirically choose $\alpha = 3$. To evaluate the effect of the adaptation law, we also define the average normal gain of the swarm $\bar{G}_n(t) := \frac{1}{N} \sum_{i=1}^N G_{n,i}(t)$.

In Figure 4.11, we report the time evolution of e_L , e_θ , and of \bar{G}_n for a representative simulation. First, we notice that the average normal gain \bar{G}_n eventually settles to a constant value. Moreover, comparing the results with the case in which the gains $G_{n,i}$ are static (see Section 4.3.2 and Figure 4.5j), here T_θ , T_L and t_{ss} are larger (meaning longer convergence time), but e_θ^{ss} and e_L^{ss} are smaller (meaning better regularity and compactness

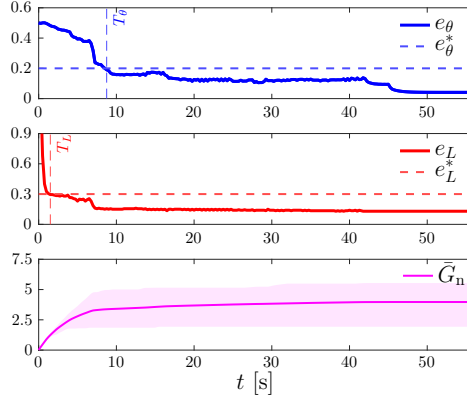


Figure 4.11: Pattern formation using the adaptive tuning law (4.23). Initial conditions are the same as those of the simulation in Figure 4.5. The shaded magenta area is delimited by $\max_{i \in \mathcal{S}} G_{n,i}$ and $\min_{i \in \mathcal{S}} G_{n,i}$, while the average across all agents is depicted by a solid magenta line. Here, $L = 4$, and $G_r = 15$.

performance).

4.4.1 Robustness analysis

Next, we test robustness to faults, flexibility, and scalability for the adaptive law (4.23), similarly to what we did in Section 4.3.3.

We ran a series of agents removal tests. For the sake of brevity, we report the results of one of such tests with $L = 4$ in Figure 4.12. At $t = 30$ s, 30% of the agents are removed; yet, after a short time the swarm reaggregates to recover the desired lattice. As observed in Section 4.3.3, at $t = 30$ s both metrics increase, but, after a short transient, they converge again below the respective thresholds.

When performing the flexibility test (time-varying reference lattice), a non trivial issue was how to set the value of the control gains, as, in a single trial, the algorithm was tested over different patterns to form dynamically. For the sake of simplicity, we tested our solution using the optimal gains found for the formation of square lattices, as this structure is intrinsically more complex than the triangular one. We re-propose here the same experiment, but considering adaptive tuning of the normal gain.

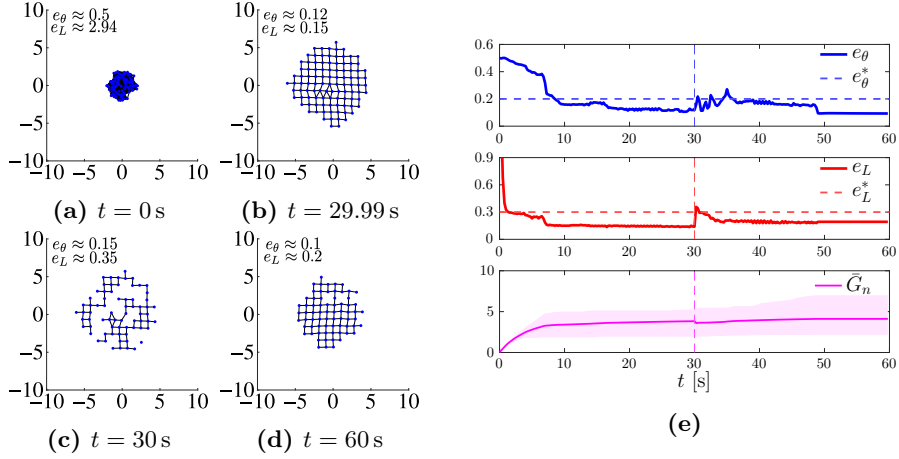


Figure 4.12: Robustness to agents removal using the adaptive tuning law (4.23). The simulation starts with $N = 100$ agents, 30 agents are removed at $t = 30$ s. Initial conditions are the same as those of the simulation in Figure 4.6. (a–d) snapshots of the agents’ configurations at different time instants. (e) time evolution of the metrics e_θ and e_L , and of the adaptive gain G_n (the shaded magenta area is delimited by $\max_{i \in S} G_{n,i}$ and $\min_{i \in S} G_{n,i}$ while the average value of the gain over all the agents is shown as a solid magenta line). Dashed vertical lines denote the time instant when agents are removed. Here $L = 4$.

Specifically, we set $G_r = 18.5$ (that is the average between the optimal gain for square and triangular patterns), and set $G_{n,i}$ according to law (4.23), resetting all $G_{n,i}$ to 0 when L is changed. The results are shown in Figure 4.13. When compared to the non-adaptive case (Figure 4.8), here e_θ^{ss} and e_L^{ss} are smaller (better pattern formation), but T_θ and T_L are larger (slower), especially when forming square patterns. Interestingly, when $L = 4$, \bar{G}_n settles to about 5, while when $L = 6$ it settles to about 0.3, a much smaller value.

Finally, we repeated the test in Section 4.3.3, setting again the sensing radius R_s to $3R$ and assessing performance while varying the size N of the swarm; results are shown in Figure 4.14. First, we notice that the larger the swarm is, the larger the steady state value of \bar{G}_n is. Comparing the results with those obtained for static gains shown in Figure 4.9b, here we observe a slight improvement of performance, with a slightly smaller e_θ^{ss} , while we verified that the convergence time is comparable to the one observed for the static policy.

4.5 Experimental validation

To further validate our control algorithm, we tested it in a real robotic scenario, using the open access *Robotarium* platform; see [97, 142] for further details. The experimental setup features 20 differential drive robots (GRITSBot [98]), that can move in a $3.2 \text{ m} \times 2 \text{ m}$ rectangular arena. The robots have a diameter of about 11 cm, a maximum (linear) speed of 20 cm/s, and a maximum rotational speed of about 3.6 rad/s. To cope with the limited size of the arena, distances $\|\mathbf{r}_{ij}\|$ in (4.7) are doubled, while control inputs \mathbf{u}_i are halved. The Robotarium implementation includes a collision avoidance safety protocol and transforms the velocity inputs (4.5) into appropriate acceleration control inputs for the robots. Moreover, we run an initial routine to yield an initial condition in which the agents are aggregated as much as possible at the centre, similarly to what considered in Section 4.3.

As a paradigmatic example, we performed a flexibility test (similarly to what done in Sec 4.3.3 and reported in Figure 4.8). During the first 33 seconds, the agents reach an aggregated initial condition. Then we set $L = 4$ for $t \in [33, 165)$, $L = 6$ for $t \in [165, 297)$, and $L = 4$ for

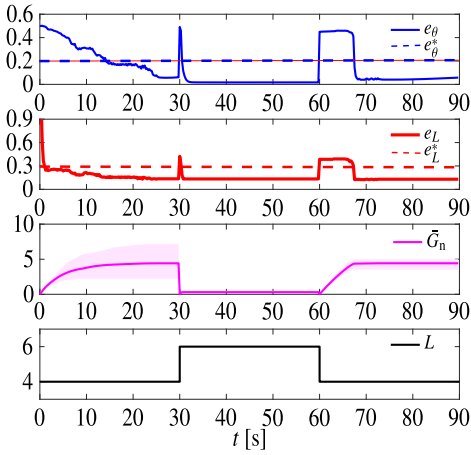


Figure 4.13: Flexibility tests using the adaptive tuning law (4.23). Initial conditions are the same as those of the simulation in Figure 4.8. The shaded magenta area is delimited by $\max_{i \in \mathcal{S}} G_{n,i}$ and $\min_{i \in \mathcal{S}} G_{n,i}$. Here $L = 4$, and $G_r = 15$, $R_s = 3$ m.

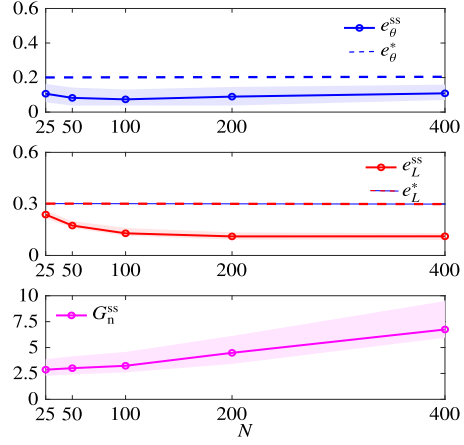


Figure 4.14: Scalability test using the adaptive tuning law (4.23). e_θ^{ss} and e_L^{ss} are averaged over $M = 30$ trials with varying N . $R_s = 3$ m; agents' initial positions are drawn with uniform distribution from a disk with radius $r = \sqrt{N/25}$. $G_n^{\text{ss}} := \bar{G}_n(t_{\text{ss}})$. The shaded areas represent the maximum and minimum values over the trials. $L = 4$, $G_r = 15$.

$t \in [297, 429]$, ending the simulation. We used the static control law (4.5)-(4.6) and (4.9), and to comply with the limited size of the arena, we scaled the control gains to the values $G_r = 0.8$ and $G_n = 0.4$, selected empirically.

The resulting movie is available online at <https://github.com/diBernardoGroup/SwarmSimPublic/tree/SwarmSimV1/Media>, while representative snapshots are reported in Figure 4.15, with the time evolution of the metrics. The metrics qualitatively reproduce the behaviour obtained in simulation (see Figure 4.8). In particular, we obtain $e_\theta^{\text{ss}} < e_\theta^*$, with both triangular and square patterns. On the other hand, we obtain $e_L^{\text{ss}} < e_L^*$ when forming square patterns, but $e_L^{\text{ss}} > e_L^*$ with triangular patterns; this does not mean that the pattern is not achieved, as it can be seen in Figure 4.15c showing the pattern is successfully achieved. This minor performance degradation is due to (i) the reduced number of agents, (ii) unmodelled dynamics of the differential-drive robots such as non-holonomic constraints and finite acceleration, and (iii) additional constraints such as the finite size of the arena and the size of the robots themselves.

The overall outcome is that, with a relative small effort, our solution can be implemented in real robots and is capable to overcome the simulation to reality gap.

4.6 Discussion

In this Chapter we presented a middle level control algorithm (see Section 2.3) to solve problems of geometric pattern formation, specifically our solution uses virtual forces to form square and triangular lattices. The proposed solution is distributed and displacement-based, i.e. only requires distance sensors and a compass, and does not need communication between the agents. We showed via exhaustive simulations and experiments that the strategy is effective in achieving both triangular and square lattices. As a benchmark, we also compared it with the well established distance-based strategy in [120], observing better performance particularly when the goal is that of achieving square lattices. Additionally, we showed that the control law is robust to failures of the agents and to noise, it is flexible to changes in the desired lattice and scalable with respect to the number of agents. We also presented a simple yet effective gain adaptation law to automatically tune the gains so as to be able to switch the goal pattern in

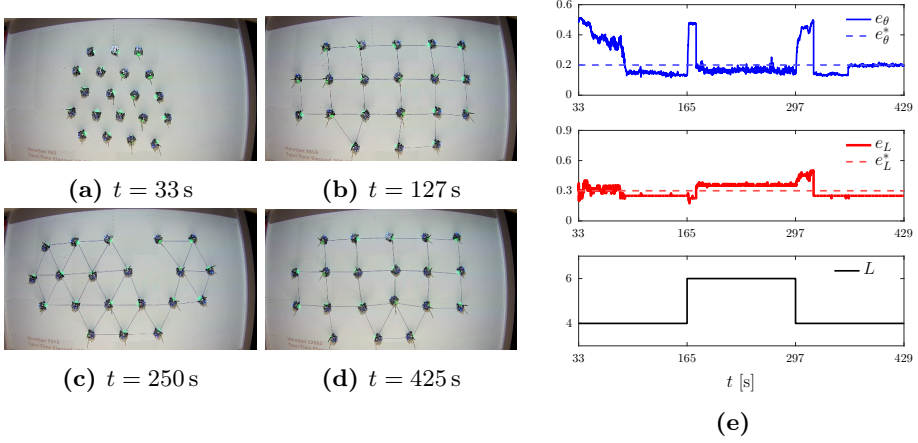


Figure 4.15: Flexibility test in the Robotarium. (a–d) swarm at different time instants. (e) time evolution of the metrics and the parameter L . The gains are set as $(G_r, G_n) = (G_r^*, G_n^*)_{L=4}$. During the first 33 seconds, the agents reach an aggregated initial condition.

real-time.

On the other hand, a complete proof of the convergence for this control algorithm is still lacking, mostly because of the difficulties generated by the interplay between the two virtual forces (i.e. radial and normal). Nevertheless, in the simplified case where only the radial force is present, such proof was indeed obtained and will be discussed in the next Chapter.

Chapter 5

Convergence toward geometric patterns

As introduced in Chapter 3, proving formally the stability of distributed control algorithms for geometric pattern formation is a challenging yet crucial task. In this Chapter, which follows the contents and the structure of our work [44], we complement the numerical and experimental results presented in Chapter 4, with a novel analytical result. Specifically, we revisit the problem of geometric pattern formation using *attraction/repulsion* virtual forces with the aim of bridging a gap in the existing literature and deriving a general proof of convergence when considering the formation of *rigid lattice* configurations in multi-dimensional spaces. When compared to previous work, our stability results (i) can be applied to most control laws based on virtual forces (or potentials), rather than only to a specific algorithm [71], (ii) are sufficient rather than necessary conditions, as, e.g., in [57], (iii) characterize the asymptotic configuration of the agents, rather than just proving its boundedness [42], and (iv) guarantee the emergence of rigid lattices rather than less regular ones, e.g., the α -lattices studied in [95], which allow for disconnected graphs and the coexistence of heterogeneous patterns (e.g., triangular and square).

5.1 Problem formulation

Let us formulate the problem by recalling the concepts of *swarm*, a set of N mobile agents as introduced in Section 3.2.3, and of *adjacency set* (Definition 3.10). Then set $R_{\min} = 0$ and the *maximum link length* $R_{\max} = R_a$, so that two agents are connected by a link (see Definition 3.11) if and only if their distance is at most R_a ; see Figure 5.1a.

Now let us recall the *infinitesimal rigidity* of *frameworks* (Definitions 3.4 and 3.7), to introduce the central topic of this Chapter, i.e. the *rigid lattice*.

Definition 5.1 (Rigid lattice). *Given a swarm with framework $\mathcal{F}(\bar{\mathbf{x}}^*)$, we call $\bar{\mathbf{x}}^*$ a rigid lattice configuration if*

- (A) $\mathcal{F}(\bar{\mathbf{x}}^*)$ is infinitesimally rigid, and
- (B) $\|\mathbf{r}_{ij}\| = R, \forall (i, j) \in \mathcal{E}(\bar{\mathbf{x}}^*)$,

where $R \in \mathbb{R}_{>0}$ denotes the desired link length.

Figs. 5.1b, 5.1c portray examples of rigid lattices, for $d = 2$ and $d = 3$: a tessellation of triangles, and one of tetrahedra and octahedra, respectively. It is immediate to verify that rigid lattices are characterized by connected graphs where each agent has at least d links, yielding robustness to link failure. A similar structure is the α -lattice from [95], which requires (B) but not (A) (hence, a rigid lattice is an α -lattice, but the converse is false). Thus, α -lattices can display more heterogeneous structures, containing different polytopes (e.g., squares, cubes), or even be disconnected, which can be unsuited for applications such as region coverage or distributed sensing. Note however that *vacancies*, i.e. holes in the lattice, can be present in both rigid and α -lattices.

In a rigid lattice, we denote by R_{next} the minimum distance between two not directly connected agents (e.g., $R_{\text{next}} = R\sqrt{3}$ if $d = 2$ and $R_{\text{next}} = R\sqrt{2}$ if $d = 3$). Here, we assume that $R_a \in]R; R_{\text{next}}[$, so that, when the swarm is in a rigid lattice configuration, the adjacency set (Definition 3.10) of any agent includes only the agents in its immediate surroundings, and all the links (Definition 3.11) have length R (see Figure 5.1). Moreover, defining $\mathcal{T} \subset \mathbb{R}^{dN}$ as the set of all rigid lattice configurations; it is immediate to verify that \mathcal{T} is unbounded and disconnected.

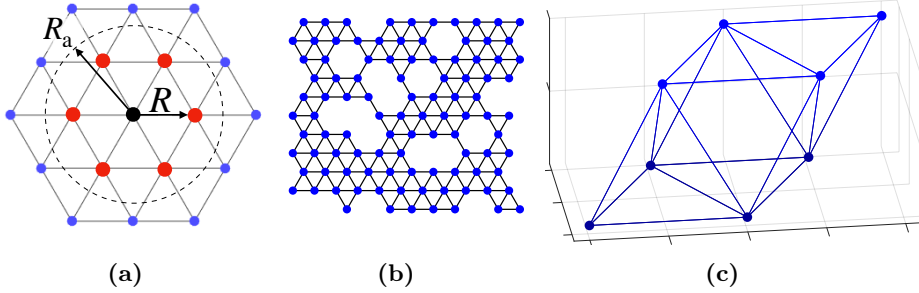


Figure 5.1: (a) Adjacency set (red) of an agent (black). (b) A rigid lattice with $d = 2$ and $N = 100$. (c) A rigid lattice with $d = 3$ and $N = 8$.

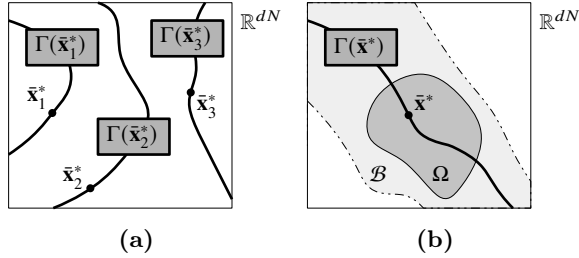


Figure 5.2: (a) Sets of rigid lattices configurations. (b) Sets used in the proof of Theorem 5.1.

Also, note that any configuration congruent to a rigid lattice (according to Definition 3.13) is a rigid lattice itself, formally $\bar{\mathbf{x}}^* \in \mathcal{T} \Leftrightarrow \Gamma(\bar{\mathbf{x}}^*) \subset \mathcal{T}$ (see Figure 5.2a), and

$$\mathcal{T} = \bigcup_{\bar{\mathbf{x}}^* \in \mathcal{T}} \Gamma(\bar{\mathbf{x}}^*). \quad (5.1)$$

Now, consider a swarm \mathcal{S} and assume the agents' dynamics is described by

$$\dot{\mathbf{x}}_i(t) = \mathbf{u}_i(t), \quad \forall i \in \mathcal{S}, \quad (5.2)$$

where $\mathbf{x}_i(t) \in \mathbb{R}^d$ is the position of agent i , and $\mathbf{u}_i(t) \in \mathbb{R}^d$ its velocity input. We aim to select and validate a distributed control law to compute this input and let the swarm achieve a rigid lattice configuration. Therefore, let us recall the interaction set $\mathcal{I}_i(t)$ from Definition 3.9 and assume

that $R_s \geq R_a$, so that

$$\mathcal{A}_i \subseteq \mathcal{I}_i, \quad \forall i \in \mathcal{S}. \quad (5.3)$$

We can now select $\mathbf{u}_i(t)$ in (5.2) as the distributed *virtual forces* control law

$$\mathbf{u}_i(t) := \sum_{j \in \mathcal{I}_i(t)} f(\|\mathbf{r}_{ij}(t)\|) \hat{\mathbf{r}}_{ij}(t), \quad (5.4)$$

where $f : \mathbb{R}_{>0} \rightarrow \mathbb{R}$ is the *interaction function*.

Let us introduce our first result, which slightly extends [42, Lemma 1], and will be used in the next Section.

Lemma 5.1. *The position of the center of the swarm, say \mathbf{x}_c , under the control law (5.4) is invariant, that is $\dot{\mathbf{x}}_c = \mathbf{0} \forall \bar{\mathbf{x}} \in \mathbb{R}^{dN}$.*

Proof. Exploiting (5.2) and (5.4), the dynamics of the center of the swarm is given by $\dot{\mathbf{x}}_c := \frac{1}{N} \sum_{i=1}^N \dot{\mathbf{x}}_i = \frac{1}{N} \sum_{i=1}^N \mathbf{u}_i = \frac{1}{N} \sum_{i=1}^N \sum_{j \in \mathcal{I}_i} f(\|\mathbf{r}_{ij}\|) \hat{\mathbf{r}}_{ij}$. Since the existence of any link (i, j) implies the existence of link (j, i) (see Definition 3.10), for any term $f(\|\mathbf{r}_{ij}\|) \hat{\mathbf{r}}_{ij}$ there exists a term $f(\|\mathbf{r}_{ji}\|) \hat{\mathbf{r}}_{ji} = -f(\|\mathbf{r}_{ij}\|) \hat{\mathbf{r}}_{ij}$ (because $\|\mathbf{r}_{ij}\| = \|\mathbf{r}_{ji}\|$ and $\hat{\mathbf{r}}_{ij} = -\hat{\mathbf{r}}_{ji}$). Therefore, the sum of the two is zero, yielding the thesis. \square

5.2 Convergence to a rigid lattice configuration

We can now state our main result, i.e., that, given an interaction function f (in (5.4)) generating short range repulsion and long range attraction, the set of rigid lattice configurations is locally asymptotically stable (see Definition 3.2). For this result to hold we need the following assumption on the interaction function.

Assumption 5.1. *f (in (5.4)) is such that:*

- (a1) $f(R) = 0$,
- (a2) $f(z) > 0$ for $z \in]0; R[$ and $f(z) < 0$ for $z \in]R; R_a[$,
- (a3) $f(z)$ is continuous in $]0; R_a[$,
- (a4) $f(z) = 0$ for any $z > R_a$.

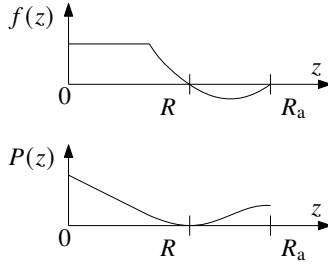


Figure 5.3: An interaction function f satisfying Assumption 5.1 and its potential P .

An exemplary interaction function fulfilling the assumption above is portrayed in Figure 5.3.

Without loss of generality, we further assume that, under Assumption 5.1, in a sufficiently small neighborhood of a rigid lattice configuration, all other equilibria are also rigid lattice configurations (supporting evidence showing that this assumption is not restrictive is reported in Section 5.4).

Theorem 5.1. *[Stability of rigid lattices] Let Assumption 5.1 hold. Then, for any rigid lattice configuration $\bar{\mathbf{x}}^*$, $\Gamma(\bar{\mathbf{x}}^*)$ is a locally asymptotically stable equilibrium set. Consequently, \mathcal{T} is also a locally asymptotically stable equilibrium set.*

Proof. Let us consider *any* rigid lattice configuration $\bar{\mathbf{x}}^* \in \mathcal{T}$, with center $\mathbf{x}_c^* := \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i^*$ and relative positions \mathbf{r}_{ij}^* , and the set $\Gamma(\bar{\mathbf{x}}^*)$ of its congruent configurations. Recalling Definition 5.1.(B) and (a1), we have that $\bar{\mathbf{x}}^*$ is an equilibrium point of (5.2)–(5.4); thus, $\Gamma(\bar{\mathbf{x}}^*)$ and \mathcal{T} are equilibrium sets, according to Definition 3.1. Next, we will prove local asymptotic stability of $\Gamma(\bar{\mathbf{x}}^*) \subset \mathcal{T}$, which implies local asymptotic stability of \mathcal{T} through (5.1).

