Giorgio Oliveri

Designing Optimal Behaviour in Mechanical and Robotic Metamaterials

DESIGNING OPTIMAL BEHAVIOUR IN MECHANICAL AND ROBOTIC METAMATERIALS

Ph.D. Thesis, Eindhoven University of Technology, March 2021
Designing optimal behaviour in mechanical and robotic metamaterials
Giorgio Oliveri

Coverimage: Andrea Garcia Negrete

A catalogue record is available from the Eindhoven University of Technology Library
ISBN: 978-90-386-5230-6

This work is part of the Dutch Research Council (NWO). It is part of the research program Innovational Research Incentives Scheme Veni from NWO with project number 15868 (NWO).

The work described in this thesis was performed at AMOLF,
Science Park 104, 1098XG, Amsterdam, The Netherlands.

This thesis was printed on FSC© certified materials.
A digital version of this thesis can be downloaded from
http://amolf.nl and https://research.tue.nl/

Designing optimal behaviour in mechanical and robotic metamaterials

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus prof.dr.ir. F.P.T. Baaijens, voor een commissie aangewezen door het College voor Promoties, in het openbaar te verdedigen op vrijdag 19 maart 2021 om 11:00 uur

door

Giorgio Oliveri

geboren te Catania, Italië.

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt: :

voorzitter	prof.dr. L.P.H. de Goey
1^{e} promotor	dr.ir. J.T.B. Overvelde
2^{e} promotor	prof.dr. H. Nijmeijer
Promotiecommissieleden:	dr.ir. J.T.B. Overvelde
	prof.dr. H. Nijmeijer
	prof.dr. C. Storm
	dr.ir. E. Steur
	prof.dr. M. van Hecke (Universiteit Leiden)
	prof.dr. A.R. Studart (ETH Zurich)

Het onderzoek of ontwerp dat in dit thesis wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

"Raise your words, not your voice.
It is rain that grows flowers, not thunder."

Jalal ad-Din Rumi

To Andrea

Contents

CHAPTER 1

Introduction

Humankind has been morphing materials to their needs since its first appearance, and the adoption of different types of materials became a reference measure of our evolution. While the importance of shape in materials has been vastly harnessed – e.g. a metal wheel as opposed to a metal knife – the idea that geometry alone could help surpass the limits of known laws in physics is relatively new. While traditional materials posses properties depending on their chemical composition and physical properties, metamaterials exhibit unusual behaviours simply because of their geometry. The *meta* prefix, which translate from Greek to *beyond*, was added to highlight that these novel materials display properties which are not found in Nature, and go beyond the material properties of their constituents. The surprising behaviour that metamaterials show was first demonstrated in 1898, when Jagadis Chunder Bose was able to rotate the polarization plane of microwaves, by interposing twisted structures between the source and the receiver [1]. From there, the study of metamaterials grew to revolutionize various fields, from optics to acoustics, from electronics to material science, from solid state physics to mechanics.

Within the branch of mechanics, the novelty of metamaterials can be understood with a simple yet meaningful experiment, by considering an elastic material such as a common piece of rubber. As one might know from experience, when compressing it along a certain direction, it will expand in an orthogonal direction as shown in Fig. 1.1. While the lateral expansion due to the compression happens with all isotropic materials we find in Nature, by patterning the same material with holes, after some initial expansion, the material will shrink transversely instead, see Fig. 1.2. This qualitatively opposite behaviour produced by the same material with a different geometry, can be assessed by measuring the Poisson's ratio, which quantifies the relation between longitudinal and transversal strain [2]. Traditional isotropic materials have positive Poisson's ratio, with typical values of 0.5 for rubbers, 0.27 for steel and ~ 0 for cork [3]. On the other hand, materials behaving like the one presented in Fig. 1.2 exhibit a negative Poisson's ratio. These novel mechanical metamaterials are also refered to as *auxetic* materials.

Following the intuition that inverted honeycomb lattices (Fig. 1.3a) could grant auxetic behaviour [4, 5], the very first example of auxetic mechanical metamaterial was introduced in 1987 [6]. It is a polymer foam, whose internal structure consisted of unit cells with inward-protruded ribs as shown in Fig. 1.3b. In this example, the connection between adjacent unit cells, together with the inward-leaning ligaments, induces a coupling between vertical and side compression.

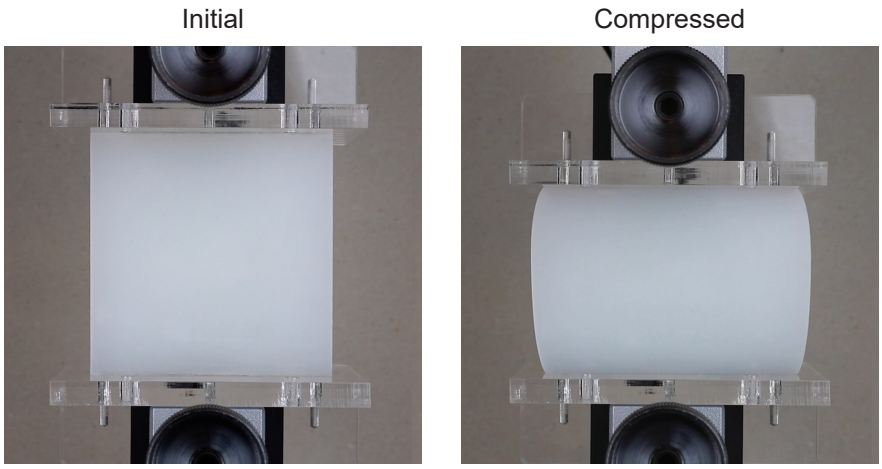

FIGURE 1.1: Initial and deformed shape for a uniaxial compression of elastomeric material (Smooth-On Dragon Skin FX-Pro).

Contrarily, when the material is pulled, the unit cells expand isotropically. Since then, this design has served as inspiration for multiple auxetic metamaterials both at the macroscale [7–10], and the microscale [11, 12]. While the negative Poisson's ratio might be the most evident characteristic of auxetic metamaterials, this feature - because of the volumetric compaction when uniaxially compressed - also enhances other mechanical properties such as material hardness, indentation strength, shear resistance, toughness and energy absorption [13–18]. These enhanced mechanical properties, together with novel geometrical features such as synclastic properties [13, 19] (as opposed to traditional anticlastic), allowed these engineered materials to be deployed in a range of products spanning from high performance sport equipment to prosthesis [18, 20, 21].

In particular, the auxetic behaviour in some of these materials is achieved by harnessing nonlinearities which originate from microscopic instabilities [22]. These materials in fact, present relatively slender internal elements which can buckle, snap or bend when loaded axially [23], see Fig. 1.2 and Fig. 1.3a-c. At the macroscopic level, the buckling of the internal elements can be detected by noticing a sharp inflection of the force-deformation response. While these buckling phenomena have previously been avoided since they are associated with catastrophic failures in structures [24], the discovery of mechanical matematerials caused an inversion of trend, moving from *buckliphobia* to *buckliphilia*, and being able to leverage and tune instabilities became the key for new exotic materials [25]. In addition, since such instabilities are characterized by sudden large deformations at relative small load increments, they enable the design of materials with sharp

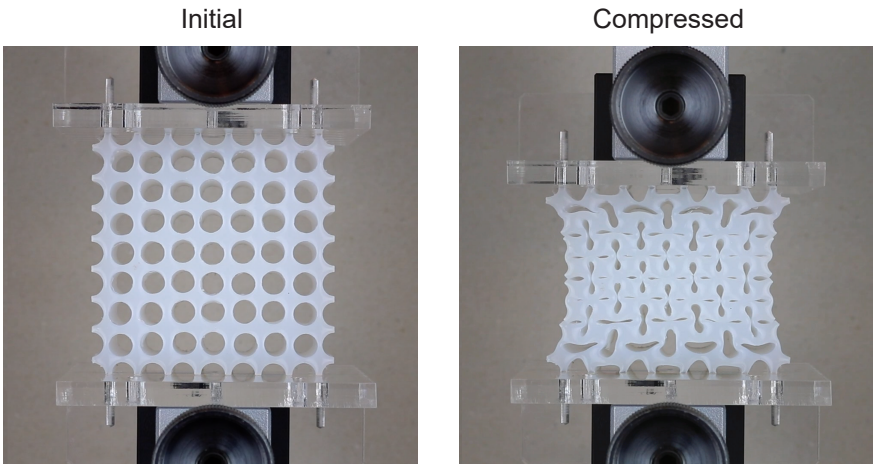

FIGURE 1.2: Initial and deformed shape for a uniaxial compression of mechanical metamaterial with circular pores (Smooth-On Dragon Skin FX-Pro).

pattern transformations [26–32], allowing for materials with rapid shift in properties [29, 33, 34], non commuting behaviour [35] and mechanical computation [36, 37].

1.1 Design of building blocks

In contrast to the rich and complex behaviour that these mechanical metamaterials show, their architecture is remarkably simple, see e.g. the circular pores on a square array in Fig. 1.2. To be able to achieve novel functionalities in fact, individual material portions need to be frustration-free, so that they deform cooperatively exhibiting a collective behaviour. This is often achieved by adopting cellular designs consisting of repetitions of identical unit cells. While natural cellular materials such as bones or bamboo have been observed to show great toughness and strength, these properties are limited to the linear elastic or plastic regimes [7]. The development of novel elastomeric solids, which can stretch up to multiple times their length, made the exploration of cellular designs within highly nonlinear regimes an exiting path to follow.

A largely explored research direction has been devoted to understand the role of the building blocks' geometry on the exotic behaviour that these material show. In [30, 33] it was shown that simply varying the pore geometry, see e.g. Fig 1.3c, can lead to tunable responses in terms of stiffness, critical strain and Poisson's ratio. The individual pore size, in relation to the unit cell dimension, can

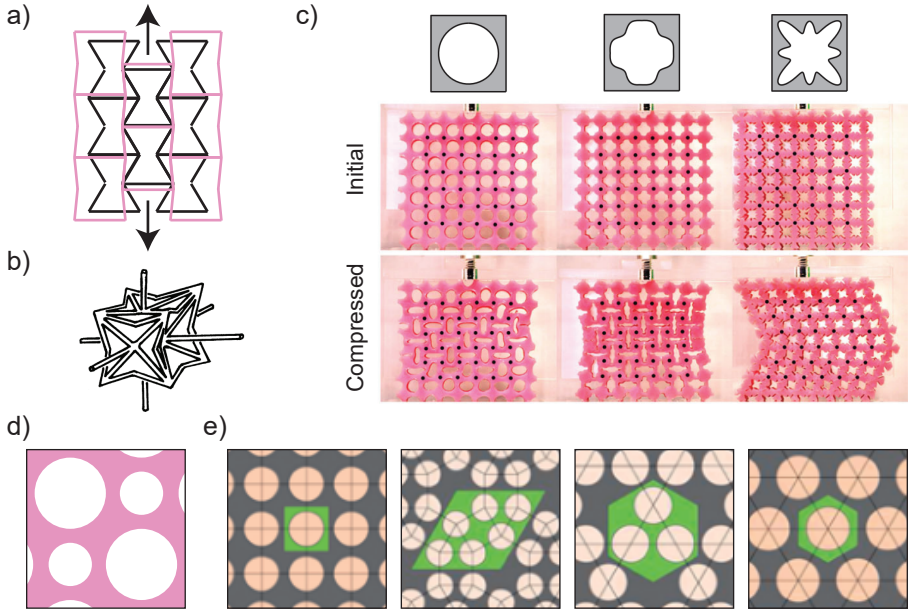

FIGURE 1.3: a), Schematic of inverted honeycomb lattice with negative Poisson's ratio [7]. The structure expand transversely when pulled longitudinally. b, Auxetic foam unit cell schematic, from [6]. Reprinted with permission from AAAS. c) Uniaxial compression of metamaterials with different hole shapes. Reprinted from [30], with permission from Elsevier. d, Biholar unit cell granting monothonic force-deformation curve in uniaxial compression [40]. e, Square, rhombitrihexagonal, trihexagonal and triangular tilings of circular holes. Republished with permission of ROYAL SOCIETY OF CHEMISTRY, from [34]; permission conveyed through Copyright Clearance Center, Inc.

be chosen to alter the wavelength of the buckling pattern [33, 38], or modify the critical strain required to obtain the pattern transformation associated with the Poisson's ratio transition from positive to negative [39]. Considering larger and more articulated elemental building blocks such as biholar unit cells (Fig. 1.3d), which break one of the symmetries that exists in the identical pore case, allows for obtaining a monotonic force-deformation curve, removing the sharp buckling transition these materials usually exhibit. In this case, the removal of one of the symmetries also strengthens the coupling between horizontal and vertical strain, enabling to calibrate the vertical snapping behaviour by controlling the horizontal confinement [29, 40].

Nevertheless, in spite of the large variety of behaviours and mechanical properties achieved to this date, most of the unit cells are designed via trial and error

processes, and a clear connection between unit cell geometry and material properties is not yet established. This created the opportunity for the adaptation of existing inverse design tools for automated unit cells designs [41–44].

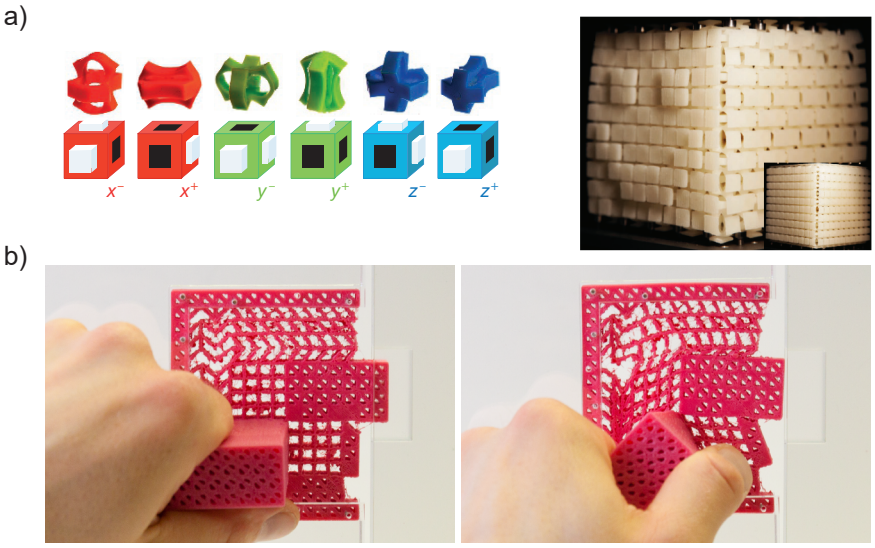

FIGURE 1.4: Heterogeneous mechanical metamaterials. a, Unit cells schematic and three-dimensional metacube material with smiley face made of $10 \times 10 \times 10$ unit cells. Reprinted by permission from Copyright Clearance Center, Inc.: Springer Nature [37], Copyright 2016. b, Mechanisms metamaterial consisting of door handle, mechanism and latch [45]. Image courtesy of A. Ion, Carnegie Mellon University.

1.2 From materials to machines

Beyond the geometry of a single unit cell, mechanical behaviour is also influence by how unit cells are assembled together to form a material. In fact, going beyond square lattices, the exploration of triangular or trihexagonal tiling patterns of unit cells containing circular pores, showed the emergence of chiral buckling patterns, together with a great diversity in achievable mechanical properties in terms of stiffness, critical load and strain, and Poisson's ratio [34].

Furthermore, the adoption of identical unit cells to form a homogeneous architecture is not the only approach to obtain frustration-free metamaterials. The choice of heterogeneous metamaterial designs comes at the cost of taking into account multiple compatibility requirements between unit cells, which demand for automated design routines. An example is the combinatorial design of textured

metamaterials, a three-dimensional cubic metamaterial consisting of an ad-hoc combination of unit cells of three different kinds, see Fig 1.4a [37]. This material has been inversely designed to display a smiley face on one of his sides, when compressed with a specific pattern on another side. One could argue that this material is actually performing some sort of computation, since the smiley face only appears if the correct compression pattern is applied as input. This example pushes the capabilities of mechanical metamaterials beyond a bare mechanical response, towards complete functionalities. One could imagine completely replacing the complex combinations of materials and structures making up the mechanisms of a door handle with a lock, with one material providing the same functionalities as shown in Fig 1.4b [45]. An updated version of this material, is able to verify that the correct numerical 'code' is inserted, before transferring the motion of the handle to the latch, opening the door [36]. These are just a few examples of how metamaterials, with the aid of automated design, are blurring the boundaries between materials and rudimental machines.

1.3 Active building blocks

Despite the design process used to accomplish a functionality, the exotic properties these material exhibit are fixed: one design has a set of sought properties or functions. What if materials could be made of active building blocks whose features, going beyond the mere geometrical mass distribution, could affect the material properties or behaviour altogether? One could imagine a unit cell which changes shape as a function of time, modifies the interactions with the neighbouring units, or performs actions, locally altering the system. To do so, each unit would require elements to sense the environment, actuators to perform actions, and controllers (or 'brains') to interface the two. This novel and emerging field, which since the last couple of years has been referred to as *Active* or *Robotic metamaterials*, is broadening the definition of what is traditionally thought of as a material [46–49].

The use of active units within materials include the potential of locally adding and extracting energy to the system, this is something that is not possible in common materials, allowing for the observation of non-Hermitian effects [50, 51]. The implementation of non-reciprocal unit cells (Fig 1.5a), which e.g. behave differently if a stimulus acts clockwise instead of counterclockwise, is enough to create a material featuring unidirectional and non-reciprocal amplification of pulse [49]. The adoption of active elements like gyroscopes, together with their placement on regular lattices (Fig 1.5b), was shown to allow for the emergence of topological edge modes, where energy can be propagated only along the edge of the lattice with high resistance to disorder [52].

These examples endorse how systems which exhibit new properties such as unidirectional transmission of energy or topological features, can be now engineered for energy localization and trapping, communication and sensing applications. Yet,

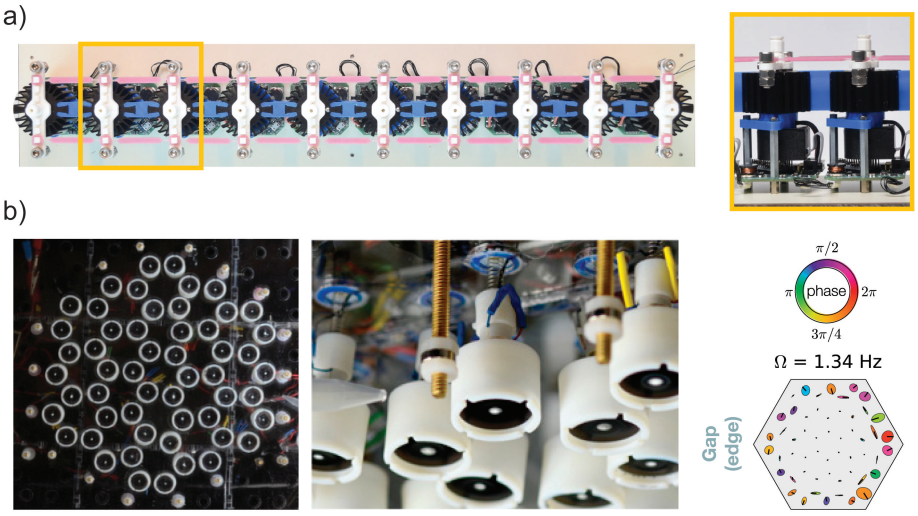

FIGURE 1.5: Robotic metamaterials. a, Non-reciprocal robotic metamaterial [49]. b, Topological gyroscopic metamaterial [52]. Images adapted and reproduced with permissions from the respective publishers.

in these examples unit cells are reactive and respond to external stimuli following a carefully predefined set of rules. But what should these rules be? How should the unit behave to get the properties we seek? These are million-dollar questions.

1.4 Research objectives

In the field of mechanical metamaterials, researchers have shown how cellular design has been vastly harness to produce novel properties emerging across scales. Specifically, the majority of these peculiar properties are obtained by leveraging instabilities at the microscale. Because most of the automated design strategies fail to correctly tune instabilities, or have a cumbersome implementation, the largest number of designs is still produced via trail and error approaches. Furthermore, since the challenging numerical description of such instabilities is shared among different fields, the impact of existing algorithms remained limited to a handful of applications. As a consequence, the first objective of this thesis is:

> *Objective 1. Develop a general inverse design algorithms able to deal with disjoint optimization landscapes.*

This objective is primarily addressed by focusing on designing the unit cell geometry (i.e. the unit cell features) such that a targeted global buckling behaviour is obtained.

Differently, when considering robotic metamaterials, a unit cell feature is also the relation between sensed inputs and the local response, which grants these materials surprisingly novel behaviours. As a consequence, the same robotic metamaterial platform can display a plethora of behaviours, by simply updating the input-output relation of its units. A natural question to ask is what should these features be, in order to obtain a certain global behaviour? Can we apply a similar inverse design strategy to robotic metamaterials, such that they can improve their behaviour by themselves, by only relying on the information sensed at the unit scale? This would create a metamaterial with capability of learning. If this was possible, these metamaterial could dynamically adapt to changing circumstances while maintaining optimal behaviour. To set a general direction in answering these questions, we start with the assumption that a measure of the behaviour we are interested in is available to individual active unit cells. Then, the second objective of this thesis can formalized as:

> *Objective 2. Develop and characterize general and minimal learning strategies for a multi-agent systems.*

This objective translates into defining the unit cells inherent dynamic behaviour, and establish how the learning occurs based on the information sensed by each unit. In the characterization phase, we are interested in assessing how the learning capabilities are affected by the environmental changes and the system size, while also defining its limitations. Nevertheless, within the framework of metamaterials, such learning strategies require a physical platform where its efficacy can be tested when facing real-life circumstances. The transition to a physical platform also highlights that these learning strategies need to be simple enough, such that they can be potentially embedded in very small physical units with limited computational power. In order to make an incremental step, we aim at designing a platform where unit-to-unit interactions are limited to a line configuration. The next objective can be defined as follows:

> *Objective 3. Design and develop a robotic metamaterial platform, and its corresponding numerical model, with the aim of assessing different learning strategies under various real-life circumstances.*

This objective entails the definition of a common target which is representative of the global sought behaviour, which can be sensed independently by the units. Specifically, the global behaviour envisioned here is the forward locomotion of the robotic metamaterial, where the units' behaviours need to adapt to the environmental circumstances in order to sustain the motion.

Next, to increase the complexity of the robotic platform to accommodate for general real-life operating conditions such as e.g. motions in two-dimensional space, we need to move away from the chain-like configuration towards two-dimensional assembly of unit cells. This implicates that the robotic metamaterial can exhibit more articulated behaviours, and create the conditions to test the learning strategies with novel challenges. The last objective of this thesis can be formulated as:

> *Objective 4. Design a simplified, but more general model of a robotic metamaterial to test more complex objectives and situations.*

1.5 Thesis outline

The contributions towards the objectives defined in the previous section, are conveyed throughout the chapters of this thesis as described here.

In Chapter 2 we developed a general, Monte Carlo-based inverse design algorithm which is able to deal with disjoint optimization problems. While we mostly look at the design of buckling-based mechanical metamaterials, where we e.g. tailored buckling properties within a wide range of values, we also tackle a different benchmark problem of compliance optimization. Furthermore, by combining simulation and experiments, we show that with the tuning of higher buckling modes, we can alter their interaction avoiding unstable and unpredictable material behaviours. While most of the results are focused on the vertical compression of a beam-like geometry, we also look at 2D lattices with multiple load cases.

In Chapter 3 we move on to robotic metamaterials, where we design a one-dimensional material made of independent active units with sensors, actuators and controllers that do not exchange information. Here, the conceived material behaviour is the capability of learning locomotion in a predefined direction in spite of the external circumstances. This is done by implementing Monte Carlo-based learning strategies on each unit, to adjust the unit's behaviour depending on the sensed data from the environment. Supported by both in-real-life and simulated experiments, we show that one of the two learning strategy is able to make the system resistant to damage, changes in the environment and rearrangement of cells.

Next, in Chapter 4 we developed a simulation framework where the robotic metamaterial's physical interaction with the environment is simplified to the evaluation of some idealized environment functions, so we can efficiently test the learning strategies without e.g. solving the equation of motion of our system. By assessing how the learning performances are affected by system size or variation in the environment, it helps us define the limiting factor of our learning strategies. Specifically, we find that within the settings considered in out simulated

framework, the rate at which the environment is changing, as perceived by the individual party, is a crucial aspect for a successful learning process.

Finally, in Chapter 5 we extend one learning strategy to a two-dimensional material. Here, with further experimental validation in mind, we reintroduce the system's physical interaction with the environment, and developed a simplified simulation framework to study the emergence of locomotion in larger and arbitrary-shaped bidimensional systems. Despite the increase in complexity deriving from motion on the plane or diverse units assembly, we show that one of the introduced learning strategy is suitable for the control of such robotic metamaterial systems. In fact, while the system is not be able to exploit good locomotion patterns, it preserves the ability to learn and adapt to variable circumstance while maintaining a positive whole-body locomotion towards the target, even with the interposition of obstacles.

Overall, the results presented in this thesis aim at designing novel metamaterials, where complex behaviours can be engineered and learnt bottom-up from the simpler unit-to-unit interactions. Specifically in the robotic metamaterial prospect, we believe that the results presented here offer a leap forward for the development of thriving ubiquitous autonomous systems.

CHAPTER 2

Inverse design of mechanical metamaterials that undergo buckling

Published as:

Oliveri, G., Overvelde, J.T.B. 2020. *Inverse Design of Mechanical Metamaterials That Undergo Buckling.* Advanced Functional Materials 2020, adfm.201909033.

Metamaterials are man-made materials which get their properties from their structure rather than their chemical composition. Their mesostructure is specifically designed to create functionalities not found in nature. However, despite the broad variety of metamaterials developed in recent years, a straightforward procedure to design these complex materials with tailored properties has not yet been established. Here we tackle the inverse design problem by introducing a general optimization tool to explore the range of material properties that can be achieved. Specifically, we apply a stochastic optimization algorithm and demonstrate its applicability to disjoint problems, in which we focus on optimizing the buckling properties of mechanical metamaterials, including experimental verification of our predictions. Besides this problem, our algorithm can be applied to a large variety of systems that, because of their complexity, would be challenging otherwise. Potential applications range from the design of optomechanical resonators, acoustic band gap materials, to dielectric metasurfaces.

2.1 Introduction

Metamaterials are materials that derive their properties from their structure, not only from their chemical composition. The microstructure of these materials is specifically designed to create new functionalities not found in nature, such as materials characterized by a negative index of refraction [53, 54], negative Poisson's ratio [6], and a very high stiffness to weight ratio [55], or structures that enable optical [56] and mechanical [57] cloaking. While most of the properties of these metamaterials are fixed, compliance, resulting from the use of soft or relatively thin materials, can be used as a paradigm to design reconfigurable metamaterials with tunable functionality. Applications range from materials with adaptive auxetic behaviour [30, 33, 39, 58], tunable stiffness [59], or ad-hoc optical [60, 61], phononic [26, 62] and acoustic [63] properties, to tunable surface properties such as the drag coefficient [64], wettability [65] and chemistry [66].

A particularly interesting avenue has been to harness mechanical instabilities in the design of reconfigurable metamaterials, which changes the continuous nature of the transformation to a discrete response originating from bifurcation.[1] Interestingly, these instabilities can be used to increase the sensitivity to external loads, and enable multistability and hysteretic behaviour [40]. In contrast to the complex and unstable behaviour that these mechanical metamaterials show, their architecture is often surprisingly simple. One of the iconic examples is an elastomeric material patterned with a square lattice of circular pores [39]. Upon compression, a collective buckling instability suddenly changes the Poisson's ratio from positive to negative, and in a similar fashion changes the phononic behaviour by opening and closing band gaps [26]. While several studies focused on the effects of pore shape [30], pore distribution [69] and material loading direction [70], the mechanical properties have only been tuned within limits dictated by a few geometrical parameters. Here, design optimization approaches could play a key role in solving the inverse problem to design mechanical metamaterials with specific properties, and explore the bounds of achievable functionality.

A specific inverse design approach that has been successfully applied to a large variety of problems is called topology optimization. This method has been developed to allow for complete design freedom by varying the local density of the structure, and therefore does not require a description and parametrization of the geometry beforehand [71]. While topology optimization was initially used to

[1] When a material is mechanically loaded, its configuration in terms of e.g. the applied load and its deformation, describes the equilibrium path. Materials can have multiple and distinct equilibrium paths for the same applied load. Knowing a material's equilibrium path is as important as knowing the stability of that paths [67, 68]. A configuration on an equilibrium path is considered stable if after a perturbation it is applied to it, the system returns to the same configuration. Differently, the configuration is considered unstable when the system will move away from it. In this case, the system will try to move to the closest, compatible equilibrium path which is stable. When two or more equilibrium paths cross, we find a bifucation point.

solve mechanical design problems such as maximizing structural stiffness using a limited amount of material [72–74], it gradually expanded towards other research areas such as optics [75–78], phononics [79], material science [44, 80, 81], and fluid mechanics [82]. Importantly, most of the algorithms use gradient information of the objective function and constraints to reach a local or global minimum. Therefore, the optimization problem needs to be continuous and differentiable with respect to the design variables. While some work has been done to include buckling behaviour in the optimization problem either through constraints [83] or the objective function [74, 84], the requirement of gradient information in the applied methods complicates the applicability and generalization of these approaches for more nonlinear problems. Moreover, the presence of multiple local minima makes the optimization highly dependent on the initial conditions [85] and search algorithm. Besides these gradient-based methods, other approaches to optimize the behaviour of mechanical networks exist, including stochastic optimization approaches [86, 87] and evolutionary approaches that are inspired by mechanical allostery [88, 89].

Here, we apply stochastic optimization to design the buckling response of periodic structures, in which the objective functions do not require objective function differentiability. To reduce the search space and to generate smooth structures, we introduce a heuristic subroutine inspired by the ferromagnetic Ising model. In order to determine the effectiveness of such an approach, we reduce computational time by considering the linear buckling response, such that we can perform multiple optimizations and compare their results (i.e. > 500 solutions per objective). More specifically, we show how it is possible to design structures with maximum buckling load, but also allow tailoring to a predefined buckling force within a wide range of values. We furthermore show that by controlling the occurrence of higher modes, we can effectively remove multi-mode interactions that occur for nearly degenerate bifurcations. Finally, we validate our optimized designs using compression experiments on elastomeric samples. While the performances of gradient based topology optimization algorithms have been proven to be unmatched [85], here we show that stochastic algorithms, due to their flexibility and relative simplicity of implementation, provide a general approach beneficial to explore more complex problems.

Problem description

To emphasize how the morphology of a structure can alter its buckling behaviour in a discontinuous fashion, we performed a buckling analysis on a range of beams with different pore shape. Each beam is composed of $n = 10$ vertically placed

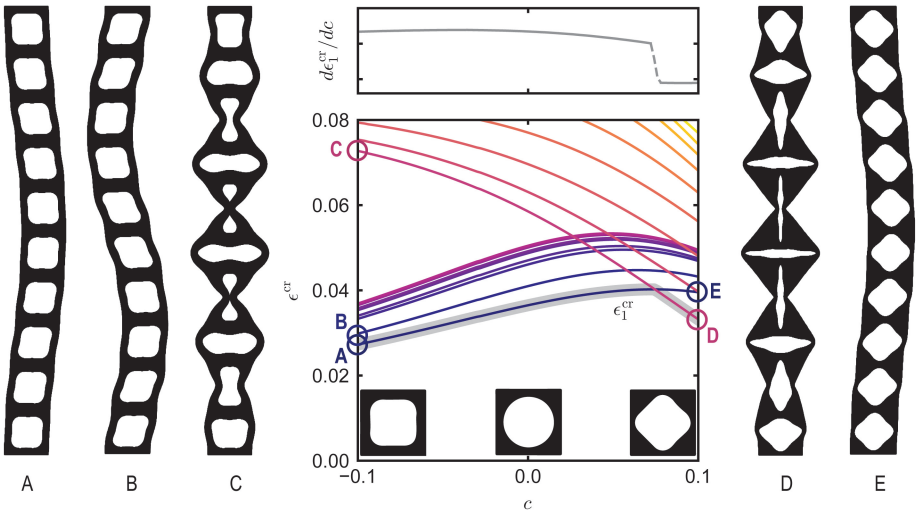

FIGURE 2.1: FEA simulation results showing the effect of pore shape on the buckling behaviour of periodic porous beams under vertical compression. Here the first twenty modes are shown. Since mode switching occurs at $c = 0.075$, the buckling strain of the first mode ϵ_1^{cr} is not continuously differentiable, such that gradients based optimization algorithms cannot be applied to optimize buckling behaviour.

square unit cells that contain a pore defined by a geometrical parameter c. The radius of the pore shape is defined as [30]

$$r(\theta) = \frac{L\sqrt{2\phi}}{\sqrt{\pi(2+c^2)}}\left[1 + c\,\cos(4\theta)\right], \tag{2.1}$$

in which $0 \leq \theta \leq 2\pi$, L is the size of the unit cell and ϕ the porosity. We applied a vertical compressive strain on beams with a porosity $\phi = 0.5$, and ran linear buckling analyses for $-0.1 \leq c \leq 0.1$ using the Finite Element Analysis (FEA) package Abaqus, in which we discretized the structure using biquadratic plane stress elements (CPE8R). In Figure 2.1 we show the critical strain ϵ^{cr} for the first 20 buckling modes as a function of the pore shape.[2] By monitoring the displacement of 11 nodes along the longitudinal axis of the beams, we were able to correlate the mode shape between the different beams via bivariate correlation [90]. This enables us to identify mode switching that arises from changes in pore geometry.

We next focus on the buckling mode with the lowest critical strain, since this mode will occur upon compression of the beam. We find that for $c < 0.075$ the

[2]In this case, the buckling modes represent the shapes that the beams assume when losing stability in the primary compression path, for a given critical strain ϵ^{cr} and pore shape c.

buckling mode of the beam is characterized by a typical macroscopic buckling mode with a wavelength of $2nL$ that is equal to twice the length of the beam, while for $c > 0.075$ a microscopic buckling mode occurs characterized by a wavelength of $2L$ equal to twice the size of the unit cell [38, 91]. When we would perform an optimization to e.g. maximize the lowest critical strain of the beam according to

$$\max_{c} \quad \epsilon_1^{\text{cr}}, \tag{2.2}$$

we find that the derivative of the objective function $d\epsilon_1^{\text{cr}}/dc$ is discontinuous due to the change in buckling behaviour (Fig. 2.1). While this specific problem can be tackled with gradient-based algorithms by rewriting Eq. 2.2 using a bound formulation [84, 92], a perturbation-based sensitivity analysis should be performed for each individual eigenvalue by solving sub-optimization problems at each optimization step. For the specific case shown in Fig. 1 in which the optimal solution lies exactly at the point where the first two modes switch order, gradient-based methods would require taking progressively smaller steps in the proximity of mode swaps in order to guarantee smoothness. This, together with the difficulties associated to tuning for a specific target buckling value ϵ_1^{cr}, strengthens the need of optimization techniques that do not rely on gradient information to seek the optimum.

2.2 Stochastic optimization

In order to find structures with interesting and specific buckling behaviour, we developed a density-based stochastic topology optimization approach based on the geometrical description used in the SIMP interpolation method [93, 94]. The topology of the structure, which is divided into square elements, can be changed by varying the density of the elements between $0 \leq \rho_i \leq 1$. Differently, in our stochastic optimization implementation we can use a discrete representation of the design since we do not require intermediate unrealistic densities to make our design continuously differentiable [86]. As such, we assume that each element can take a density ρ_i of $\rho_{\text{min}} \approx 0$ (void) or $\rho_{\text{max}} = 1$ (solid). When assuming linear elastic behaviour of each element, we can evaluate the initial mechanical and buckling response of the structures by using a linear FEA code implemented in Matlab (details in Sec. 2.6), in which the stiffness and Poisson's ratio of each element are given by $E_i = \rho_i E$ and $\nu_i = \rho_i \nu$ to account for the geometry of the structure.

Starting from an initial random density distribution with a solid/void ratio of ϕ, we alter the geometry by randomly picking and exchanging two elements with different density. After each change in geometry, the objective function $\Phi(\rho_i)$ is evaluated. Following a simulated annealing optimization approach [95], the variation in objective function $\Delta\Phi(\rho_i)$ between current and previous iteration is

used to determine the acceptance probability, $P(\Delta\Phi(\rho_i))$ of the new candidate solution according to

$$P\left(\Delta\Phi(\rho_i)\right) = e^{-\frac{\Delta\Phi(\rho_i)}{T}}, \tag{2.3}$$

where T is a parameter often referred to as the *Temperature*, which can be used to tune the acceptance probability of optimization steps that do not improve the objective function, in order to reach global optima. Here we use a fixed number of iterations for the optimization, N_{iter}, and assume an exponential decay of the temperature [95, 96]. We let the temperature decrease from T_{max} to T_{min}, such that the temperature at a given iteration n is given by

$$T(n) = T_{\text{max}} \left(\frac{T_{\text{min}}}{T_{\text{max}}}\right)^{n/N_{\text{iter}}}. \tag{2.4}$$

A test problem: compliance optimization

While our main goal is to optimize buckling behaviour, we will first test our method using a typical topology optimization problem that is computationally less expensive. We focus on compliance optimization (i.e. stiffness maximization), and do this by replicating the MBB-beam optimization problem [97, 98] (see schematic in Fig. 2.2a). This allows us to compare our implementation with previous work. If we consider a domain with $n_{\text{x}} \times n_{\text{y}} = 60 \times 20$ elements, the optimization problem can be formulated as:

$$\min_{\rho_i} \qquad \Phi(\rho_i) = C = \{\boldsymbol{D}\}^T[\boldsymbol{K}]\{\boldsymbol{D}\}, \tag{2.5a}$$

$$s.t. \qquad \{\boldsymbol{F}\} = [\boldsymbol{K}]\{\boldsymbol{D}\}, \tag{2.5b}$$

$$\frac{V_{\text{solid}}}{V_\Omega} = \phi^* = 0.5, \tag{2.5c}$$

where $\{\boldsymbol{D}\}$ and $\{\boldsymbol{F}\}$ are vectors containing the displacement and reaction forces at the nodes, respectively, which are related by the stiffness matrix $[\boldsymbol{K}]$ (see Sec. 2.6). Moreover, the volume constraint of Eq. 2.5c enforces that the ratio between the volume of material elements, V_{solid}, and the total domain volume, V_Ω, is constant. Note that in our approach this volume constraint is automatically satisfied, and depends on the initial solid/void fraction. Finally, we normalize the compliance of our designs C by the compliance of a homogeneous structure C_{ref} with $\rho_i = 0.5$, such that $\tilde{C} = C/C_{\text{ref}}$.

We start by identifying proper temperature bounds (T_{max} and T_{min}) for the simulated annealing algorithm (Eq. 2.4). First, to determine the objective function sensitivity to the temperature, we run 10 optimizations with a total number of iterations equal to $N_{\text{iter}} = 1.5 \times 10^4$, during which the temperature is decreased according to Eq. 2.4 between $T_{\text{max}} = 10^2$ and $T_{\text{min}} = 1 \times 10^{-9}$. By looking at the

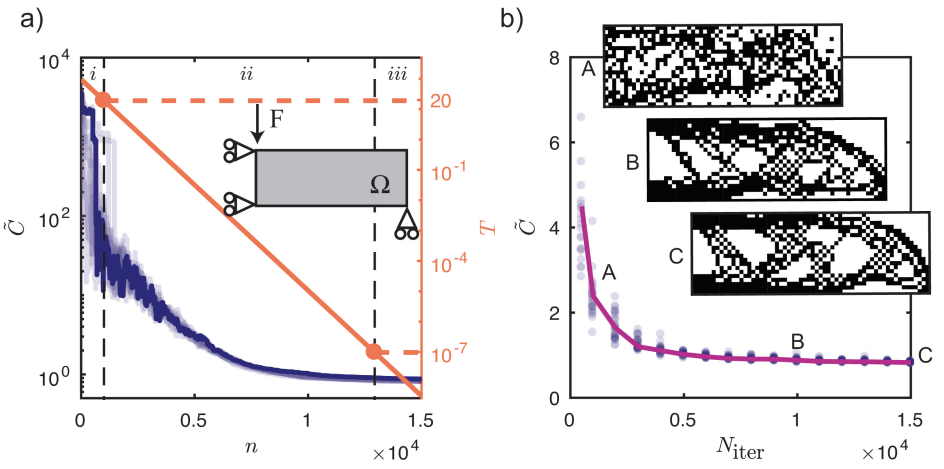

FIGURE 2.2: Compliance optimization of the MBB-beam problem using our stochastic optimization algorithm without the Ising-inpired subroutine. a) Selecting the optimal temperature bounds (T_{max} and T_{min}) for the simulated annealing algorithm for actual optimization runs by performing ten different optimizations. b) Dependency of the final optimized solutions on the number of iterations N_{iter}. For each value of N_{iter}, 30 optimizations have been performed, where the insets show one of the final optimized solutions for $N_{\text{iter}} = 10^3$, 10^4 and $1.5 \cdot 10^4$.

objective function evolution for decreasing T (Fig. 2.2a), we identify three regimes. (*i*) For high temperatures ($T > 20$) any candidate solution is accepted, such that the objective function fluctuates around the same value and there is no convergence to an optimum. (*ii*) Intermediate temperatures ($1 \times 10^{-7} \leq T \leq 20$) result in convergence towards a minimum, with a probability to reach a global minimum. (*iii*) For $T < 1 \times 10^{-7}$ only candidate solutions with lower objective function are accepted, the algorithm behaves as a random search with steepest descent and therefore only convergences to a local minimum. For optimal behaviour of the simulated annealing algorithm, we focus on regime (*ii*), such that $T_{\text{max}} = 20$ and $T_{\text{min}} = 1 \times 10^{-7}$.

Using these specific bounds for the temperature, we next focus on finding the required number of iterations N_{iter}. To do so, we performed several optimizations with $500 \leq N_{\text{iter}} \leq 15000$. In Figure 2.2b we show the final objective function values for 30 optimizations per data point. As expected, we find that a larger number of iterations (i.e. a slower cooling rate) benefits the objective function optimization. This can also be seen by looking at the topologies associated with different cooling rates (Fig. 2.2b), which become more clustered given enough time. Balancing the time required for the optimization with the final objective function value, $N_{\text{iter}} = 10^4$ seems to be a good choice for the number of iterations, for which the typical $\tilde{C} \approx 0.9$.

2.3 Ising-inspired subroutine

As can be seen from the compliance optimization examples (Fig. 2.2b), while the solutions are converging, the current approach leads to structures with fragmentation and local checkerboard patterns. This is a typical problem in topology optimization, and is the result of artificial stiffening [99, 100]. As a result, the optimized geometries highly depends on the choice of mesh density. Moreover, apart from introducing numerical artifacts, these local patterns make the fabrication impossible.

Typically, this problem is solved by introducing local averaging [101], which is not suitable for our optimization given that we do not take into account intermediate densities. Differently, for discrete grids a potential solution to overcome checkerboard patterns is e.g. by introducing additional constraints on the local connectivity of material elements during each step of the optimization [102, 103], or by removing the checkerboard pattern by random perturbations during a finite number of the optimization iterations [86]. These methods however only prevent checkerboard patterns, or loosely connected structures, and it is not possible to control more global features in the final optimized structures such as interface length or other feature sizes. Therefore, we introduce an approach to heuristically reduce the probability of certain geometrical features by drawing only specific candidate solutions. Here, we take inspiration from the 2D ferromagnetic Ising Model [104, 105], and implement a subroutine to prioritize clustering of the material, as it is a well known that for the Ising model the interface length can be tuned by varying the Ising temperature. While the Ising model describes the behaviour of a system formed by magnetic dipole moments of different spins interacting among each other, we translate the spin diversity into material diversity. If we relate the spin to the density of each element according to $s_i = 2\rho_i - 1$, using the Von Neumann neighborhood the total energy in our system can be written as

$$E = -\frac{1}{2}\sum_{i=1}^{N_{\text{elem}}} s_i \left[s_{\text{up}(i)} + s_{\text{down}(i)} + s_{\text{left}(i)} + s_{\text{right}(i)} \right], \tag{2.6}$$

where the spin of each element s_i is only affected by the spins of the four adjacent elements. Therefore, if an element of a certain spin s_i is inverted, the system will experience a variation in Ising Energy equal to

$$\Delta E_i = 2\, s_i \left[s_{\text{up}(i)} + s_{\text{down}(i)} + s_{\text{left}(i)} + s_{\text{right}(i)} \right], \tag{2.7}$$

which, according to the 2D Ising Method, is only accepted with a probability equal to

$$P(\Delta E_i) = e^{-\frac{\Delta E_i}{T_{\text{Ising}}}}. \tag{2.8}$$

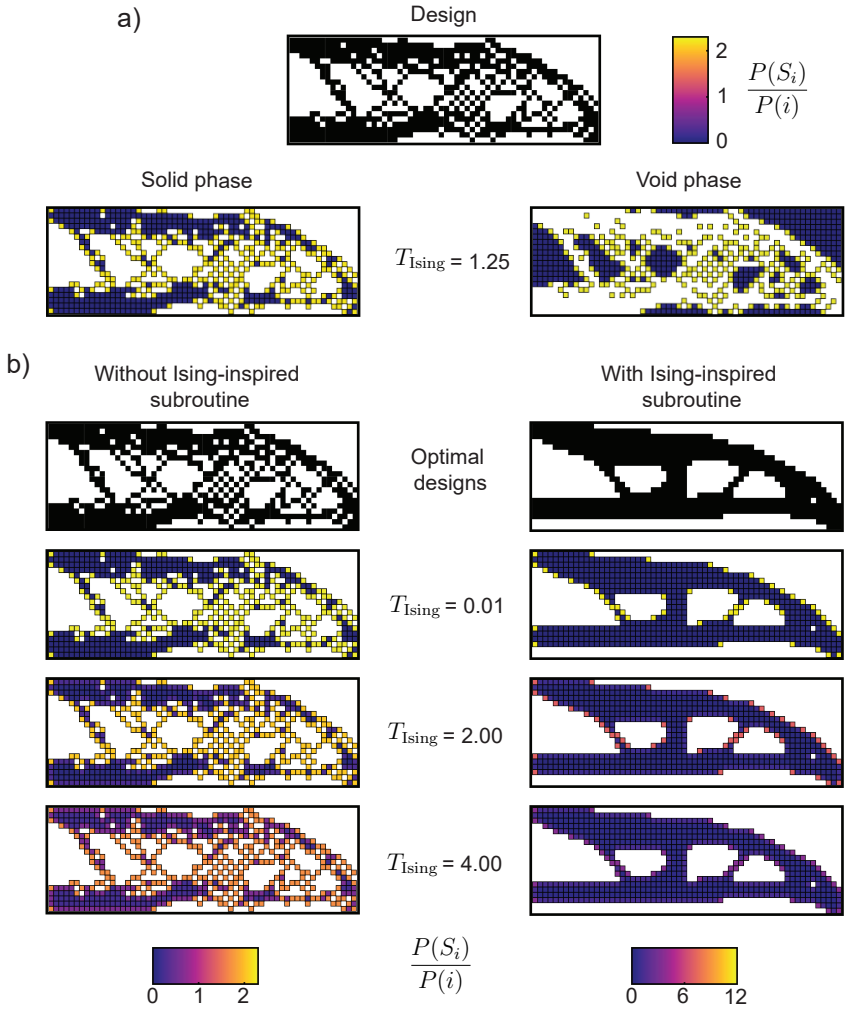

FIGURE 2.3: Compliance optimization of the MBB-beam problem using our stochastic optimization algorithm without and with Ising-inpired subroutine. a) Probability distribution of the Ising-inpired subroutine to change the density of an element as determined from Eq. (2.9) within a solid or void phase, relative to the optimal solution B in Fig. 2.2b. b) Effect of Ising temperature on the swapping probability of element in the solid phase for optimized solutions found without and with the Ising-inspired subroutine.