Step 1 (Lyapunov function) Given a configuration $\bar{\mathbf{x}} \in \mathbb{R}^{dN}$ with center \mathbf{x}_c and inducing the links in $\mathcal{E}(\bar{\mathbf{x}})$ according to Definition 3.11, let $m := |\mathcal{E}(\bar{\mathbf{x}})|$ and order the links in $\mathcal{E}(\bar{\mathbf{x}})$ arbitrarily, so that $\mathbf{r}_1, \dots, \mathbf{r}_m$ refer to the relative positions \mathbf{r}_{ij} for $(i, j) \in \mathcal{E}(\bar{\mathbf{x}})$. Recalling (a3), we can define the potential function $P :]0, R_a] \rightarrow \mathbb{R}$ given by $P(z) = -\int_R^z f(y) dy$ (see

Figure 5.3). Note that $P(R) = 0$, $\frac{dP}{dz}(z) = -f(z)$, and, from (a2),

$$P(z) > 0 \quad \forall z \in \mathbb{R}_{>0} \setminus \{R\}. \quad (5.5)$$

Then, let us consider the candidate Lyapunov function

$$V(\bar{\mathbf{x}}) := \|\mathbf{x}_c^* - \mathbf{x}_c\|^2 + \sum_{k \in \mathcal{E}(\bar{\mathbf{x}})} P(\|\mathbf{r}_k\|). \quad (5.6)$$

By (5.5), it holds that $V(\bar{\mathbf{x}}) \geq 0 \quad \forall \bar{\mathbf{x}} \in \mathbb{R}^{dN}$, and $V = 0$ if and only if both $\mathbf{x}_c = \mathbf{x}_c^*$ and Definition 5.1.(B) holds.

Step 2 (Properties of V) $V(\bar{\mathbf{x}})$ is discontinuous over \mathbb{R}^{dN} (because $\mathcal{E}(\bar{\mathbf{x}})$ changes when links (dis-)appear). However, $V(\bar{\mathbf{x}})$ is continuous and differentiable in any subset of \mathbb{R}^{dN} where the set $\mathcal{E}(\bar{\mathbf{x}})$ of links is constant. To find such a set, we seek conditions on $\bar{\mathbf{x}}$ such that $\mathcal{E}(\bar{\mathbf{x}}) = \mathcal{E}(\bar{\mathbf{x}}^*)$ (see Definitions 3.10 and 3.11), i.e.,

$$\|\mathbf{r}_{ij}\| < R_a, \quad \forall (i, j) \in \mathcal{E}(\bar{\mathbf{x}}^*), \quad (5.7a)$$

$$\|\mathbf{r}_{ij}\| > R_a, \quad \forall (i, j) \notin \mathcal{E}(\bar{\mathbf{x}}^*). \quad (5.7b)$$

(5.7a) means that all links in $\mathcal{E}(\bar{\mathbf{x}}^*)$ are preserved in $\mathcal{E}(\bar{\mathbf{x}})$, while (5.7b) means that no new links are created in $\mathcal{E}(\bar{\mathbf{x}})$ with respect to $\mathcal{E}(\bar{\mathbf{x}}^*)$. With simple algebraic manipulations it is possible to show that (5.7a) and (5.7b) hold if $\bar{\mathbf{x}} \in \mathcal{B}$, where

$$\mathcal{B} := \{\bar{\mathbf{x}} \in \mathbb{R}^{dN} : \|\|\mathbf{r}_{ij}\| - \|\mathbf{r}_{ij}^*\|\| < \beta, \quad \forall i, j \in \mathcal{S}\}, \quad (5.8)$$

and $\beta < \min_{i,j \in \mathcal{S}} |R_a - \|\mathbf{r}_{ij}^*\||$; \mathcal{B} can be intended as a “neighborhood” of $\Gamma(\bar{\mathbf{x}}^*)$ with “width” β (see Figure 5.2b). Thus, $\mathcal{E}(\bar{\mathbf{x}}) = \mathcal{E}(\bar{\mathbf{x}}^*)$ in \mathcal{B} and, hence, V is continuously differentiable in \mathcal{B} .

Step 3 (Analysis of \dot{V}) Now, we restrict our analysis to the set \mathcal{B} to study the attractivity of $\Gamma(\bar{\mathbf{x}}^*)$. We start by studying the dynamics of the agents. From (5.2)–(5.4), we have $\dot{\mathbf{x}}_i = \sum_{j \in \mathcal{I}_i} f(\|\mathbf{r}_{ij}\|) \hat{\mathbf{r}}_{ij}$. Hypothesis (a4) and (5.3) imply that $\sum_{j \in \mathcal{I}_i} f(\|\mathbf{r}_{ij}\|) \hat{\mathbf{r}}_{ij} = \sum_{j \in \mathcal{A}_i} f(\|\mathbf{r}_{ij}\|) \hat{\mathbf{r}}_{ij}$. Hence, using

the incidence matrix \mathbf{B} (Definition 3.3) of the swarm graph, we get

$$\dot{\mathbf{x}}_i = \sum_{j \in \mathcal{A}_i} f(\|\mathbf{r}_{ij}\|) \hat{\mathbf{r}}_{ij} = \sum_{k=1}^m [\mathbf{B}]_{ik} f(\|\mathbf{r}_k\|) \hat{\mathbf{r}}_k. \quad (5.9)$$

Moreover we can write the dynamics of the relative positions along a link k as $\dot{\mathbf{r}}_k = \sum_{i=1}^N [\mathbf{B}]_{ik} \dot{\mathbf{x}}_i$. Therefore, exploiting (5.6), Lemma 5.1, and (5.9), we get

$$\begin{aligned} \dot{V}(\bar{\mathbf{x}}) &= \sum_{k=1}^m \frac{\partial V}{\partial \|\mathbf{r}_k\|} \frac{\partial \|\mathbf{r}_k\|}{\partial \mathbf{r}_k} \dot{\mathbf{r}}_k = \sum_{k=1}^m P'(\|\mathbf{r}_k\|) \hat{\mathbf{r}}_k^\top \sum_{i=1}^N [\mathbf{B}]_{ik} \dot{\mathbf{x}}_i \\ &= - \sum_{i=1}^N \sum_{k=1}^m f(\|\mathbf{r}_k\|) [\mathbf{B}^\top]_{ki} \hat{\mathbf{r}}_k^\top \dot{\mathbf{x}}_i = - \sum_{i=1}^N \dot{\mathbf{x}}_i^\top \dot{\mathbf{x}}_i = -\dot{\bar{\mathbf{x}}}^\top \dot{\bar{\mathbf{x}}} \leq 0, \end{aligned} \quad (5.10)$$

where we also used that $P' = -f$ and that $\frac{\partial \|\mathbf{r}_k\|}{\partial \mathbf{r}_k} = \hat{\mathbf{r}}_k^\top$. We can hence conclude that $\dot{V}(\bar{\mathbf{x}}) = 0$ if and only if $\dot{\bar{\mathbf{x}}} = \mathbf{0}$, i.e., in correspondence of equilibrium configurations.

Choosing β in (5.8) small enough, we exclude the presence of equilibrium configurations not belonging to $\Gamma(\bar{\mathbf{x}}^*)$, and hence

$$\begin{cases} \dot{V}(\bar{\mathbf{x}}) = 0, & \text{if } \bar{\mathbf{x}} \in \Gamma(\bar{\mathbf{x}}^*), \\ \dot{V}(\bar{\mathbf{x}}) < 0, & \text{if } \bar{\mathbf{x}} \in \mathcal{B} \setminus \Gamma(\bar{\mathbf{x}}^*). \end{cases} \quad (5.11)$$

Step 4 (Applying LaSalle's invariance principle) To complete the proof, we define a forward invariant neighborhood of $\bar{\mathbf{x}}^*$ and then apply LaSalle's invariance principle. Given some $\omega \in \mathbb{R}_{>0}$, let Ω be the largest connected set containing $\bar{\mathbf{x}}^*$ such that $V(\bar{\mathbf{x}}) \leq \omega \forall \bar{\mathbf{x}} \in \Omega$ (see Figure 5.2b). In particular, we select ω small enough that $\Omega \subseteq \mathcal{B}$.¹ Since $V(\bar{\mathbf{x}}) \leq \omega$ and $\dot{V}(\bar{\mathbf{x}}) \leq 0$ for all $\bar{\mathbf{x}} \in \Omega$, then Ω is forward invariant. Moreover, Ω is closed, because V is continuous in Ω , and Ω is the inverse image of the closed set $[0, \omega]$. Ω is also bounded because (i) translations too far from

¹ Such ω exists because \mathcal{B} is a "neighborhood" of $\Gamma(\bar{\mathbf{x}}^*)$ (in the sense of (5.8)) and, by the rigidity of framework $\mathcal{F}(\bar{\mathbf{x}}^*)$ (Definition 3.8), any continuous motion of the vertices that changes the distance between any two vertices also changes the length of at least one link, causing V to increase.

$\bar{\mathbf{x}}^*$ cause V to increase beyond ω (see (5.6)), and (ii) $\Omega \subseteq \mathcal{B}$ implies that the deformations of the framework are bounded (see (5.8)).

As Ω is closed, bounded (thus compact) and forward invariant, we can apply LaSalle's invariance principle [65, Theorem 4.4], and noting that, in Ω , $\dot{V}(\bar{\mathbf{x}}) = 0$ if and only if $\bar{\mathbf{x}} \in \Gamma(\bar{\mathbf{x}}^*)$ (see (5.11)), we get that all the trajectories starting in Ω converge to $\Gamma(\bar{\mathbf{x}}^*) \cap \Omega$. This and the forward invariance of Ω imply that $\Gamma(\bar{\mathbf{x}}^*)$ is locally asymptotically stable (see Definition 3.2), and so is \mathcal{T} because of (5.1). \square

Moreover, this proof opens the door to the following ancillary results.

Proposition 5.1. *[Collision avoidance] Let $P^0 := \lim_{z \searrow 0} P(z)$. (i) No collisions between agents occur if $P^0 = \infty$. (ii) In a sufficiently small neighborhood of a rigid lattice configuration, no collisions occur if $\bar{\mathbf{x}}(0)$ is such that $\sum_{k \in \mathcal{E}(\bar{\mathbf{x}})} P(\|\mathbf{r}_k\|) < P^0$.*

Proof. When a collision occurs, at least one \mathbf{r}_k becomes zero and thus, from (5.5), $\sum_{k \in \mathcal{E}(\bar{\mathbf{x}})} P(\|\mathbf{r}_k\|) \geq P^0$. Equations (5.6) and (5.10) yield the first statement. The second statement is obtained by recalling that $\Omega \subseteq \mathcal{B}$ and that Ω is forward invariant (see Step 4 of the proof of Theorem 5.1). \square

Remark 5.1. *[Path tracking] Path tracking can be obtained by adding a velocity term $\mathbf{w}(t)$ on the right hand side of (5.2). Theorem 5.1 still holds, as the analysis can be carried out on new states \mathbf{y}_i , with $\mathbf{y}_i(t) = \mathbf{x}_i(t) - \int_0^t \mathbf{w}(\tau) d\tau$ and $\dot{\mathbf{y}}_i = \mathbf{u}_i$.*

Remark 5.1 only aims to show feasibility of path tracking; clearly, more sophisticated strategies can be designed.

Remark 5.2. *[Second order dynamics] It is possible to show that the results in Theorem 5.1 also hold in the case of second order nonlinear dynamics, that is $\dot{\mathbf{x}}_i = \mathbf{v}_i$, $\dot{\mathbf{v}}_i = g(\|\mathbf{v}_i\|)\hat{\mathbf{v}}_i + \mathbf{u}_i$, where \mathbf{x}_i and \mathbf{v}_i are the position and velocity of agent i , and $g : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\leq 0}$ is a friction term with $g(z) = 0 \Leftrightarrow z = 0$ and such that $\mathbf{v}_c(t) := \sum_{i=1}^N \mathbf{v}_i(t) \rightarrow \mathbf{0}$. Namely, the proof of Theorem 5.1 can be adapted by using the function $V = \sum_{k \in \mathcal{E}(\bar{\mathbf{x}})} P(\|\mathbf{r}_k\|) + \frac{1}{2} \sum_{i=1}^N \mathbf{v}_i^\top \mathbf{v}_i$ in (5.6) and exploiting that $\mathbf{x}_c(t) := \sum_{i=1}^N \mathbf{x}_i(t)$ remains bounded, to apply LaSalle's invariance principle.*

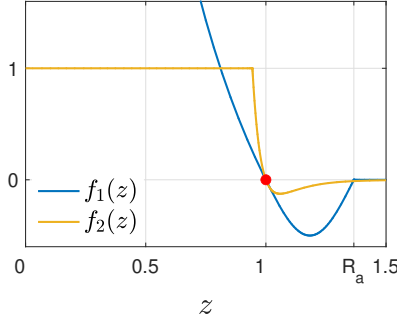


Figure 5.4: Interaction functions (5.12) and (5.13) in the case $d = 2$. The red dot highlights the zero of the functions in $z = R$.

5.3 Numerical validation

In this section, we validate numerically the result presented in Section 5.2 and estimate the basin of attraction of \mathcal{T} .

5.3.1 Simulation setup

We set the number of agents to $N = 100$, the desired link length to $R = 1$, the sensing radius $R_s = 3$, and the maximum link length to $R_a = (1 + R_{\text{next}})/2$ (i.e. $R_a \approx 1.37$ if $d = 2$; $R_a \approx 1.21$ if $d = 3$). We validate our strategy using two interaction functions, depicted in Figure 5.4. The first one is

$$f_1(z) = \begin{cases} g \left(\frac{1}{z} - \frac{1}{R} \right) \frac{\pi R^2}{R_a - R} & \text{if } z \in]0; R], \\ -g \sin \left((z - R) \frac{\pi}{R_a - R} \right) & \text{if } z \in]R; R_a], \\ 0 & \text{if } z > R_a; \end{cases} \quad (5.12)$$

with $g = 0.5$. f_1 satisfies Assumption 5.1, is smooth in $]0; R_a[$ and $\lim_{z \searrow 0} f_1(z) = \infty$, which, through Proposition 5.1, guarantees the absence of collisions. The second interaction function f_2 is the Physics-inspired Lennard-Jones

function [13], i.e.,

$$f_2(z) = \min \left\{ \left(\frac{a}{z^{2c}} - \frac{b}{z^c} \right), 1 \right\}, \quad (5.13)$$

where we select $a = b = 0.5$ and $c = 12$ when $d = 2$ and $c = 24$ when $d = 3$; see Figure 5.4. f_2 saturates to 1 as $z \searrow 0$ to comply with possible actuator saturation. Moreover, f_2 satisfies (a1), (a2) and (a3) in Assumption 5.1 exactly, but (a4) only approximately. This is intentional as it allows to account for long range attraction between the agents, which is frequently required in swarm robotics applications [42]. Notice that f_2 is equivalent to the radial interaction function (4.7) used in previous Chapter for the formation of triangular and square lattices.

To assess if the swarm is in a rigid lattice configuration, we check conditions (A), (B) in Definition 5.1. To evaluate (A) we use Theorem 3.1. To evaluate (B), we define the *error* $e(t) := \max_{k \in \mathcal{E}(t)} \left| \|\mathbf{r}_k(t)\| - R \right|$, which is zero when (B) holds. Also, as long as $e(t)$ stays strictly lower than $R_a - R$, links in the configuration of interest are neither created nor destroyed.

For each simulation, the initial positions of the agents are obtained by picking a random rigid lattice configuration and then applying, to each agent, a different random displacement drawn from a uniform distribution over a disk (when $d = 2$) or a sphere (when $d = 3$), having radius $\delta \in \mathbb{R}_{\geq 0}$.

All simulation are run in MATLAB using the agent-based simulator SwarmSim V2 we developed (for more information see Appendix A or visit <https://github.com/diBernardoGroup/SwarmSimPublic>). In particular, agents' dynamics (5.2)–(5.4) are integrated using the forward Euler method with a fixed time step equal to 0.01 s.

5.3.2 Numerical results

To validate Theorem 5.1, in Figure 5.5 we report the time evolution of the error $e(t)$ for 10 simulations where the swarm starts from a perturbed rigid lattice configuration. Simulations are presented for $d \in \{2, 3\}$ and for both interaction functions f_1 and f_2 . In all cases, infinitesimal rigidity is preserved and $e(t)$ converges to zero, denoting local stability of the lattice.

To estimate the basin of attraction of the set of rigid lattice configu-

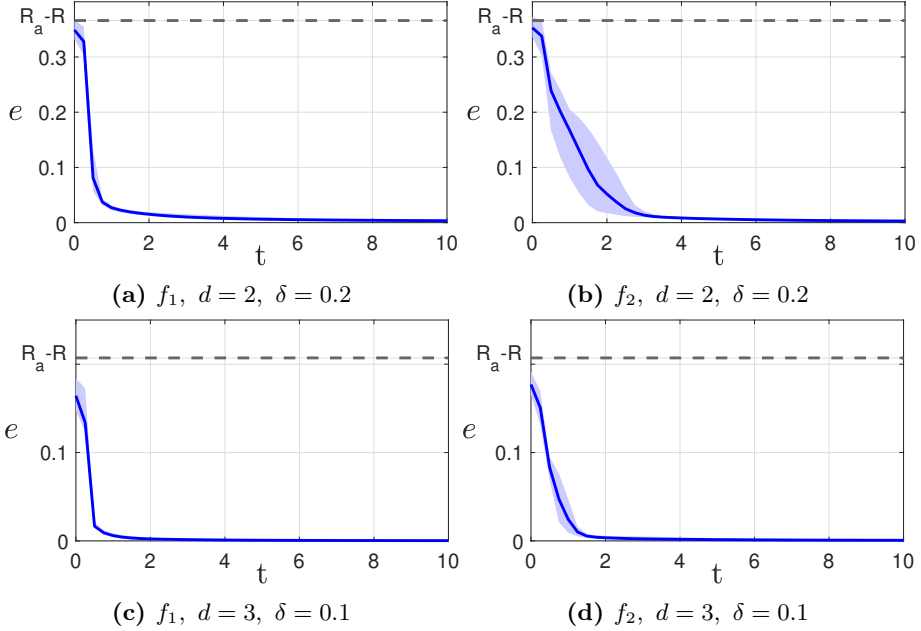


Figure 5.5: Time evolution of e for various interaction functions and values of d . In each panel, 10 simulations with random initial conditions are shown; the solid line is the mean; the shaded area is the minimum and maximum.

rations, we performed extensive simulations for various values of δ , and characterize the steady state configurations in Figure 5.6. Namely, for δ smaller than 0.25 for $d = 2$ and 0.2 for $d = 3$ all simulations converge to a rigid lattice configuration. Then, as δ increases, fewer simulations converge to rigid lattices, until none does. Note that $e(0) \leq 2\delta$, therefore $\delta = 0.25$ (respectively $\delta = 0.2$) corresponds to a perturbation of up to 50% (respectively 40%) of the initial link length, giving an estimation of the basin of attraction of \mathcal{T} .

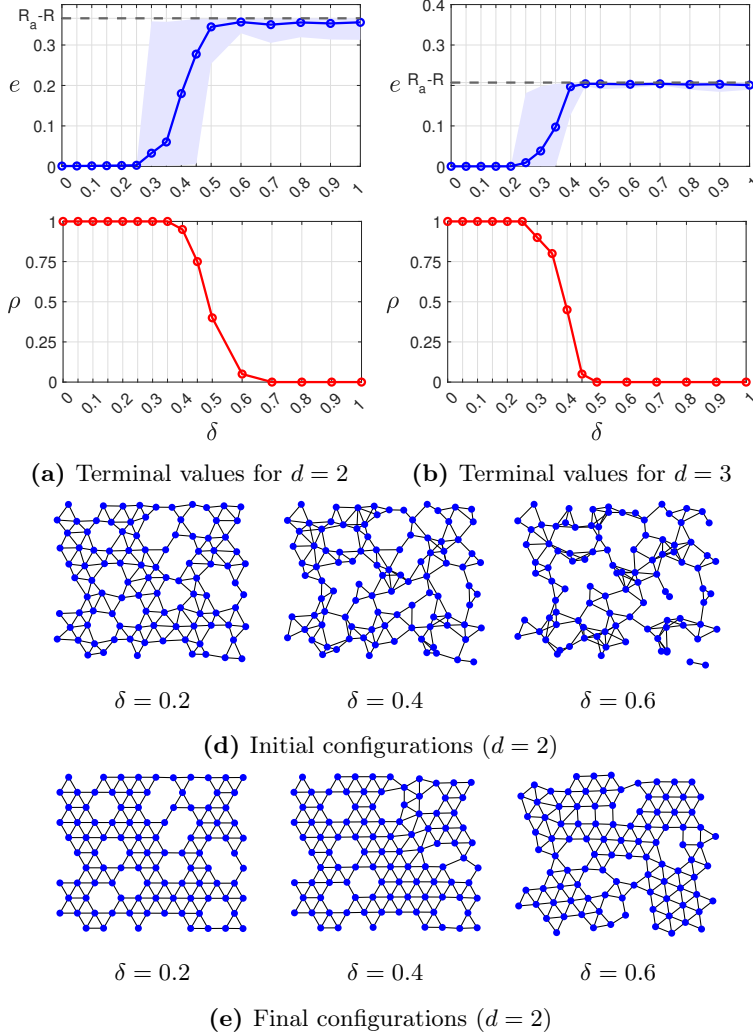


Figure 5.6: Simulations for different values of δ and interaction function f_2 . (a), (b): Terminal values of e and ρ , respectively for $d = 2$ and $d = 3$. ρ is the fraction of simulations converging to an infinitesimally rigid configuration. For e , the solid line is the mean; the shaded area is the minimum and maximum. 20 simulations with random initial conditions are performed for each value of δ , and last 20 s. (c), (d): Initial and final configurations of representative simulations for specific values of δ in the case that $d = 2$.

5.4 Further results

To further confirm the effectiveness of our theoretical results, we provide below complementary semi-analytical evidence that the set of rigid lattice configurations \mathcal{T} is locally asymptotically stable, which also excludes the presence of other equilibria in an arbitrarily small neighborhood of it. To do so, we linearize system (5.2)–(5.4) around a rigid lattice configuration, say $\bar{\mathbf{x}}^*$, obtaining $\dot{\bar{\mathbf{x}}} \approx \mathbf{J}(\bar{\mathbf{x}}^*)(\bar{\mathbf{x}} - \bar{\mathbf{x}}^*)$, with $\mathbf{J}(\bar{\mathbf{x}}^*) \in \mathbb{R}^{dN \times dN}$ derived as follows.