Here, the Ising temperature T_{Ising} can be used to tune the acceptance probability of certain topological variations. Lower T_{Ising} will result in the formation of more clustered candidates solutions, while higher T_{Ising} will not have much effect on the

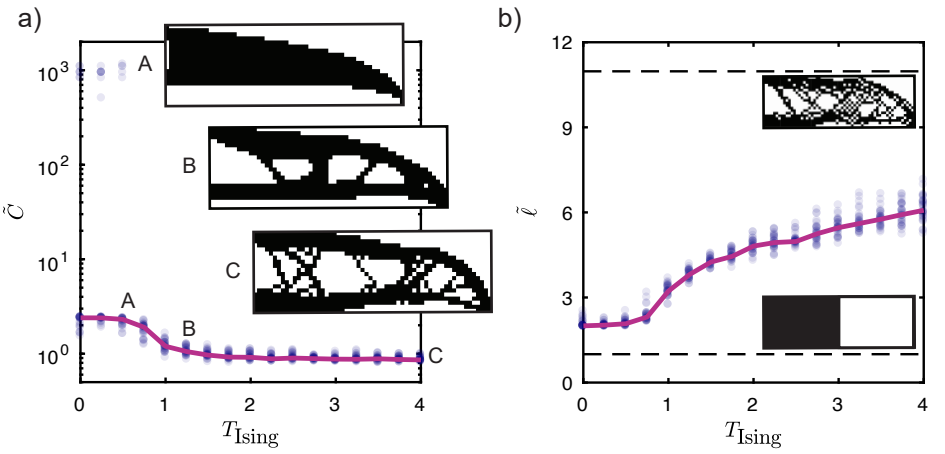

FIGURE 2.4: Compliance optimization of the MBB-beam problem using our stochastic optimization algorithm with the Ising-inpired subroutine. a) Dependency of the final optimized solutions on the Ising temperature T_{Ising}. For each value of T_{Ising}, 30 optimizations have been performed, where the insets show one of the final optimized solutions for $T_{\text{Ising}} = 0.5$, 1.25 and 4. b) Length of the material-void interface for the optimized solutions. The insets represent the bounds for the length.

selection of the candidate solution and therefore will result in topologies similar to those shown in Fig. 2.2b.

Note that for the Ising model, especially for low temperatures, it often occurs that no spins changes are made during a step. To make sure that our design changes in each iteration step, we instead draw an element from the probability distribution specified by

$$P(S_i) = \frac{P(\Delta E_i)}{\sum_{k=1}^{N_{\text{elem}}} P(\Delta E_k)}, \tag{2.9}$$

which depends on how the Ising acceptance probability of an element $P(\Delta E_i)$ relates to the total acceptance probability $\sum_{k=1}^{N_{\text{elem}}} P(\Delta E_k)$ of the current state. To ensure that the volume constraint is satisfied, each design change is divided in two steps in which we first draw a solid element, update the density and probability distribution, and then we draw a void element. In Figure 2.3a we show the material and void probability distributions of an optimized design with a checkerboard pattern, in which the density distributions are normalized with respect to the probability of a random element selection $P(i)$ (i.e. without using the Ising-inpired subroutine). Moreover, in Fig. 2.3b we show the effect of T_{Ising} on the probability of making a solid element void for the given configuration. We observe that for low T_{Ising}, the elements located at the solid-void interface (e.g. checkerboard regions) have a high probability of being swapped, while the elements within

homogeneous regions have almost zero probability. Increasing the temperature will increase the probability of selecting elements within the homogeneous regions, which in the limit of $T_{\text{Ising}} \to \infty$ results in a fully random selection not influenced by the Ising-inspired subroutine.

We next apply this subroutine to the same compliance optimization problem considered previously, using the same parameters for the simulated annealing algorithm, and study the effect of T_{Ising} on the optimized topology. We do so by running 30 simulations per T_{Ising} value. As can be seen in Fig. 2.4a, for low T_{Ising} the material clusters dramatically, and prevents full exploration of the design space. As a result, the optimized designs have a relatively high compliance of $\tilde{C} \approx 2.4$. Moreover, for $T_{\text{Ising}} < 0.5$ we find that some optimized designs are disconnected from the boundary conditions (i.e. connected via low density elements), leading to very high objective values of $\tilde{C} \approx 10^3$. Better results are obtained for $T_{\text{Ising}} \geq 1$, for which all the optimizations converge to similar objective function values of $\tilde{C} \approx 0.9$, comparable to the results obtained without Ising-inspired subroutine.

Interestingly, while the objective functions for $T_{\text{Ising}} > 1$ found with our stochastic optimization approach are similar, the optimized topologies depend greatly on T_{Ising}. Two examples for which we have used $T_{\text{Ising}} = 1.25$ and $T_{\text{Ising}} = 4$ are shown in Fig. 2.4a. Clearly, lower T_{Ising} results into more clustered solutions and mitigates the checkerboard patterns, while high T_{Ising} creates thin material connections with sharp material-void interfaces affected by numerical errors. This effect of T_{Ising} on the topology can be shown by considering the final length $\tilde{\ell} = \ell/\ell_{\text{min}}$ of the solid-void interface as shown in Fig. 2.4b, where the lower dashed line represents the minimum length ℓ_{min} the given domain can achieve, and the upper dashed line is the length of the optimal design obtained without the Ising-inspired subroutine shown in Fig. 2.3a. For $T_{\text{Ising}} > 1$ we find a direct relation between the Ising temperature and the interface length. Similarly, we counted the number of edges that each material element has (Fig. 2.11), and find that for $T_{\text{Ising}} < 1$ the number of elements with two or less neighbors (i.e. which are needed for checkerboard patterns) remains below 7%. Taking both effects into account, we select $T_{\text{Ising}} = 1.25$ as the temperature in the following studies. Note that the results obtained for $T_{\text{Ising}} = 1.25$ are $\sim 30\%$ higher, and therefor not as good as, the optimal material distribution obtained with the SIMP method [98], see Section 2.6.

Furthermore, besides reducing the interface length and removing checkerboard patterns, introducing the Ising-inspired subroutine heuristically reduces the number of candidate solutions that are considered during each iteration of the optimization. This is shown in Fig. 2.3b, where we plot the probability of selecting elements of an optimal design obtained with the Ising-inspired subroutine. Notice how the elements with high probability of being swapped are mostly at the material-void interface. Therefore, the optimization first explores the interesting part of the design space, associated with the material interface rather than considering the full domain at each iteration.

2.4 Buckling behaviour optimization

The strength of our algorithm becomes visible when dealing with disjoint problems, where conventional gradient-based approaches fail, have a limited versatility, or complicate the implementation. In this section we will focus on determining the topology of a periodic structure that undergoes buckling. We focus on a beam-like domain composed of $n_{\mathrm{x}}^{\mathrm{uc}} \times n_{\mathrm{y}}^{\mathrm{uc}} = 2 \times 10$ unit cells, in which each unit cell is made of $n_{\mathrm{x}} \times n_{\mathrm{y}} = 20 \times 20$ elements that are used as the design variables. We fix the top and bottom of the beam, and apply a vertical compression. Moreover, to better estimate the bifurcation buckling we impose mirror symmetry with respect to the vertical axis for the density distribution [106] (see schematic in Fig. 2.5a).

Optimizing for a specific buckling load

We start by determining the optimal topology of the periodic beam for a predefined buckling load. Given a target critical force, the optimization problem can be defined as:

$$\min_{\rho_i} \quad \Phi(\rho_i) = \sqrt{(\lambda_1 - \lambda_{1T})^2} \tag{2.10a}$$

$$s.t. \quad ([\boldsymbol{K}] - \boldsymbol{\lambda}[\boldsymbol{K_\sigma}])\{\boldsymbol{\delta D}\} = \{\mathbf{0}\}, \qquad \{\boldsymbol{\delta D}\} \neq \{\mathbf{0}\} \tag{2.10b}$$

$$\boldsymbol{\lambda} = \{\lambda_1,\ \lambda_2,\ \dots,\ \lambda_n\}^T \tag{2.10c}$$

$$\frac{V_{\text{solid}}}{V_\Omega} = \phi^* = 0.5, \tag{2.10d}$$

where λ_1 is the first eigenvalue, λ_{1T} is the target eigenvalue, and $[\boldsymbol{K}]$ and $[\boldsymbol{K_\sigma}]$ are the stiffness matrix and the geometrical stiffness matrix, respectively (see Sec. 2.6). The eigenvalues $\boldsymbol{\lambda}$ are coefficients proportional to the buckling force of the structure. Note that we consider only positive eigenvalues related to compression of the beam, and ignore negative eigenvalues related to tension. Moreover, while here we used $\Phi(\rho_i) = \sqrt{(\lambda_1 - \lambda_{1T})^2}$ for our objective function, similar results might be obtained by using $\Phi(\rho_i) = (\lambda_1 - \lambda_{1T})^2$, although this would affect the specific temperatures that need to be set for the optimization. Furthermore, by performing the same parameter study as done in previous sections for the compliance optimization problem, we obtain suitable temperatures for this problem given by $T_{\text{max}} = 0.01$, $T_{\text{min}} = 10^{-8}$ and $N_{\text{iter}} = 3000$. Finally, we use $T_{\text{Ising}} = 1.25$ as determined previously.

In Figure 2.5a we show the results of 900 optimizations for $\tilde{\lambda}_{1T} = \lambda_{1T}/\lambda_{\text{ref}}$ in the range $[0.2, 3]$, where λ_{ref} is the first buckling load of a solid beam with equal volume fraction ϕ^*. We find that our algorithm can consistently optimize for targeted buckling forces up to $\tilde{\lambda}_{1T} \approx 1.3$ (i.e. the buckling force is improved by 30% with respect to a solid beam made with the same mass), where we find that

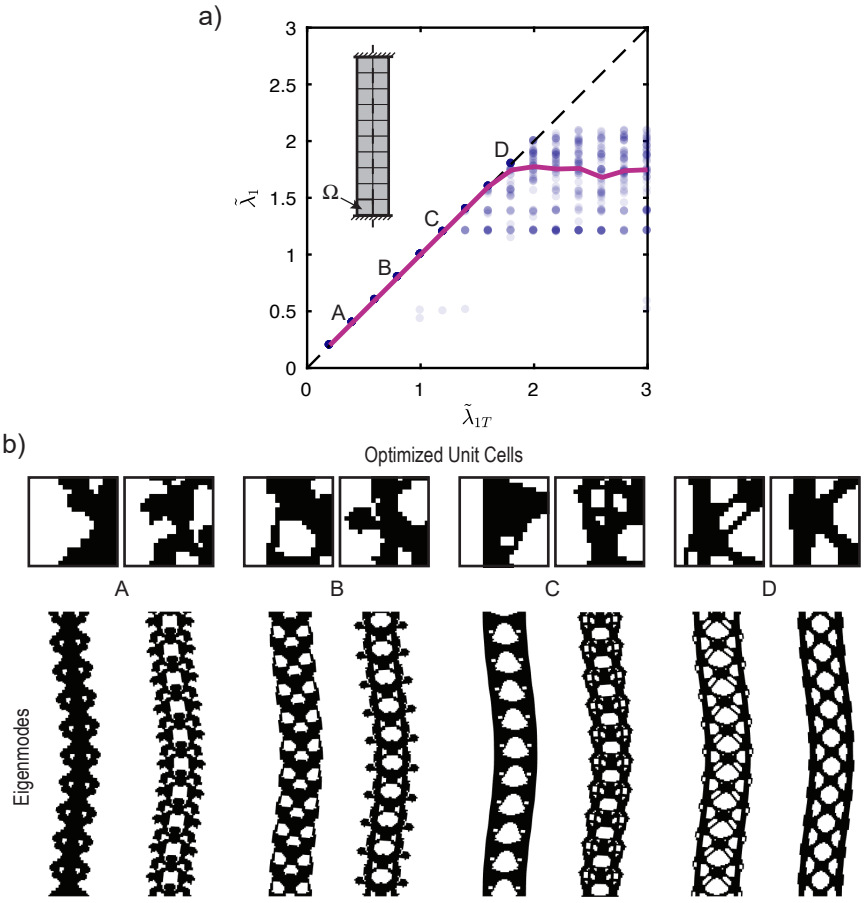

FIGURE 2.5: Optimization to a target buckling load $\tilde{\lambda}_{1T}$. a) Results of 900 optimizations (60 per selected $\tilde{\lambda}_{1T}$) as a function of the target value $\tilde{\lambda}_{1T}$. b) Selection of optimal density distributions and the corresponding buckling modes for $\tilde{\lambda}_{1T} = 0.4, 0.8, 1.2$ and 1.8.

approximately 1% percent of simulation converges to a load different than the objective. Interestingly, for each target buckling force we find multiple optimal solutions, as shown by the examples in Fig. 2.5b. However, for $\tilde{\lambda}_{1T} > 1.3$ (i.e. $\tilde{\lambda}_1 > 1.3$) we see an increase in diversity, resulting from solutions that do not converge to the target objective. In fact, these solutions are likely the result of optimizations that converge to local minima. Note that for $\tilde{\lambda}_{1T} > 2$ we find that all solutions convergence to a value lower than the target, with a median value at $\tilde{\lambda}_1 \approx 1.8$.

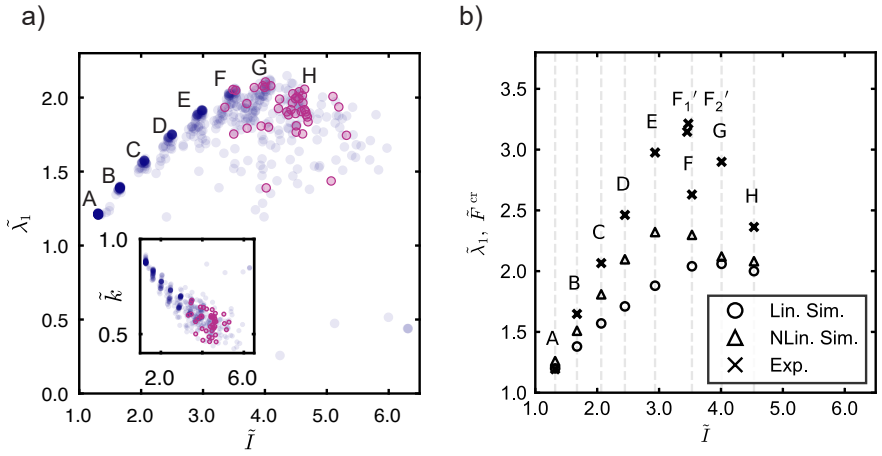

FIGURE 2.6: Maximum buckling load optimization. a) The buckling load $\tilde{\lambda}_1$ of 600 optimization in function of the optimized structures' normalized moment of inertia $\tilde{I}$. Structures with first modes characterized by long ($2n_y^{uc}L$) or short ($2L$) wavelengths are indicated by the blue and pink markers, respectively. The insets shows the effective stiffness $\tilde{k}$ of the optimized solutions. b) Comparison between prediction of critical buckling load found with linear and nonlinear simulations ($\tilde{\lambda}_1$) and experiments ($\tilde{F}^{cr}$).

Maximizing buckling load

To understand whether the maximum buckling load we achieved for this domain is $\tilde{\lambda}_1 \approx 2$ as we showed in the previous section, or that this limit is an artefact of the convergence to local minima, we next introduce an optimization problem to maximize the buckling load. This is achieved by setting the target objective buckling force in Eq. (2.10a) to an arbitrarily high value of $\tilde{\lambda}_{1T}$=10, which effectively will never be reached. Movie 1 and Movie 2 show the accepted candidate solutions of two optimizations, with and without Ising-inspired subroutine, respectively. It is important to note that during both optimization paths we encounter buckling modes of different wavelengths, indicating the disjoint nature of the problem.

We perform 600 optimizations, each starting from different random initial configurations. The results are shown in Fig. 2.6a, in which we plot the normalized buckling force $\tilde{\lambda}_1$ and average moment of inertia $\tilde{I} = \langle I \rangle / \langle I_{\text{ref}} \rangle$ for each optimized beam, where $\langle I \rangle = \iiint \rho_i x_i^2 dA\, dy$. As expected, we find that most of the results are clustered within a few local minima indicated by points A-H in Fig. 2.6a. Representative topologies for these local minima are shown in Fig. 2.7a. We find that in order to achieve higher buckling force the material distribution needs to maximize the moment of inertia, which follows Timoshenko beam theory [23]. As a consequence, an increase in buckling force lowers the stiffness of the structures as

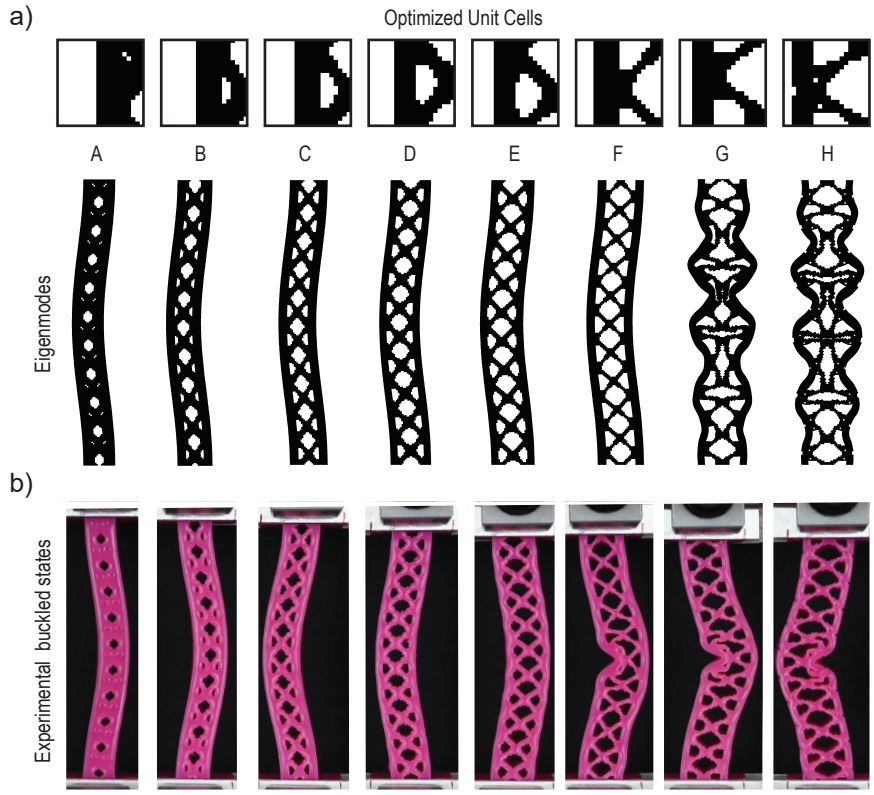

FIGURE 2.7: a-b, Simulated and experimental buckling modes of structures A-H in Fig. 2.6a.

shown in the inset of Fig. 2.6a. For beams with lower moment of inertia (e.g. beams A-F) we find that all optimized beams buckle with a wavelength equal to $2n_{\mathrm{y}}^{\mathrm{uc}}L$. However, we also observe solutions (e.g. beams G-H) in which the first buckling mode has a wavelength equal to double the size of the unit cell ($2L$), similar to the observation made in Fig. 2.1. This is the result of widening of the beams, which has a direct impact on the thickness and buckling behaviour of the internal features. By performing centroid linkage hierarchical clustering on the eigenvector results [107], we are able to assess the wavelengths of optimized beams, and distinguish beams with wavelengths equal to $2n_{\mathrm{y}}^{\mathrm{uc}}L$ from $2L$ as indicated by using blue and pink markers in Fig. 2.6a. The appearance of these two modes seems to limit the maximum buckling load that can be achieved, and we find a maximum buckling load equal to 2.04 times that of a solid beam with the same weight.

Experimental validation To quantify how the optimized topologies (A-H in Fig. 2.7a) compare with experiments, we fabricated the beams using an elastomeric rubber (Zhermack Elite Double 8). Each beam was casted in a single step in a 3D-printed mold (Stratasys Eden260VS), and has a thickness of $40mm$ and a nominal length of $140mm$, associated with a unit cell dimension of $14 \times 14mm$. We then performed five compression cycles at a rate of $30mm/min$ to a maximum compression of $20mm$ using a materials testing machine (Instron 5965L9510). From the force-displacement response we can then determine the buckling point F^{cr}, which is found at the intersection between the regression lines of the force-strain curves at small strain $\epsilon \approx 0$ and right after buckling as indicated by the sharp transition (Fig. 2.13a). To be able to compare the results with experiments we normalize each response by the experimental buckling force of the reference geometry (solid vertical beam), i.e. $\tilde{F} = F/F^{\text{cr}}_{\text{ref}}$.

The results are summarized in Fig. 2.6b, where we show the difference between the experimental buckling force $\tilde{F}_{\text{cr}}$ and simulation results $\tilde{\lambda}_1$ in function of the normalized moment of inertia of the structures $\tilde{I}$. While beam A, which is closest to a solid beam, is in agreement with the simulation results, wider beams with a more defined porous structure start to show higher buckling forces compared to the numerical results. Interestingly, for beams A-E we see an approximately linear deviation from the predicted buckling load $\tilde{\lambda}_1$ for increasing $\tilde{I}$. This is likely due to the relatively large strains that need to be applied before the beams buckle (i.e. up to $\epsilon = 0.12$), such that the beams considerably widen and shorten, resulting into an effective increase in the experimental buckling load. To limit the computation time in our simulations, these nonlinear geometrical effects have not been taken into account in the model.

However, to make sure that the deviation between theory and experiments is due to nonlinear effects, we performed additional nonlinear FEA to determine the buckling load of the optimized 2D beams under plane stress conditions. Note that we have used the same mesh as in our optimization, except for the low density elements that we removed for the nonlinear analysis. As indicated in Fig. 4c, we find that nonlinear effects such as bulging indeed seems to increase the moment of inertia of the beam, resulting in higher buckling loads. However, the obtained results do not fully agree with the experiments, in which we see an even larger increase in the buckling load. This could be the effect of additional geometrical effects that arise in the fully 3D samples. However, additional effects could also play a role, such as differences in boundary conditions, misalignment of the sample, imperfection (or e.g. increase in ligament thickness) due to the fabrication process, viscoelastic behaviour of the material, or numerical effects related to e.g. the mesh density.