Jacobian of (5.2)–(5.4) System (5.2)–(5.4) can be recast as

$$\dot{\bar{\mathbf{x}}} = ((\mathbf{BFG}^{-1}\mathbf{B}^\top) \otimes \mathbf{I}_d)\bar{\mathbf{x}} = ((\mathbf{BHB}^\top) \otimes \mathbf{I}_d)\bar{\mathbf{x}}, \quad (5.14)$$

where $\mathbf{F}, \mathbf{G}, \mathbf{H} \in \mathbb{R}^{m \times m}$ are diagonal matrices; $[\mathbf{F}]_{ii} := f(\|\mathbf{r}_i\|)$, $[\mathbf{G}]_{ii} := \|\mathbf{r}_i\|$, and $\mathbf{H} := \mathbf{FG}^{-1}$. The Jacobian of (5.14) is

$$\mathbf{J} = \left(\mathbf{B} \frac{\partial \mathbf{H}}{\partial \bar{\mathbf{x}}} \mathbf{B}^\top \otimes \mathbf{I}_d \right) \bar{\mathbf{x}} + (\mathbf{BHB}^\top) \otimes \mathbf{I}_d =: \mathbf{J}_1 + \mathbf{J}_2, \quad (5.15)$$

where $\frac{\partial \mathbf{H}}{\partial \bar{\mathbf{x}}} \in \mathbb{R}^{m \times m \times dN}$ is a tensor, and $[\frac{\partial \mathbf{H}}{\partial \bar{\mathbf{x}}}]_{::,k} = [\frac{\partial \mathbf{H}}{\partial \bar{\mathbf{x}}}]_{::,k} \mathbf{B}^\top \in \mathbb{R}^{m \times N}$, with notation $[\cdot]_{::,k}$ denoting the matrix obtained by fixing the third index of the tensor. From (a1), for all rigid lattice configurations we have $\mathbf{J}_2 = (\mathbf{BHB}^\top) \otimes \mathbf{I}_d = \mathbf{0}$. Then, $[\mathbf{J}_1]_{:,k} = \left(\mathbf{B} [\frac{\partial \mathbf{H}}{\partial \bar{\mathbf{x}}}]_{::,k} \mathbf{B}^\top \otimes \mathbf{I}_d \right) \bar{\mathbf{x}}$. From [61, p. 20] we have $\frac{\partial \|\mathbf{r}_i\|^2}{\partial [\bar{\mathbf{x}}]_k} = 2[\mathbf{M}]_{i,k}$ (see Definition 3.6), that is $\frac{\partial \|\mathbf{r}_i\|}{\partial [\bar{\mathbf{x}}]_k} = \frac{1}{\|\mathbf{r}_i\|} [\mathbf{M}]_{i,k}$, and thus

$$\begin{aligned} \left[\frac{\partial \mathbf{H}}{\partial \bar{\mathbf{x}}} \right]_{i,i,k} &= \frac{\partial [f(\|\mathbf{r}_i\|)/\|\mathbf{r}_i\|]}{\partial \|\mathbf{r}_i\|} \frac{\partial \|\mathbf{r}_i\|}{\partial [\bar{\mathbf{x}}]_k} \\ &= [f'(\|\mathbf{r}_i\|) \|\mathbf{r}_i\| - f(\|\mathbf{r}_i\|)] \|\mathbf{r}_i\|^{-3} [\mathbf{M}]_{i,k}, \end{aligned} \quad (5.16a)$$

$$\left[\frac{\partial \mathbf{H}}{\partial \bar{\mathbf{x}}} \right]_{i,j,k} = 0, \quad \text{if } i \neq j. \quad (5.16b)$$

Numerical analysis We set $R = 1$ and generated 1520 random rigid lattice configurations (10 per each $N \in \{25, 26, \dots, 100\}$, and each $d \in$

$\{2, 3\}$). For each of these configurations, assuming f (in (5.4)) is in the form (5.13), we computed \mathbf{J} using (5.15)–(5.16) and found that in all cases \mathbf{J} has $d(d+1)/2$ zero eigenvalues with eigenvectors $\{\mathbf{w}_i^0\}_i$, and $dN - d(d+1)/2$ negative eigenvalues with eigenvectors $\{\mathbf{w}_j^\pm\}_j$. Moreover, $\mathbf{M}\mathbf{w}_i^0 = \mathbf{0}$ and $\mathbf{M}\mathbf{w}_j^\pm \neq \mathbf{0}$; thus, from Definition 3.7, the span of $\{\mathbf{w}_i^0\}$ corresponds to roto-translations and is a hyperplane locally tangent to $\Gamma(\bar{\mathbf{x}}^*)$ (see Definition 3.13), while $\{\mathbf{w}_j^\pm\}$ correspond to other motions. Therefore, the *center manifold theorem* [68, Theorem 5.1] yields that $\Gamma(\bar{\mathbf{x}}^*)$ is a *center manifold* of system (5.2)–(5.4). Moreover, as expected from Theorem 5.1, the *reduction principle* [68, Theorem 5.2] confirms that the dynamics locally converge onto the equilibrium set $\Gamma(\bar{\mathbf{x}}^*)$, and excludes the presence of other equilibria in an arbitrarily small neighborhood of it.

5.5 Discussion

We proved analytically local asymptotic stability of rigid lattice configurations, in multi-dimensional spaces, for swarms under the action of a distributed control law based on virtual attraction/repulsion forces. The theoretical results were supported by exhaustive numerical simulations, providing also an estimate of the basin of attraction, and further semi-analytical derivations. The mild hypotheses required on the interaction function allow for wide applicability of the theoretical results. This result addresses an important gap in the Literature on geometric pattern formation, providing a theoretical support for numerous existing solutions. The analytical characterization of the basin of attraction of \mathcal{T} , and the extension of the results to other geometric lattices, such as square, remain open.

In conclusion, Part I presented our contribution to the control of geometric pattern formation in LS-MAS. In particular, the results presented in Chapters 4 and 5 represent an advancement in both the design and the validation of distributed control algorithms for the formation of geometric patterns. The next Part will deal with the study of spatial behaviours of microorganisms, and how these emerge from the movement of the agents and their response to external stimuli.

Part II

Spatial behaviours of microorganisms

Introduction to spatiotemporal control of microorganisms

Understanding how natural [LS-MAS](#) navigate the environment and organize their spatial distribution is as relevant as implementing these behaviours in artificial systems, as discussed in [Part I](#). Controlling the movement and the spatial distribution of microorganisms, cells or artificial micro-agents is a crucial goal to improve our ability to control the microscopic world, with applications ranging from tumor treatment [\[51\]](#) to wound healing [\[141\]](#), from soil and water remediation [\[134, 146\]](#) to population control [\[47\]](#). All these applications require the ability to control the spatial distribution of the agents (i.e. microorganisms or micro-robots) or the expression of a specific behaviour (e.g. expression of a certain protein). Specifically, *spatiotemporal* control aims to regulate some quantity of interest in both space and time, so as, for example, to implement density regulation by prescribing the agents to uniformly distribute in the environment and then, at a specific time, converge towards a certain region. To influence the behaviour of the agents of interest some actuation input is needed. Chemical inputs (e.g. antibiotics, sugars or molecules such as IPTG) have been traditionally, and are still today, used as the main control inputs to steer the cell behaviour. Given the ubiquity of chemical reactions, the right choice of the input molecules allows to influence vir-

tually any biological organism. Moreover, chemicals are easily stored and administered by human operators. On the other hand, chemical inputs, because of diffusion and degradation dynamics, provide poor resolution, both in space and time. In many applications such tight control is not required, for example the spatial patterns generated by swarming bacteria (e.g. *Proteus mirabilis*) have been used to encode information [29]. Combinations of molecules in different concentrations were used as inputs, encoded in the features of the swarming pattern and then decoded using computer vision.

Instead, when precise actuation is required chemical inputs can be replaced by physical ones, such as light [27, 38, 69] or magnetic fields [40]. Both allow unparalleled temporal resolution, with actuation times in the order of milliseconds. Especially light, thanks to the use of digital light projector (DLP), provides high resolution in both space and time, together with fine quantization (usually 1/256) and multiplexing capabilities by combining different colors (usually red, green and blue). Moreover, light can transmit energy and many microorganisms naturally react to it [90], changing, for example, their movement, while others can be genetically engineered to show light induced behaviours [79]. We therefore focused our research on spatiotemporal control of microorganisms using light inputs.

In the following, we will discuss how microorganisms move, and, crucially, how their motion can be modelled. Then, we will deal with the influence of light on their movement and how this can be exploited to achieve spatiotemporal control of cellular populations.

6.1 Movement of microorganisms

Microorganisms can propel themselves using two main solutions, either *cilia*, or one or multiple *flagella* [12, 37] (see Figure 6.1). Flagella are long filaments that protrude from the posterior part of the body. Despite some relevant differences in structure and composition this solution is widespread across Bacteria (e.g. *E. coli*), Archaea and microscopic Eukaryotes (e.g. *Euglena*). Cilia, instead, are exclusive of Eukaryotes. These are small filaments that cover the cell membrane and can address various functions. In particular *ciliate* microorganisms (e.g. *Paramecium*) use cilia for locomotion [80].

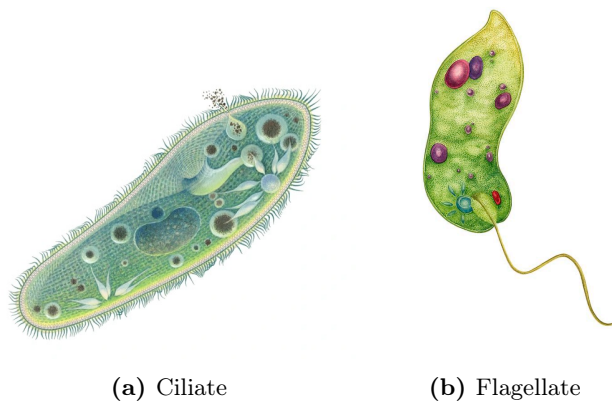


Figure 6.1: Graphical representation of ciliate and flagellate microorganisms.

We are now interested in abstracting the details of locomotion, such as the propulsion mechanism or the biochemistry behind it, to focus on a higher level description, so as to uncover and mathematically describe the kinematic features of the movement and the resulting trajectories. Such stochastic dynamics cannot be described by classical dynamical models using Ordinary Differential Equations (ODEs), therefore stochastic models are introduced to include randomness [20] (see Chapter 2 for more about modelling multi-agent systems).

6.1.1 Run and tumble dynamics

The motion of many microorganisms is often described in terms of “run and tumble” dynamics [8]. This is mostly the case of bacteria, such as *E. coli* [8], but also unicellular Eukaryotes, such as the microscopic alga *Chlamydomonas reinhardtii* [100]. This behaviour is characterized by almost straight movements (runs) alternated by sudden changes of direction (tumbles). Runs are usually longer and executed at a relatively high constant speed. On the contrary, tumbles occur over shorter time and produce a negligible displacement. A similar behaviour is the run-reverse-flick (adopted for example by some marine bacteria such as *Vibrio alginolyticus*) [115]. In this case only 180 degrees (reverse) and ~ 90 degrees (flick)

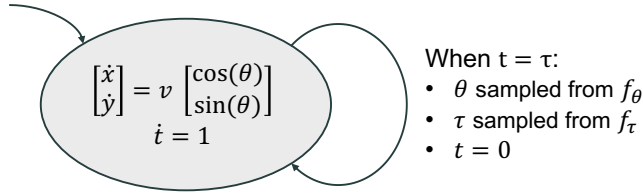


Figure 6.2: Hybrid dynamical model of the Lévy walker.

rotations are present.

Both run and tumble, and run-reverse-flick behaviours can be mathematically described by Lévy walks [145]. This random motion resembles the biological run and tumble. Indeed, the agent moves with a constant velocity v for a randomly distributed time span τ , then randomly selects a new direction θ and start moving again with the same speed. This can be represented as the hybrid dynamical system [54] depicted in Figure 6.2. In this representation, (x, y) is the position of the agent in a 2D space, and $f_\tau : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ and $f_\theta : [-\pi; +\pi] \rightarrow \mathbb{R}_{\geq 0}$ are the probability density distributions of τ and θ respectively.

In the most classic case the duration of each *run* is sampled from either an exponential or a power-law distribution [31], while the new direction after a *tumble* is assumed to be uniformly distributed (i.e. $f_\theta = U_{[-\pi; +\pi]}$). Naturally, many variations of this simple model have been proposed to describe richer behaviours. For example the selection of a new direction might depend on the the current one [9], this allows to model the run-reverse-flick behaviour or the fact that smaller rotations might be more likely (e.g. the average angle between two consecutive movements of *E. coli* is 68° [8]). Other variations have been proposed to model tumbles with non-zero duration [9], or to introduce randomness in the run phase, for example considering some additive random vibration [144]. Moreover, these models can be *continuified* to obtain a macroscopic description (see Section 2.2), that captures the evolution of the ensemble of agents [20, 21].

6.1.2 Persistent Turning Walkers

Many other (micro)organisms do not show such abrupt turns, but smoother trajectories. This is the case of microscopic algae such as *Volvox*

[96] and *Chlamydomonas nivalis* [56], but also animals, such as fishes [41, 149]. To model such smooth and continuous turns it is more suitable to consider the heading direction and the angular velocity as state variables. This is the case of Persistent Turing Walker (PTW) models [41], in which the evolution of the angular velocity is described by a SDE. Such equations extend the framework of ODEs to include randomness. In the most simple case the longitudinal speed v is assumed to be constant, while the evolution of the angular velocity ω is described by a Ornstein-Uhlenbeck process, which is a type of linear SDE, also known as Vasicek models and often used to model the evolution of interest rates or dynamical systems with noise [10]. The resulting model then is

$$\begin{cases} dv &= 0 \\ d\omega &= -\theta\omega dt + \sigma dW. \end{cases} \quad (6.1)$$

where the parameters θ and σ represent respectively the *rate* and the *volatility* of the Ornstein-Uhlenbeck process; while W is a Wiener process. Extensions of this model have been proposed to include a dynamic evolution of the speed in the form of a second Ornstein-Uhlenbeck process, allowing also for dependencies between the speed and the angular velocity [149]. A similar model, specifically developed to describe the 3D motion of *Euglena gracilis*, assumes the agent moves with constant speed while rotating around an axis fixed to its own body frame [128].

The main difference between Lévy walk and PTW models, besides the different smoothness of the trajectories, is in the resulting distributions of the angular velocity. Indeed measuring the speed and the angular velocity of Lévy walkers at some time instants we expect to observe two clearly distinct behaviours, either high longitudinal speed and approximately zero angular velocity or low speed and significant angular velocity. Contrary, for PTW agents we expect the angular velocity to be continuously distributed over a range of values. A second difference is given by the structure of the models, indeed in the PTW model, the evolution of the state variables is explicitly described by SDEs, that can be easily modified to accommodate non-linear terms or exogenous inputs. Last, while Lévy walks allow a straightforward extension to the 3D case, this requires non-trivial adjustments for PTW models, as in [128]. Figure 6.3 shows the comparison

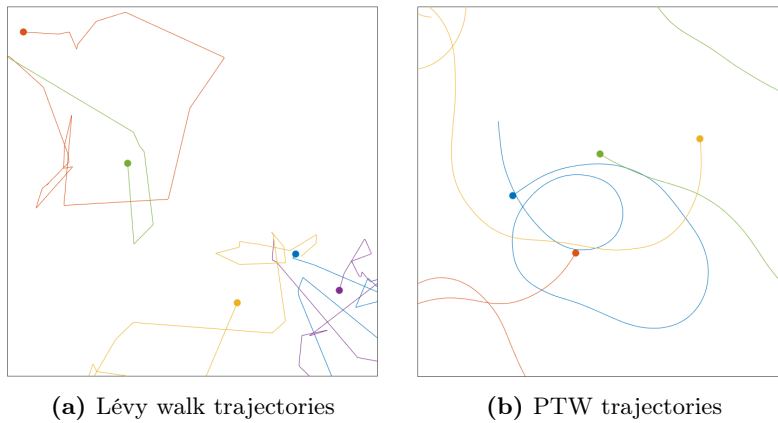


Figure 6.3: Example trajectories generated by Lévy and persistent turning walkers.

between trajectories generated by these two models.

Finally, it is worth to notice that some microorganisms might even show a combination of these two behaviours, with both smooth and abrupt turns. Besides the movement of individual agents, the interactions between them can induce more complex behaviours at the population level. For example, inter-cellular chemical communication can trigger bacterial swarming [7], that is a rapid and highly coordinated movement of colonies of flagellate bacteria; while hydrodynamic interactions between swimming starfish embryos can make them assembly into active chiral crystals [124].

6.2 Light induced behaviours

As previously discussed, light represents an ideal control input to influence the behaviour of micro-agents, either biological or artificial, mainly because high temporal and spatial resolution can be easily achieved [38]. Therefore, we are interested in understanding how light can influence the movement of such motile agents, and, once more, we will disregard biochemical and mechanical details, to focus on the resulting kinematic effects of such light inputs.

Some microorganisms can perform *photomovements*, i.e. react to light and modify their motion in a variety of ways (see [60, 90], or [17] for a

discussion about the nomenclature), namely

- *Photokinesis*: change of the movement speed. Positive if the speed increases with light, negative otherwise.
- *Photoklinokinesis*: change of the turning rate or in the frequency of direction changes. Positive if the turns increase with light, negative otherwise.
- *Phototaxis* or *Phototactic orientation*: alignment with unidirectional light source. Positive if the alignment points toward the light source, negative otherwise.
- *Photophobic response*: sudden stop followed by a change of direction. Such response can be further classified as:
 - *Step-down*: triggered by light decrease.
 - *Step-up*: triggered by light increase.

Multiple of these responses are often present at the same time, in different proportions and possibly induced by different light intensities and wavelengths [50], or depending on light polarization [143]. The combination of the these individual responses can lead to the emergence of macroscopic density regulation behaviours at the population level [110],

- *Photoaccumulation*: accumulation of microorganisms in illuminated areas. It is promoted by step-down photophobic, negative photokinesis and positive photoklinokinesis.
- *Photodispersion*: accumulation of microorganisms in darker areas. It is promoted by step-up photophobic and positive photokinesis and negative photoklinokinesis.

This landscape of behaviours is further enriched by the presence of adaptation, indeed many microorganisms, such as *Paramecium bursaria* [60] or *Volvox* [30], show some adaptation to constant light inputs. This implies that not only the intensity and the color, but also the duration, of light inputs can influence the organisms' response.

6.3 Control applications & experimental platforms

Many applications of spatiotemporal control at the micro-scale have been proposed, and various types of micro-agents have been used. Examples include both biological agents, such as microscopic algae, bacteria or mammalian cells, and artificial micro-robots. Here we discuss some of these applications based on light inputs, with a focus on the proposed experimental platforms.

The movement of microscopic algae *Euglena* (see Section 7.2.1 for details about this microorganism) can be influenced exploiting its light response. In [58] a prototype experimental platform is proposed, which embeds a microscopic camera for vision, while light actuation is provided by four LEDs placed around the sample. These LEDs can be used to induce phototaxis in the *Euglena* and steer their movement direction toward one of the sides. The main features of this platform are the possibility to be used remotely and to automatically handle the loading of the sample, making it more accessible; nevertheless no automatic control application is proposed, either in the original or in subsequent works, such as [139]. A similar solution is proposed in [69]. This platform lacks the remote access but embeds, besides the four LEDs, a digital light projector (DLP), which provides more flexible actuation. Indeed here the phototactic response of *Euglena* can be coupled with the photophobic one, induced by highly bright areas generated by the projector. A simple strategy is proposed to remove the microorganisms from a certain area of the sample, but the feedback is minimal, as it does not take into account any information on the movement of the specimens, and the control strategy consists only in the projection of a constant light pattern and the lighting of one of the LEDs. Therefore, more advanced control strategies may easily improve the performance and solve more general tasks. Another experimental platform, called the Dynamic Optical Micro-Environment (DOME), was introduced in [27] (see Section 7.1 for details about this platform). It lacks the lateral actuation LEDs present in [58, 69], but the embedded projector provides flexible actuation. Indeed, the DOME has been used to influence the movement of microscopic algae *Volvox*, to promote migration of human cells [141], and its use has also been proposed to control micro-robots [131]. Specifically, it was shown in [27] that the movement of *Volvox* (see Section 7.2.1 for

details about this microorganism) can be influenced using repeating light inputs to induce negative photokinesis and, thus, reducing their speed. This was first demonstrated in [27] using an open loop strategy, and then improved in [28] with the use of Q-learning. Also motile micro-robots, made of contractile and light-reactive units, are currently being developed and tested in the DOME [131].

Spatiotemporal control has also been applied to bacteria. The speed and the spatial distribution of genetically engineered *E. coli* were controlled in [84] using light inputs. Specifically the bacteria were engineered to show positive photokinesis. A computer vision algorithm was then used to detect the microorganisms and a bright spot was projected next to each agent, towards the desired direction of movement, so that only the agents moving in the desired direction exhibited augmented speed. This was shown to produce an overall flow of bacteria towards the desired area. Light sensitive bacteria were also used to propel micro-robots. In [121] multiple specimens of *Serratia marcescens* were attached on a star-shaped plate. The propulsion generated by the bacteria induces a rotation of the micro-robot, whose velocity can be influenced via UV light acting on the swimming of the bacteria. Acting directly on the movement of bacteria is not the only way of controlling their spatial distribution. For example the relative distribution of two bacterial populations is spatially controlled in [135] by the use of selective light-induced antibiotics. In some application the physical distribution of the bacteria might not be the only quantity of interest, for example spatiotemporal control of the expression of green fluorescence protein, is implemented in [79], using genetically engineered *E. coli* and light inputs of different colors.

Light can also be used to influence the migration of mammalian cells. The DOME, modified to project UV light, was used in [141] to induce DNA damage in the cells at the edges of a wound, accelerating their migration and promoting healing of the wound. While sub-cellular lighting was used in [126] to influence the steering of migrating human cells.

6.4 Discussion

In this Chapter we described the concept of spatiotemporal control of microorganisms and how to achieve it. Specifically, we discussed how

microorganisms move in their environment, and two different approaches to mathematically model their stochastic motion. We then discussed how light influences both, the movement of the individual organisms, and the macroscopic behaviour of the population. Finally, we reviewed the existing solutions that exploit light to achieve spatiotemporal control of micro-agents, including microscopic algae, bacteria and mammalian cells. The main outcome of this analysis is that, despite various approaches having been proposed to control the movement of such micro-agents, a comprehensive framework is still missing. Indeed, most control strategies are designed heuristically and specific to a single species. This may be explained by the lack of mathematical models describing the motion of such microorganisms, and how it is influenced by light.

In the following we propose a general methodology to build such models, without a priori knowledge on the behaviour of a specific species, but using experimental data. The next Chapter will present our experimental setup, describing the experimental platform and the species of microorganisms we studied. Moreover, it will discuss the experiments we executed, and how the resulting data allow to characterize the movement and the light response of the microorganisms. Chapter 8 will describe our modelling approach, and how experimental data can be used to estimate the values of the parameters and validate the resulting model. Finally, we will discuss how these models allow to improve our spatiotemporal control over the microorganisms, and ultimately achieve density regulation.

Experimental setup and data processing

As discussed in the previous Chapter characterizing the behaviour of microorganisms is a crucial step towards their spatiotemporal control. Specifically, we are interested in studying the movement of different species of motile microorganisms, together with their response to light. Such study starts from experimental data to characterize and model the most relevant features of these behaviours, and then allow the design of feedback control laws.

In this chapter we will describe the experimental setup used to study the movement of microorganisms and their response to light, the algorithmic pipeline to analyse the data, and the results of such analysis.

7.1 DOME: the experimental platform

Our research was carried out using the experimental platform called the Dynamic Optical Micro-Environment (**DOME**). This low-cost and open-access equipment was designed by Denniss et al. [27] at the Bio-ComputeLab (www.biocomputelab.github.io) to study the behaviour of microscopic agents (e.g. protozoa, micro-robots [131], tissue samples [141], etc.) under light inputs. Such platform extends the functionalities of a classic digital microscope by implementing, besides the imaging capabilities, the possibility of acting on the sample with light inputs, and embedded

computing. These features, thanks to both the hardware design and the software we developed, allow high-throughput data acquisition and the execution of feedback control strategies.

7.1.1 Hardware

The **DOME** is made of two main blocks arranged around the sample, the sensing and the actuation blocks (see Figure 7.1). The first unit is responsible for the acquisition of the microorganisms images. It is made up of an HD camera, a light filter, a magnifying tube lens and a magnifying objective lens. The objective lens are optional and replaceable, allowing to adjust the magnification level depending on the size of the microscopic agents to be studied. The tube lens provides a first magnification and can be moved vertically to adjust the back-focus distance from the camera sensor. The light filter is an optional and easily replaceable component that can be chosen to image only specific wavelengths. For example, a long-passing filter, with the right cutting wave length, will absorb green and blue light, allowing only red light to reach the camera. The second unit allows for the spatiotemporal control of the light inputs delivered to the organisms. It consists of a condenser lens, a projector and a Raspberry Pi board. The Raspberry Pi ZERO board drives the digital light projector (**DLP**), while the condenser lens focuses the projected light on the sample. The two blocks are connected by a Raspberry Pi 4, which is connected to the camera, runs the software to acquire the images, process them and compute the light pattern to be projected. This is sent via an ad-hoc WiFi connection to the Raspberry Pi ZERO driving the projector. The sample stage is placed between the sensing and actuation blocks. Its vertical position can be adjusted to focus the camera on the sample, while a caliper controls the horizontal position. The liquid sample is loaded on either a Petri dish or a microscopy slide. Specifically, we used microscopy slides fitted with a 3D printed plastic frame, allowing the use of larger liquid samples (up to 300 μL) [27]. For further details about the **DOME** refer to [27] or visit www.theopendome.org.

The projection and imaging specifications of the **DOME** are reported in Table 7.1, while the components used in our setup are listed in Table 7.2. Additionally, to interact with the **DOME**, we used a USB keyboard, a mouse, and an HDMI monitor connected to the Raspberry Pi 4. Note

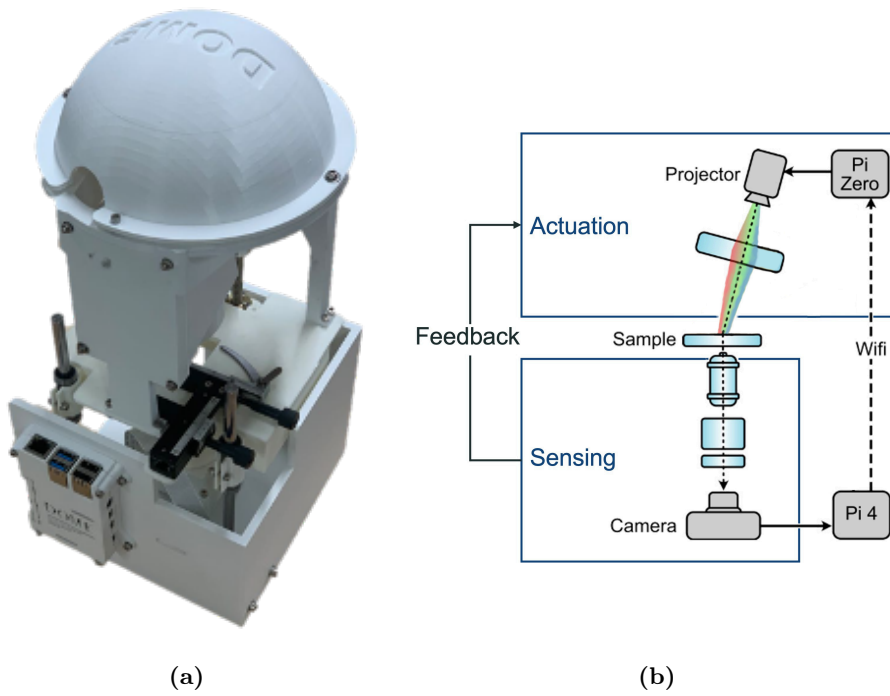


Figure 7.1: DOME experimental platform: (a) picture and (b) schematic adapted from [27].

Total mag- nification	Imaging		Imaging	Projection		Projection
	pixel [μm]	size	resolution [px]	pixel [μm]	size	resolution [px]
9X	4.04 \times 4.04		1080 \times 1920	30 \times 30		480 \times 854
90X	1.25 \times 1.25					

Table 7.1: Projection and imaging specifications of the [DOME](#) in our setup.

that we used the [DOME](#) in two different configurations, either with a 10X objective lens (90X total magnification) or without (9X total magnification), this allowed us to work with microorganisms of very different sizes. Moreover, we used a long wavelength passing filter to screen the camera sensor from blue light.

7.1.2 Software

To readily use the [DOME](#) we developed a software package. The package is written in Python and made of two parts. The first includes scripts working on Raspberry Pi OS to control both the Raspberry Pi boards embedded in the [DOME](#). It manages the WiFi communication between the two boards, and allows to acquire images and video from the camera, project light patterns, run experiments and save the data in a structured way. The second part collects scripts, that can be executed on any PC, to read the data acquired during the experiments, perform the automatic tracking (see [Appendix B](#)) of the microorganisms from the images and analyze the resulting trajectories. This package is currently being refined and will soon be made available.

7.2 Experiments with microorganisms

7.2.1 Microorganisms

For our experiments we selected four species of microorganisms (see [Figure 7.2](#)), two *Paramecia* (*P. caudatum* and *P. bursaria*) and two microscopic algae (*Volvox* and *Euglena gracilis*). Together these species represent a significant sample of motile and light sensitive eukaryotic microor-

	Component	Description	Manufacturer
Optics	Projector	DLP evaluation module (DLPM2000EVM)	Texas Instruments
	Condenser Lens	Ø50mm×44mm FL, PCX condenser lens	Edmund Optics
	Tube Lens	9X tube lens	Edmund Optics
	Objective Lens	10X DIN semi-plan standard objective	Edmund Optics
	Long Passing Filter	Red light passing filter, Ø25.4mm to cut out blue light from the camera	Edmund Optics
	Camera Lens	Ø25mm uncoated glass window to seal the camera.	Edmund Optics
	O-ring	23.5-25.5mm O-ring to seal the camera	
Electronics	Raspberry Pi 4	Model B, 4GB	Raspberry Pi
	Raspberry Pi ZERO	Raspberry Pi ZERO W	Raspberry Pi
	Camera	Raspberry Pi High Quality Camera Module	Raspberry Pi
	Camera Connector	Flexible cable for Raspberry Pi Camera - 300mm	Raspberry Pi
	MicroSD Card	MicroSD Card (Class 10 A1) 32GB	SanDisk
	Raspberry Power Supply	Official Raspberry Pi 4 power supply (5.1V, 3A)	Raspberry Pi
	Power Supply	Plug-in power supply (20W, 5V, 4A)	RS
Interface PCB	Custom board to connect the Pi ZERO adapter with the projector		
Mechanical	3D printed parts	Custom 3D printed parts	
	Linear Rail Set	Set of lead screw, linear bearing, rod rail support	Glvinc
	X-Y Caliper	Caliper for the stage	
	Linear Ball Bearing	Long linear motion ball bearings	

Table 7.2: Components of the **DOME** in our setup. For up to date information visit www.theopendome.org.

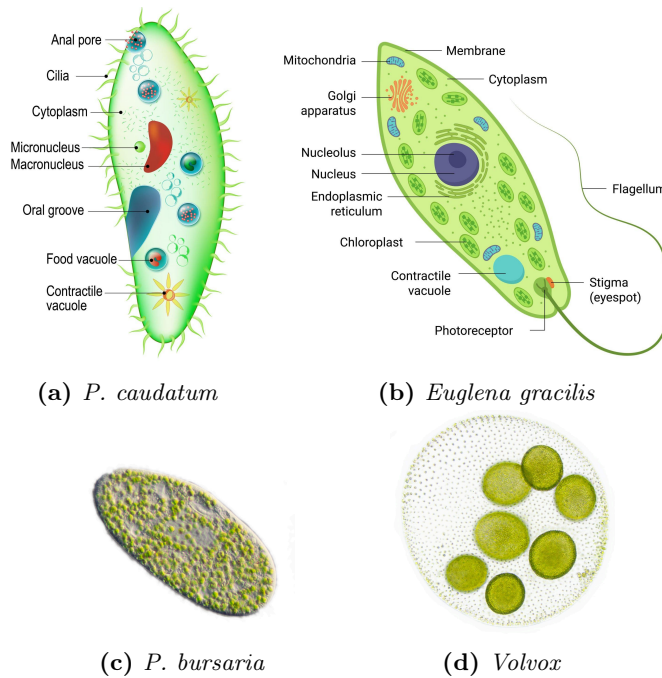


Figure 7.2: Species of microorganisms used in the experiments. (a) Diagram of *P. caudatum*. (b) Diagram of *Euglena gracilis*. (c) Picture of *P. bursaria*, where symbiotic *Chlorella* are easily visible. (d) Picture of a *Volvox* colony.

ganisms, that will allow to test the wide adaptability of both the experimental platform and the analytical pipeline.

Paramecium caudatum is a common species of paramecium widespread in stagnant and fresh water environments [140]. The locomotion along the main axis is characterized by a spiral path and powered by the cilia. It shows photophobic step-down response [94] (see Section 6.2 for an introduction to light responses of microorganisms), with peak response to UV light (300-330nm) and a linear correlation between the intensity of light and the percentage of responding agents. Compared to the other three species *P. caudatum* is large (200-300 μm) and fast moving (1100-1500 $\mu\text{m/s}$ [94]), but its light response is weaker and less documented.

Paramecium bursaria is another ciliate fresh water species [140]. It is characterized by the symbiotic relationship with the *Chlorella* microscopic

algae, which can live inside the these microorganisms and influences their response to light stimuli. Specifically, the light-adapted (i.e., cultured in a lit environment for a few hours before the experiment) *Chlorella*-containing paramecium show a combination of step-down photophobic response, negative photokinesis and positive klinokinesis [25], resulting in photoaccumulation. Conversely, *Chlorella*-containing dark-adapted (i.e., cultured in a dark environment for a few hours before the experiment) and *Chlorella*-free specimens perform step-up photophobic response, resulting in photodispersion [60]. Both behaviors are predominantly induced by blue and green light (440-680nm). The complex light response of *P. bursaria* results in a large literature, but often unclear and even contrasting results [25, 60, 107]. Moreover, the small size (85-150 μm) combined with a low population density (0.4-1.0 specimens/ μL) and a fast movement (1500-1900 $\mu\text{m/s}$ [60]) make the simultaneous imaging of multiple organisms problematic.

Euglena gracilis is a fresh water unicellular alga. It swims rolling around its main axis and is propelled by a flagellum. It shows strong light responses and, therefore, is commonly used in studies on light-responsive microorganisms [58, 69, 139]. Specifically it shows phototaxis (positive when light intensity is low and negative otherwise [49]), and a strong step-up photophobic response [128], resulting in a clear photodispersion behaviour, with peak response induced by blue light (420-480nm). It is the smallest within the species we considered (50-90 μm), but the relatively slow movement (30-120 $\mu\text{m/s}$ [88, 128]) combined with a high population density (50-70 specimens/ μL) implies that a significant amount of organisms can be contained in a relatively small volume and be observed with higher magnification.

The last species we worked with is *Volvox*. These unicellular algae form spherical colonies of up to 50,000 cells. The colonies, despite being almost spherical, have a main axis dictating the internal structure and the direction of movement. The propulsion is generated by the coordinated motion of the cilia of the cells on the surface of the colony. They show negative photokinetic response to green light (470-530nm) [27], resulting in photoaccumulation [130] when the light intensity is not too high, otherwise a photodispersion behaviour emerges [30]. *Volvox* is the largest species we worked with (350-500 μm), moreover their spherical shape and the relatively slow movement with respect to the body size (300-600 $\mu\text{m/s}$

Species	Size [μm]	Speed [$\mu\text{m/s}$]	Peak light sens. [nm]	Ligth response
<i>P. caudatum</i>	200-300	1100-1500	300-330	Step-down photophobic.
<i>P. bursaria</i>	85-150	1500-1900	440-680	Light-adapted: step-down photophobic, negative photokinesis and positive klinokinesis, photoaccumulatio. Dark-adapted or <i>Chlorella</i> -free: step-up photophobic response, photodispersion
<i>E. gracilis</i>	50-90	30-120	420-480	Phototaxis (negative or positive depending on the intensity), step-up photophobic, photodispersion.
<i>Volvox</i>	350-500	300-600	470-530	Negative photokinesis, photoaccumulation.

Table 7.3: Species of microorganisms used in the experiments and their main features.

[96, 114]) facilitate the automatic detection and tracking.

These species show remarkably different features (i.e. size, movement speed, light response, etc), and together represent a significant sample of motile eukaryotic microorganisms. See Table 7.3 for a summary.

P. caudatum, *Volvox* and *Euglena* were sourced from BladesBio UK (www.blades-bio.co.uk), while *P. bursaria* were sourced from Carolina Biological Supply (www.carolina.com). *Volvox* were cultured in Alga Grow medium, *Euglena* in its ad-hoc medium and both *Paramecium* species in Protozoan Pellet medium. *Euglena*, *Volvox* and *P. bursaria* were kept under an artificial light bank with a daily activation time of 12h¹.

7.2.2 Experimental protocol

To collect a rich data-set and to be able to characterize both the movement in a dark environment and the light response of the sample microor-

¹Culturing and media preparation protocols were taken from www.carolina.com/teacher-resources/Document/protozoa-invert-care-handling-instructions/tr10466.tr.

Species	Magnification	Density [μL^{-1}]	Sample volume [μL]	Plastic frame
<i>P. caudatum</i>	9X	0.19-0.95	100-150	Yes
<i>P. bursaria</i>	9X	0.4-1.0	20-100	Only for larger volumes
<i>E. gracilis</i>	90X	50-70	15-20	No
<i>Volvox</i>	9X	0.2-0.8	120-150	Yes

Table 7.4: Experimental parameters.

ganisms we performed a set of open loop experiments, projecting predefined spatial and temporal light inputs. Each experiment lasts 3 minutes with the acquisition of an image every 0.5s. The sampling time was chosen due to constraints on the execution frequency given by the platform, while the duration of the experiments was selected large enough to collect abundant data and be able to observe slow dynamics, such as adaptation, while keeping a reasonable burden in terms of storage memory and time.

For each experiment a given volume (see Table 7.4) is taken from the culture jar and placed on the microscopy slide, eventually in the well made by the plastic frame. The sample is then placed on the sample stage and the DOME covered with a dark hood to screen external, not controlled, light. The projector is then set to shine constant and low intensity red light (640nm, 5% of the projector’s maximum red brightness), providing the illumination for dark-field imaging. Blue light (460nm), being within the sensibility range of most microorganisms, is then used as input. The presence of a red light filter prevents blue light from reaching the camera sensor, guaranteeing a constant illumination for better image acquisition². Due to the different sizes of the species involved in the experiments we used 90X magnification, when working with *Euglena*, while the other species were imaged at 9X magnification, see Section 7.1 for details about the components.

We run different experiments to characterize (i) the movement of the microorganisms in absence of light inputs, and (ii) their response to light inputs of different intensities and duration. Specifically we run experi-

²The light filter was not present during preliminary experiments, resulting in uneven quality images due to sudden changes in luminosity.

	Experiment	Description
Default	No input	Constant OFF, only red background illumination.
Intensity	30% Intensity	1min OFF, 1min with 30% input, 1min OFF.
	60% Intensity	1min OFF, 1min with 60% input, 1min OFF.
	100% Intensity	1min OFF, 1min with 100% input, 1min OFF.
	Ramp	10s OFF, linearly increasing input from 0% to 100%, 10s with constant 100% input.
Frequency	Switch 10s	10s OFF, repeating ON-OFF input with 20s period and 50% duty cycle.
	Switch 5s	10s OFF, repeating ON-OFF input with 10s period and 50% duty cycle.
	Switch 1s	10s OFF, repeating ON-OFF input with 2s period and 50% duty cycle.

Table 7.5: Types of experiments executed to characterize the movement and light response of microorganisms.

ments with light inputs of different duration, from 1 to 60 seconds, and different intensity, ranging from 0% to 100% of the projector’s maximum blue brightness. Table 7.5 and Figure 7.3 show a short description and a graphical representation of the different types of experiments, while Figure 7.4 shows examples of the images acquired during the experiments. Each experiment was repeated, for each species, for a minimum of 3 biological and 2 technical replicates, resulting in more than 300 experiments.

7.3 Data extraction and analysis

To study the movement of microorganisms we need to extract the relevant kinematic variables (position, velocity, orientation, etc.) from the images acquired during the experiments. Therefore, we developed a software that elaborates these images, automatically detects the microorganisms and tracks their position over time (see Appendix B for more details about the tracking algorithm).

A preliminary processing of the recorded trajectories was performed

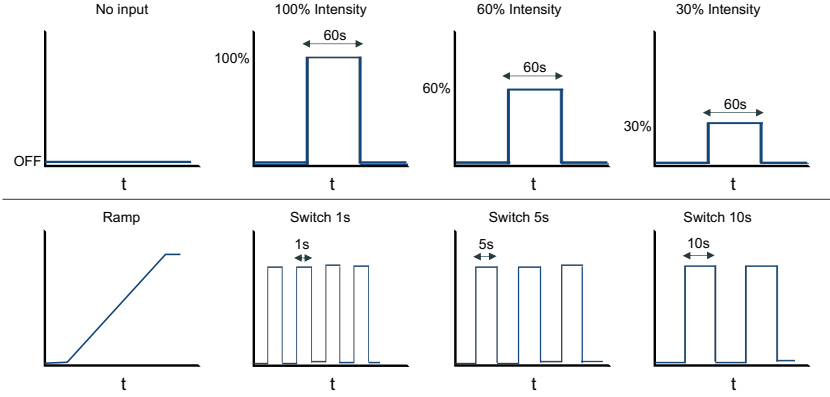


Figure 7.3: Graphical representation of the types of experiments executed to characterize the movement and light response of microorganisms.

prior of the data analysis. Specifically, the trajectories are selected according to their duration, and those shorter than 5s are discarded. This helps in removing erroneously detected trajectories, and ensures remaining ones have a sufficient number of points. The remaining trajectories are then smoothed with a moving average of window size 3 to reject noise or high frequencies movements. The velocity vectors are computed using second order accurate central differences, to then extract the longitudinal speed (i.e. the magnitude of the velocity vector) and the angular velocity (i.e. the rate of change of the movement direction). Specifically, given a single trajectory of points $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ in the camera frame $[0; 1080] \times [0; 1920]$, sampled every ΔT seconds, the velocity is computed as

$$\mathbf{v}_k = \begin{cases} \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{\Delta T} & \text{if } k = 1; \\ \frac{\mathbf{x}_k - \mathbf{x}_{k-1}}{\Delta T} & \text{if } k = n; \\ \frac{\mathbf{x}_{k+1} - \mathbf{x}_{k-1}}{2\Delta T} & \text{otherwise} \end{cases} \quad (7.1)$$

and measured in px/s. Then the speed v_k is simply given by the norm of \mathbf{v}_k , while the angular velocity ω_k , measured in rad/s, is computed as

$$\omega_k = \frac{1}{\Delta T} \operatorname{atan2}\left(\frac{\mathbf{v}_k \times \mathbf{v}_{k+1}}{\mathbf{v}_k \cdot \mathbf{v}_{k+1}}\right) \quad (7.2)$$

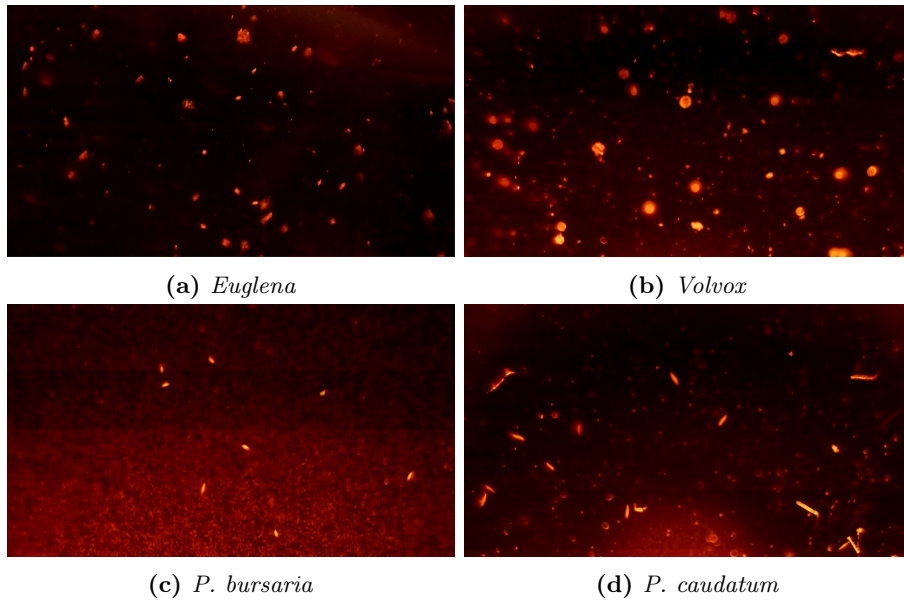


Figure 7.4: Example images acquired during the experiments. The experimental conditions are those reported in Table 7.4.

Each of these time series is then further smoothed by applying again the moving average of size 3.

We then check possible tracking errors, by applying Algorithm 7.1, with $m = 2.5$, to detect the outliers in the resulting speed data. For each outlier the corresponding agent and time instant are reported to the user that can visually check the tracking video, and, in case of errors, correct it by removing erroneous trajectories or merging multiple trajectories belonging to the same microorganism.

Note that methods described in this and the following pages hold for any of the species we considered, but, for the sake of time, we have, up to this point, only analysed the data from the experiments with *Euglena*.

7.3.1 The behaviour in a dark environment

First we characterized the behaviour of microorganisms without light inputs (i.e., in a dark environment), with only low intensity red illumina-

Algorithm 7.1 Given a data-set (vector or multidimensional array) d and a positive threshold m , the function `detect_outliers(d, m)` identifies the outliers in the d .

```

function detect_outliers( $d, m$ )
  require  $m > 0$            # Check threshold
  outliers = empty list # Initialize empty list
  for  $k$  in indices( $d$ ) do
    # Compute normalised variation
     $s = |d[k] - \text{median}(d)| / (|d - \text{median}(d)|)$ 
    if  $s > m$  then
      outliers.append( $k$ ) # Insert  $k$  in the list of outliers
    end if
  end for
  return outliers

```

tion. Figure 7.5 shows the time evolution of speed and angular velocity from an experiment with *Euglena* (see Section 7.2 for details about the experiments). It can be easily observed that the data are very variable, but their macroscopic features (i.e., mean and variance) are consistent over time.

We then statistically characterized these data. Specifically, we computed the mean speed and mean absolute angular velocity of each agent, and observed a clear negative correlation between these quantities (see Figure 7.6a), with faster agents having a lower angular velocity. Nevertheless, we notice this might be a byproduct of discrete time sampling and numerical differentiation. Indeed, if during a certain time step an agent turns, while keeping a constant longitudinal velocity, at the next sampling time the displacement from the initial position will be smaller, and numerical differentiation will result in a smaller velocity.

Moreover, we studied the deviation of agents from their average behaviour. Figure 7.6b shows a non monotonic dependence between the standard deviation and the mean of the speed of the agents, while Figure 7.6c shows the expected linear correlation between standard deviation and mean value of the angular velocity. Therefore, we can conclude that faster agents have more consistent motion, with lower angular velocity and proportionally less variable speed.

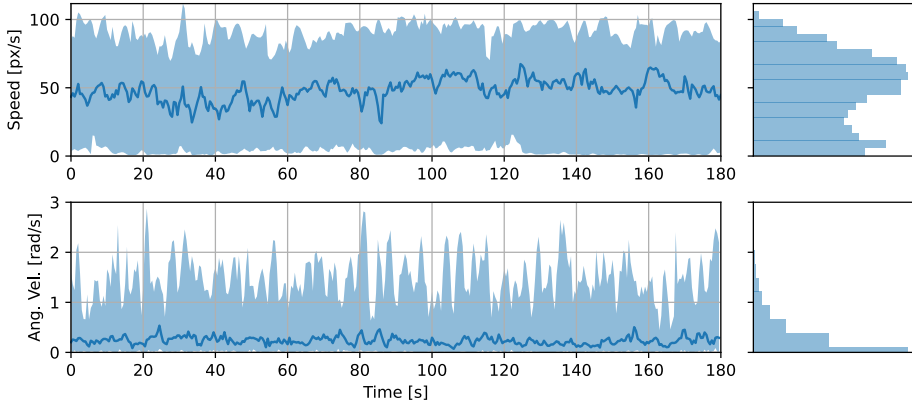


Figure 7.5: Longitudinal speed and absolute angular velocity of *Euglena* specimens in a dark environment. The solid lines show the sample’s median, while the shaded areas show the minimum and the maximum.

Moreover, none of the plots in Figure 7.6 shows clearly separated clusters, this suggests the presence of a single homogeneous, yet variable, population; rather than the coexistence of more populations with clearly different behaviours.

7.3.2 The influence of light

We then characterized the response of the microorganisms to light inputs. Figure 7.7 shows the behaviour of *Euglena* with light inputs of different duration, respectively 60s, 10s and 5s, and spatially uniform (see Section 7.2 for details about the experiments). The time evolution of the quantities of interest clearly shows a rapid (i.e. in approximately one second) decrease of the speed and a simultaneous increase of the absolute angular velocity when the blue light is switched on. In the following ~ 10 s both quantities return back, close to the original values, showing a strong adaptation capability of this species. This implies that a constant illumination or very fast bursts (i.e. during a second or less) do not cause large effects in the long term, while inputs with intermediate duration (i.e. during between 5 and 10 seconds) do (see Figure 7.9a). These effects are compatible with the well known step-up photophobic response of *Euglena*

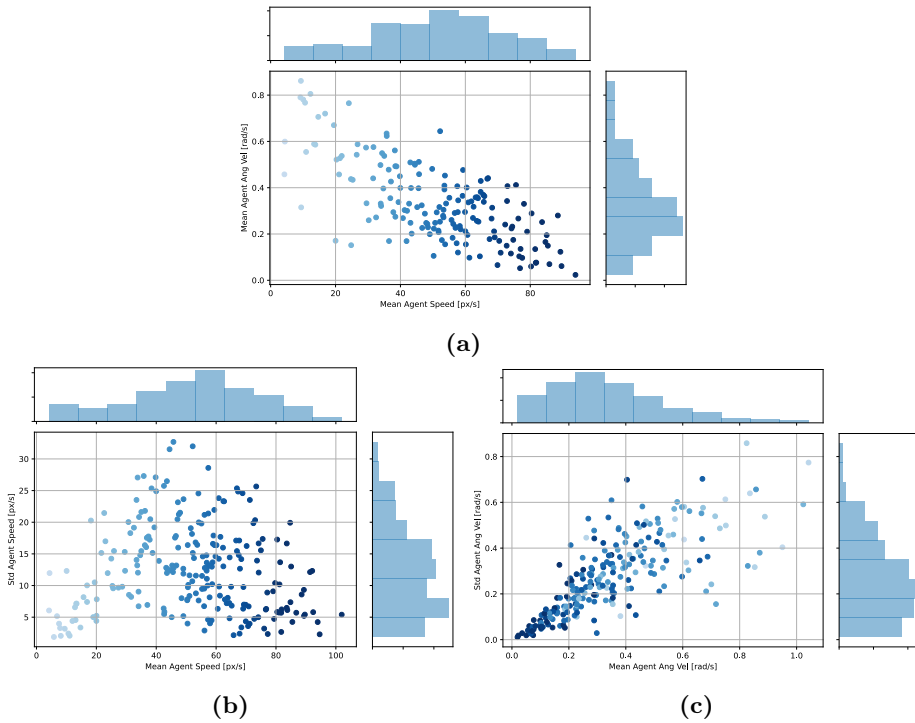


Figure 7.6: Characterization of the motion *Euglena* specimens in a dark environment. (a) Average speed and average absolute angular velocity of the agents. (b) Average and standard deviation of the speed of the agents. (c) Average and standard deviation of the absolute angular velocity of the agents. Each dot represents one agent and is colored according to the mean speed of the agent, so that darker points correspond to faster agents.