Finally, this nearly linear relation between buckling force and moment of inertia does not hold for beams F-H. To determine where this deviation is coming from, we show in Fig. 2.7b the experimentally obtained buckled states for all beams

right after buckling has occurred. While for beams A-E the beams all buckle in the predicted buckling mode, for beams F-H we observe a localization of the mode and a sudden drop in force (Fig. 2.13a). This localization indicates a creasing instability [108], and is also observed in the nonlinear FEA (Fig. 2.14). Note that beams D-E also show a similar localization later along the loading path, and therefore undergo a second instability during loading. Importantly, it seems that due to the increase in buckling load of the predicted buckling mode, a mode switch occurs in beams F-H, such that for higher values of the moment of inertia the expected buckling load can no longer be reached.

The sensitivity of the buckling mode to the finite size deformations that occur during loading suggests that the critical values of higher modes are not well separated from the first mode. In Figure 2.8 we show the difference between the critical force of the first and second mode $\tilde{\lambda_2} - \tilde{\lambda_1}$, as a function of the moment of inertia $\tilde{I}$. Interestingly, we find a sudden transitions at $\tilde{I} = 3$ that separates structures that have a well-defined first mode (e.g. A-E), and structures for which multiple modes happen at the same critical force (e.g. F-H). Note that all the structures that exhibit potential microscopic buckling (i.e. have a buckling wavelength of $2L$ as indicated by the pink marker in Fig. 2.8) lay on $\tilde{\lambda_2} - \tilde{\lambda_1} \approx 0$, such that we were not able to obtain such a localized mode experimentally.

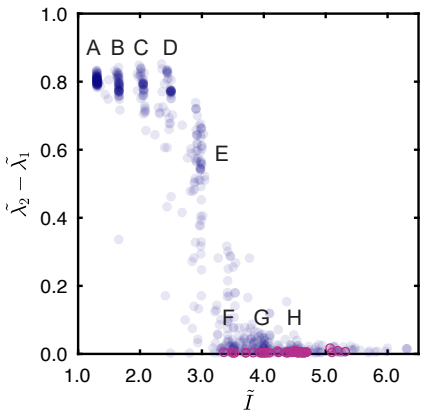

FIGURE 2.8: Difference between the buckling load of the first and second mode $\tilde{\lambda}_2 - \tilde{\lambda}_1$ of the optimized structures from Fig. 2.6, as a function of the moment of inertia $\tilde{I}$. Structures with first modes characterized by long ($2n_y^{uc}L$) or short ($2L$) wavelengths are indicated by the blue and pink markers, respectively.

Penalizing higher modes

To investigate whether mode switching influences the experimental post buckling behaviour of our optimized geometries, we next introduce a penalty in our optimization problem that enforces a minimum separation between the first and second critical buckling force. To do so, we rewrite the objective function from Eq. (2.10a) as

$$\min_{\rho_i} \qquad \Phi(\rho_i) = \sqrt{\left(\tilde{\lambda}_1 - \tilde{\lambda}_{1T}\right)^2} + \alpha \frac{\delta\tilde{\lambda} - \delta\tilde{\lambda}_T}{\delta\tilde{\lambda}_T}, \tag{2.11}$$

in which α is the penalty factor, equal to

$$\alpha = \begin{cases} 1/5, & \text{if} \quad \delta\tilde{\lambda} < \delta\tilde{\lambda}_T \\ 0, & \text{if} \quad \delta\tilde{\lambda} \geq \delta\tilde{\lambda}_T. \end{cases} \tag{2.12}$$

For further optimization studies we chose $\delta\tilde{\lambda}_T = \tilde{\lambda}_2 - \tilde{\lambda}_1 = 0.3$ as the minimum distance between the first and second buckling mode. Rerunning the previous optimization study now with the penalty applied to the buckling force of the second mode has a considerable effect on the results. This becomes clear in Fig. 2.9b, in which we show the difference between the buckling force $\delta\tilde{\lambda}$ for the optimized structures in function of the moment on inertia $\tilde{I}$. The sharp transition at $\tilde{I} \approx 3$ is still present, however the results levels off for $\tilde{I} > 3$ at the specified minimum of $\delta\tilde{\lambda} = 0.3$. It is important to note that by using this penalty we only find structures that exhibit a macroscopic buckling mode with a wavelength of $2n_{\mathrm{y}}^{\mathrm{uc}}L$. Moreover, while we are still able to achieve structures similar to A-F found previously, Fig. 2.9a shows that after the introduction of the penalty, we no longer observe clusters of structures with a microstructure similar to G and H (Fig. 2.6a).

To asses whether the separation of the first and second mode has has an effect on the buckling behaviour of our structures, we fabricated two different geometries belonging to the F′ group, and used to same protocol as before to test them in a compression machine. In Figure 2.9c we show the buckling modes of the optimized geometries F_1′ and F_2′ obtained using both simulations and experiments. While the designs are nearly identical to structure F (Fig. 2.7a), these structures undergo macroscopic buckling as predicted by our simulations, before exhibiting a creasing instability, see Fig. 2.13b. In fact, by adding the experimental results in terms of buckling force $\tilde{F}^{\mathrm{cr}}$ to Fig. 2.6b, we find that the structures exhibit the same geometrical stiffening as predicted. Therefore, we conclude that the second mode for structures F_1′ and F_2′ is separated well enough to prevent mode switching.

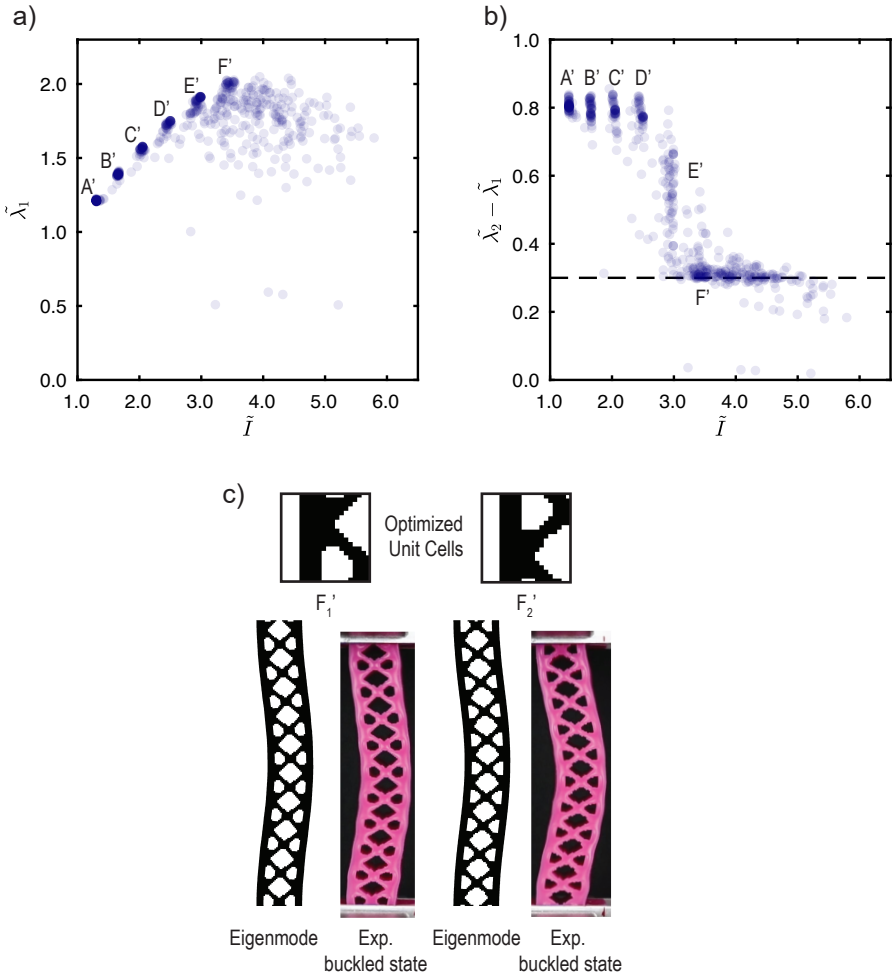

FIGURE 2.9: Maximum buckling load optimization with penalization of the second buckling mode according to Eq. 2.11 with $\delta\tilde{\lambda}_T = 0.3$. a) Buckling load $\tilde{\lambda}_1$ of 600 optimizations in function of average moment of inertia $\tilde{I}$. Most of the results cluster in the same minima A'-F' as previous maximization study without penalization, however, structures G and H are no longer present (Fig. 2.6a). b) Difference between the buckling load of the first and second mode $\lambda_2 - \tilde{\lambda}_1$ as a function of the average moment of inertia $\tilde{I}$. c) Comparison between buckling modes obtained using simulations and experiments for two representative geometries belonging to cluster F'.

Maximum mode separation

Finally, to explore the maximum mode separation that can be achieved between the first two modes, we introduce a different objective function given by:

$$\min_{\rho_i} \qquad \Phi(\rho_i) = \sqrt{\left(\tilde{\lambda}_1 - \tilde{\lambda}_{1T}\right)^2 + \left(\tilde{\lambda}_2 - \tilde{\lambda}_{2T}\right)^2}. \tag{2.13}$$

We ran a total of 2104 optimizations with target eigenvalues ranging between $\tilde{\lambda}_{1T} = [0.4, 2.4]$ and $\tilde{\lambda}_{2T} = \left[\tilde{\lambda}_{1T}, 3\right]$. In Figure 2.10 we show the results in term of the optimized buckling forces $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$. Interestingly, we find a bounded region of buckling behaviour that can be achieved, indicated by the gray area. In fact, while we can optimize for structures with coincident critical points $\tilde{\lambda}_1 \approx \tilde{\lambda}_2$ resulting in so-called frustrated structures [28], there is a maximum separation $\tilde{\lambda}_2 - \tilde{\lambda}_1$ which can be achieved. This value seems to be dependent on the $\tilde{\lambda}_1$, since for $\tilde{\lambda}_1 < 1$ the maximum separation achievable $\tilde{\lambda}_2 - \tilde{\lambda}_1 \propto \tilde{\lambda}_1$, and for $\tilde{\lambda}_1 > 1$, $\tilde{\lambda}_2 - \tilde{\lambda}_1 \propto \tilde{\lambda}_1 + 0.85$. These results are coherent with the results shown in Fig. 2.8 and Fig. 2.9b. Therefore, the gray area shown in Fig. 2.10 represents a projection of the design space onto the control variables of our inverse design problem.

2.5 Conclusions

In this work we have applied stochastic optimization to tackle disjoint topology optimization problems. Furthermore, in order to create feasible optimized geometries, we proposed an Ising-ispired subroutine to effectively guide the optimization and cluster material. After testing our method and reproducing the results of a widely-used, differentiable, compliance optimization problem, we approached novel optimization problems dealing with the buckling behaviour of mechanical metamaterials. Specifically, in this paper we focused on the optimization of the first two buckling modes of a beam, and their interaction. While here we focused on the optimization of beams that exhibit 1D periodicity, our method can also be applied on structures with higher periodicities. For example, Fig. 2.15 demonstrates the optimization of a 2D lattice, with multiple load cases. As such, our method opens up new avenues for the exploration of more fundamental questions regarding frustration and mode interaction that occur in these and more complex mechanical metamaterials.

While in current work the Ising energy has been defined by penalizing interfaces (i.e. acting effectively as a surface tension) by reducing the probability of drawing these solutions, other energy potentials with e.g. longer range interactions or physics based potentials could be used to alter the candidate selection. This could potentially vary the minimum feature size or incorporate specific geometrical features in the final solution.

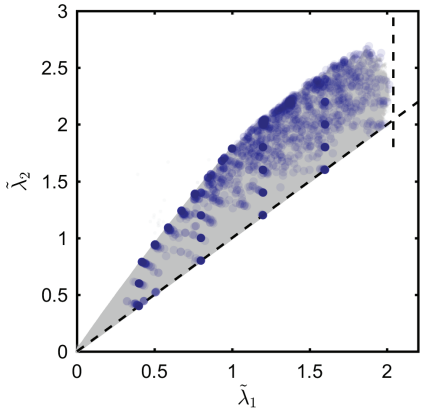

FIGURE 2.10: Optimization results for 56 different combinations of target buckling values $(\tilde{\lambda}_{1T}, \tilde{\lambda}_{2T})$, where $\tilde{\lambda}_{1T} = [0.4, 2.4]$ and $\tilde{\lambda}_{2T} = \left[\tilde{\lambda}_{1T}, 3\right]$. The blue markers indicate all optimized solutions, while the gray area shows all the evaluated solutions during the optimization, indicating the feasible design space for our optimization problem. The diagonal dashed lines represent the determined bounds, specified by $\tilde{\lambda}_1 = \tilde{\lambda}_2$, and $\tilde{\lambda}_1 < 2.1$.

Moreover, because our method does not require objective function differentiability, it can be applied to different problems that would be impossible otherwise. These can range from optimization of the buckling shape of mechanical metamaterials or the band gap behaviour of periodic and aperiodic photonic, and acoustic metamaterials, to their fully nonlinear, transient and unstable behaviour. Therefore, we believe that the flexibility and simplicity of our method is a good addition to existing gradient-based optimization problems, and is able to deal with more complex problems that have been left unexplored so far.

2.6 Supplementary information

Finite Element Analysis of buckling behaviour

To determine the buckling behaviour of 2D structures, we implemented a linear Finite Element Analysis (FEA) model made of a geometry that is discretized using square 2D bilinear quadrilateral element (Q4) [68]. The displacement $(u, v)_i$ of a coordinate (x, y) within an element i can be obtained, as a function of the nodal displacements $\{\boldsymbol{d}\}_i$, via linear interpolation using the shape functions $[\boldsymbol{N}]$, expressed as

$$\{u, v\}_i^T = [\boldsymbol{N}]\{\boldsymbol{d}\}_i, \tag{2.14}$$

in which

$$[\boldsymbol{N}] = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix}, \tag{2.15}$$

$$\{\boldsymbol{d}\}_i = \begin{bmatrix} u_1 & v_1 & u_2 & v_2 & u_3 & v_3 & u_4 & v_4 \end{bmatrix}^T, \tag{2.16}$$

and u_j and v_j ($j = 1, .., 4$) are the horizontal and vertical displacements of the four nodes corresponding to each element. In our topology optimization approach we only apply square elements with sides of length $2a$, such that the shape functions can be expressed as

$$\begin{aligned} N_1 &= \frac{1}{4}\frac{(a-x)(a-y)}{a^2}, \quad N_2 &&= \frac{1}{4}\frac{(a+x)(a-y)}{a^2}, \\ N_3 &= \frac{1}{4}\frac{(a+x)(a+y)}{a^2}, \quad N_4 &&= \frac{1}{4}\frac{(a-x)(a+y)}{a^2}. \end{aligned} \tag{2.17}$$

The element stiffness matrix $[\boldsymbol{K}^{\mathrm{e}}]_i$ is defined as

$$[\boldsymbol{K}^{\mathrm{e}}]_i = \int_{-a}^{a}\int_{-a}^{a} [\boldsymbol{B}]_i^T [\boldsymbol{E}_{\mathrm{m}}]_i [\boldsymbol{B}]_i \; t \; dx \; dy, \tag{2.18}$$

where t is the element thickness and $[\boldsymbol{B}]_i$ and $[\boldsymbol{E}_{\mathrm{m}}]_i$ are the strain-displacement and elasticity matrices, respectively, which can be written out as

$$[\boldsymbol{B}]_i = \begin{bmatrix} N_{1,x} & 0 & N_{2,x} & 0 & N_{3,x} & 0 & N_{4,x} & 0 \\ 0 & N_{1,y} & 0 & N_{2,y} & 0 & N_{3,y} & 0 & N_{4,y} \\ N_{1,y} & N_{1,x} & N_{2,y} & N_{2,x} & N_{3,y} & N_{3,x} & N_{4,y} & N_{4,x} \end{bmatrix}, \tag{2.19}$$

where terms correspond to the partial derivatives of the shape function, e.g. $N_{1,x} = \frac{\delta N_1}{\delta x}$. Moreover

$$[\boldsymbol{E}_{\mathrm{m}}]_i = \frac{E_i}{1-\nu_i^2} \begin{bmatrix} 1 & \nu_i & 0 \\ \nu_i & 1 & 0 \\ 0 & 0 & \frac{1-\nu_i}{2} \end{bmatrix}, \tag{2.20}$$

where we have assumed plain-stress conditions. Moreover, the elements Young's modulus E_i and Poisson's ratio ν_i are defined in function of the element density ρ_i, such that $E_i = \rho_i E$ and $\nu_i = \rho_i \nu$. Material element have $\rho_i = \rho_{\max} = 1$, and void element have $\rho_i = \rho_{\min} = 1 \times 10^{-5}$, while $E = 1$ and $\nu = 0.49$. The element stiffness matrices $[\boldsymbol{K}^{\mathrm{e}}]$ are numerically integrated via second order Gaussian quadrature using four sampling points and assembled to form the global stiffness matrix $[\boldsymbol{K}]$. Using this global stiffness matrix we can then determine the displacement of the global nodes $\{\boldsymbol{D}\}$ of the structure given certain boundary conditions, by solving

$$\{\boldsymbol{F}\} = [\boldsymbol{K}]\{\boldsymbol{D}\}. \tag{2.21}$$

To determine the buckling behaviour, we need to additionally solve for the stability of the structures. We do this by considering the element geometrical stiffness matrix $[\boldsymbol{K}_\sigma^{\mathrm{e}}]_i$, which is equal to

$$[\boldsymbol{K}_\sigma^{\mathrm{e}}]_i = \int_{-a}^{a}\int_{-a}^{a} [\boldsymbol{G}]_i^T [\boldsymbol{S}]_i [\boldsymbol{G}]_i \, t \, dx \, dy, \tag{2.22}$$

where $[\boldsymbol{G}]_i$ and $[\boldsymbol{S}]_i$ are defined as

$$[\boldsymbol{G}]_i = \begin{bmatrix} N_{1,x} & 0 & N_{2,x} & 0 & N_{3,x} & 0 & N_{4,x} & 0 \\ N_{1,y} & 0 & N_{2,y} & 0 & N_{3,y} & 0 & N_{4,y} & 0 \\ 0 & N_{1,x} & 0 & N_{2,x} & 0 & N_{3,x} & 0 & N_{4,x} \\ 0 & N_{1,y} & 0 & N_{2,y} & 0 & N_{3,y} & 0 & N_{4,y} \end{bmatrix}, \tag{2.23}$$

and

$$[\boldsymbol{S}]_i = \begin{bmatrix} \sigma_x & \tau_{xy} & 0 & 0 \\ \tau_{xy} & \sigma_y & 0 & 0 \\ 0 & 0 & \sigma_x & \tau_{xy} \\ 0 & 0 & \tau_{xy} & \sigma_y \end{bmatrix}. \tag{2.24}$$

Here, the stress $\{\boldsymbol{\sigma}\}_i$ can be determined from

$$\{\boldsymbol{\sigma}\}_i = \{\sigma_x, \, \sigma_y, \, \tau_{xy}\}_i^T = [\boldsymbol{E}_{\mathrm{m}}]_i [\boldsymbol{B}]_i \{\boldsymbol{d}\}_i. \tag{2.25}$$

Once the global geometrical stiffness matrix $[\boldsymbol{K}_\sigma]$ is assembled similarly to $[\boldsymbol{K}]$, the critical loads $\boldsymbol{\lambda}$ can be determined by solving the following eigenvalue problem

$$([\boldsymbol{K}] - \boldsymbol{\lambda}[\boldsymbol{K}_\sigma])\{\boldsymbol{\delta D}\} = \{\boldsymbol{0}\} \quad \{\boldsymbol{\delta D}\} \neq \{\boldsymbol{0}\}. \tag{2.26}$$

Associated with each eigenvalue λ_k, the eigenvectors $\{\boldsymbol{\delta D}\}_k$ represent the structural buckling modes. Note that for low density elements, the terms in $[\boldsymbol{K}]$ are very small compared to the respective terms in $[\boldsymbol{K}_\sigma]$ leading to buckling modes localized into low density regions. To avoid the occurrence of such artificial buckling modes, we neglected the contribution of the low density elements when assembling the geometric stiffness matrix $[\boldsymbol{K}_\sigma]$ [74].

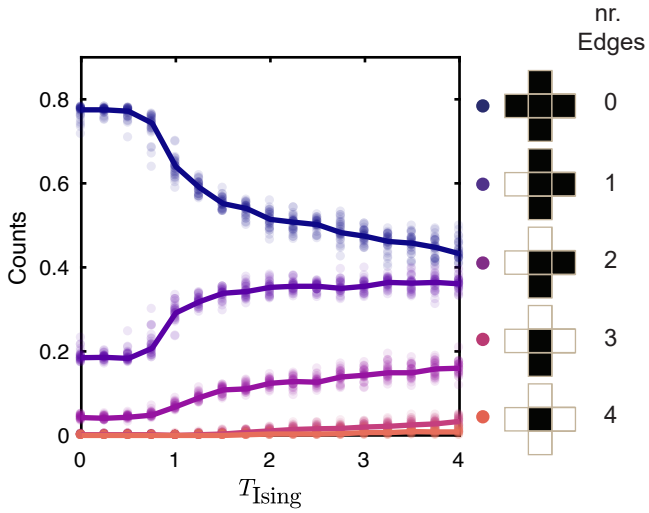

FIGURE 2.11: Number of edges per material element for each optimization as shown in Fig. 2.4a-b

Influence of Ising temperature on local element connectivity

In contract to the global effect of the Ising temperature on the final topologies as showed in Fig. 2.4b, to understand its influence on the local element connectivity we counted the relative number of material cells for each kind of direct connection for all the structures optimized for compliance, see Fig. 2.11. For $T_{\text{Ising}} < 1$ most of the optimized results show either no material-void interface, or elements that have only one edge. For $T_{\text{Ising}} \geq 1$ the structures start to exhibit more elements with 1 or 2 edges, while connections with 3 or 4 edges only account for $\approx 1\%$ at $T_{\text{Ising}} = 2$ of the total material elements. It is important to note that diagonal connections can be made only when cells with 2 or more edges are placed adjacently with a 45 degrees angle. While we do not directly control diagonal connections, as shown by above results we can control their occurrence by defining an energy potential on the nearest neighborhood.

Comparison with SIMP results

In order to compare our compliance optimization results with the SIMP results, we run the code presented in [98], using $n_{\text{x}} \times n_{\text{y}} = 60 \times 20$ elements, volume fraction $\phi^* = 0.5$, penalization power $p = 3$ and filter size of 1.5 resulting into the optimized geometry shown in Figure 2.12a, characterized by a compliance $\tilde{C} \approx 0.81$. We then evaluate the compliance of the design obtained via our stochastic

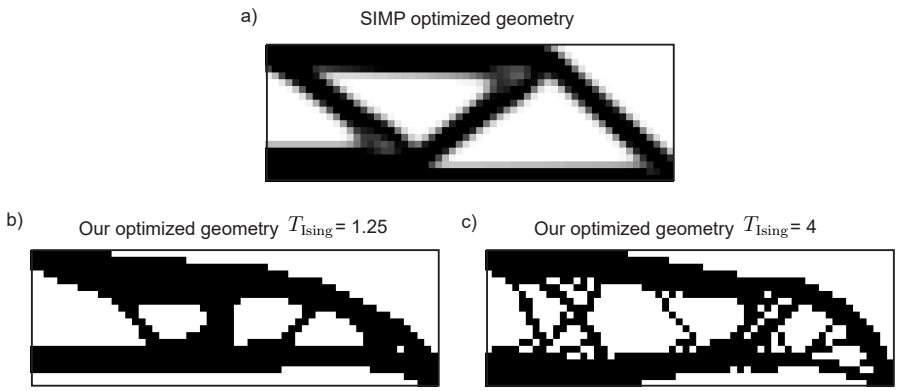

FIGURE 2.12: Compliance optimized geometries. a) Result obtained via SIMP algorithm [98]. b) Optimized geometry of our stochastic optimization algorithm using $T_{\text{Ising}} = 1.25$. c) Optimized geometry of our stochastic optimization algorithm using $T_{\text{Ising}} = 4$.

optimization algorithm, shown in Figure 2.12b, using the same code provided in [98] obtaining a compliance $\tilde{C} \approx 1.05$: our results obtained for $T_{\text{Ising}} = 1.25$ present a worse compliance, $\sim 30\%$ higher, than the optimal material distribution obtained with the SIMP method [98]. On the other hand, the results obtained for higher T_{Ising}, e.g. $T_{\text{Ising}} = 4$, while exhibiting fragmented material distribution (see Figure 2.12c), have a compliance $\tilde{C} \approx 0.86$ comparable to the SIMP results.