Scenario	Speed [px/s]			Angular velocity [rad/s]		
	OFF	ON		OFF	ON	
No input	48.1±25.7	————		0.35±0.40	————	
30% Intensity	49.6±27.3	49.6±27.0	+0%	0.40±0.47	0.43±0.47	+6%
60% Intensity	46.6±27.3	46.2±26.9	-1%	0.40±0.47	0.46±0.49	+14%
100% Intensity	51.4±27.9	49.5±26.9	-4%	0.34±0.43	0.39±0.45	+14%
Switch 1s	45.8±26.0	46.8±25.9	+2%	0.41±0.44	0.44±0.46	+7%
Switch 5s	46.9±29.0	35.0±24.9	-25%	0.38±0.50	0.59±0.56	+57%
Switch 10s	51.4±28.2	39.2±26.7	-24%	0.32±0.43	0.53±0.55	+67%

Table 7.6: Experimental results on the movement of *Euglena*. Data are obtained aggregating the results of three biological replicates per case. Shown values are mean \pm standard deviation and, for light ON case, percentage variation of the mean value with respect to the light OFF case.

to blue light (see Table 7.3), and, potentially, weaker negative photokinesis and positive photoklinokinesis.

We then studied how the light intensity affects the response of *Euglena*. Figure 7.8 shows the effect of light inputs with different intensities, respectively 100%, 60% and 30%, and spatially uniform. The results suggest that the intensity of light do not affect significantly the response, as even the minimum tested intensity is sufficient to trigger the photophobic response. Figure 7.9b further supports this finding, and shows that, regardless of input intensity, long exposure does not cause significant effects in the long term. These results are summarised in Table 7.6, where can be easily observed that switching inputs generate the largest effects, with up to 25% speed decrease and 67% increase in the angular velocity.

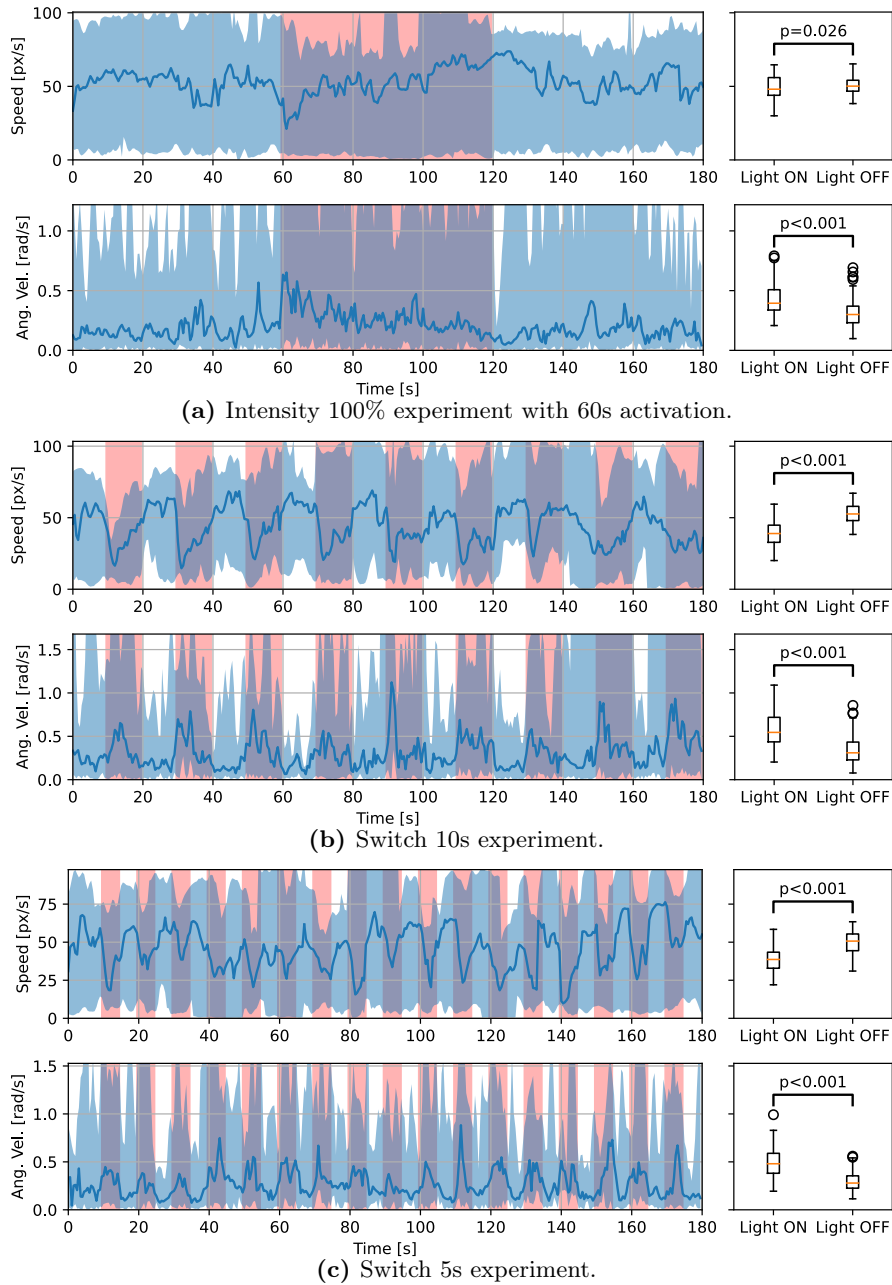


Figure 7.7: Response of *Euglena* to light inputs of different duration. The red bands indicate the periods in which the light inputs were active. Boxplots show the distribution of the mean speeds (or angular velocity) of the agents. Statistical significance is computed by Mann-Whitney U test [83].

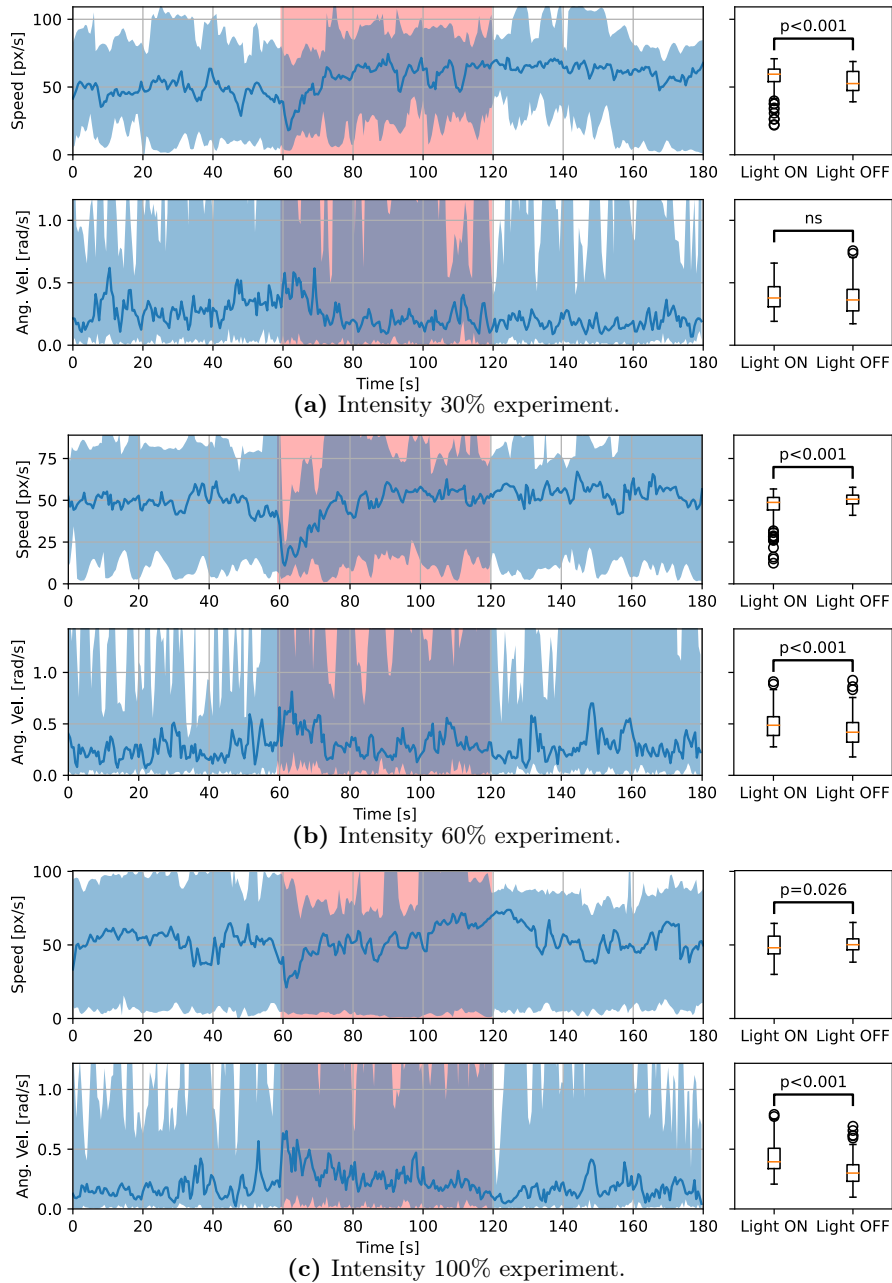


Figure 7.8: Response of *Euglena* to light inputs of different intensities. The red bands indicate the periods in which the light inputs were active. Boxplots show the distribution of the mean speeds (or angular velocity) of the agents. Statistical significance is computed by Mann-Whitney U test [83].

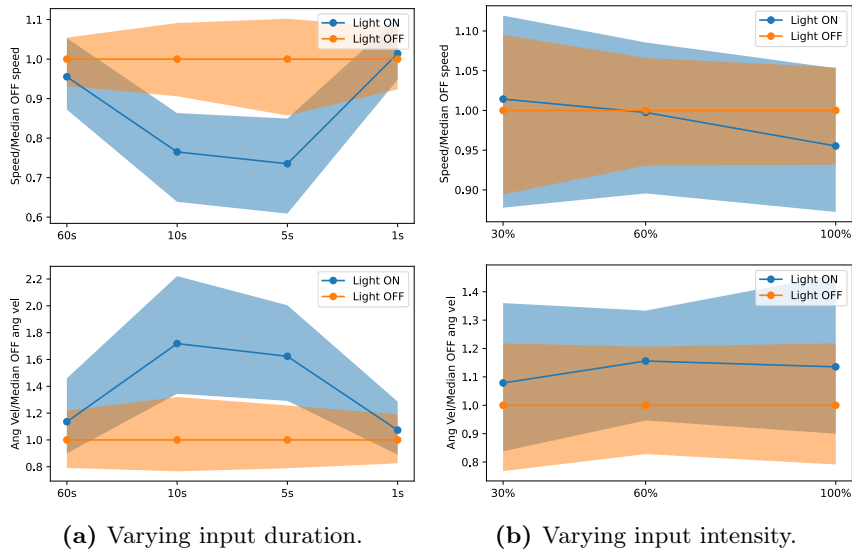


Figure 7.9: Response of *Euglena* to light inputs of different (a) duration and (b) intensities. The plots are obtained by aggregating the results from three replicates for each case. For each experiment the average over the agents is computed, and normalized on the median value recorder when the light inputs are absent. The solid lines show the median, while the shaded areas show the range between first and third quartiles.

7.4 Discussion

In this Chapter we introduced our experimental setup, and characterized the movement and the light response of *Euglena*. We started by describing the **DOME**, an innovative experimental platform that was crucial for our research. In particular, the ability to stimulate the microorganisms with light inputs allowed to study their response. Then, we introduced the four species of motile microorganisms used in our study, and the series of experiment designed to investigate their response to light stimuli. Finally, we analysed the kinetics of the movement of *Euglena*, studying their behaviour in a dark environment and the modifications induced by light. In particular, we observed these microorganisms to strongly react to sudden luminosity changes by slowing down and increasing their turning rate. These results will be used in the next Chapter to direct the modelling of the motion of these microorganisms.

Data-driven modelling

Mathematical models allow to achieve better understanding of the experimental results and new insights on the studied system, through both simulation or analysis. Moreover, dynamical models are a fundamental tool for the design of control strategies. Nevertheless, as discussed in Chapter 6, general approaches to model the movement and the light response of microorganisms are currently lacking, hampering the development of effective control strategies. Therefore, this Chapter will introduce our data-driven approach to model stochastic mobile agents, which also allows to include inputs and might be used to describe the movement of a variety of microorganisms, together with their response to light. Specifically we will use the data presented in Section 7.3 to parameterize and validate the model, using *Euglena* as a representative example. Finally, we will present some preliminary results in the direction of controlling the spatial distribution of microorganisms.

8.1 Stochastic model of motion

Models of dynamical systems usually consist of ODEs, nevertheless, these descriptions cannot grasp the nature of highly stochastic behaviours, as often found in biology. In Section 6.1, we discussed two of the most common approaches to model the stochastic movement of swimming organisms, namely the Lévy walk [8] and the Persistent Turing Walker (PTW) [41]. Given that we observed *Euglena* to be continuously moving and turning

for most of the time, that we are only interested in modelling 2D trajectories, and that the PTW model can be easily adapted to include inputs, we adopted this modelling framework. Specifically, we use two Stochastic Differential Equations (SDEs) to model, respectively, the time evolution of the speed v_i and the angular velocity ω_i [149]. So that, for each agent i , we have

$$\left\{ \begin{aligned} dv_i &= [\theta_{v,i}(\mu_{v,i} - v_i) + \alpha_{v,i}u_i + \beta_{v,i} \max(\dot{u}_i, 0)]dt + \sigma_{v,i}dW_{v,i} & (8.1a) \\ d\omega_i &= [-\theta_{\omega,i}\omega_i + \text{sign}(\omega_i)(\alpha_{\omega,i}u_i + \beta_{\omega,i} \max(\dot{u}_i, 0))]dt + \sigma_{\omega,i}dW_{\omega,i} & (8.1b) \end{aligned} \right.$$

where $\theta_{\cdot,i}$ represents the *rate*, $\mu_{\cdot,i}$ the *mean*, $\sigma_{\cdot,i}$ the *volatility*, $\alpha_{\cdot,i}$ and $\beta_{\cdot,i}$ the input gains of the corresponding equation; while u_i is the intensity of the light input measured by agent i and \dot{u}_i its time derivative; finally, $W_{\cdot,i}$ represent independent standard Wiener processes. Notice that we assumed zero mean for the angular velocity (i.e. $\mu_{\omega,\cdot} = 0$), giving the agents no preference between left or right turns, but, for all the other parameters, we are allowing different values for each agent.

Concerning the agents' reaction to light inputs, we included two terms in each equation, describing the effects of light intensity and its time derivative. In particular, by using the term $\max(\dot{u}_i, 0)$ we only considered the effect of positive variations of the inputs, that can indeed trigger the step-up photophobic response of these microorganisms¹. Moreover, the term $\text{sign}(\omega_i)$ in (8.1b) keeps the consistency between the parameters $\alpha_{\omega,i}$ and $\beta_{\omega,i}$, describing the variation of the absolute value of ω_i , and ω_i itself, being either positive or negative. All together this model allows to capture any combination of photokinetic (through $\alpha_{v,i}$), photoklinokinetic (through $\alpha_{\omega,i}$), and step-up photophobic responses (through $\beta_{v,i}$ and $\beta_{\omega,i}$) (see Section 6.2 for more information about possible light responses of microorganisms).

Compared with the original PTW model [41], which assumed constant speed, our model can capture richer dynamics, allowing for fluctuations of the longitudinal speed. This improvement was already proposed in [149], which also introduced an explicit dependency between the speed and the angular velocity. While, for the sake of simplicity, we assumed the two variables to be independent, we introduced input terms in both

¹Here we decided to only model the step-up photophobic response, nevertheless the extension to also include the step-down response is straightforward. It can be implemented by inserting an additional term $+\gamma_{\cdot,i} \min(\dot{u}_i, 0)$ in both equations.

equations (8.1). This, allows our model to capture the influence of external factors, such as light, that can, indeed, induce a coupling between speed and angular velocity. A similar model, specifically designed to capture the step-up photophobic response of *Euglena g.*, was proposed in [128]. This, while describing 3D movement and some light responses, assumes constant longitudinal speed, so that some crucial aspects, such as velocity fluctuations and photokinesis, cannot be captured.

8.2 Model parametrization

Model 8.1 is linear in the parameters, allowing for a relatively straightforward parametrization. Therefore, we extended the calibration technique described in [133] to identify the values of the parameters.

We need to identify the parameters in both (8.1a) and (8.1b), which have the same structure,

$$dx = [\theta(\mu - x) + \alpha u]dt + \sigma dW. \quad (8.2)$$

Assume we can acquire data points $\{x_1, x_2, \dots, x_n\}$ and $\{u_1, u_2, \dots, u_n\}$ from x and u , with a sampling time step ΔT . Then, Ordinary Least Squares (OLS) regression [39] can be used to fit the straight line $x_{k+1} = ax_k + bu_k + c$ to these data, obtaining estimates for the parameters a , b and c , and the residuals $\epsilon_k = x_{k+1} - (ax_k + bu_k + c)$.

Also, discretizing (8.2) we have

$$x_{k+1} = e^{-\theta\Delta T} x_k + \mu(1 - e^{-\theta\Delta T}) + \int_0^{\Delta T} e^{-\theta\tau} b d\tau u_k + \sigma \sqrt{\frac{1 - e^{-2\theta\Delta T}}{2\theta}} N_{0,1}, \quad (8.3)$$

where $N_{0,1}$ is a normally distributed random variable. By comparing it to the equation of the fitting line one gets the parameters of the original

continuous time SDE, namely

$$\theta = -\frac{\ln(a)}{\Delta T}, \quad (8.4)$$

$$\mu = \frac{c}{1-a}, \quad (8.5)$$

$$\alpha = \frac{\ln(a)}{\Delta T(a-1)}b \quad (8.6)$$

$$\sigma = \text{std}(\epsilon) \sqrt{\frac{-2\ln(a)}{(1-a^2)\Delta T}}. \quad (8.7)$$

Given the time series from an experiment we apply this procedure to calibrate both the equations (8.1). Specifically, for each agent i and independently for the recorded speed v_i and the absolute angular velocity $|\omega_i|$, we identified the parameters $\theta_{\cdot,i}$, $\mu_{\cdot,i}$, $\alpha_{\cdot,i}$, $\beta_{\cdot,i}$, and $\sigma_{\cdot,i}$. As inputs u_i and \dot{u}_i we used, for all the agents, the intensity of blue light given by the projector, divided by 255 to be in the range $[0; 1]$, and its approximate time derivative (numerical differentiation is described in (7.1)). We then discarded all the agents for which $\theta_{v,i}$ or $\theta_{\omega,i}$ was negative, because that would result in unstable dynamics. Moreover, we discarded all the agents for whom any of the identified parameters represents an outlier in the set of values of that parameter (according to Algorithm 7.1 and with a threshold $m = 5$). This ensures robustness of the identification procedure, which involves highly non-linear functions and would otherwise be susceptible to noise or uncertainties in the data. Notice that this procedure can also identify $\mu_{\omega,i}$, nevertheless we assumed the angular velocity to have zero mean, hence this parameter will not be used.

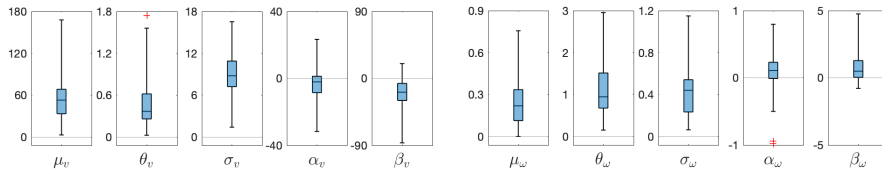
Figure 8.1a shows the resulting estimated values of the parameters for a representative experiment with light inputs switching every 10 s. We can observe a good qualitative agreement between these data and the results presented in Section 7.3, in particular α_v being mostly negative describes the observed negative photokinesis, similarly α_ω being mostly positive describes the observed positive photoklinokinesis, while β_v and β_ω being respectively negative and positive capture the photophobic response. Figure 8.1b shows the pairwise correlation between the estimated values of the parameters. Notice that, while most pairs of parameters show weak

correlation, the relations observed in the experimental data (see Section 7.3), such as negative correlation between speed and angular velocity (see (μ_v, μ_ω)), or positive correlation between the mean and the variability of the angular velocity (see $(\mu_\omega, \sigma_\omega)$), are here retrieved. Moreover we can observe new properties, for example faster agents are less responsive to light inputs (see the negative correlations between (μ_v, α_v) and (μ_v, β_v)).

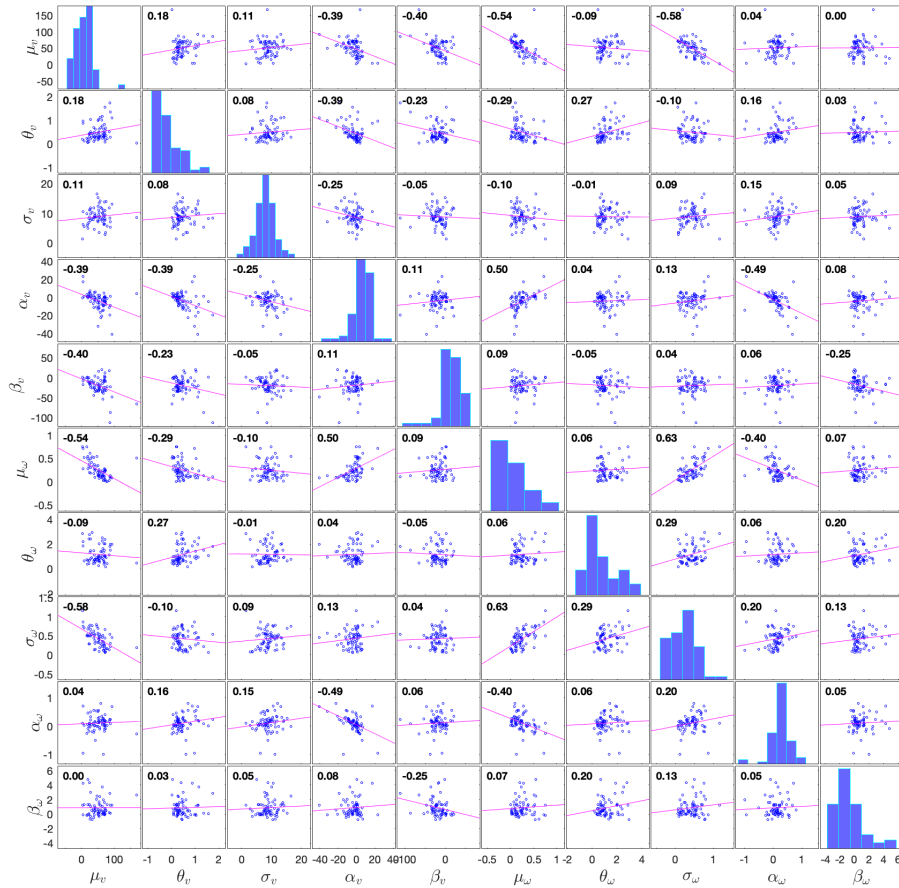
8.3 Numerical validation

To validate our model (8.1) and the parametrization procedure we implemented them in the agent based simulator we developed, called SwarmSim (for more information about SwarmSim see Appendix A or visit www.github.com/diBernardoGroup/SwarmSimPublic). Specifically, starting from the trajectories recorded during an experiment, we applied the identification procedure to all N' detected agents, obtaining a collection of $N'' \leq N'$ valid sets of parameters. By randomly sampling sets of parameters from this collection we can instantiate N virtual agents² in our simulator, and then compare the synthetic data to the experimental ones. Figure 8.2 shows real and simulated trajectories, which, indeed appear to be qualitatively similar. Moreover, Figure 8.3 shows the comparison between experimental and synthetic data, for two experiments, one with no inputs and the other in the presence of switching inputs. Both simulations are obtained by simulating the agents identified from the experimental data with switching inputs. The remarkable similarity between experimental and simulated data, also validated by Mann-Whitney U statistical test [83] (see Figures 8.3e and 8.3f), confirms the effectiveness of both the model (8.1) and the identification procedure described in the previous pages. In particular Figure 8.3b shows how the simulated model reproduces the complex light response observed in the real organisms, made of a combination of step-up photophobic, photokinetic and photoklinokinetic responses, followed by an adaptation in the next ~ 10 s.