Experimental Validation

From the experimental force-strain response of each structure, we determined the buckling point F^{cr}, found at the intersection between the regression lines of the force-strain curves at small strain $\epsilon \approx 0$, and right after buckling has occurred. We use the buckling load of the reference geometry (solid beam) as normalization factor such that $\tilde{F} = F/F^{\text{cr}}_{\text{ref}}$. In Figure 2.13 we show the average force-strain curve of the last 4 compression cycles for the representative geometries obtained via maximum buckling load optimization, with and without penalty respectively.

NonLinear simulations FEA

To account for the geometrical changes occurring in the experiments, we performed nonlinear Finite Element Analysis (FEA) in ANSYS to determine the buckling load of 2D beams under plane stress conditions. Here we used the same mesh as used in the linear buckling analysis (ANSYS element type PLANE182), however, we removed the void elements. We modeled the structures with a nearly

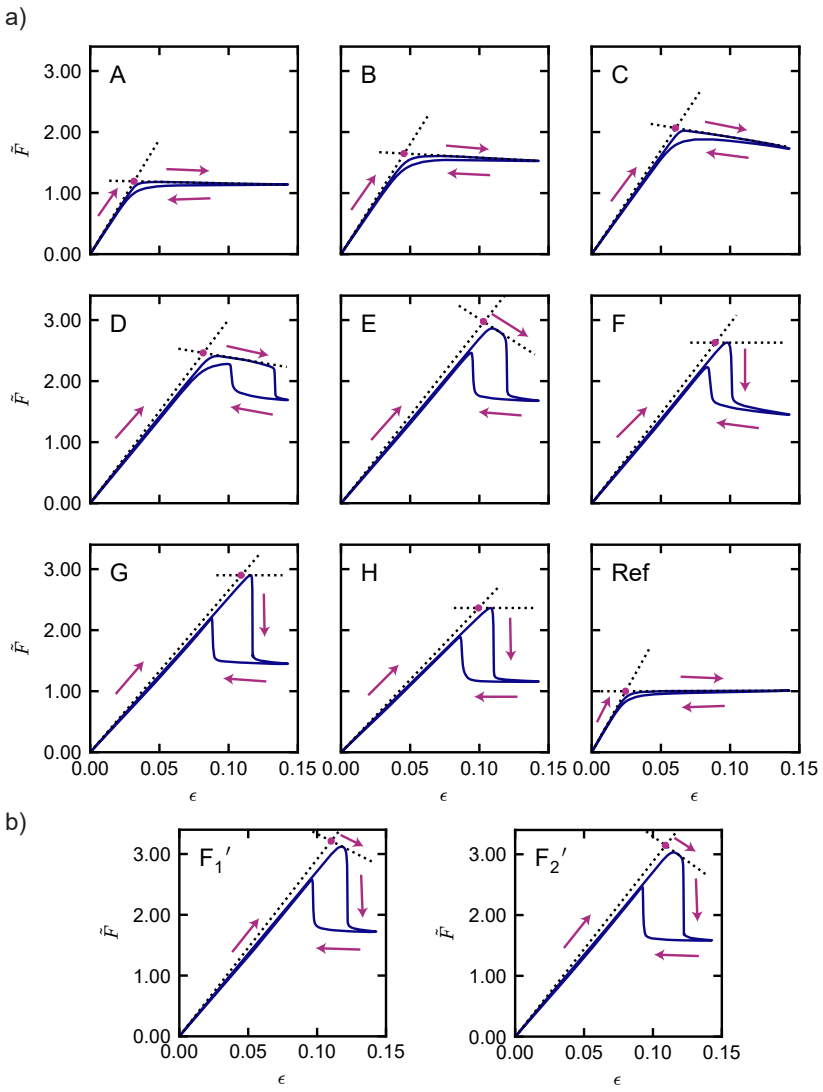

FIGURE 2.13: Normalized force-strain curves for representative beams obtained with maximum buckling load optimization. Dashed lines represent the regression lines before and after buckling, used to find the experimental buckling load shown with the pink marker. The results of all beams are normalized with respect to the buckling force of the reference (Ref) geometry. a) Normalized force-strain curves for representative beams A-H (Fig. 2.7b) obtained with maximum buckling load optimization. b) Force-strain curves of geometries F_1' and F_2' obtained via maximum buckling load optimization with penalty for the second buckling mode, $\delta\tilde{\lambda}_T = 0.3$ (Fig. 2.9).

incompressible Neo-Hookean material model, in which we used a shear modulus of $\mu = E/3$ and a bulk modulus of $K = 2\mu(1+\nu)/(3(1-2\nu))$, in which $\nu = 0.49$. Moreover, we applied an imperfection of $.1\%$ to all nodes according to the first five buckling modes of the beam. In Fig. 2.14 we show the force-strain curves obtain by the nonlinear simulation, in which the buckling force is estimated similar to the experiments at the intersection between the regression lines of the force-strain curves at small strain $\epsilon \approx 0$, and right after buckling as indicated by the sharp transition.

Multi-Objective optimization to 2D cellular materials

To show the flexibility of our topology optimization algorithm and demonstrate that our strategy can be extended to complete cellular solids, we released the symmetry constraint and studied the buckling behaviour of a $2D$ material made of 7×7 unit cells of $n_{\text{x}} \times n_{\text{y}} = 20 \times 20$ elements. We implemented a multi-objective optimization aimed at maximizing the buckling force along the x and y direction, in which we use an objective function given by

$$\min_{\rho_i} \quad \Phi_{MO}(\rho_i) = \sqrt{\Phi_x(\rho_i) \cdot \Phi_y(\rho_i)}, \tag{2.27}$$

in which $\Phi_x(\rho_i)$ and $\Phi_y(\rho_i)$ are the sub-objective functions determined according to Eq. (2.10a), obtained by applying a force independently in the x and y direction. In Fig. 2.15a we show a schematic of the enforced periodicity in relation to the total domain Ω, while Fig. 2.15b illustrates the two load cases for which the complete system is evaluated. In Fig. 2.15c we show two distinct optimized solution for which the buckling load along x and y are simultaneously maximized. These results are obtaining by annealing the temperature between $T_{\text{max}} = 0.01$ and $T_{\text{min}} = 10^{-8}$ in $N_{\text{iter}} = 6000$, with a $T_{\text{Ising}} = 0.8$.

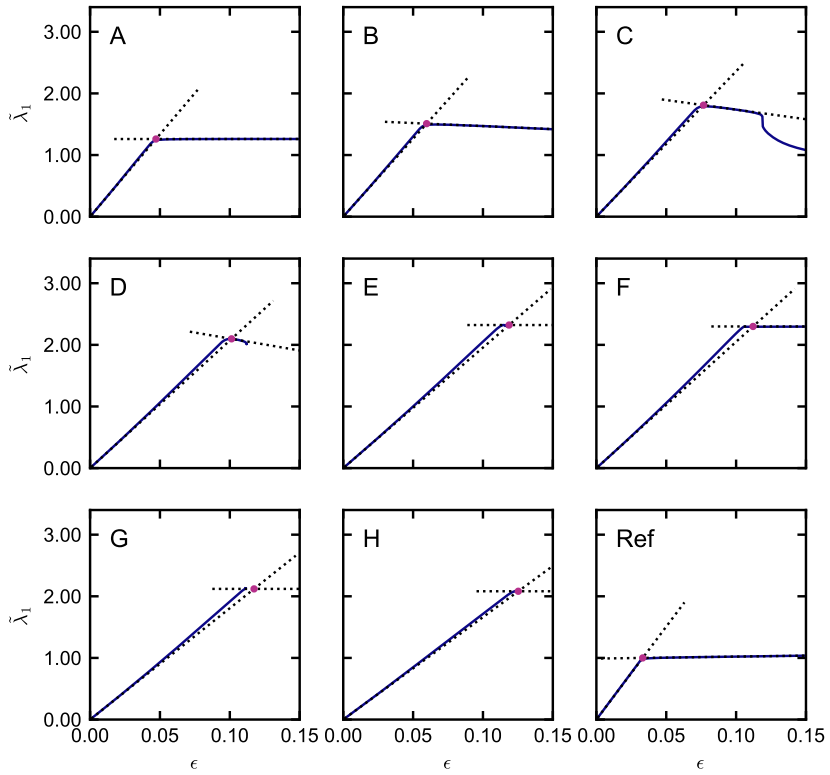

FIGURE 2.14: Normalized force-strain curves for the structures A-H shown in Fig. 2.7a, obtained by performing nonlinear FEA (ANSYS). Dashed lines represent the regression lines before and after buckling, used to find the buckling load shown with the pink marker. The results of all beams are normalized with respect to the buckling force of the reference (Ref) geometry.

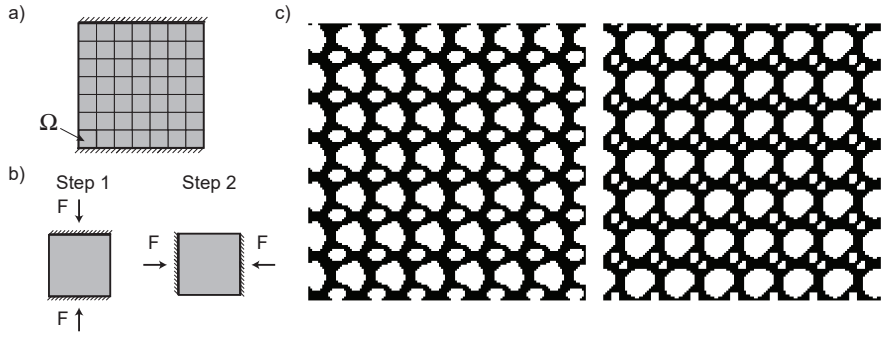

FIGURE 2.15: Multi-objective topology optimization for a lattice with 2D periodicity. a) Schematic of material made of 7×7 unit cells. b) Schematic of the two loading conditions that are applied for multi-objective optimization, in which the objective is given by Eq. (2.27), and $\Phi_x(\rho_i)$ and $\Phi_x(\rho_i)$ are taken from Eq. (2.10a). c) Two examples of optimized structures with maximized buckling force in both the x and y direction simultaneously.

CHAPTER 3

Continuous learning of emergent behaviour in robotic matter

In review as:

Oliveri, G., van Laake, Lucas C, Carissimo, C., Miette, C., Overvelde, J.T.B. 2020. *Continuous Learning of Emergent behaviour in Robotic Matter*

One of the main challenges in robotics is the development of systems that can adapt to their environment and achieve autonomous behaviour. Current approaches typically aim to achieve this by increasing the complexity of the centralised controller by, e.g., direct modeling of their behaviour, or implementing machine learning. In contrast, we simplify the controller using a decentralised and modular approach, with the aim of finding specific requirements needed for a robust and scalable learning strategy in robots. To achieve this, we conducted experiments and simulations on a specific robotic platform assembled from identical autonomous units that continuously sense their environment and react to it. By letting each unit adapt its behaviour independently using a basic Monte-Carlo scheme, the assembled system is able to learn and maintain optimal behaviour in a dynamic circumstances as long as its memory is representative of the current environment, even when incurring damage. We show that the physical connection between the units is enough to achieve learning, and no additional communication or centralised information is required. As a result, such a distributed learning approach can be easily scaled to larger assemblies, blurring the boundaries between materials and robots, paving the way for a new class of modular 'robotic matter' that can autonomously learn to thrive in dynamic or unfamiliar situations, for example encountered by soft robots or self-assembled (micro-)robots in various environments spanning from the medical realm to space explorations.

3.1 Introduction

Traditional robots typically exhibit well-defined motions, and are often controlled by a centralised controller that uses sensor data to provide feedback to its actuators [109, 110]. While these systems are capable of operating in well-known and controlled environments combining high-speed with precision, adapting their behaviour to various highly dynamic and unforeseen circumstances remains a challenge in the design of autonomous robots [111]. Towards this goal, reinforced learning strategies can be used to deal with more complex tasks that cannot directly be programmed in the behaviour of the robot. By sensing their environment, robots can build models of themselves [112], or optimize man-made models with machine learning strategies to improve their behaviour [109, 110, 112–116]. Apart from a clear division between training and task-execution, these systems rely on a centralised architecture, that - with an increasing number of active components - demands advanced models and higher computational power.

A recent alternative idea to simplify the computational task, while allowing for adaptability to the environment, is to make soft robots. These are robots fabricated with materials with stiffness resembling human tissue [111, 117–120], that exhibit so-called 'embodied intelligence' [121–123], and adjust their shape when subjected to forces resulting from interactions with their environment. While this physical adaptability enhances the robustness of the robot's behaviour, allowing it to e.g. operate in more complex environments, it is not guaranteed that they operate optimally, or even operate at all. For example, a soft robot that is able to walk on various surfaces still requires a different actuation sequence to crawl through a narrow opening [124]. At the moment, and because of the unpredictable nature of the robot's behaviour in varying environments, this sequence is externally (and manually) adjusted. A natural question to ask is if such higher levels of decision-making and learning can also be embodied in robots, similar to e.g. the decentralised neural networks of squids [125].

In fact, decentralised approaches are already applied in swarm robotics, where an increasing number of active components can cooperatively achieve complex behaviour when sensing or communication capabilities are incorporated into the individual units [126–129]. In such a robotic system, behaviour emerges from local interactions, rather than being centrally controlled. Inspired by social animals such as ants, bees or birds, researchers have been able to achieve emerging properties out of simple unit-to-unit interactions, such as adaptation to mechanical stimuli [130], self-assembly [129, 131], construction [132], and locomotion [127, 133]. Interestingly, such behaviour does not have to be programmed, and could also be learned using the same decentralised architecture [134, 135].

In this work, in order to explore similar decentralised strategies, we develop and perform experiments on modular robots that are connected to form a single body. Using this platform, our aim is to simplify the learning strategy, while

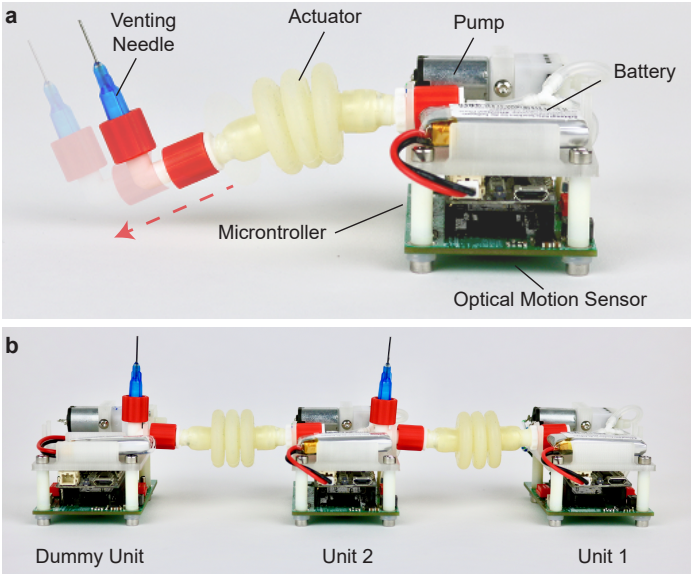

FIGURE 3.1: Robotic unit design and assembled robot consisting of two active units. a, Design of a single robotic unit in its relaxed and extended state. b, Robot assembled from two active units and one dummy unit.

focusing on the fundamental challenge to ensure robust and optimal behaviour[1] under varying external conditions and damage. We do so by having each element continuously learn and adapt it own behaviour to changing circumstances. To that end, we develop a fully decentralised and universal reinforced learning strategy that can be easily implemented, is scalable, and allows the robotic system to autonomously achieve and maintain specific emergent behaviour.

3.2 Individual unit behaviour

Such an approach holds potential for a wide variety of robotic systems, and we illustrate this here with a robot made of several identical units connected by soft actuators that periodically extend and contract along one direction on a circular track, ensuring that 1D motion. Each robotic unit operates independently and

[1]Under the assumption that information on the working condition is partially available, optimal behaviour is here intended in a heuristic term, where "good enough" performance is optimal. Such that the task is executed, even if not with best performance. Considering that circumstances are dynamic, and as a consequence the optimal behaviours change over time, a robust behaviour refers to the ability of the learning strategy to find and assume the optimal behaviour relative to the current circumstances.

consists of a micro-controller, optical motion sensor, pump, pneumatic actuator and battery (Fig. 3.1a and Sec. 3.8). The actuator is activated by cyclically turning the pump on and off at a ratio of $\alpha = t_{on}/t_{\text{cycle}} = 0.4$ with a total cycle duration of $t_{\text{cycle}} = 2\ s$. By continuously venting the air in the actuator through a needle, the actuator extends and contracts (Fig. 3.2a). Note that the units are not capable of locomotion by themselves and require to be physically attached to other units to move. For this reason, forward locomotion is what we refer to as the emergent behaviour that our assembled robot exhibits.

Importantly, each unit aims to move as quickly as possible in the same predefined direction. To achieve this, it can adapt its behaviour by varying the phase of actuation $\phi_i t_{\text{cycle}}$. As a first trial for a potential learning algorithm that is capable of learning emergent behaviour, we implement a Monte-Carlo scheme [136] in each unit. At the beginning of every learning step, the unit perturbs its phase according to $\phi_i' = \phi_i + \epsilon\Delta s$, in which ϵ is a random number drawn from a uniform distribution on the interval $[-1, 1]$, and $\Delta s = 0.1$ is the stepsize indicating the maximum change in phase allowed per learning step. Then, after performing $n_{\text{act}} = 2$ cycles the unit determines its average velocity U_i' via the motion sensor, and compares it with the velocity U_i stored in memory. The unit will update the phases and velocities in its memory with probability $e^{(U_i'-U_i)/T}$, in which T is a 'temperature' that can be used to tune the acceptance probability when $U_i' - U_i < 0$. We refer to this algorithm as the 'Thermal algorithm' (see pseudocode in Sec. 3.8).

3.3 Learning of emergent behaviour

Two active units

In order to test if the system can optimize its movement in a predefined direction, we first focus on the most basic assembly of two active units (Fig. 3.1b). Note that we also need one dummy unit without an actuator at the end of the assembly to make sure all the other actuators are connected on both sides. We place the units on a circular track and initialize both the active units with a random actuation phase ϕ_i (Fig. 3.2a), and let the system run for $n_{\text{learn}} = 100$ learning iterations. At the beginning of each learning step the units update their phases independently, without exchanging any information. The pushing and pulling forces applied by the pneumatic actuators are the only interactions between the units (Movie 1). The results in Fig. 3.2b-c show that the assembled robot learns to move forward, and reaches its fastest speed after just 80 s (i.e. 20 learning iterations). This grants each unit a forward motion of ≈ 4.5 mm/cycle (see SI for specifics on the velocity calculation). For the remaining 320 s the units try to continuously increase their speed, resulting in random perturbations around the optimal behaviour. We then repeat this learning experiment 56 times with the same settings, but with different random initial phases.

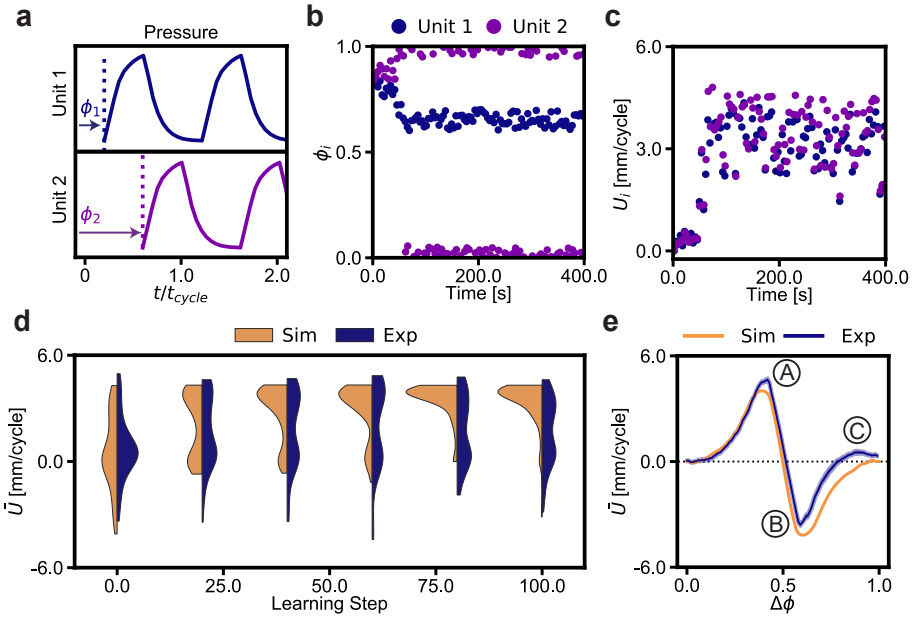

FIGURE 3.2: Initial learning experiments for an assembled robot consisting of two active units. a, Each active unit can vary its phase $\phi_i t_{\text{cycle}}$, which affects the actuation timing as shown by the cyclic pressure in the soft actuators. b-c, Evolution of phases ϕ_i and measured velocity U_i in the active units as a function of time for a single learning experiment. d, Distribution of average robot velocity $\bar{U}$ as a function of the learning step for 112 learning simulations and 56 learning experiments. e, Simulated and experimental velocity as function of phase difference $\Delta\phi = \phi_2 - \phi_1$, exemplified by three experiments in Fig.3.3.

To complement our experimental results with simulations, we additionally model the robot behaviour with a one dimensional mass-spring system, where the masses experience static friction with the environment. Each active robotic unit is modeled with a mass with an adjacent spring, see Fig. 3.9a. To model the cyclic actuation that units perform, each spring has a time-dependent preferred spring length which depends on the phase ϕ_i, resulting in a pushing force on two consecutive masses, (see Sec. 3.8 for further details). By using this model, we then perform 112 learning simulations and compare the results with experiments. The distribution of the average robot velocities $\bar{U} = \sum_i U_i$ during learning is shown in Fig. 3.2d. We find that, while the average speed after learning is distributed around $\bar{U} \approx 4.5$ mm/cycle for both simulations and experiments, in the latter case we also find an additional optimum for $\bar{U} \approx 1$ mm/cycle. The distribution stabilizes at these two values after ≈ 20 learning steps.

To determine the underlying cause for this bi-modal velocity profile shown in experiments, we map the behaviour that this assembled robot can exhibit. Although each unit updates its phase individually, the overall motion of the assembled robot depends only on the phase difference $\Delta\phi = \phi_2 - \phi_1$ between the two units. Fig. 3.2e shows a scan of the potential velocities $\bar{U}$ the robot can exhibit, obtained in experiments by setting $\phi_1 = 0$ and varying ϕ_2 between 0 and 1. From these results we find that the highest speed is achieved at a phase difference of $\Delta\phi \approx 0.4$, at which the robot reaches a velocity of ≈ 4.5 mm/cycle. This is same phase difference we observed in our first learning experiment (Fig. 3.2d). At this optimal locomotion pattern, the system seems to undergo peristaltic motion, where the first unit's actuator is expanding for $t \leq \alpha 2\ s$ pushing the first unit to right (see e.g. Fig. 3.1b). Then, once the first unit's actuation has ceased, the second unit actuator becomes active pushing on both the second and dummy unit. At the same time, the first unit's actuator is relaxing, favoring the positive displacement of the second unit to the right. Finally, when also the actuators of the second unit stops, its spring back force has a larger effect on the dummy unit since the actuator is also contracting. To further understand the relation between optimal phase combination and α, we performed a parametric study and found an approximately linear relation $\Delta\phi \approx \alpha$ (Fig. 3.16).

Similarly, for $\Delta\phi = 0.6$ the robot moves in the opposite direction with a reversed actuation pattern with a velocity of ≈ -3 mm/cycle, while at $\Delta\phi \approx 0.9$ we find a local maximum in experiments with lower speed of ≈ 1 mm/cycle. The total displacement caused by these three locomotion patterns are demonstrated in Fig. 3.3, where we show both the starting and the end position after 60 s. It is important to note that while we expected symmetric behaviour between forward and backward motion, as also demonstrated by the mass-spring model (Fig. 3.2e), in experiments we observe asymmetric motion. We believe that this asymmetry arises from a reduction of the friction due to the vibration of the pumps, which only occurs in the active units, see SI for further details. The placement of the dummy unit therefore influences the behaviour the robot can achieve. Also note that when applying the learning strategy in the simulations (Fig. 3.2d-e), we find that the system only stabilizes around the maximum at $\Delta\phi \approx 0.4$, since no significant local optima are observed.

Three active units

While the results for a robot assembled from two active units seem promising, the system's average score function $\bar{U}$ is relatively simple (Fig. 3.2e). Therefore, we next increase the complexity by performing experiments on a robot assembled from three active units and one dummy unit (Movie 1). In Fig. 3.4a we show the distribution of the average velocities $\bar{U}$ during learning, of 112 simulations and 56 experiments. We observe both in experiments and simulations an initial improve-

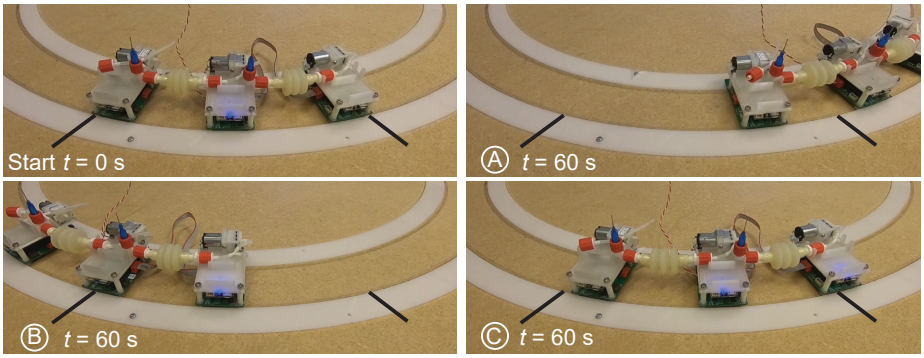

FIGURE 3.3: Start and end positions for three experiments with fixed phase combinations, $\Delta\phi = 0.4, 0.6$ and $\Delta\phi = 0.9$ respectively, as indicated in Fig. 3.2e

ment of $\bar{U}$ during the first few ≈ 50 learning iterations, similar to the assembled robot containing two active units. Interestingly, while the simulations stabilize at a $\bar{U}$ with a mean positive value and an acceptance rate of ≈ 0.25 (Fig. 3.4b), for experiments we observe a decrease of the mean $\bar{U}$ after ≈ 200 learning steps. This is accompanied by a decrease in the average acceptance rate of new phases, which approaches zero soon after the learning starts. To understand the qualitative difference between simulations and experiments, and specifically the decrease in $\bar{U}$ in experiments, we measured $\bar{U}$ for all potential phase combinations in both simulations (Fig. 3.4c) and experiments (Fig. 3.4d). As expected, the results show a diverse range of possibilities, with multiple maxima. While our relatively simple model seems to capture the behaviour of the assembled robot, there are two main differences in experiments. First, in experiments we observe significant noise, which could, e.g., be the result of how the measurements are taken (see SI). Second, in experiments the performance of the robot seems to depend on where it is located on the circular track, which could be the result of the variations in track width and the resulting friction (SI and Fig. 3.17). This is supported when considering the standard deviation of the observed $\bar{U}$ measured over 20 experiments, which exhibits non-negligible variations (Fig. 3.4e and Fig. 3.13a). For example, when running an experiment with $\phi_2 - \phi_1 = 0.4$ and $\phi_3 - \phi_1 = 0.2$ (bottom circle in Fig. 3.4d-e), we find that for some parts of the track this assembled robot is capable of achieving $\bar{U} \approx 2$ mm/cycle, while the robot comes to a complete stop when reaching different parts of the tracks (Fig. 3.4f and Movie 2). In contrast, other regions in the phase space appear to be robust against track variations, see e.g. the phase combination $\phi_2 - \phi_1 \approx 0.4$ and $\phi_3 - \phi_1 \approx 0.8$ (top circle in Fig. 3.4d-e), for which the robot displacement $\bar{U} \approx 4.5$ mm/cycle is at a maximum (Fig. 3.4f and Movie 2).

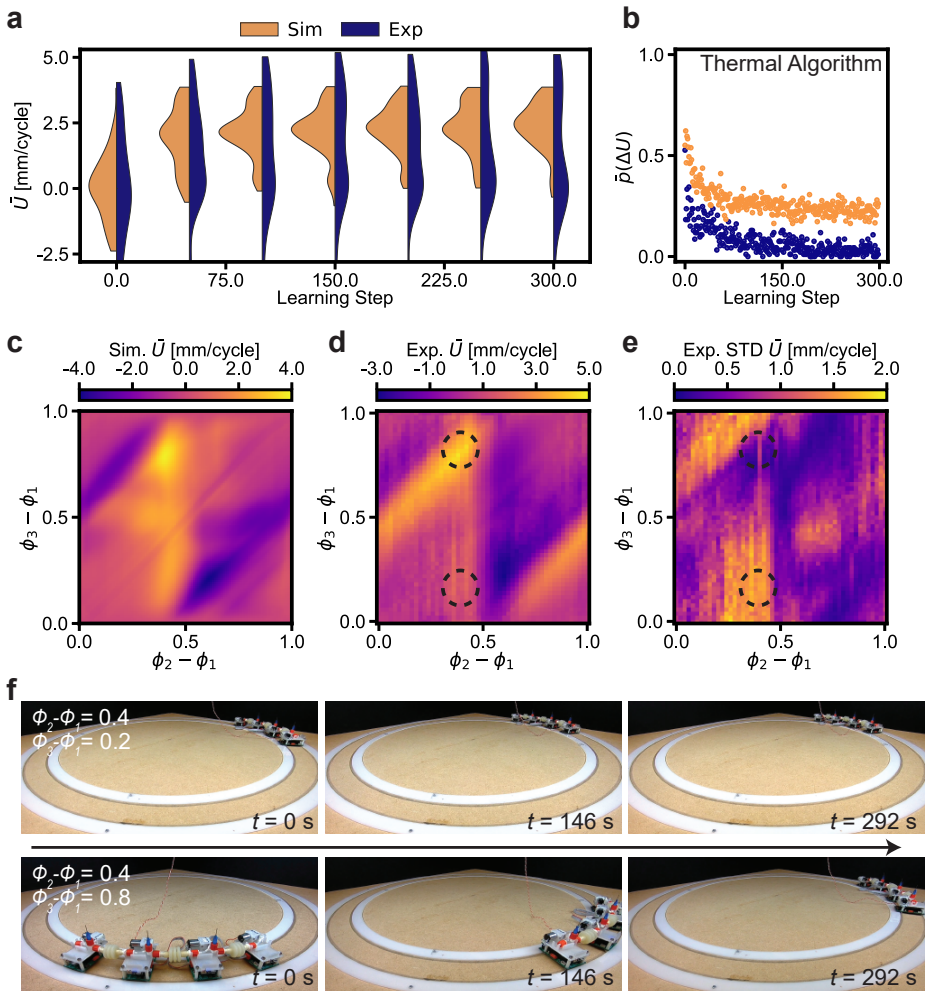

FIGURE 3.4: Adaptability to variations in the environment of an assembled robot consisting of three active units. a-b, Distribution of average velocities $\bar{U}$ and average acceptance rate $\bar{p}$ for the Thermal algorithm, as a function of the learning step, for 112 simulations and 56 experiments. c-d, Average velocity $\bar{U}$ obtained in simulations and experiments as a function of all the possible combinations of phases $\phi_3 - \phi_1$ and $\phi_2 - \phi_1$ (with $\phi_1 = 0$). e, Standard deviation of the average velocity $\bar{U}$ observed over 20 experimental runs. f, Two different experiments with fixed phases, to highlight the effect of the track on the robot's behaviour.

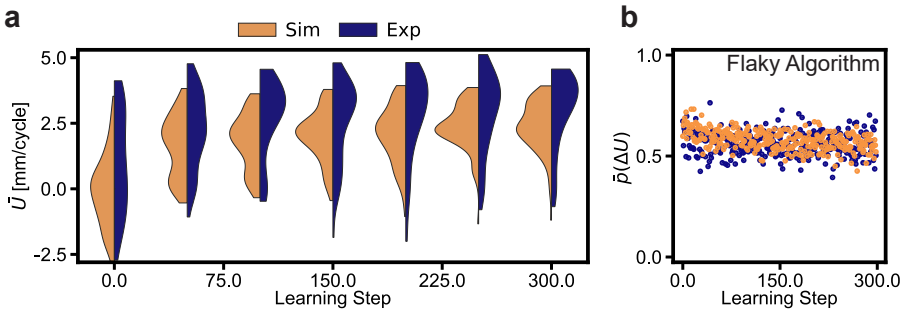

FIGURE 3.5: a-b, Distribution of average velocities $\bar{U}$ and average acceptance rate $\bar{p}$ for the Flaky algorithm as a function of the learning step, for 112 simulations and 56 experiments.

3.4 A more robust learning strategy

To understand the effect that noise and changes in environment can have on our proposed decentralised learning algorithm and the reason that the algorithm stops accepting new phases, we need to consider how the memory is updated. At the moment, each unit only updates its U_i that is stored in memory when achieving a higher U_i', with the exception of a probability to accept lower U_i' depending on the specified temperature T. If in the previous step the noise caused an overestimation of the U_i that was stored in memory, the probability to accept realistic U_i' becomes smaller. Similarly, if changes in the environment cause a decrease in U_i', it will become less likely that the value of U_i that is stored in memory is updated. This will influence the overall acceptance probability (Fig. 3.4b), and reduce the ability to adapt to variations and noise as also demonstrated by including noise in the simulations (Fig. 3.14a-b).

Therefore, while the Thermal algorithm performs well in a noiseless or static environment, in order to improve the adaptability[2] of our system to real dynamic experimental conditions where sensor data might not be always reliable or environment has changed, we need to modify how memory is stored in each active unit. So far, we stored the last accepted phase ϕ_i and corresponding displacement U_i in memory. Instead, we propose an improved algorithm in which we update the memory according to the newly obtained velocity U_i', while still only updating the phase to ϕ_i' in case of acceptance. We refer to the this new strategy as the 'Flaky algorithm' (see pseudocode in Sec. 3.8). As such, with this new formulation each unit is constantly aware of the landscape and does not rely on past information that may not be reliable anymore (Movie 1).

[2]We refer to adaptability as the capacity of our learning algorithm to adapt to different circumstances, such that it is still able to improve the current performance on the task, and explore the range of optimal behaviours given the ongoing circumstances.

To determine the performance of this updated algorithm, we repeated the learning study and gathered data from 112 simulations and 56 experiments. The results in Fig. 3.5a show that with the Flaky algorithm the active units are now capable of adapting to variations in their environment, achieving velocities of $\bar{U} \approx 4$ mm/cycle. Interestingly, the units now accept $\approx 60\%$ of all new candidates, independent of the learning step (Fig. 3.5b). As a result, the robot is able to adapt its behaviour to noise and variations in the environment, which significantly improves its overall robustness. The difference in adaptivity between the Thermal and Flaky algorithm can also be illustrated by simulations in which we implement a sudden reduction in friction during learning (Fig. 3.18 and Fig. 3.19). Moreover, simulations indicate that the acceptance rate of the Flaky algorithm is not affected by the noise level, in contrast to the Thermal algorithm (Fig. 3.14c).

While the Flaky algorithm seems more suitable for dynamic environments than the Thermal algorithm, we can still observe certain conditions for which the Flaky algorithm is starting to have problems. For example, when increasing the friction in this system as shown in Fig. 3.17e-f we observe potential behaviour that changes more abruptly with a change in actuation phase. As such, the derivative of the score with respect to the phases is close to zero everywhere, except for a few boundaries between domains. In this case the memory in the Flaky algorithm can be reset almost instantaneously, after which it will accept any other phase combination. As such, the Flaky algorithm requires some notion of smoothness, with variations occurring over domains that are larger then the stepsize.

3.5 Adaptivity to damage

With the adoption of the Flaky algorithm, we next explore if and how the assembled robot adapts to suddenly applied damage. In order to show this, we consider a learning experiment with three active units and one dummy unit, in which we intentionally damage the robot after ~ 300 learning steps by removing the venting needle in the second unit (Fig. 3.6a-b and Movie 3). This dramatically alters the possible behaviours the robot can exhibit, as indicated by the contour plots of the score before and after damage (Fig. 3.6c-d and Fig. 3.13). Once damaged, the robot must find a new locomotion pattern. It should be noted that although the second actuator cannot inflate after damage, we still observe the presence of a global optimum at $\phi_2 - \phi_1 \approx \phi_3 - \phi_1$, for which unit two and three are active simultaneously. While one would expect no influence of the damaged unit's phase on the overall score (Fig. 3.20, Fig. 3.21), we find that by turning the pump on and off, unit two seems to control its friction properties and as a result still plays a role in the optimal behaviour.

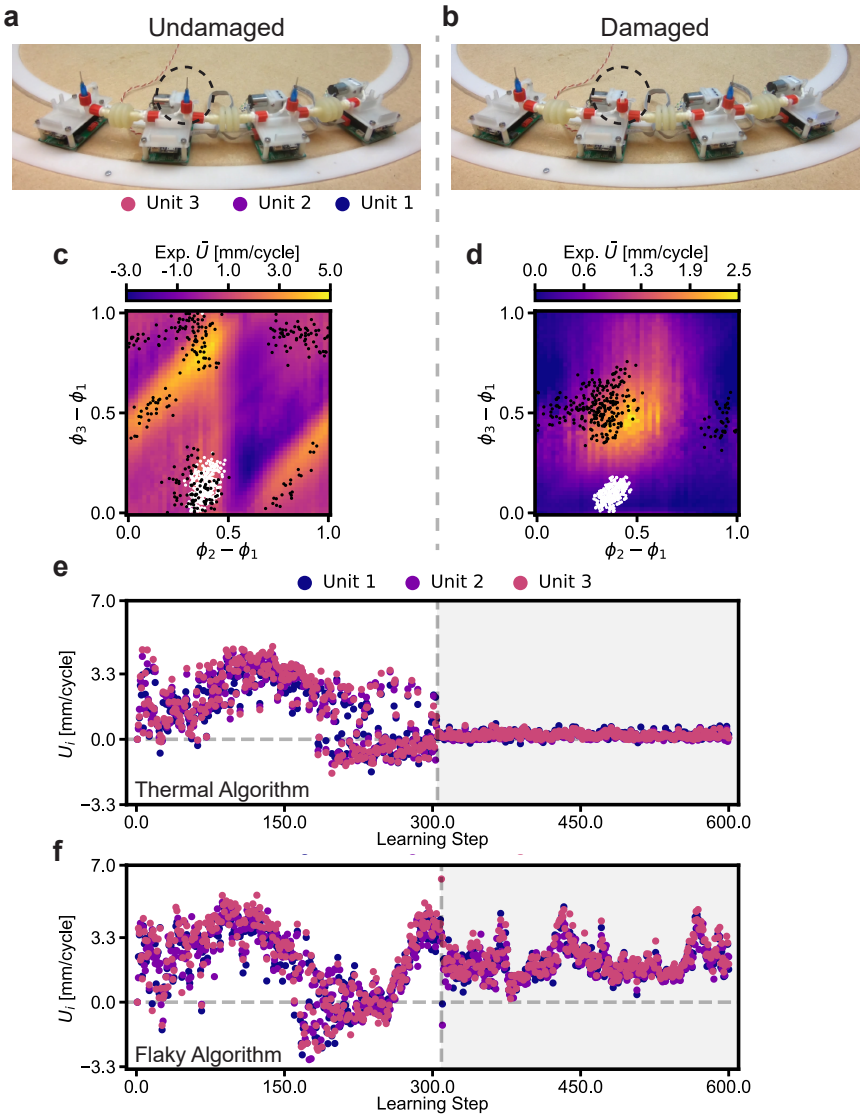

FIGURE 3.6: Adaptability of the assembled robot to damage using the Thermal and Flaky algorithm. a-b, An assembled robot consisting of three active units is damaged by removing one of the needles. c-d, Average velocity $\bar{U}$ obtained in experiments for an intact and damaged robot, as a function of the possible range of combinations of phases $\phi_3 - \phi_1$ and $\phi_2 - \phi_1$ (with $\phi_1 = 0$). The white and black dots represent the tried phases in two learning experiments in which the robot is damaged after 300 learning steps, using the (e) Thermal and (f) Flaky algorithm, respectively.

Importantly, when directly comparing the experiment and the simulations results of the Thermal and the Flaky algorithm as shown in Movie 3, Fig. 3.6e-f, Fig. 3.20 and Fig. 3.21, we find that the Thermal algorithm is not able to adapt to the damage and gets stuck at a certain phase, while the Flaky algorithm is able to overcome damage by completely adapting its behaviour.

3.6 Scalability of the learning behaviour

Finally, to verify if our learning strategy is scalable and can be applied to assembled robots consisting of more than three active units, we performed 112 simulations and 56 experiments on a robot consisting of seven active units and one dummy unit (Fig. 3.7a and Movie 4). In Fig. 3.7b we show the distribution of average velocity $\bar{U}$ as a function of the learning step. While the average displacement $\bar{U} \approx 2.5$ mm/cycle is lower than before, the initial learning occurs within a comparable number of learning steps. Note that during learning the assembled robot occasionally does reach higher speeds up to $\bar{U} \approx 4$ mm/cycle, however it is not able to maintain these speeds, likely due to the random guesses made by each unit during normal operation. To better understand how the system's behaviour is affected by its size, we performed 112 simulations for sizes up to 20 active units. To determine the equilibrium velocity $\bar{U}_{\text{eq}}$ as shown in Fig. 3.7c, we fitted an exponential function to the average learning behaviour (Fig. 3.15a). We find that the equilibrium velocity decreases for larger number of units, but appears to asymptotically reach a value close to $\bar{U} \approx 1.7$ mm/cycle.

Here, it should be noted that scalability could refer to two things, which is the behaviour that the robotic platform can exhibit (in the sense that it can physically move with more units connected together), and the ability to learn "good enough" or optimal behaviour. In the results shown in Fig. 3.7c we are investigating both factors for scalability simultaneously. One could argue that the velocity associated with the peristaltic locomotion should not depend on the number of units, such that the system is apparently not able to find the optimal behaviour for larger system size. This might be explained by the fact that our learning algorithm is exploring the behaviour in a probabilistic way, such that equilibrium relates to some extent to the standard deviation of the system's potential behaviour (Fig. 3.22). However further studies should be performed to support this statement.

Interestingly, we do find that an increase in the number of units does not affect the rate of learning, as indicated by Fig. 3.7d, and that on average the units only need 35 learning steps to reach speeds within 80% of the equilibrium velocity $\bar{U}_{\text{eq}}$. Furthermore, we can increase the equilibrium velocity $\bar{U}_{\text{eq}}$ at the cost of learning rate by decreasing the steps size used to update the phases (Fig. 3.7c-d), which underpins the assumption that better behaviour is attainable but depends on the stochastic nature of the algorithm.

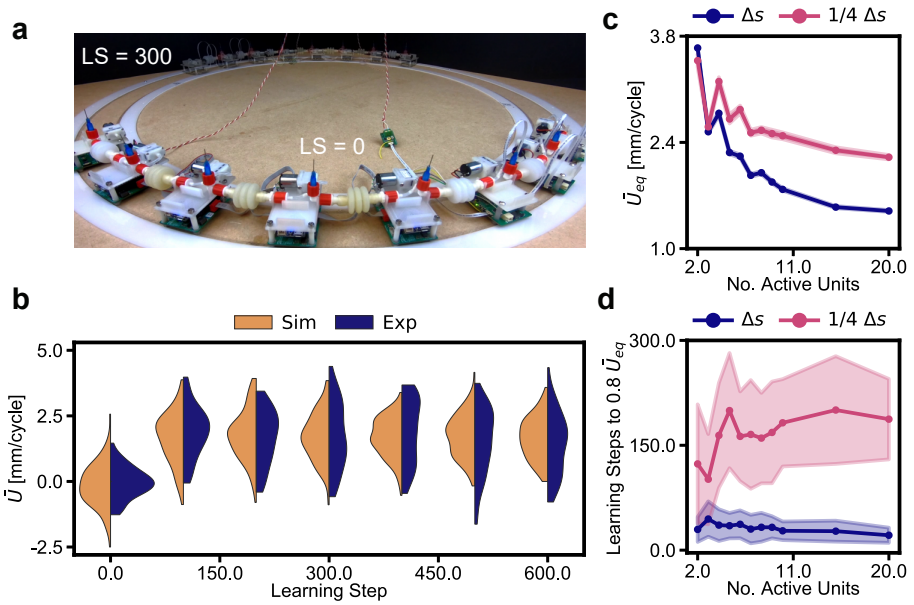

FIGURE 3.7: Scalability of the learning behaviour for an assembled robot ranging from 2 to 20 active units. a, Initial and final position after 300 learning steps (1200 s) of an assembled robot consisting of seven active units. b, Distribution of average velocities $\bar{U}$ as a function of the learning step, for 112 simulations and 56 experiments. c, Relation between the number of active units and the equilibrium velocity $\bar{U}_{\text{eq}}$, obtained using simulations, for $\Delta s = 0.1$ and $\Delta s = 0.1/4$. d, Average number of learning steps, and standard deviation, needed to reach equilibrium (i.e. $0.8\ \bar{U}_{\text{eq}}$) as a function of the number of active units. Here we only consider the simulation which reach the $0.8\ \bar{U}_{\text{eq}}$ threshold within the assigned number of learning steps. While for $\Delta s = 0.1$ all the simulations reach the threshold, for $\Delta s = 0.1/4$ we find that a percentage of simulations do not reach $0.8\ \bar{U}_{\text{eq}}$ (Fig. 3.15b).