²By allowing for repetitions, N can be larger than N'' .



(a) Distribution of the estimated parameters.



(b) Pairwise correlation between the estimated parameters.

Figure 8.1: Estimated values of the parameters (after the selection procedure) (a), and their distribution and pairwise Pearson correlation coefficient (b). The data comes from an experiment with light inputs switching every 10s. Input: 78 time series of speed and absolute angular velocity. Output: 71 valid sets of parameters.

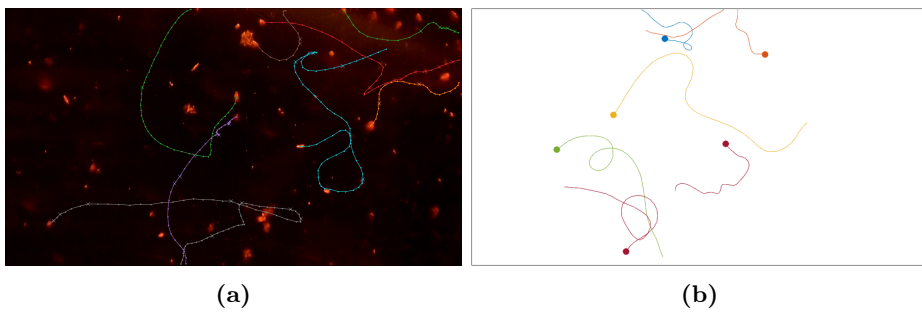


Figure 8.2: Comparison between real (a) and simulated (b) trajectories of *Euglena* in a dark environment.

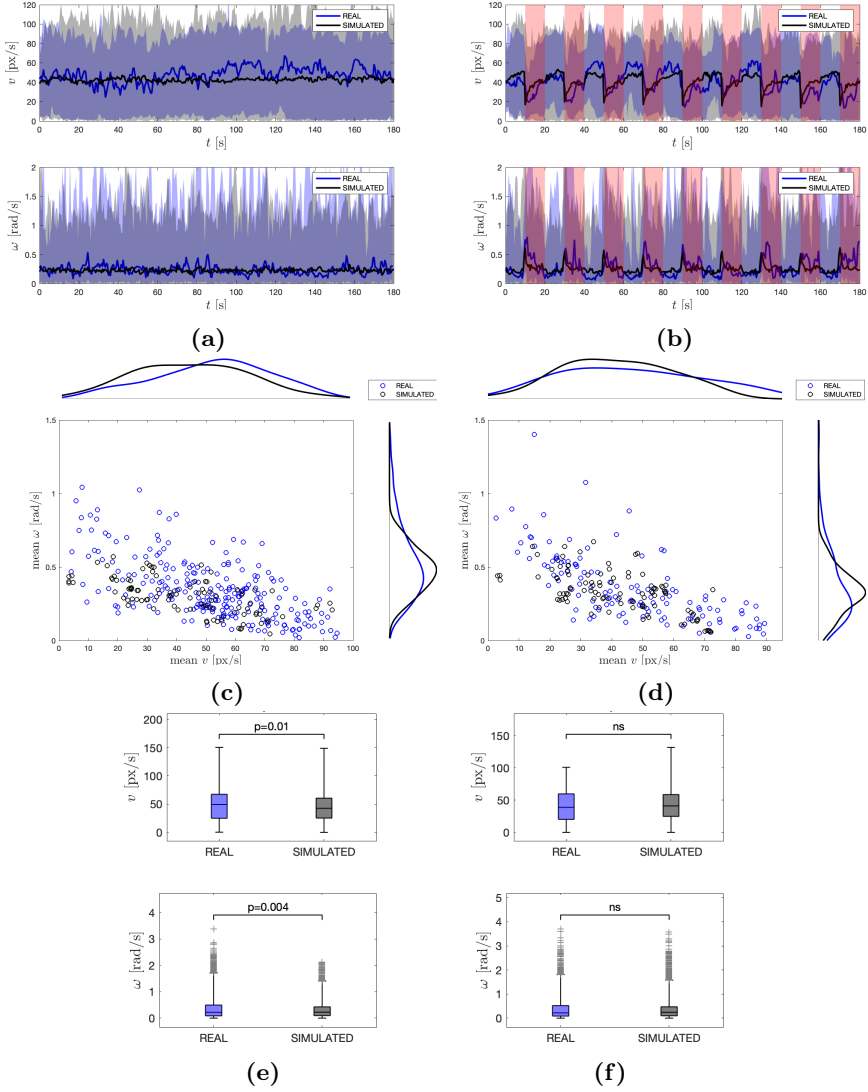


Figure 8.3: Comparison between experimental and synthetic data, for two experiments without inputs (a, c, e) and with switching input (b, d, f). (a, b) Time evolution of speed and absolute angular velocity. Red shaded area indicate light input is on. (c, d) Each dot represents an agent, either real or simulated, and the position is given by its mean speed and absolute angular velocity. Marginal distributions are obtained by Kernel Density Estimation. (e, f) Data points, for either real or simulated agents. Statistical significance is computed by Mann-Whitney U test. All real and simulated experiments lasted 180 s, with a sampling time of 0.5 s. Both simulations use the virtual agents identified from the experimental data with switching inputs.

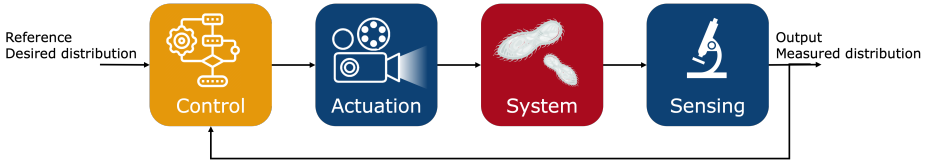


Figure 8.4: Control scheme to implement spatial control in the [DOME](#).

8.4 Towards spatial control

The final goal of our project is using the [DOME](#) to implement spatial control of the studied microorganisms, according to the scheme in [Figure 8.4](#). Specifically, we will use the mathematical model developed in this Chapter to design a feedback control strategy able to steer the movement of the microorganisms, so to obtain the desired distribution and achieving controlled density regulation. Such control strategy may leverage the flexibility offered by the [DLP](#) to combine spatial inputs with single-cell ones, and exploit the knowledge provided by the model to achieve unseen performance. Moreover, it will be easily applied to any species of microorganisms whose movement can be captured by our model.

To this aim we run preliminary experiments, complementary to those described in [Section 7.2](#), to study the response of the microorganisms to spatial inputs. These experiments are described in [Table 8.1](#) and depicted in [Figure 8.5](#), while some representative results are shown in [Figure 8.6](#). It can be easily seen that, as known, *Euglena* shows a strong photodispersion behaviour, and something similar is observed for *Volvox*. Instead, the response of *Paramecia* is less clear. These observations allowed to test on *Euglena* a simple open-loop approach, based on the static projection of a dark patten on a lit background. This naive approach, while occasionally providing satisfactory results (see [Figure 8.7](#)), is limited to a single specie and proved very sensible to the experimental conditions.

A thorough investigation of the control aspects will be carried out next, but is beyond the scope of this thesis and will be presented in future publications.

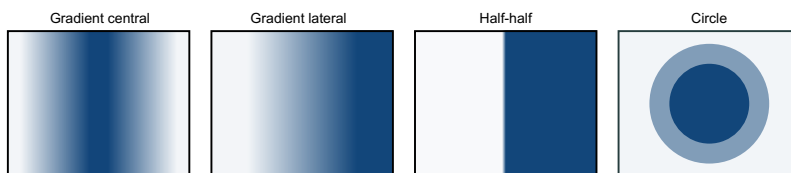


Figure 8.5: Graphical representation of the experiments executed to characterize the spatial response of the microorganisms.

	Experiment	Description
Spatial inputs	Half-Half	10s OFF, half sample OFF, half ON.
	Gradient lateral	10s OFF, linear light gradient from one side of the sample to the other.
	Gradient center light	10s OFF, linear light gradients from the sides of the sample to the center.
	Gradient center dark	10s OFF, linear light gradients from the center of the sample to the sides.
	Circle light	10s OFF, lit circle on dark background.
	Circle dark	10s OFF, dark circle on lit background.

Table 8.1: Experiments executed to characterize the spatial response of the microorganisms.

8.5 Discussion

In this Chapter we introduced a new model for stochastic mobile agents, extending the classic Persistent Turing Walker model with a second Stochastic Differential Equation modelling the longitudinal speed, and inputs to take into account the effect of light. This allowed to capture both the movement of microorganisms in a dark environment, and their complex response to light inputs, showing a combination of step-up photophobic, negative photokinetic and positive photoklinokinetic responses, further enriched by adaptation. The model has a generic structure and can be easily parameterized from the experimental data, meaning that it may suite a variety of mobile micro-agents and effectively describe their motion. Finally, we discussed how to proceed in the direction of developing new strategies to control the distribution of microorganisms in both space and time.

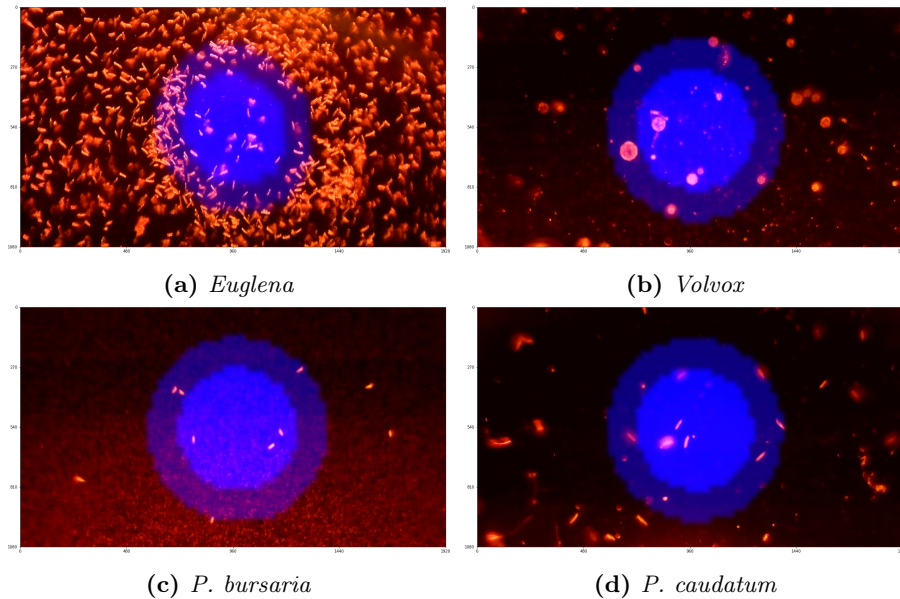


Figure 8.6: Response of microorganisms to a lit circle on dark background. The experimental conditions are those reported in Table 7.4. The projected blue light pattern is not present in the original images, thanks to the red light filter, and was added digitally.

Crucially, relatively simple but descriptive models, which allow both simulations and model based design, can be instrumental for the development of more effective feedback strategies.

To date, the model has been tested and validated only on *Euglena*, but we are currently working to extending the application of this methodology to obtain models also for the other species in our data-set, i.e. *Volvox*, *Paramecium bursaria* and *Paramecium caudatum*. Moreover, we aim to study how macroscopic density regulation (i.e. photoaccumulation or photodispersion) emerges from the individual responses of the organisms.

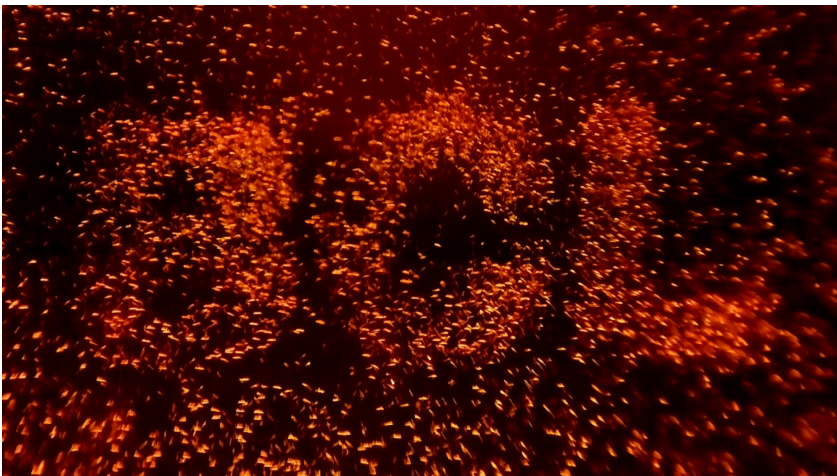


Figure 8.7: Spatial control with static input. A colony of *Euglena* forms the letters “BCL”, for BioCompute Lab.

Conclusions

This thesis explored the emerging behaviours displayed by Large-Scale Multi-Agent Systems (*LS-MAS*), particularly focusing on their spatial organization. The ubiquity of such systems, made of several interacting agents, spans from natural phenomena to artificial systems. These reflected into the two application domains discussed in this work, spatial control of biological populations and swarm robotics. The motivation stemmed from the many open challenges and the untapped possibilities, generated by the intricate dynamics of such systems.

Contributions. Our study began in Chapter 2 by surveying the existing Literature to establish a robust foundation and better understand the peculiarities of *LS-MAS*. This preliminary step helped identify critical gaps and challenges in modeling, analyzing, and controlling their emergent behaviors.

Then, in Part I, focusing on geometric pattern formation, we devised a novel distributed control algorithm to achieve self-organization of swarms into specific geometric patterns, like triangular and square lattices, representing a valuable advancement in the field of spatial organization of swarming robots. After introducing and validating the novel algorithm, we developed a formal proof of the convergence of swarming systems towards rigid lattices, when these are driven by suited virtual forces. This validates the stability of many existing solutions, thereby enhancing their applicability and reliability. In Part II, parting from geometric patterns,

we ventured into the study of spatial behaviours of biological systems, particularly the movement of microorganisms and their response to light stimuli. Starting from experimental data, this led to the quantitative characterization of the key features of the kinematics of the microscopic algae *Euglena gracilis*, and how its motion is influenced by light. The resulting insights represented the starting point for the development of a novel data-driven mathematical model that clearly captures the behavior of the studied microorganisms, both in terms of motion and light response. This model lays the foundation for future research on the spatial control of light sensitive cellular populations.

Methodological Insights. Our approach merged formal and numerical tools, together with experiments. A formal approach, based on graph and Lyapunov theories, allowed the development of formal guarantees for the stability of rigid lattices. The experiments were carried out leveraging innovative platforms, particularly the Robotarium was used to validate our distributed control strategy on real swarming robots, while the [DOME](#) allowed the study and the collection of a rich data-set on a range of light sensitive microorganisms. Lastly, numerical tools were used throughout the thesis to analyze data and carry out simulations. This resulted in the development of a new software platform, SwarmSim, to simulate mobile multi-agent systems, and the software to use the [DOME](#), which also includes a computer vision package to automatically detect and track moving objects. These tools were used to validate our results, while also paving the way for future advancements in the study of [LS-MAS](#) and their spatial behaviours.

Implications and Future Directions. The advancements made in this thesis have significantly contributed to understanding and addressing the challenges posed by [LS-MAS](#). However, certain critical aspects remain as open questions, presenting avenues for future research and exploration. As regards geometric pattern formation, we contributed to the development of both novel control algorithms and formal proofs; nevertheless much remains to be addressed, aiming for a general control approach and the corresponding proof of convergence, so as to guarantee the formation of any required lattices. Looking at the behaviour of biological

agents, we presented our first results, that point in the right direction, towards the development of integrated methodologies to model and control the movement and the emerging spatial behaviours of light sensitive microorganisms. To this aim, it is needed to extend and improve our model, and the connected parametrization procedure, so as to be able to describe a wider range of micro-agents, and finally to control their distribution. In a more general prospective, despite the numerous advancements achieved by researchers, obtaining a full understanding of the multiple scales in which [LS-MAS](#) operate, and their interplay, remains a crucial goal.

Closing Remarks. In conclusion, this thesis, by addressing critical challenges in modeling, analyzing, and controlling emergent behaviors of Large-Scale Multi-Agent Systems, represents a step forward in unraveling their inherent complexity. The work presented therein stemmed from a three years long journey that took place between the Sincro research group, at the University of Naples Federico II, and the BioCompute Lab at the University of Bristol. Specifically, the work reported in [Part I](#) was developed in collaboration with Giancarlo Maffettone, Marco Coraggio and Davide Fiore, while the experimental work presented in [Part II](#) was carried out during a nine months period at the BioCompute Lab (University of Bristol), under the supervision of Dr. Thomas Gorochowski.

Now, as we look towards the horizon, the journey continues, brimming with opportunities to explore, innovate, and refine our understanding of Large-Scale Multi-Agent Systems, propelling us towards a future where their potential is fully realized across diverse domains.

SwarmSim: a framework for Multi-Agent Systems simulation

SwarmSim is a MATLAB software package crafted for the simulation of swarms of mobile agents, in both 2D and 3D environments. We developed it to provide researchers with an intuitive and flexible platform to analyze the intricate dynamics of multi-agent systems and test new control algorithms. This chapter summarizes the utility, functionalities, and significance of SwarmSim. The up to date software and the complete documentation are available online at www.github.com/diBernardoGroup/SwarmSimPublic. Figure A.1 depicts the logo of SwarmSim.

A.1 Core functionalities and practical utility

The software operates on a modular architecture that provides flexibility and adaptability. Central to its functionality is the ability to manipulate both the dynamics and the interactions governing agents' behavior. Users can either implement new dynamical models, tailored to their specific needs, or use pre-existing models embedded within the system, such as first and second order integrators, Lévy walkers [145], Persistent Turing Walker agents [41] (see Figure A.2a), etc. A central aspect of SwarmSim is its support for distributed control laws. Also in this case, users can ei-



Figure A.1: Logo of SwarmSim

ther leverage embedded control algorithms based on virtual forces, such as those for geometric pattern formation discussed in Chapters 4 and 5 (see Figures A.2b and A.2c) or in [120, 125], or implement new ones. This enables the observation and analysis of emergent phenomena stemming from the interactions among agents.

SwarmSim also facilitates the evaluation of performance metrics, providing quantitative measures to assess and compare different scenarios (see Figure A.2d). This capability is instrumental in gauging the efficacy of control strategies, evaluating their robustness, scalability and flexibility. Moreover, the software supports extensive parallel simulation, allowing to easily scrutinize stochastic effects, explore diverse initial conditions, and investigate the influence of varying parameters on the resulting system's behavior (see Figure A.2e). The integration of local stability analysis through linearization further enriches the software, by enabling the exploration the system's stability around specific configurations. Finally, SwarmSim includes functions to visualize all relevant information, such as the position of the agents, their trajectories and interactions, the time evolution of relevant quantities or the relation between selected parameters and output metrics. This software was used for all the simulations showed in this thesis, to validate control algorithms for geometric pattern formation in Chapters 4 and 5, and to simulate the movement of microorganisms in Chapter 8.

Compared to other simulators, such as SwarmLab [117], tailored on specific applications (i.e. flocking) and pre-defined dynamics, SwarmSim offers improved flexibility, in terms of both agents dynamics and emerging

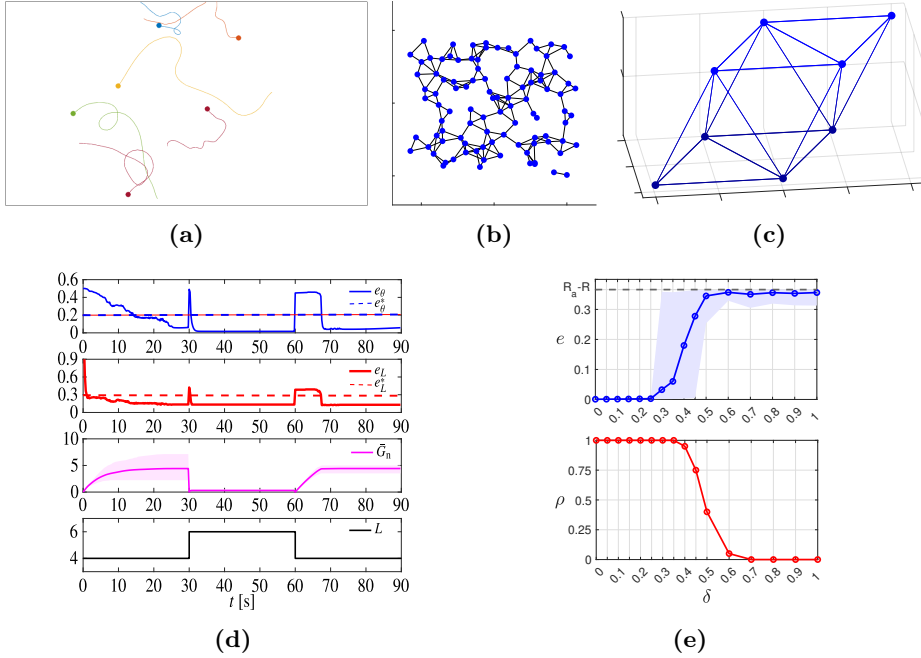


Figure A.2: Example of figures generated by SwarmSim. (a) Trajectories of Persistent Turing Walkers. (b) Agents in 2D space with interaction network. (c) Agents in 3D space with interaction network. (d) Time evolution of selected quantities of interest. (e) Dependency of two selected quantities of interest with respect to a parameter.

behaviour.

A.2 Discussion

In conclusion, SwarmSim supports researchers by enabling exploration and analysis of emergent behaviors in multi-agent systems. Furthermore, the software serves as a valuable tool to test and validate novel distributed control algorithms. SwarmSim, while being developed for swarm robotics, can result useful for different disciplines, such as biology and complex systems analysis. Additionally, its intuitive interface makes it a suitable tool for introducing students to the simulation of multi-agent systems.

Appendix B

Tracking software

Being able to automatically extract relevant information from images and video is a critical feature in many recent applications, including our research. Therefore, together with the [DOME](#) software, presented in Section 7.1.2, we developed an automatic tracking software. It allows to analyze the images acquired from a static camera to automatically detect the moving objects and track their movement over time. It combines computer vision techniques to detect the moving objects, with estimation and optimization algorithms, to assign them consistent ids over time. This software is written in Python using the OpenCV library.

The algorithm is made of three main steps, *background modelling*, *objects detection* and *objects tracking*, as depicted in Figure B.1. In the following we discuss the function and the structure of each of these steps.

B.1 Background modelling

Given a collection of images acquired during an experiment the software first builds a model of the background, that is an image representing all and only the elements that are static.