3.7 Conclusions

In this work, we implemented a basic decentralised reinforced learning algorithm in a modular robot to improve and adapt its emergent global behaviour. We demonstrated that such a decentralised learning approach is an effective way to adapt the robot's behaviour to a changing environment or damage. One of the most important findings of this work is the requirement on how previous scores (measured by the sensors) should be stored in each unit, where we find that the memory should be representative of the current environment in order for the algorithm to be adaptive. We implemented this in the Flaky algorithm by always keeping the last measured displacement, in contrast to the more classic Thermal

algorithm that keeps the best of all previously measured displacements, and therefore can only overcome sudden changes in conditions according to a specific temperature parameter. Interestingly, our proposed system was able to learn peristaltic locomotion gaits [127, 137, 138] without any prior knowledge, making this decentralised learning strategy suitable for the exploration of environments whose governing physical conditions are unknown.
Given that each unit in the robot is identical, no electrical connections are needed, and there is no centralised controller, the robot can be reconfigured into any shape and maintain its capability to learn, for example after shuffling the units (Movie 5). As a result, each module represents a unit cell capable of learning, such that the learning capability becomes essentially a material property, e.g., when cutting the robot in two both parts would maintain the ability to learn. We therefore refer to this kind of system as 'robotic matter' [47, 49, 139]. Although we performed 1D experiments to simplify the potential behaviours, the robot does not need to be confined to a track (Movie 5). Importantly, we believe that our approach is applicable to a large range of robotic systems with different architectures, actuators, and sensors, including (soft) robots with many degrees of freedom that are assembled in 2D patterns [127], or that fully deform in 3D [124]. Furthermore, the simplicity of our learning strategy and the fact that it can be coded using only a few lines makes it suitable for (micro-sized) robots that have limited computational power and dedicated electronic circuits [140–142], and could potentially even be realized through other means than an electronic circuit such as microfluidics [117, 143], and even material behaviour [144]. The simplicity and robustness of our proposed robotic matter is an important step towards robotic systems that can autonomously learn to thrive in a broad range of applications at different length-scales, spanning from *in vivo* healthcare, to disaster relieve and space exploration.

3.8 Supplementary information

Fabrication of units

Each unit consists of a micro-controller (ESP32), a motion sensor (ADNS-9800) and a pressure sensor (Honeywell MPRLS0015 PG0000SA), mounted on a custom made PCB embedded in a 3D printed frame with a total dimension of $68 \times 44 \times 42$ mm. The frame is designed to support a micro-pump (Huizhou Yingyi Motor Co. YYP032), a 3.7V battery (Carson 500608131), and has two mechanical snap connectors to attach the soft actuators.

The soft actuators consist of three bellows of 25 mm outer diameter, a length of 35 mm and a total inner volume of ≈ 2 ml, which were moulded using a soft elastomer (Smooth-ON Dragon Skin 30). The mould consists of two external components printed on a polyjet printer (Stratasys Eden260VS), and a soluble PVA core printed on a FDM printer (Ultimaker 3). After casting, the soft actuators were

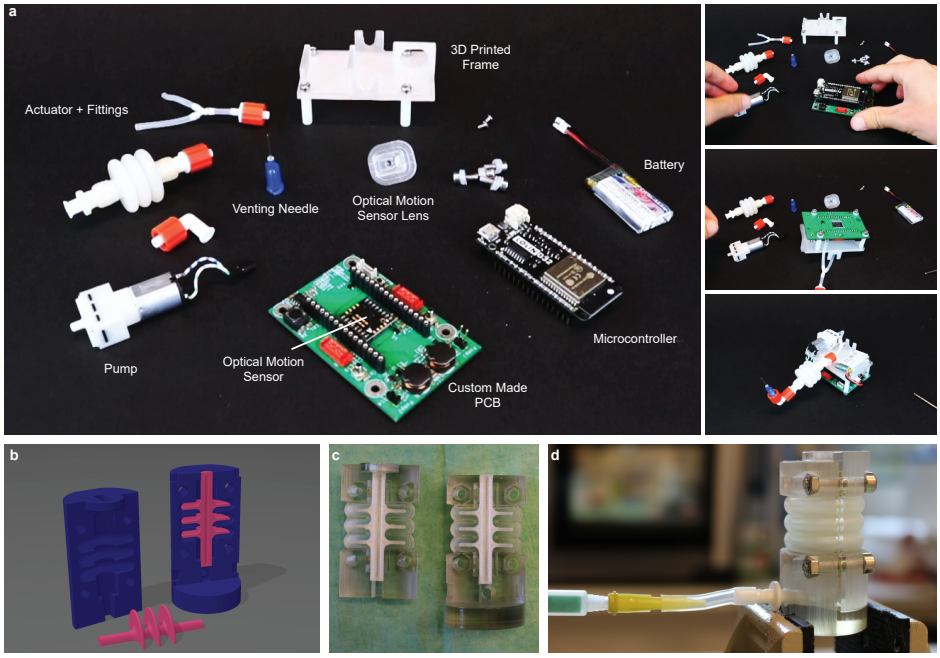

FIGURE 3.8: Unit assembly and bellow actuator fabrication. a, Overview of unit components and assembly process. b, CAD models of bellow actuator moulds. The outer mould (in blue) was printed in VeroClear (Stratasys Eden260VS), and the inner core (in pink) was printed in soluble PVA (Ultimaker 3). c, 3D printed parts before casting. d, Injection casting of bellow actuator with Dragon Skin 30 (Smooth-ON).

connected to a water pump (Eheim 1250-790) to flush out the soluble core over a period of approximately two days. The actuators were connected on both sides to luer lock couplings (Nordson Medical FTLL035-1 and MLRL035-1) with heat-shrink tubes (RS-458-068). One side of the actuator is then connected to a venting needle (Metcal 9922050-TE gauge 22, inner diameter of 0.412 mm and length of 11 mm), via a male-female luer elbow coupling (Nordson Medical LE87-1), and two luer lock caps (Nordson Medical FSLLR-3). The other side of the actuator is connected to the pump and the pressure sensor via a silicone tube (outer and inner diameter 3 mm and 1.5 mm, respectively), fitted with a barb to luer lock connector (Nordson Medical MLRL007-1).

The mass m of each unit is equal to 63.7 g, including 10.2 g for the complete actuator and 10.8 g for the battery. After connecting two units using a soft actuator, the distance at rest between the two adjacent units equals 62 mm. Importantly, the assembled units touch the surface using four 5.5×8 mm screws. To maximize the accuracy of the motion sensor, the distance of the PCB from the surface was set to be ≈ 5 mm, achieved with a screw head of 3 mm thickness, and four nylon rings of 0.54 mm thickness per screw. The width of the units at the place they touch the ground (i.e. width between the screws), which fits into the track, has dimension of 55.4 mm. An overview of the components and assembly is given in Fig. 3.8.

Experimental setup

The learning experiments are launched by initiating each unit at a random phase ϕ_i. The initialization is done via a WiFi connection, and it is made with each unit sequentially. While the WiFi connection is not needed to perform the experiments, the connection is used to gather sensor data during the experiments. After initialization, the units try out new phases after every $n_{\text{act}} = 2$ cycles of $t_{\text{cycle}} = 2$ s duration. During these two cycles, the phase ϕ_i is kept constant. After performing two cycles, each unit determines its average velocity U' that is used in the learning algorithm. Each unit takes displacement measurements at a 10 Hz rate as shown in Fig. 3.23, and the unit displacement for the learning step is determined by comparing the absolute displacement at the beginning and at the end of the second actuation cycle (the measurements that are located at the dashed lines). The velocity can then be determined by dividing the displacement by the actuation cycle duration of $t_{\text{cycle}} = 2$ s.

The change in phase between two consecutive learning steps, is implemented in the first actuation cycle. The change to the phase is made by scaling the total duration of the cycle according to $1-(\phi_i'-\phi_i)/t_{\text{cycle}}$, which results in an elongating or shortening of the duration of the first cycle. Importantly, for both cycles the pump is only turned on in the first $\alpha = 0.4$ portion of each cycle. Note that the implementation of the phase change, which results in an elongating or shortening

of the duration of the first cycle, introduces some differences in measured velocity between units of the same robots at the same learning step, because the evaluation periods start and finish at different times.

By comparing the measured velocity U' with the velocity U stored in memory, each unit determines the acceptance probability of the current phase ϕ' according to $p(\Delta U) = e^{(U'-U)/T}$, in which $T = 0.1$. Using the phase that is stored in memory ϕ, each unit then perturbs its phase according to $\phi' = \phi + \epsilon\Delta s$ to obtain a new candidate phase, in which ϵ is a random number drawn from a uniform distribution on the interval$[-1, 1]$, and $\Delta s = 0.1$ is the used stepsize. Note that while the units are initialized at the same time, the internal clocks of the units are not further synchronized and will go out of sync over time. While this is not corrected during learning (which should also not be necessary as the units should be able to adapt to this), for plotting purposed we take the learning step as reference. While the units can operate on battery, to allow for continuous testing most of the experiments were run by powering the units with a common power supply, attached to one of the active units. Power is then delivered to the other units via a unit-to-unit connection placed below the soft actuator. Moreover, the experiments were performed on a lasercut circular POM track screwed on a MDF surface. The diameter of the track is $D \approx 0.854$ m, and the track width is $w = 56.4 \pm 0.4$ mm.

The learning algorithms

Each unit runs an identical algorithm. The pseudecode for the two algorithms (Thermal and Flaky) tested in this work are given are given below.

Algorithm 1: Thermal algorithm in pseudocode

```
Initialize random phase φ
Evaluate velocity U
for Step in n_learn do
    Perturb phase stored in memory φ' = φ + ε·Δs
    Evaluate new velocity U'
    Determine acceptance probability p(ΔU) = e^((U'−U)/T)
    if p(ΔU) ≥ rand(0,1) then
        Store new phase in memory φ = φ'
        Store new velocity in memory U = U'
    end
end
```

Algorithm 2: Flaky algorithm in pseudocode

```
Initialize random phase φ
Evaluate velocity U
for Step in n_learn do
    Perturb phase stored in memory φ' = φ + ϵΔs
    Evaluate new velocity U'
    Determine acceptance probability p(ΔU) = e^{(U'−U)/T}
    if p(ΔU) ≥ rand(0, 1) then
        Store new phase in memory φ = φ'
    end
    Store new velocity in memory U = U'
end
```

Numerical model

We model the robot's behaviour with a one dimensional mass-spring system, where masses exhibit friction with the environment (Fig. 3.9a). Each i-th active unit is modeled as a mass m_i, with an adjacent spring with stiffness k_i and initial length L_i. To model the actuation forces applied by the pneumatic actuator, this spring has a time-dependent rest length $l_i(t) = L_i + \Delta L_i(t)$, which depends on the actuation phase ϕ_i. Furthermore, we assume that each unit experiences a nonlinear friction $F(\dot{x}_i)$ with the surface that depends on the mass velocity $\dot{x}_i$. We can then write the system's equation of motion as follows,

$$\mathbf{M}\ddot{\mathbf{x}} = -\mathbf{K}\mathbf{x} - \mathbf{F}(\dot{\mathbf{x}}) + \mathbf{A}(t), \tag{3.1}$$

where $\mathbf{M}$ is a mass matrix, $\mathbf{K}$ is a stiffness matrix, $\mathbf{F}$ is a friction force vector and $\mathbf{A}$ a vector containing the actuation forces exerted by the actuators on the connected masses. Specifically, the matrices for the assembled system containing $(n-1)$ active units (i.e. n masses) are equal to

$$\mathbf{M} = \begin{bmatrix} m_1 & & & \\ & m_2 & & \\ & & \ddots & \\ & & & m_n \end{bmatrix},$$

$$\mathbf{K} = \begin{bmatrix} k_1 & -k_1 & & & \\ -k_1 & k_1+k_2 & -k_2 & & \\ & \ddots & \ddots & \ddots & \\ & & -k_{n-2} & k_{n-2}+k_{n-1} & -k_{n-1} \\ & & & -k_{n-1} & k_{n-1} \end{bmatrix},$$

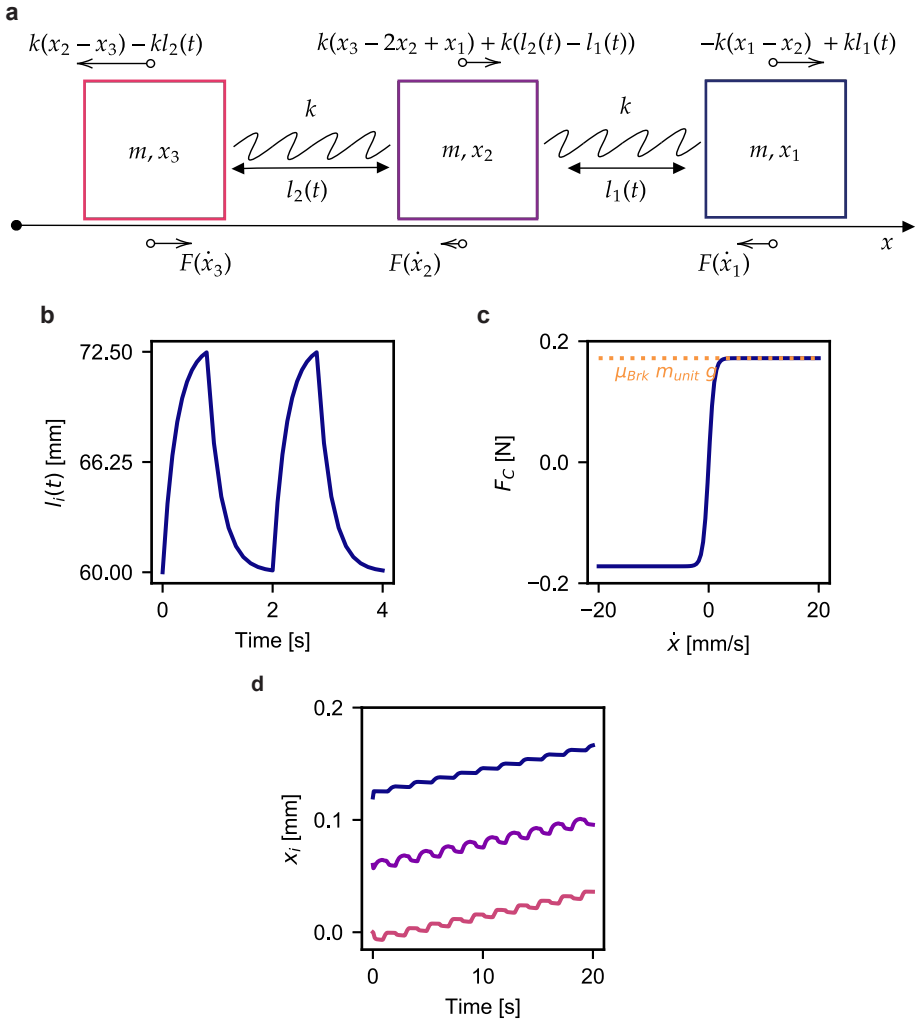

FIGURE 3.9: Numerical model schematic and implementaion. a, One dimensional mass-spring model schematic for an assembled robot consisting of two active units (and one dummy unit). b, Actuator's preferred spring length $l(t)$ as a function of time for the specific case of $\phi = 0$. c, Modelled Coulomb friction F_C as a function of the relative velocity $\dot{x}$ to the ground. d, Example of positions x_i for the three masses in a learning step of $n_{\text{act}} = 10$ actuation cycles, with $\phi_1 = 0$ and $\phi_2 = 0.4$.

$$\mathbf{A}(t) = \begin{bmatrix} k_1 l_1(t) \\ k_2 l_2(t) - k_1 l_1(t) \\ \vdots \\ k_{n-2} l_{n-2}(t) - k_{n-1} l_{n-1}(t) \\ k_{n-1} l_{n-1}(t) \end{bmatrix}.$$

As an example, if we assume that all units are identical (i.e. $m_i = m$ and $k_i = k$), for two active units we have

$$m \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix} \begin{bmatrix} \ddot{x_1} \\ \ddot{x_2} \\ \ddot{x_3} \end{bmatrix} = -k \begin{bmatrix} 1 & -1 & \\ -1 & 2 & -1 \\ & -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} - \begin{bmatrix} F(\dot{x_1}) \\ F(\dot{x_2}) \\ F(\dot{x_3}) \end{bmatrix} + k \begin{bmatrix} l_1(t) \\ -l_2(t) - l_1(t) \\ l_2(t) \end{bmatrix}. \tag{3.2}$$

In experiments, the pump is cyclically turned on and off for t_{on} and t_{off} seconds, respectively. When the pump is turned on, the flow into the soft actuator is approximately constant. Simultaneously, the actuator vents air through a needle. To model the inflation and deflation cycle of the actuator characterized by the actuation function $l_i(t)$ in Eq. (3.1), we turn to an electronic analogy [145]. If we assume the pump acts like a current source, the actuator as a capacitor, and the needle as a resistor (with parameters I, C and R, respectively), the voltage V in the capacitor during one full cycle (representing the pressure in the actuator) is given by

$$V = \begin{cases} V_{\text{off}} + (IR - V_{\text{off}})(1 - e^{-t/(RC)}) & \text{for } 0 \leq t \leq t_{on} \\ V_{\text{on}} e^{-t/(RC)} & \text{for } t_{\text{on}} \leq t \leq t_{\text{off}} + t_{\text{on}}, \end{cases} \tag{3.3}$$

where V_{off} is the voltage in the capacitor at the end of the previous deflation step, and V_{on} is the voltage in the capacitor at the end of the inflation step. We next transform this voltage to extension of an actuator by assuming $\Delta L = \beta V$, such that

$$l_i(t) = L_i + \beta V_i, \tag{3.4}$$

where we assumed a linear relation between the pressure inside the actuator and the extension of the actuator.

We model the friction force $F_i(\dot{x}_i)$ for each mass as a combination between a Coulomb $F_{\text{C},i}(\dot{x}_i)$ and a viscous $F_{\text{v},i}(\dot{x}_i)$ friction [146]. The Coulomb friction is given by

$$F_{\text{C},i}(\dot{x}_i) = (\mu_{\text{Brk}} m_i g) \tanh\left(\frac{\dot{x}_i}{v_{\text{C}}}\right), \tag{3.5}$$

in which μ_{Brk} is the breakaway friction coefficient between each mass and the surface, g is the gravitational constant, and v_{C} is the Coulomb velocity threshold which scales the speeds at which the friction barrier is broken. Moreover, the viscous friction equals

$$F_{\text{v},i}(\dot{x}_i) = c\dot{x}_i, \tag{3.6}$$

where c is the viscous damping coefficient. The total friction force that acts on each mass equals

$$F_i(\dot{x}_i) = F_{\text{C},i}(\dot{x}_i) + F_{\text{v},i}(\dot{x}_i). \tag{3.7}$$

Note that the μ_{Brk} parameter for the Coulomb friction can be observed experimentally. In contrast, from an experimental perspective v_{C} is ideally equal to zero. However, in simulations we used $v_{\text{C}} = 0.001$ to smooth the transition and prevent numerical problems. Moreover, we included the viscous term $F_{\text{v},i}$ to numerically dampen the system and avoid possible numerical instabilities or high-frequency oscillations. We noticed that the choice of parameter c only effects the quantitative agreement between simulations and experiments shown in Fig. 3.2e, and we found a value of $c = 5$ to be appropriate.

Simulation setup

To determine the motion of the assembled system of equations, we numerically solve Eq. (3.1) using the implicit Radau method implemented in a custom script in Python with the SciPy integration library (version 1.4.1). To perform a learning simulation, in the setup phase a worm object is instantiated and all parameters are set to their initial values. This includes the resting length of the springs which is always set to a constant value of $L = 60$ mm. A random phase ϕ_i is assigned to each unit, along with a fixed number of learning steps and a simulation time of 20 s per learning step, during which the phases ϕ_i are kept constant. The phase ϕ_i is used to determine the time-shift of the preferred spring length l_i for each spring. E.g. for a phase $\phi_i = 0$, the preferred spring length starts increasing at $t = 0$, as shown in Fig. 3.9b. After the last learning step, units calculate their score U' as the average distance travelled relative to the simulated time of 20 s. Units use this score to update their phases as done in experiments. Similarly, the phases are also perturbed in the same way after each learning step, and new candidate phases ϕ_i' are set for the next learning step. Note that at the beginning of every learning step, we reset the displacements of the units to their rest length, such that the system undergoes transient behaviour during the first actuation cycle, as can be seen in the first seconds in Fig. 3.9d. This is the result of the difference between the natural rest length of the spring 60 mm, and the new rest length as determined by the new actuation phase ϕ_i'. This does not significantly affect the U', as we simulate a relatively large period of time.

Nevertheless, when we compute the system velocity as a function of the phase combinations (e.g. Fig. 3.2e), the simulation settings with fix time does include some asymmetry between forward and backward motion, since we for example set $\phi_1 = 0$ for all our simulations. This cause that a phase combination of $\phi_1 = 0$ and $\phi_2 = 0.4$ produces a locomotion that is not exactly the same as for $\phi_1 = 0$ and $\phi_2 = 0.6$. For example, in the two active unit case, we observe that the backward maximum velocity ($\Delta\phi = 0.6$) is 0.3 [mm/cycle] higher than the forward velocity ($\Delta\phi = 0.4$). I.e., considering the velocity in forward locomotion as the real value, since in this case both actuations have ceases at the end of the simulated time, we measure an asymmetry (in terms of relative error) of -8%. To further verify this, we also compute the same phase difference sweep where learning steps have a duration $80\ s$. In this case, we see that the asymmetry drops to -2%, where we measure a velocity difference < 0.1 [mm/cycle].

Model parameter measurements

To characterize the behaviour of the soft actuator (i.e. stiffness k and the extension ΔL), we measured the force-displacement response of the actuator using a mechanical testing machine (Instron 5965L9510). To determine the stiffness, we start with an actuator that is open to the surroundings and perform a pulling experiment four times, in which we deform the actuator at a constant deformation rate of 120 mm/min to an extension of 20 mm, return to the initial position, and compress 3 mm beyond the initial position (Fig. 3.10a). Based on these results, we extract the slope of the curve to determine the stiffness k of the actuator, as shown in Fig. 3.10b.