To do this a subset of images is selected, specifically we used N_{BG} images uniformly distributed during the experiment. Each of these images is converted to grey scale, according to the `channel` parameter, which allows to select either one of the three color channels (i.e. red, green and blue) or the combined brightness. Then the brightness is adjusted so that

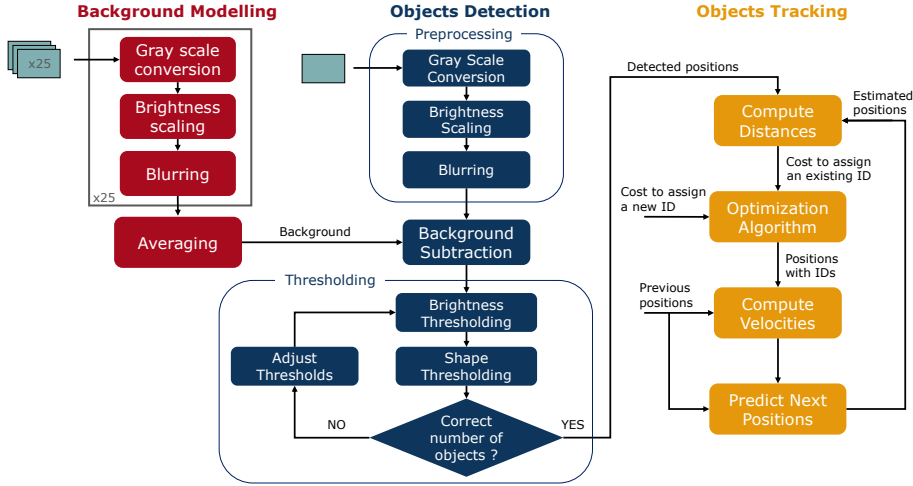


Figure B.1: Schematic of the tracking algorithm.

the darkest pixel becomes the new zero, while the brightest one becomes the new one. Finally, a median blurring is applied to remove noise and smooth the contours of the objects.

These image are then collapsed into a single one, by a pixel-wise median over the time dimension. The resulting grey scale image represents the model of the background, that will be used for the background subtraction. Figure B.2 shows an example of this process.

B.2 Objects detection

After the background modelling, the software analyses the images one by one, in temporal order, to get the positions of the moving objects. Each frame undergoes the same pre-processing steps applied during the background modelling, namely grey scale conversion, brightness adjustment and blurring. This ensures the images stay consistent with the background. Then, the foreground, that represents only the moving objects, is computed by pixel-wise *background subtraction* as $foreground = \max(0, frame - background)$.

The actual image segmentation is then performed by pixel-wise bright-

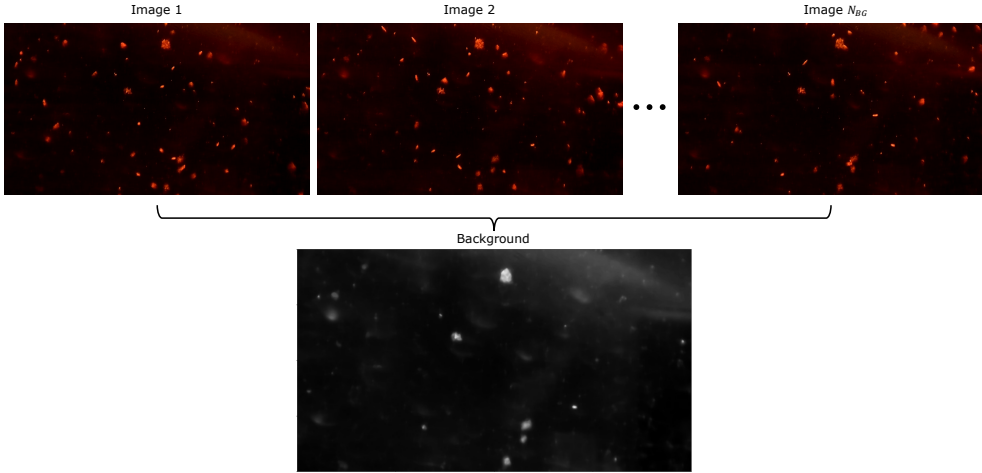


Figure B.2: Example of background modelling.

ness thresholding, that is $mask = foreground \geq b_{min}$. The `findContours` function is then applied to the resulting black and white mask, to detect the contours of the objects. For each of the detected objects we compute the area a , the perimeter p and the Polsby–Popper compactness measure [101], as $c = (4\pi a)/(p^2) \in [0; 1]$. A second selection happens by retaining only the objects whose area and compactness fall in the given ranges, $[a_{min}; a_{max}]$ and $[c_{min}; c_{max}]$ respectively. This allows to discern the objects we are looking for from debris or elements of the background not completely removed. Figure B.3 shows the objects detection process step by step.

The number of detected objects is then compared with the expected one, if it is not farther than δ from the expected one the detection is considered successful, and the positions of the centers of the detected objects are returned. Otherwise, all the thresholds a_{min} , a_{max} , c_{min} , c_{max} and b_{min} are relaxed or tightened by a factor λ , depending on whether the number of detected objects is smaller or larger than the expected one. The detection is then repeated with the new thresholds. This iterative procedure, should the condition on the number of detected objects not be satisfied, is forcibly stopped after a maximum number of iterations. For the first frame the expected number of objects is set equal to the number of detected ones, so

that the condition is automatically satisfied, while for subsequent frames the expected number of objects is set to the number of objects detected in the previous frame. This automatic adaptation of the thresholds helps in keeping a more consistent detection, also in the presence of changes in the illumination.

B.3 Objects tracking

The last step consists in assigning ids to the detected objects and keeping these consistent over time. Given the current detected positions and the estimated ones the distance between the two, for all the possible couples, is computed. The matching cost is then defined as the square of the distance. Moreover, a cost for the allocation of a new id is computed for each of the detected objects, depending on the distance from the edges of the camera, so that objects close to an edge have a lower cost for the allocation of a new id.

The Jonker-Volgenant optimization algorithm [26] is then used to solve the assignment problem, while minimizing the total cost. Specifically, each detected object will be given an id, either an existing or a new one.

Finally, for each object the position at the next time step is estimated as $\mathbf{p}_{k+1} = \mathbf{p}_k + \gamma(\mathbf{p}_k - \mathbf{p}_{k-1})$, where $\gamma \in \mathbb{R}_{\geq 0}$ is the inertia parameter. The estimated position will be used at the next time step to perform the id matching.

When an existing id is not assigned to any of the detected objects it is not discarded and its estimated position is updated as usual, but its *inactivity counter* is incremented by one. The inactivity counter is then used when computing the matching costs, so that longly inactive objects have a larger matching cost. Specifically this additional cost increases with the square of the inactivity counter. This allows to keep a consistent tracking of the object even if it “disappears” for one or more consecutive frames, that might happen when it overlaps with another object or something in the background. Figure B.4 shows an example application of this algorithm.

Parameter	Description	Value
<code>channel</code>	Color channel used for gray-scale conversion	Red
N_{BG}	Number of images used for background modelling	25
b_{min}	Brightness threshold	85/255
a_{min}	Minimum area	175 px
a_{max}	Maximum area	1500 px
c_{min}	Minimum compactness	0.55
c_{max}	Maximum compactness	0.90
δ	Tolerance on the number of detected objects	25%
λ	Thresholds adjustment gain	2%
γ	Inertia parameter for position estimation	0.9

Table B.1: Parameters used to track *Euglena*.

B.4 Discussion

This software allows to automatically detect and track moving objects, using images acquired from a static camera. Despite not leveraging the most modern techniques, based on machine learning, it provides satisfactory results, with an open door for further improvements. It was originally developed to analyze data from the [DOME](#) experiments, but the generality of the algorithm makes it suitable for different applications. In particular it was designed to detect bright objects moving on a dark background, but could be easily adapted to work in the opposite situation. [Table B.1](#) shows the values of the parameters used to track the *Euglena*, using the images from the [DOME](#). Together with the rest of the software developed to use the [DOME](#), this is currently being refined for future publication.

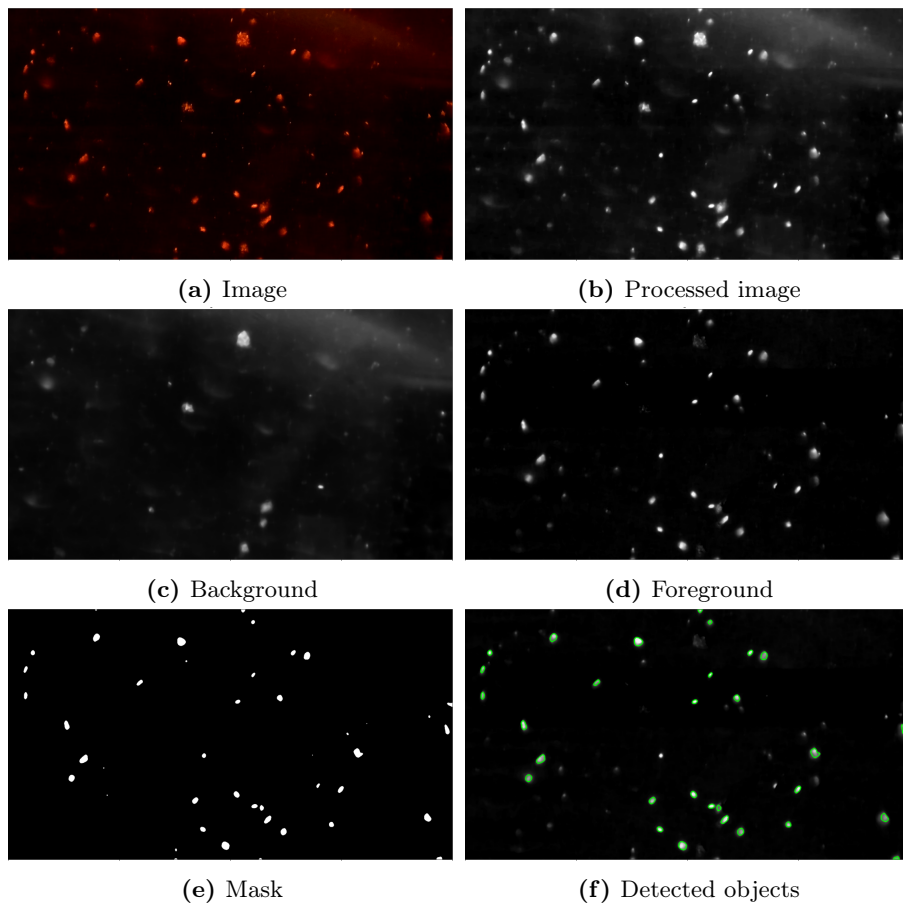


Figure B.3: Step by step image elaboration for object detection. (a) Original image. (b) Processed image by selecting the red channel, adjusting the brightness and applying the blur. (c) Model of the background. (d) Foreground obtained by subtracting the background from the processed image. (e) Black and white mask obtained by thresholding the brightness of the pixels. (f) Contours of the detected objects, after the selection based on area and compactness. These images were acquired by the [DOME](#), during an experiment with *Euglena*.

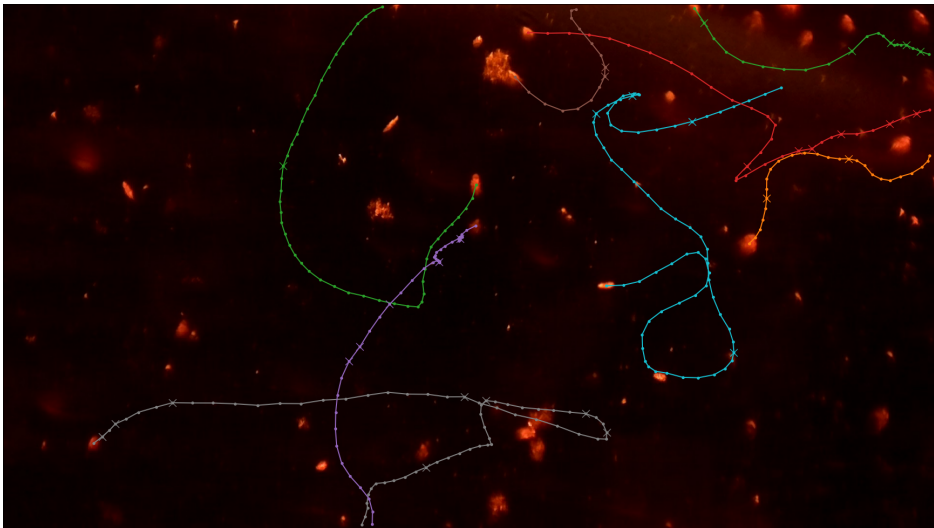


Figure B.4: Trajectories generated by the tracking software during an experiment with *Euglena*.

The End

Bibliography

- [1] Fabrizia Auletta, Davide Fiore, Michael J. Richardson, and Mario di Bernardo. Herding stochastic autonomous agents via local control rules and online target selection strategies. *Autonomous Robots*, 46(3):469–481, 2022. Cited on page 10.
- [2] He Bai, Murat Arcak, and John Wen. *Cooperative Control Design A Systematic, Passivity-Based Approach*. 2011. Cited on page 10.
- [3] T. Balch and M. Hybinette. Behavior-based coordination of large-scale robot formations. *Proceedings - 4th International Conference on MultiAgent Systems, ICMAS 2000*, pages 363–364, 2000. Cited on pages 19 and 21.
- [4] Tucker Balch and Maria Hybinette. Social potentials for scalable multi-robot formations. *Proceedings-IEEE International Conference on Robotics and Automation*, 1(April):73–80, 2000. Cited on pages 19 and 21.
- [5] M. Ballerini, N. Cabibbo, R. Candelier, A. Cavagna, E. Cisbani, I. Giardina, V. Lecomte, A. Orlandi, G. Parisi, A. Procaccini, M. Viale, and V. Zdravkovic. Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study. *Proceedings of the National Academy of Sciences of the United States of America*, 105(4):1232–1237, 2008. Cited on page 1.
- [6] Levent Bayindir. A review of swarm robotics tasks. *Neurocomputing*, 172:292–321, 2016. Cited on page 10.
- [7] Avraham Be’er and Gil Ariel. A statistical physics view of swarming bacteria. *Movement Ecology*, 7(1):1–17, 2019. Cited on page 71.
- [8] Howard C. Berg. *Random Walks in Biology*. 1993. Cited on pages 68, 69, and 96.

-
- [9] Howard C. Berg. *E.Coli in motion*, volume 28. Springer, 2004. Cited on page 69.
- [10] Jaya P.N. Bishwal. *Parameter Estimation in Stochastic Differential Equations*, volume 2084. Springer, 1923. Cited on page 70.
- [11] M. Bodnar and J. J.L. Velasquez. Derivation of macroscopic equations for individual cell-based models: A formal approach. *Mathematical Methods in the Applied Sciences*, 28(15):1757–1779, 2005. Cited on page 7.
- [12] Karen Grace Bondoc-Naumovitz, Hannah Laeverenz-Schlogelhofer, Rebecca N Poon, Alexander K Boggon, Samuel A Bentley, Dario Cortese, and Kirsty Y Wan. Methods and Measures for Investigating Microscale Motility. *Integrative And Comparative Biology*, 63(6):1485–1508, 2023. Cited on page 67.
- [13] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41, mar 2013. Cited on pages 1, 2, 5, 6, 13, 24, 31, and 60.
- [14] Steven L. Brunton, Joshua L. Proctor, J. Nathan Kutz, and William Bialek. Discovering governing equations from data by sparse identification of non-linear dynamical systems. *Proceedings of the National Academy of Sciences of the United States of America*, 113(15):3932–3937, 2016. Cited on pages 2 and 9.
- [15] Francesco Bullo. *Contraction theory for dynamical systems*. 2022. Cited on page 10.
- [16] Francesco Bullo, Jorge Cortés, and Sonia Martínez. *Distributed Control of Robotic Networks*. 2008. Cited on page 8.
- [17] A. H. Burr. Photomovement Behavior in Simple Invertebrates. *Photoreception and Vision in Invertebrates*, pages 179–215, 1984. Cited on page 71.
- [18] Daniel Carrillo-Zapata, James Sharpe, Alan Frank T. Winfield, Luca Giuglioli, and Sabine Hauert. Toward Controllable Morphogenesis in Large Robot Swarms. *IEEE Robotics and Automation Letters*, 4(4):3386–3393, 2019. Cited on page 7.
- [19] Arnaud Casteigts, Jérémie Albert, Serge Chaumette, Amiya Nayak, and Ivan Stojmenovic. Biconnecting a network of mobile robots using virtual angular forces. *Computer Communications*, 35(9):1038–1046, 2012. Cited on pages 17, 18, 21, and 29.
-

-
- [20] Edward A. Codling, Michael J. Plank, and Simon Benhamou. Random walk models in biology. *Journal of the Royal Society Interface*, 5(25):813–834, 2008. Cited on pages [68](#) and [69](#).
- [21] Martina Conte and Nadia Loy. A Non-Local Kinetic Model for Cell Migration: A Study of the Interplay Between Contact Guidance and Steric Hindrance. *SIAM Journal on Applied Mathematics*, 0(0):S429–S451, 2023. Cited on page [69](#).
- [22] Mario Coppola, Jian Guo, Eberhard Gill, and Guido C.H.E. de Croon. Provable self-organizing pattern formation by a swarm of robots with limited knowledge. *Swarm Intelligence*, 13(1):59–94, 2019. Cited on pages [19](#) and [21](#).
- [23] Jorge Cortés. Global and robust formation-shape stabilization of relative sensing networks. *Automatica*, 45(12):2754–2762, 2009. Cited on page [18](#).
- [24] Iain D. Couzin, Jens Krause, Nigel R. Franks, and Simon A. Levin. Effective leadership and decision-making in animal groups on the move. *Nature*, 433(7025):513–516, 2005. Cited on page [7](#).
- [25] Donald Cronkite and Sue Van Den Brink. The Role of Oxygen and Light in Guiding Photoaccumulation in the Paramecium bursaria-Chlorella Symbiosis. *The journal of experimental zoology*, 217:171–177, 1981. Cited on page [82](#).
- [26] David F. Crouse. On implementing 2D rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696, 2016. Cited on page [117](#).
- [27] Ana Rubio Denniss, Thomas E. Goroehowski, and Sabine Hauert. An Open Platform for High-Resolution Light-Based Control of Microscopic Collectives. *Advanced Intelligent Systems*, 2200009:2200009, 2022. Cited on pages [67](#), [73](#), [74](#), [76](#), [77](#), [78](#), and [82](#).
- [28] Ana Rubio Denniss, Laia Freixas Mateu, Thomas Goroehowski, and Sabine Hauert. Q-learning for real time control of heterogeneous microagent collectives, 2022. Cited on page [74](#).
- [29] Anjali Doshi, Marian Shaw, Ruxandra Tonea, Soonhee Moon, Anish Doshi, Andrew Laine, Jia Guo, and Tal Danino. Engineered bacterial swarm patterns as spatial records of environmental inputs. *Nature Chemical Biology*, 2023. Cited on pages [1](#), [7](#), and [67](#).
-

-
- [30] Knut Drescher, Raymond E. Goldstein, and Idan Tuval. Fidelity of adaptive phototaxis. *Proceedings of the National Academy of Sciences of the United States of America*, 107(25):11171–11176, 2010. Cited on pages 72 and 82.
- [31] S. Duncan, G. Estrada-Rodriguez, J. Stoczek, M. Dragone, P. A. Vargas, and H. Gimplerlein. Efficient quantitative assessment of robot swarms: Coverage and targeting Lévy strategies. *Bioinspiration and Biomimetics*, 17(3), 2022. Cited on pages 7 and 69.
- [32] Karthik Elamvazhuthi, Chase Adams, and Spring Berman. Coverage and field estimation on bounded domains by diffusive swarms. *2016 IEEE 55th Conference on Decision and Control, CDC 2016*, (Cdc):2867–2874, 2016. Cited on page 7.
- [33] Karthik Elamvazhuthi and Spring Berman. Mean-field models in swarm robotics: A survey. *Bioinspiration and Biomimetics*, 15(1):ab49a4, 2020. Cited on pages 8 and 9.
- [34] Peter Engel, Louis Michel, and Marjorie Senechal. Lattice Geometry. Technical report, Institut des Hautes Etudes Scientifique, 2004. Cited on page 26.
- [35] Utku Eren and Behçet Açikmeşe. Velocity field generation for density control of swarms using heat equation and smoothing kernels. In *IFAC-PapersOnLine*, 2017. Cited on page 7.
- [36] Ernesto Estrada. What is a Complex System, After All? *Foundations of Science*, (April), 2023. Cited on page 5.
- [37] Don Fawcett. Chapter 4 - Cilia and Flagella. In Jean Brachet and Alfred E Mirsky, editors, *The Cell*, pages 217–297. Academic Press, 1961. Cited on page 67.
- [38] David Fielding, Imogen Taylor, Simon Jones, Sabine Hauert, and Edmund R Hunt. Optical Herding of Swarms: Toward Universal Control Algorithms for Microscopic Collectives. In *Proceedings of the 2023 Artificial Life Conference*, 2023. Cited on pages 67 and 71.
- [39] D Freedman. *Statistical Models: Theory and Practice*. Cambridge University Press, 2009. Cited on page 98.
- [40] Gaurav Gardi, Steven Ceron, Wendong Wang, Kirstin Petersen, and Metin Sitti. Microrobot collectives with reconfigurable morphologies, behaviors, and functions. *Nature Communications*, 13(1):1–14, 2022. Cited on pages 10, 16, and 67.
-

-
- [41] Jacques Gautrais, Christian Jost, Marc Soria, Alexandre Campo, Sébastien Motsch, Richard Fournier, Stéphane Blanco, and Guy Theraulaz. Analyzing fish movement as a persistent turning walker. *Journal of Mathematical Biology*, 58(3):429–445, 2009. Cited on pages 70, 96, 97, and 111.
- [42] Veysel Gazi and Kevin M. Passino. A class of attraction/repulsion functions for stable swarm aggregations. *Proceedings of the IEEE Conference on Decision and Control*, 77(18):2842–2847, 2002. Cited on pages 7, 51, 54, and 60.
- [43] Piergiorgio Gentile, Valeria Chiono, Irene Carmagnola, and Paul V. Hatton. An overview of poly(lactic-co-glycolic) Acid (PLGA)-based biomaterials for bone tissue engineering. *International Journal of Molecular Sciences*, 15(3):3640–3659, 2014. Cited on page 7.
- [44] Andrea Giusti, Marco Coraggio, and Mario di Bernardo. Local Convergence of Multi-Agent Systems Toward Rigid Lattices. *IEEE Control Systems Letters*, 7:2869–2874, 2023. Cited on page 51.
- [45] Andrea Giusti, Gian Carlo Maffettone, Davide Fiore, Marco Coraggio, and Mario di Bernardo. Distributed control for geometric pattern formation of large-scale multirobot systems. *Frontiers in Robotics and AI*, (September):1–16, 2023. Cited on pages 7 and 26.
- [46] Thomas E. Gorochowski, Sabine Hauert, Jan Ulrich Kreft, Lucia Marucci, Namid R. Stillman, T. Y.Dora Tang, Lucia Bandiera, Vittorio Bartoli, Daniel O.R. Dixon, Alex J.H. Fedorec, Harold Fellermann, Alexander G. Fletcher, Tim Foster, Luca Giuggioli, Antoni Matyjaszkiwicz, Scott McCormick, Sandra Montes Olivas, Jonathan Naylor, Ana Rubio Denniss, and Daniel Ward. Toward Engineering Biosystems With Emergent Collective Functions. *Frontiers in Bioengineering and Biotechnology*, 8(June):1–9, 2020. Cited on pages 1 and 2.
- [47] Nicolas E. Grandel, Kiara Reyes Gamas, and Matthew R. Bennett. Control of synthetic microbial consortia in time, space, and composition. *Trends in Microbiology*, 29(12):1095–1105, 2021. Cited on page 66.
- [48] Luis Guerrero-Bonilla, David Saldana, and Vijay Kumar. Design Guarantees for Resilient Robot Formations on Lattices. *IEEE Robotics and Automation Letters*, 4(1):89–96, 2019. Cited on page 2.
- [49] D. P. Häder. Polarotaxis, gravitaxis and vertical phototaxis in the green flagellate, *Euglena gracilis*. *Archives of Microbiology*, 147(2):179–183, 1987. Cited on page 82.
-

-
- [50] Donat P. Häder and Mineo Iseki. Photomovement in *Euglena*. In *Euglena: Biochemistry, Cell and Molecular Biology*, number 11, pages 207–235. 2017. Cited on page 72.
- [51] Sabine Hauert and Sangeeta N. Bhatia. Mechanisms of cooperation in cancer nanomedicine: Towards systems nanotechnology. *Trends in Biotechnology*, 32(9):448–455, 2014. Cited on page 66.
- [52] Mary Katherine Heinrich, Mostafa Wahby, Marco Dorigo, and Heiko Hamann. Swarm Robotics. In *Cognitive Robotics*, chapter 5, pages 77–98. The MIT Press, 2022. Cited on page 10.
- [53] Bruce Hendrickson. Conditions for unique graph realizations. *SIAM Journal on Computing*, 21(1):65–84, 1992. Cited on page 23.
- [54] Thomas A. Henzinger. Theory of hybrid automata. *Proceedings - Symposium on Logic in Computer Science*, pages 278–292, 1996. Cited on page 69.
- [55] Suranga Hettiarachchi and William M. Spears. Moving swarm formations through obstacle fields. *Proceedings of the 2005 International Conference on Artificial Intelligence, ICAI'05*, 1:97–103, 2005. Cited on pages 18 and 20.
- [56] N A Hill. Trajectories of Swimming Micro-Organisms and Continuum Models of Bioconvection. In Hans G Othmer, Philip K Maini, and James D Murray, editors, *Experimental and Theoretical Advances in Biological Pattern Formation*, chapter 12, pages 101–103. Springer US, Boston, MA, 1993. Cited on page 70.
- [57] Konrad Hinsen. Normal mode theory and harmonic potential approximations. In *Normal Mode Analysis : Theory and Applications to Biological and Chemical Systems*, chapter 1, pages 1–18. 2005. Cited on pages 20 and 51.
- [58] Zahid Hossain, Engin W. Bumbacher, Alice M. Chung, Honesty Kim, Casey Litton, Ashley D. Walter, Sachin N. Pradhan, Kemi Jona, Paulo Blikstein, and Ingmar H. Riedel-Kruse. Interactive and scalable biology cloud experimentation for scientific inquiry and education. *Nature Biotechnology*, 34(12):1293–1298, 2016. Cited on pages 73 and 82.
- [59] M. Ani Hsieh, Vijay Kumar, and Luiz Chaimowicz. Decentralized controllers for shape generation with robotic swarms. *Robotica*, 26(5):691–701, 2008. Cited on page 7.
-

-
- [60] Kenji Iwatsuki and Yutaka Naitoh. Behavioural Responses to Light in Paramecium Bursaria in Relation to its Symbiotic Green Alga Chlorella. *Journal of Experimental Biology*, 134(1):43–60, 1988. Cited on pages 71, 72, and 82.
- [61] Bill Jackson. Notes on the Rigidity of Graphs, 2007. Cited on pages 22, 23, and 63.
- [62] Meng Ji and Magnus Egerstedt. Distributed coordination control of multi-agent systems while preserving connectedness. *IEEE Transactions on Robotics*, 23(4):693–703, 2007. Cited on page 7.
- [63] Marko Jusup, Petter Holme, Kiyoshi Kanazawa, Misako Takayasu, Ivan Romić, Zhen Wang, Sunčana Geček, Tomislav Lipić, Boris Podobnik, Lin Wang, Wei Luo, Tin Klanjšček, Jingfang Fan, Stefano Boccaletti, and Matjaž Perc. Social physics. *Physics Reports*, 948:1–148, 2022. Cited on page 2.
- [64] Miquel Kegeleirs, Giorgio Grisetti, and Mauro Birattari. Swarm SLAM: Challenges and Perspectives. *Frontiers in Robotics and AI*, 8(March):1–6, 2021. Cited on pages 10 and 16.
- [65] Hassan K. Khalil. *Nonlinear Systems*, volume 122. Prentice Hall, third edition, 2002. Cited on pages 8, 10, and 58.
- [66] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 500–505, 1985. Cited on page 17.
- [67] Seungkeun Kim, Hyondong Oh, Jinyoung Suk, and Antonios Tsourdos. Coordinated trajectory planning for efficient communication relay using multiple UAVs. *Control Engineering Practice*, 29:42–49, 2014. Cited on page 16.
- [68] Yuri A Kuznetsov. *Elements of Applied Bifurcation Theory*, volume 112. Springer, second edition, 2004. Cited on pages 22 and 64.
- [69] Amy T. Lam, Karina G. Samuel-Gama, Jonathan Griffin, Matthew Loewn, Lukas C. Gerber, Zahid Hossain, Nate J. Cira, Seung Ah Lee, and Ingmar H. Riedel-Kruse. Device and programming abstractions for spatiotemporal control of active micro-particle swarms. *Lab on a Chip*, 17(8):1442–1451, 2017. Cited on pages 7, 67, 73, and 82.
- [70] Robin Lamarche-Perrin, Yves Demazeau, and Jean Marc Vincent. How to build the best macroscopic description of your multi-agent system? *Lecture*
-

- Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7879 LNAI(May):157–169, 2013. Cited on page 9.
- [71] Geunho Lee and Nak Young Chong. A geometric approach to deploying robot swarms. *Annals of Mathematics and Artificial Intelligence*, 52(2-4):257–280, 2008. Cited on pages 19, 20, 29, and 51.
- [72] Geunho Lee, Yasuhiro Nishimura, Kazutaka Tatara, and Nak Young Chong. Three dimensional deployment of robot swarms. *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, pages 5073–5078, 2010. Cited on pages 20 and 29.
- [73] Andrew J. Leverentz, Chad M. Topaz, and Andrew J. Bernoff. Asymptotic dynamics of attractive-repulsive swarms. *SIAM Journal on Applied Dynamical Systems*, 8(3):890–908, 2009. Cited on page 7.
- [74] Guannan Li, David St-Onge, Carlo Pinciroli, Andrea Gasparri, Emanuele Garone, and Giovanni Beltrame. Decentralized progressive shape formation with robot swarms. *Autonomous Robots*, 43(6):1505–1521, 2019. Cited on pages 19 and 21.
- [75] Xiang Li, M. Fikret Ercan, and Yu Fai Fung. A triangular formation strategy for collective behaviors of robot swarm. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5592 LNCS(PART 1):897–911, 2009. Cited on pages 19 and 21.
- [76] Francesco Lo Iudice, Franco Garofalo, and Pietro De Lellis. Bounded Partial Pinning Control of Network Dynamical Systems. *IEEE Transactions on Control of Network Systems*, 10(1):238–248, 2023. Cited on page 7.
- [77] Hamilton J.M. Lopes and Danielli A. Lima. Evolutionary Tabu Inverted Ant Cellular Automata with Elitist Inertia for swarm robotics as surrogate method in surveillance task using e-Puck architecture. *Robotics and Autonomous Systems*, 144:103840, 2021. Cited on page 10.
- [78] László Miklós Lovász. Large Networks and Graph Limits. In *Colloquium Publications*, 2012. Cited on page 9.
- [79] Jean-Baptiste Lugagne, Caroline M. Blassick, and Mary J. Dunlop. Deep model predictive control of gene expression in thousands of single cells, 2022. Cited on pages 67 and 74.
-

-
- [80] Denis H. Lynn. *The Ciliated Protozoa Characterization, Classification, and Guide to the Literature*. Number July. Springer, third edition, 2008. Cited on page 67.
- [81] Gian Carlo Maffettone, Alain Boldini, Mario Di Bernardo, and Maurizio Porfiri. Continuuification Control of Large-Scale Multiagent Systems in a Ring. *IEEE Control Systems Letters*, 7:841–846, 2023. Cited on page 9.
- [82] M. H.A. Majid, M. R. Arshad, and R. M. Mokhtar. *Swarm Robotics Behaviors and Tasks: A Technical Review*, volume 371. Springer International Publishing, 2022. Cited on pages 6 and 11.
- [83] H. B. Mann and D. R. Whitney. On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *The Annals of Mathematical Statistics*, 18(1):50–60, 1947. Cited on pages 92, 93, and 100.
- [84] Helena Massana-Cid, Claudio Maggi, Giacomo Frangipane, and Roberto Di Leonardo. Rectification and confinement of photokinetic bacteria in an optical feedback loop. *Nature Communications*, 13(1):1–8, 2022. Cited on pages 7 and 74.
- [85] Yan Meng, Hongliang Guo, and Yaochu Jin. A morphogenetic approach to flexible and robust shape formation for swarm robotic systems, 2013. Cited on page 7.
- [86] Mehran Mesbahi and Magnus Egerstedt. *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010. Cited on pages 10, 17, 22, and 23.
- [87] John G. Mooney and Eric N. Johnson. A Comparison of Automatic Nap-of-the-earth Guidance Strategies for Helicopters. *Journal of Field Robotics*, 31(4):637–653, 2014. Cited on page 16.
- [88] Kazuki Muku, Hiroshi Yamashita, Touya Kamikubo, Nobuhiko J. Sue-matsu, and Makoto Iima. Long-time behavior of swimming *Euglena gracilis* in a heterogenous light environment. *Frontiers in Cell and Developmental Biology*, 11(February):1–8, 2023. Cited on page 82.
- [89] Denis Nikitin, Carlos Canudas-de Wit, and Paolo Frasca. A Continuation Method for Large-Scale Modeling and Control: From ODEs to PDE a Round Trip. *IEEE Transactions on Automatic Control*, 67(10), 2022. Cited on page 9.
-

-
- [90] W Nultsch and D P Hader. Photomovement in motile microorganisms-II. *Photochemistry and Photobiology*, 47(6):837–869, 1988. Cited on pages 67 and 71.
- [91] Hyondong Oh and Yaochu Jin. Evolving hierarchical gene regulatory networks for morphogenetic pattern formation of swarm robots, 2014. Cited on page 7.
- [92] Hyondong Oh, Ataollah Ramezan Shirazi, Chaoli Sun, and Yaochu Jin. Bio-inspired self-organising multi-robot pattern formation: A review. *Robotics and Autonomous Systems*, 91:83–100, 2017. Cited on page 16.
- [93] Kwang Kyo Oh, Myoung Chul Park, and Hyo Sung Ahn. A survey of multi-agent formation control. *Automatica*, 53:424–440, 2015. Cited on pages 17 and 18.
- [94] Hiroshi Okumura. Response to Light in Paramecium, 1963. Cited on page 81.
- [95] Reza Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, 2006. Cited on pages 17, 20, 51, and 52.
- [96] T. J. Pedley, D. R. Brumley, and R. E. Goldstein. Squirmlers with swirl: A model for Volvox swimming. *Journal of Fluid Mechanics*, 798:165–186, 2016. Cited on pages 70 and 83.
- [97] Daniel Pickem, Paul Glotfelter, Li Wang, Mark Mote, Aaron Ames, Eric Feron, and Magnus Egerstedt. The Robotarium: A remotely accessible swarm robotics research testbed. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1699–1706, 2017. Cited on page 47.
- [98] Daniel Pickem, Myron Lee, and Magnus Egerstedt. The GRITSBot in its natural habitat - A multi-robot testbed. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2015)*, volume 2015-June, pages 4062–4067, 2015. Cited on page 47.
- [99] Carlo Pinciroli, Mauro Birattari, Elio Tuci, Marco Dorigo, Marco Del Rey Zapatero, Tamas Vinko, and Dario Izzo. Self-organizing and scalable shape formation for a swarm of pico satellites. *Proceedings of the 2008 NASA/ESA Conference on Adaptive Hardware and Systems, AHS 2008*, pages 57–61, 2008. Cited on pages 18 and 21.
-

-
- [100] Marco Polin. Chlamydomonas Swims with Two “Gears” in a Eukaryotic Version of Run-and-Tumble Locomotion. *Science*, 487(July):487–491, 2009. Cited on page 68.
- [101] Daniel Polsby and Robert Popper. The Third Criterion: Compactness as a Procedural Safeguard Against Partisan Gerrymandering. *Yale Law & Policy Review.*, 9(2), 1991. Cited on page 116.
- [102] Souvik Roy, Mario Annunziato, and Alfio Borzi. A Fokker–Planck Feedback Control-Constrained Approach for Modelling Crowd Motion. *Journal of Computational and Theoretical Transport*, 45(6):442–458, 2016. Cited on page 9.
- [103] Michael Rubenstein, Cornejo Alejandro, and Radhika Nagpal. Programmable self-assembly in a thousand-robot swarm. *Science*, 345(6198):795–799, 2014. Cited on page 7.
- [104] Michael Rubenstein, Adrian Cabrera, Justin Werfel, Golnaz Habibi, James McLurkin, and Radhika Nagpal. Collective transport of complex objects by simple robots. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*, pages 47–54, 2013. Cited on page 16.
- [105] Giovanni Russo and Mario di Bernardo. Contraction theory and master stability function: Linking two approaches to study synchronization of complex networks. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 56(2):177–181, 2009. Cited on pages 6 and 10.
- [106] Prabhu Sailesh, Li William, and McLurkin James. Hexagonal lattice formation in multi-robot systems. *Distributed Autonomous Robotic Systems*, 104:307–320, 2014. Cited on pages 20, 29, and 43.
- [107] M. Saji and F. Oosawa. Mechanism of Photoaccumulation in Paramecium bursaria. *The Journal of protozoology*, 21(4):556–561, 1974. Cited on page 82.
- [108] Kazunori Sakurama and Toshiharu Sugie. Generalized Coordination of Multi-robot Systems. *Foundations and Trends® in Systems and Control*, 9(1):1–170, 2021. Cited on pages 17 and 18.
- [109] Markus Schatten, Petra Grd, Mladen Konecki, and Robert Kudelić. Towards a Formal Conceptualization of Organizational Design Techniques for Large Scale Multi Agent Systems. *Procedia Technology*, 15:576–585, 2014. Cited on page 1.
-

-
- [110] Antonella Sgarbossa, Giovanni Checcucci, and Francesco Lenzi. Photoreception and photomovements of microorganisms. *Photochemical and Photobiological Sciences*, 1(7):459–467, 2002. Cited on page 72.
- [111] Barbara Shannon, Criseida G. Zamora-Chimal, Lorena Postiglione, Davide Salzano, Claire S. Grierson, Lucia Marucci, Nigel J. Savery, and Mario Di Bernardo. In Vivo Feedback Control of an Antithetic Molecular-Titration Motif in *Escherichia coli* Using Microfluidics. *ACS Synthetic Biology*, 9(10):2617–2624, 2020. Cited on page 13.
- [112] Dragoslav D. Šiljak. Stability of Large-Scale Systems. *IFAC Proceedings Volumes*, 5(1):75–85, 1972. Cited on page 10.
- [113] Herbert A. Simon. The Architecture of Complexity. In *Proceedings of the American Philosophical Society*, volume 106, pages 467–482, 1962. Cited on page 5.
- [114] Cristian A. Solari, Richard E. Michod, and Raymond E. Goldstein. *Volvox barberi*, the fastest swimmer of the Volvocales (Chlorophyceae). *Journal of Phycology*, 44(6):1395–1398, 2008. Cited on page 83.
- [115] Kwangmin Son, Filippo Menolascina, and Roman Stocker. Speed-dependent chemotactic precision in marine bacteria. *Proceedings of the National Academy of Sciences of the United States of America*, 113(31):8624–8629, 2016. Cited on page 68.
- [116] Yang Song and Jason M. O’Kane. Decentralized formation of arbitrary multi-robot lattices. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1118–1125, 2014. Cited on pages 19, 21, and 28.
- [117] Enrica Soria, Fabrizio Schiano, and Dario Floreano. SwarmLab: a MATLAB Drone Swarm Simulator. *arXiv*, pages 8005–8011, 2020. Cited on page 112.
- [118] Diana Spears, Wesley Kerr, and William Spears. Physics-Based Robot Swarms For Coverage Problems. *International Journal of Intelligent Control and Systems*, 11(3):124–140, 2006. Cited on page 7.
- [119] William M. Spears and Diana F. Gordon. Using artificial physics to control agents. In *Proceedings - 1999 International Conference on Information Intelligence and Systems*, pages 281–288. IEEE, 1999. Cited on pages 20 and 41.
-

-
- [120] William M. Spears, Diana F. Spears, Jerry C. Hamann, and Rodney Heil. Distributed, physics-based control of swarms of vehicles. *Autonomous Robots*, 17(2-3):137–162, 2004. Cited on pages [17](#), [18](#), [19](#), [20](#), [21](#), [28](#), [29](#), [32](#), [41](#), [42](#), [43](#), [44](#), [49](#), and [112](#).
- [121] E. B. Steager, D. Wong, N. Chodosh, and V. Kumar. Optically Addressing Microscopic Bioactuators for Real-time Control. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2015. Cited on page [74](#).
- [122] Robert Stephany and Christopher Earls. PDE-READ: Human-readable partial differential equation discovery using deep learning. *Neural Networks*, 154:360–382, 2022. Cited on page [2](#).
- [123] Steven H Strogatz. *Nonlinear Dynamics and Chaos*. 1994. Cited on page [6](#).
- [124] Tzer Han Tan, Alexander Mietke, Junang Li, Yuchao Chen, Hugh Higinbotham, Peter J. Foster, Shreyas Gokhale, Jörn Dunkel, and Nikta Fakhri. Odd dynamics of living chiral crystals. *Nature*, 607(7918):287–293, 2022. Cited on pages [16](#) and [71](#).
- [125] Salvatore Torquato. Inverse optimization techniques for targeted self-assembly. *Soft Matter*, 5(6):1157–1173, 2009. Cited on pages [17](#), [18](#), [20](#), and [112](#).
- [126] Jason Town and Orion Weiner. Rac negative feedback links local PIP 3 rate-of-change to dynamic control of neutrophil guidance, 2023. Cited on page [74](#).
- [127] Angelo Trotta, Marco Di Felice, Federico Montori, Kaushik R. Chowdhury, and Luciano Bononi. Joint Coverage, Connectivity, and Charging Strategies for Distributed UAV Networks. *IEEE Transactions on Robotics*, 34(4):883–900, 2018. Cited on pages [19](#) and [21](#).
- [128] Alan C.H. Tsang, Amy T. Lam, and Ingmar H. Riedel-Kruse. Polygonal motion and adaptable phototaxis via flagellar beat switching in the microswimmer *Euglena gracilis*. *Nature Physics*, 14(12):1216–1222, 2018. Cited on pages [70](#), [82](#), and [98](#).
- [129] Alan Turing. The Chemical Basis of Morphogenesis. *Philosophical Transactions of the Royal Society of London*, 237(641):683–764, 1952. Cited on page [7](#).
- [130] Noriko Ueki, Shigeru Matsunaga, Isao Inouye, and Armin Hallmann. How 5000 independent rowers coordinate their strokes in order to row into the
-

- sunlight: Phototaxis in the multicellular green alga *Volvox*. *BMC Biology*, 8, 2010. Cited on page 82.
- [131] Matthew Uppington, Pierangelo Gobbo, Sabine Hauert, and Helmut Hauser. Evolving Morphologies for Locomoting Micro-scale Robotic Agents. In *Proceedings of MARSS 2022 - 5th International Conference on Manipulation, Automation, and Robotics at Small Scales*, pages 1–5. IEEE, 2022. Cited on pages 73, 74, and 76.
- [132] Gabriele Valentini, Eliseo Ferrante, and Marco Dorigo. The best-of-n problem in robot swarms: Formalization, state of the art, and novel perspectives. *Frontiers Robotics AI*, 4(MAR), 2017. Cited on page 7.
- [133] Thijs van den Berg. Calibrating the Ornstein-Uhlenbeck (Vasicek) model, 2011. Cited on page 98.
- [134] Jan Roelof Van Der Meer and Shimshon Belkin. Where microbiology meets microengineering: Design and applications of reporter bacteria. *Nature Reviews Microbiology*, 8(7):511–522, 2010. Cited on page 66.
- [135] Willem A. Velema, Jan Pieter Van Der Berg, Wiktor Szymanski, Arnold J.M. Driessen, and Ben L. Feringa. Bacterial patterning controlled by light exposure. *Organic and Biomolecular Chemistry*, 13(6):1639–1642, 2015. Cited on page 74.
- [136] Gang Wang, Mingzhe Liu, Fengchen Wang, and Yan Chen. A Novel and Elliptical Lattice Design of Flocking Control for Multi-agent Ground Vehicles. *IEEE Control Systems Letters*, 7:1159–1164, 2022. Cited on pages 17 and 20.
- [137] Hanlin Wang and Michael Rubenstein. Shape Formation in Homogeneous Swarms Using Local Task Swapping. *IEEE Transactions on Robotics*, 36(3):1–16, 2020. Cited on page 16.
- [138] Ximing Wang, Jinsheng Sun, Zixing Wu, and Zhitao Li. Robust integral of sign of error-based distributed flocking control of double-integrator multi-agent systems with a varying virtual leader. *International Journal of Robust and Nonlinear Control*, 32(1):286–303, 2022. Cited on page 20.
- [139] Peter Washington, Karina G. Samuel-Gama, Shirish Goyal, Ashwin Ramaswami, and Ingmar H. Riedel-Kruse. Interactive programming paradigm for real-time experimentation with remote living matter. *Proceedings of the National Academy of Sciences of the United States of America*, 116(12):5411–5419, 2019. Cited on pages 73 and 82.
-

-
- [140] Ralph Wichterman. *The Biology of Paramecium*. Plenum Press, second edition, 1986. Cited on page 81.
- [141] Neshika Wijewardhane, Ana Rubio Denniss, Matthew Uppington, Helmut Hauser, Thomas E Goroehowski, Eugenia Piddini, and Sabine Hauert. Long-term imaging and spatio-temporal control of living cells using light. *Proceedings of MARSS 2020: International Conference on Manipulation, Automation, and Robotics at Small Scales*, 2022. Cited on pages 66, 73, 74, and 76.
- [142] S. Wilson, P. Glotfelter, L. Wang, S. Mayya, G. Notomista, M. Mote, and M. Egerstedt. The Robotarium: Globally Impactful Opportunities, Challenges, and Lessons Learned in Remote-Access, Distributed Control of Multirobot Systems. *IEEE Control Systems Magazine*, 40(1):26–44, 2020. Cited on page 47.
- [143] Siyuan Yang, Mingji Huang, Yongfeng Zhao, and H. P. Zhang. Controlling Cell Motion and Microscale Flow with Polarized Light Fields. *Physical Review Letters*, 126(5):58001, 2021. Cited on page 72.
- [144] V. Zaburdaev, S. Denisov, and P. Hänggi. Perturbation spreading in many-particle systems: A random walk approach. *Physical Review Letters*, 106(18):16–19, 2011. Cited on page 69.
- [145] V. Zaburdaev, S. Denisov, and J. Klafter. Lévy walks. *Reviews of Modern Physics*, 87(2):483–530, 2015. Cited on pages 69 and 111.
- [146] Dimitri Zarzhitsky, Diana F. Spears, and William M. Spears. Distributed robotics approach to chemical plume tracing. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, number June 2014, pages 4034–4039, 2005. Cited on pages 7 and 66.
- [147] Haitao Zhao, Jibo Wei, Shengchun Huang, Li Zhou, and Qi Tang. Regular topology formation based on artificial forces for distributed mobile robotic networks. *IEEE Transactions on Mobile Computing*, 18(10):2415–2429, 2019. Cited on pages 7, 16, 17, 19, 21, 29, and 41.
- [148] Yongkun Zhou, Bin Rao, and Wei Wang. UAV swarm intelligence: Recent advances and future trends. *IEEE Access*, 8:183856–183878, 2020. Cited on page 11.
- [149] Adam Zienkiewicz, David A.W. Barton, Maurizio Porfiri, and Mario di Bernardo. Data-driven stochastic modelling of zebrafish locomotion. *Journal of Mathematical Biology*, 71(5):1081–1105, 2015. Cited on pages 70 and 97.
-

Author's publications

The results presented in Chapters 4 and 5 of this thesis were published in:

1. A. Giusti, M. Coraggio and M. di Bernardo, 2023, “Local Convergence of Multi-Agent Systems Toward Rigid Lattices” in *IEEE Control Systems Letters*, vol. 7, pp. 2869-2874, doi: [10.1109/LCSYS.2023.3289060](https://doi.org/10.1109/LCSYS.2023.3289060).
2. A. Giusti, GC. Maffettone, D. Fiore, M. Coraggio and M. di Bernardo, 2023, “Distributed control for geometric pattern formation of large-scale multirobot systems” in *Frontiers in Robotics and AI*, doi: [10.3389/frobt.2023.1219931](https://doi.org/10.3389/frobt.2023.1219931).

Moreover, the content of Chapters 7 and 8 is currently being refined for publication.

External resources

For up to date information about the following items, visit the corresponding web page.

- The DOME www.theopendome.org.
- SwarmSim www.github.com/diBernardoGroup/SwarmSimPublic.
- SINCRO research group at the University of Naples Federico II
www.sites.google.com/site/dibernardogroup.
- BioCompute Lab at the University of Bristol
<https://biocomputelab.github.io>.