To determine if the stiffness of the actuator depends on the internal pressure, we next performed a second experiment where we start with an unpressurized, but closed actuator, and performed the same extension/compression testing routine. We then add 1 ml of air to the actuator using a syringe pump, leading to an increase in the internal pressure. We then closed a valve in order to disconnect the syringe during the pulling experiment and repeated this procedure for a total of four measurement cycles. In Fig. 3.10c-d we show the measured force-deformation curves and the corresponding stiffness k for the closed cases, respectively. We find that the stiffness is not significantly affected by the internal pressure when the actuator is closed to the surroundings. It is important to note that during compressive loading at small or negative extensions the actuator undergoes a global buckling instability that suddenly changes the stiffness (Fig. 3.10c-d). However, such an instability has never been observed in the actuators during normal operation of the units. Moreover, we also find that the closing of the actuator to the surroundings increases the stiffness. This is likely the result of the enclosed air that acts as an additional spring. Since the experimental situation is between open and closed,

and preliminary results showed that the response is qualitatively identical independently of the exact numerical value of stiffness, we chose $k = 100$ N/m for the numerical simulations. To determine the parameters used to specify the actuation

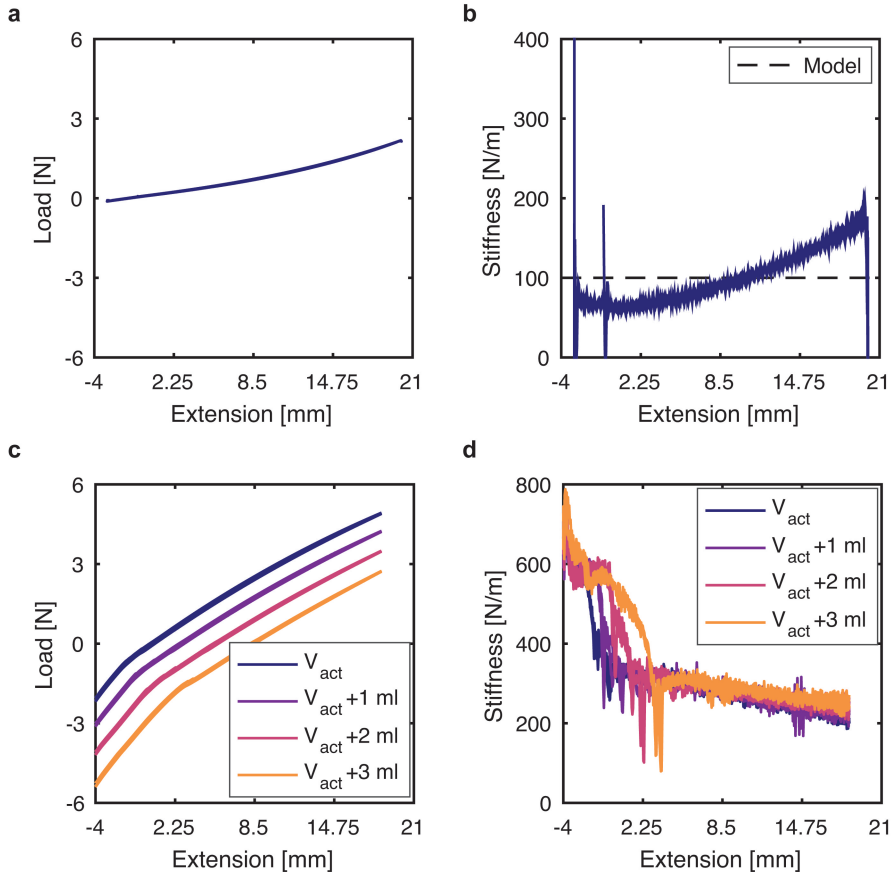

FIGURE 3.10: Characterization of a soft actuator. a, b, Force-extension and stiffness-extension response of an actuator open to the environment. The value $k = 100\ N/m$ was chosen as actuator stiffness in the numerical model. c, d, Force-extension and stiffness-extension response for an actuator closed to the environment (i.e. contains a fixed amount of air), for increasing enclosed air volumes from $V_{tot} = V_{act}$ to $V_{tot} = V_{act} + 3\ ml$, in steps of $1\ ml$.

cycle, we measure the maximum extension $\Delta L_{\max}$ and corresponding pressure $P_{\max}$ of the actuator during cyclic operation. We find that $\Delta L_{\max} = 12.5$ mm and $P_{\max} = 6$ kPa. Given that the maximum pressure in the actuator equals the

maximum voltage $V_{\text{on}} = 6$ in our fluidic-electric model (Eq. (3.4)), we find that $\beta \approx 2.08$ mm/kPa. We next tune I, R and C such that $V_{\text{on}} \approx 6$, and $V_{\text{off}} \approx 0.01 V_{\text{on}}$ (such that the actuator is empty at the end of the deflation cycle). We find that values of $I = 0.35$, $R = 18$ and $C = 0.0145$ give a realistic inflation and deflation behaviour according to Eq. (3.3), as shown in Fig. 3.9b.

To characterize the friction $F(\dot{x})$ of the unit moving on the experimental table, we attached one unit to a horizontal material testing machine (Instron 5900 equipped with 100 N load cell) as shown in Fig. 3.11a), and measured the reaction force when prescribing a displacement (Fig. 3.11b) to the unit that is moving on a similar surface as the one used in the learning experiments. Note that we used the bottom part of one of the units, and placed additional weight on it to a total of $m_{\text{Test}} = 0.811$ kg, in order to increase the resolution in our measurements. To understand if this coefficient is influenced by the relative speed of the units, we performed five tests in which a maximum displacement of 126 mm is reached after ramping up and ramping down the displacement rate as shown in Fig. 3.11b. Assuming that the friction force is equal to $\mu_{\text{Brk}}\ m_{\text{Test}}\ g$, the resulting friction coefficient μ_{Brk} is shown in Fig. 3.11c. Apart from the initial increase in friction coefficient when the unit is set into motion, we find an approximate constant friction force as a function of the unit's relative velocity. Based on this result, we chose $\mu_{Brk} = 0.24$ as the breakaway friction coefficient in our numerical model (Eq. (3.5)). The final friction behaviour is shown in Fig. 3.9c.

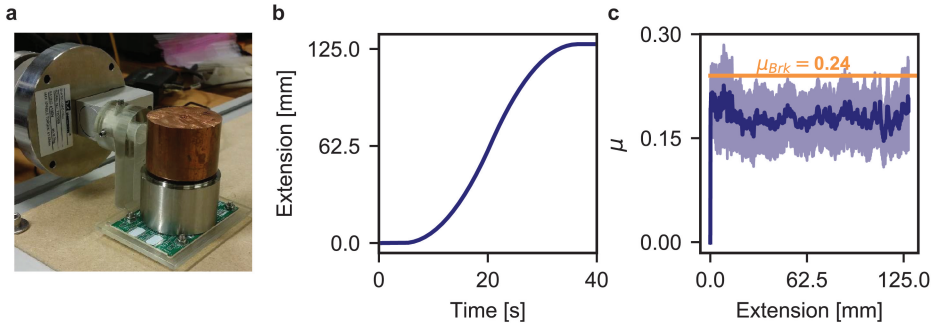

FIGURE 3.11: Friction characterization between the unit and the table. a, overview of the horizontal friction characterization experiment. Additional weight has been added to the unit to increase the magnitude of the force and reduce relative noise. b, Applied displacement (extension) profile as a function of time. c, Measured friction coefficient, and standard deviation, as function of the applied extension. The value $\mu_{Brk} = 0.24$ was chosen as breakaway friction coefficient to use in the numerical model.

Difference between experiments and simulations

Asymmetric locomotion in experiments

While we expect a symmetric velocity profile with respect to phase combinations, it is not the case for our experiments. We observed that a reduction of friction might be caused by the vibration of the pump, and since the dummy unit is never vibrating, this may cause the difference between forward and backward locomotion. This seems to be suggested also by comparing the experimental and simulated average velocity $\bar{U}$ as a function of the phases combinations, in a three active units system with damage, Fig. 3.6d and Fig. 3.21e.

To verify whether a higher friction on the dummy unit introduces the experimental asymmetry we see e.g. in Fig. 3.23, we ran simulations on a two active unit system, in which the Coulomb friction F_C on the dummy mass is multiplied of a factor β. In Fig. 3.12a, we show the velocity as a function of the phase difference $\delta\phi$, for different β, and see that the increase in friction reduces the maximum and minimum speed the system can achieve. Specifically, it seems that the backward velocity peak is affected more by the increase in β. To quantify this effect, we consider the relative difference between the maximum and minimum velocity, and find an increasing trend as β increases, despite the simulation step time of 20 or 80 s, see Fig. 3.12b. While the higher friction on the dummy unit decreases the maximum velocity in a backward motion, it still does not explain the presence of the local maximum at $\Delta\phi \approx 0.9$ we find in experiment.

Experimental noise

We believe that the noise observed in experiments is mainly caused by three effects, related to the sensing hardware, decentralisation, and environment.

First, while the position sensor is relatively precise (steps of 0.016 mm) and can handle much higher speeds (up to 150 mm/s, we do observe some rotation when the units are moving along the track. Such rotation is not captured by the sensors, and could therefore cause an integration error. This causes a relatively high noise for phases that result in approximately no movement.

Second, since measurements of the position are done separately by each unit, the moment at which the position is measured by each unit varies in time (e.g., see Fig. 3.23). Therefore, when one unit is measuring its displacement, another unit might still be moving. This effect is furthermore strengthened by the fact that units' internal clocks go out of sync as the learning progresses. This noise therefore depends on the state of the system, and the time during experiments.

Third, experimental noise is also caused by variations in the track (the track has been the same for all experiments). As mentioned in the section on experimental setup, some regions of the tracks are narrower than others, where we observe a standard deviation of ~ 0.4 mm between the 20 measurements at different track

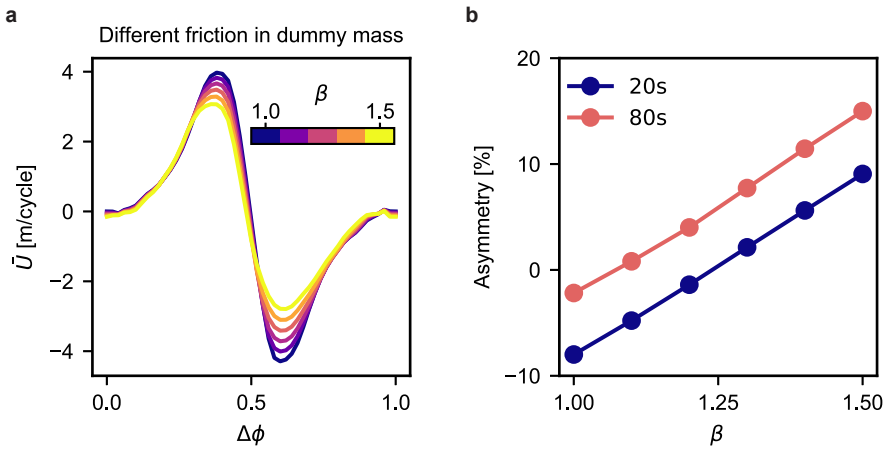

FIGURE 3.12: Simulations results with higher friction on dummy mass. a, Velocity as a function of the phase difference $\Delta\phi$ for different friction factor β. b, System locomotion asymmetry, expresses as the relative error between forward and backward locomotion, increasing friction factor β. The two curves represent results for simulations steps of 20 s (standard settings) and 80 s.

locations. This causes the units' feet to experience more play in some regions and less in others, leading to small rotations besides the tangential translation when pushed by the actuators. This is also demonstrated by the results in Fig. 2d, where we show the average measured velocities across the system for 20 measurements starting from different initial positions. In Fig. 2d we show the standard deviation between these measurements highlighting the large variation in performance for fixed phase combinations and different positions. Some of the individual experimental scan can be seen in Fig. 3.13, where the landscape is explored by fixing $\phi_1 = 0$ and ϕ_2, and varying ϕ_3. There is a significant dependency of the potential behaviour the units can exhibit and where they are on the track, and so this effect is rather a dynamic condition, rather than noise.

To determine if the velocity measurement taken by the local sensors is accurate, we also compare the units' displacement measurements with an external camera. Considering the experiment with fixed phases for three active units in Movie 2, we see that it takes approximately 1182 seconds to complete a full circle of length 2.68 meters. Comparing this to the distance measured with the sensors, we find the three motion sensors underestimate the traveled distance with relative errors of $1\%, 9.7\%$ and 10.7% respectively.

Importantly, we did not try to further reduce the above effects, but rather embrace them, as one of our goals is to design a robust learning algorithm that is capable of dealing with dynamic (and noisy) situations. Note that all three cases

have not been included in most simulations, except for a noise term similar to sensor noise that we have studied in Fig. 3.14, and changing friction and damage shown in Fig. 3.18, Fig. 3.19, Fig. 3.20 and Fig. 3.21 a, respectively.

Supplementary figures

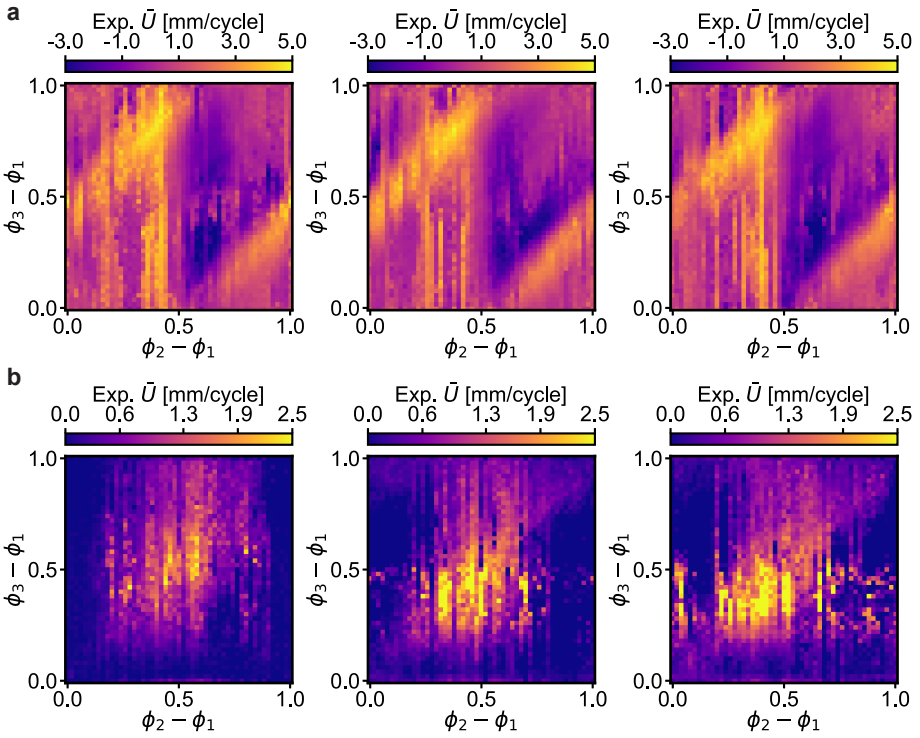

FIGURE 3.13: Selection of experimental velocity scans for an assembled robot consisting of three active units and one dummy unit. The results show three out of 20 experimental runs that were used to determine the average speed $\bar{U}$ as a function of all the possible combinations of phases $\phi_3 - \phi_1$ and $\phi_2 - \phi_1$ (with $\phi_1 = 0$), for (a) an undamaged and (b) a damaged robot. Each scan is initialized by placing the robot at different starting positions.

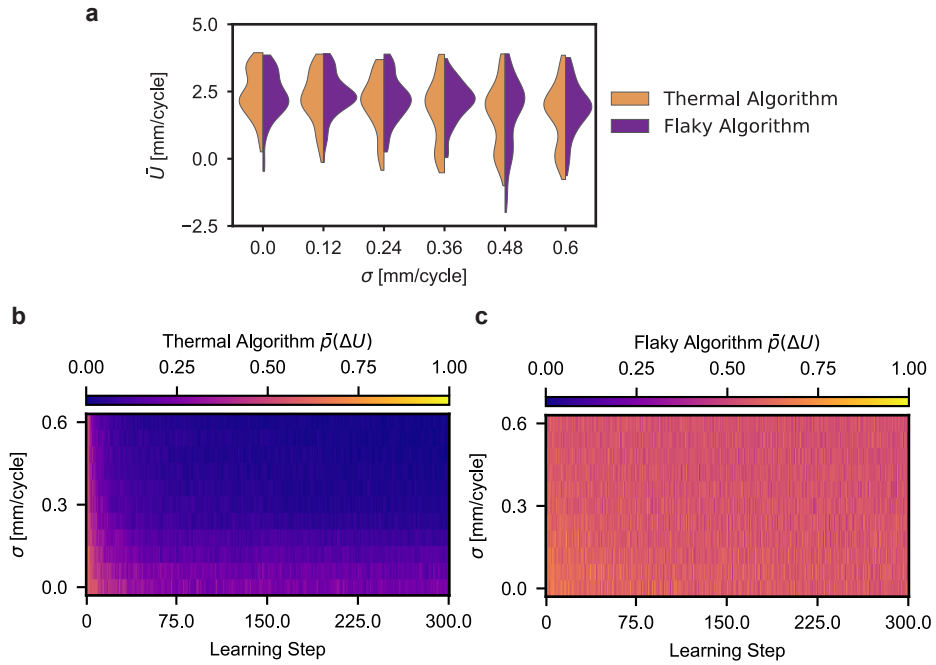

FIGURE 3.14: Effect of noise on learning strategies. To understand how the performance of the Thermal and Flaky algorithms are effected by noise, we performed simulations on a assembled robot consisting of three active and one dummy unit, in which the measured speed U_i of each unit was altered by a random value u drawn from a Gaussian distribution with standard deviation ranging from $0 \leq \sigma \leq 0.6$. For each selected value of σ we performed 100 simulations. a, Distribution of measured velocities after 300 learning steps, as a function of the standard deviation σ of the noise applied. We notice that for both algorithms, even for larger values of the standard deviation, the system learns how to move (although less effectively). However, for the Thermal algorithms and for an applied noise with $\sigma > 0.24$ [mm/cycle] we see an increasing number of simulations with $\bar{U} \approx 0$. b, Important to note is that for the Thermal algorithm the noise level has a large effect on the acceptance probability $\bar{p}(\Delta U)$, which quickly tends to go to zero for increasing number of learning steps and for $\sigma > 0.024$ [mm/cycle]. c, In contrast, noise does not seem to effect the acceptance probability $\bar{p}(\Delta U)$ when using the Flaky algorithm.

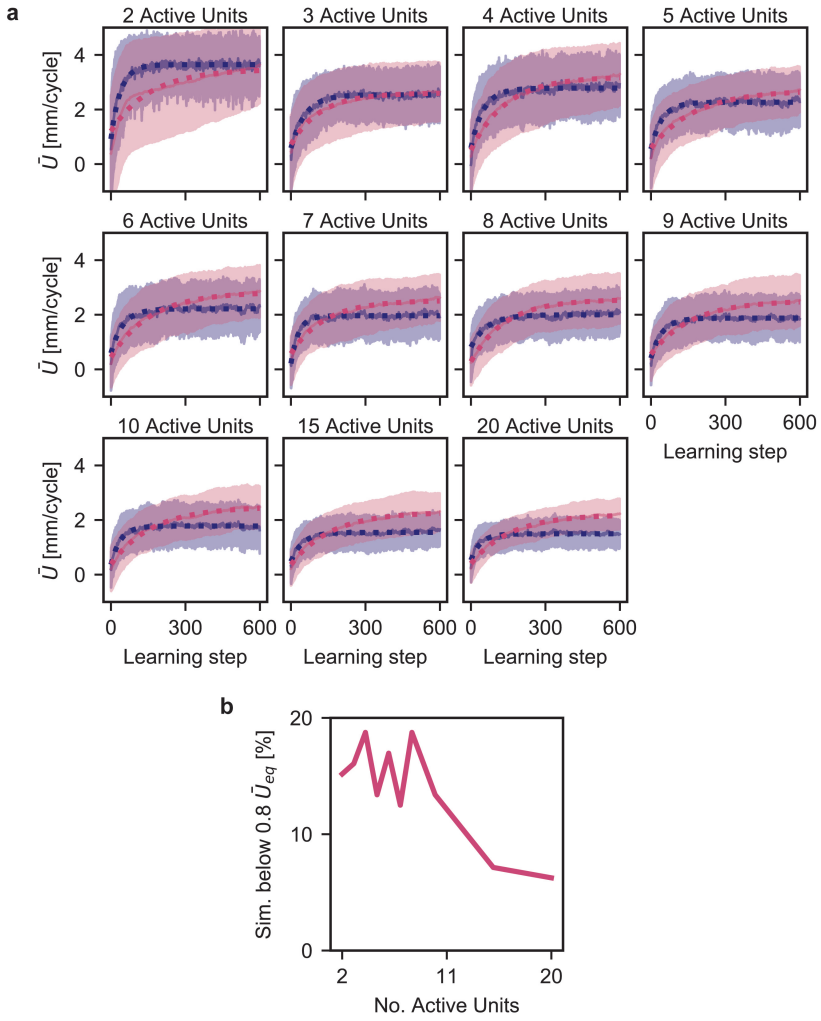

FIGURE 3.15: Scalability of the decentralised learning approach. To study the effect of the number of units on the effectiveness of our updated learning algorithm, we performed numerical learning experiments on assembled robots consisting of two to twenty active units (and one dummy unit). We performed these simulations for two different stepsizes of $\Delta s = 0.1$ and $\Delta s = 0.025$, in blue and pink respectively. a, Solid lines represents the average over 112 simulations, while the shaded area represents the standard deviation. The average curves have been fitted by an exponential function $y = \bar{U}_{eq} - \left(\bar{U}_{eq} - \bar{U}_0\right) e^{-\gamma\, x}$, shown by the dashed line. b, Percentage of simulations for $\Delta s = 0.025$ not reaching the $0.8\,\bar{U}_{eq}$ threshold within 600 learning steps. Note that for $\Delta s = 0.1$ all simulations reach $0.8\,\bar{U}_{eq}$.

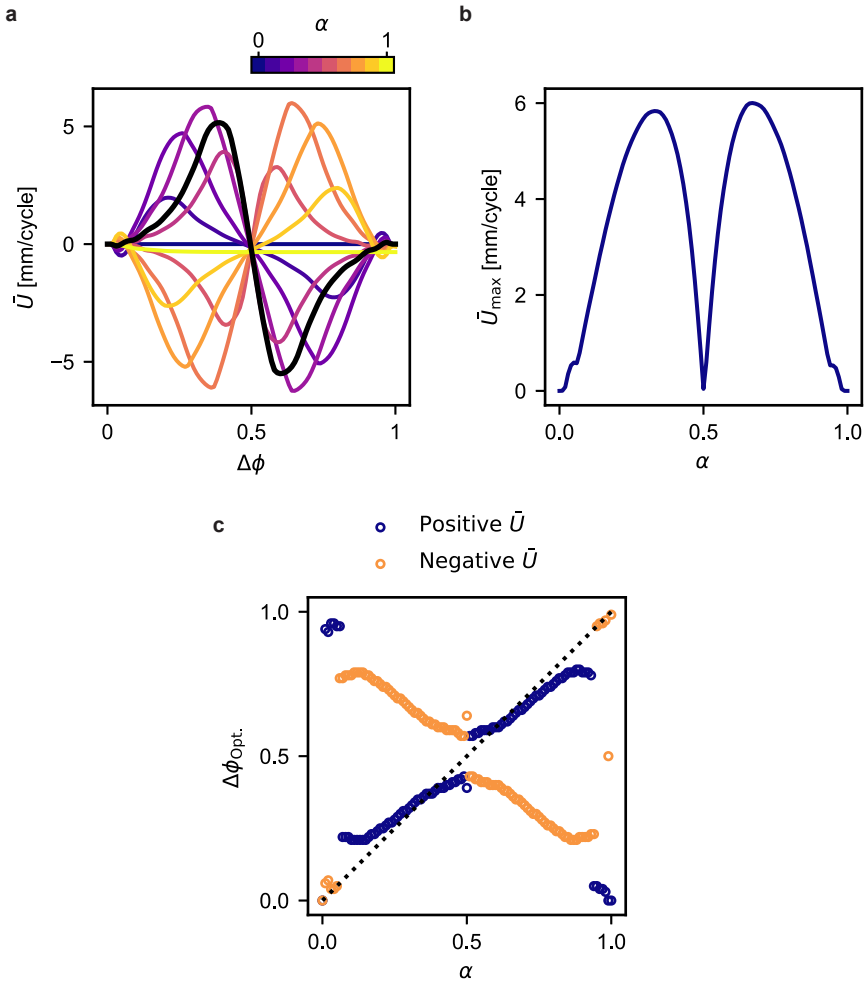

FIGURE 3.16: Simulations to determine the influence of the actuation duration α on the displacement of a two active unit robot. a, Average velocity of the assembled robot as a function of the phase difference $\delta\phi = \phi_2 - \phi_1$, for different values of α. The black line shows the response for $\alpha = 0.4$, that has been used throughout our studies. b, Maximum system velocity $\bar{U}_{max}$ as a function of α. c, Optimal phase difference $\Delta\phi_{Opt.}$ (for both positive and negative $\bar{U}$) as a function of α.

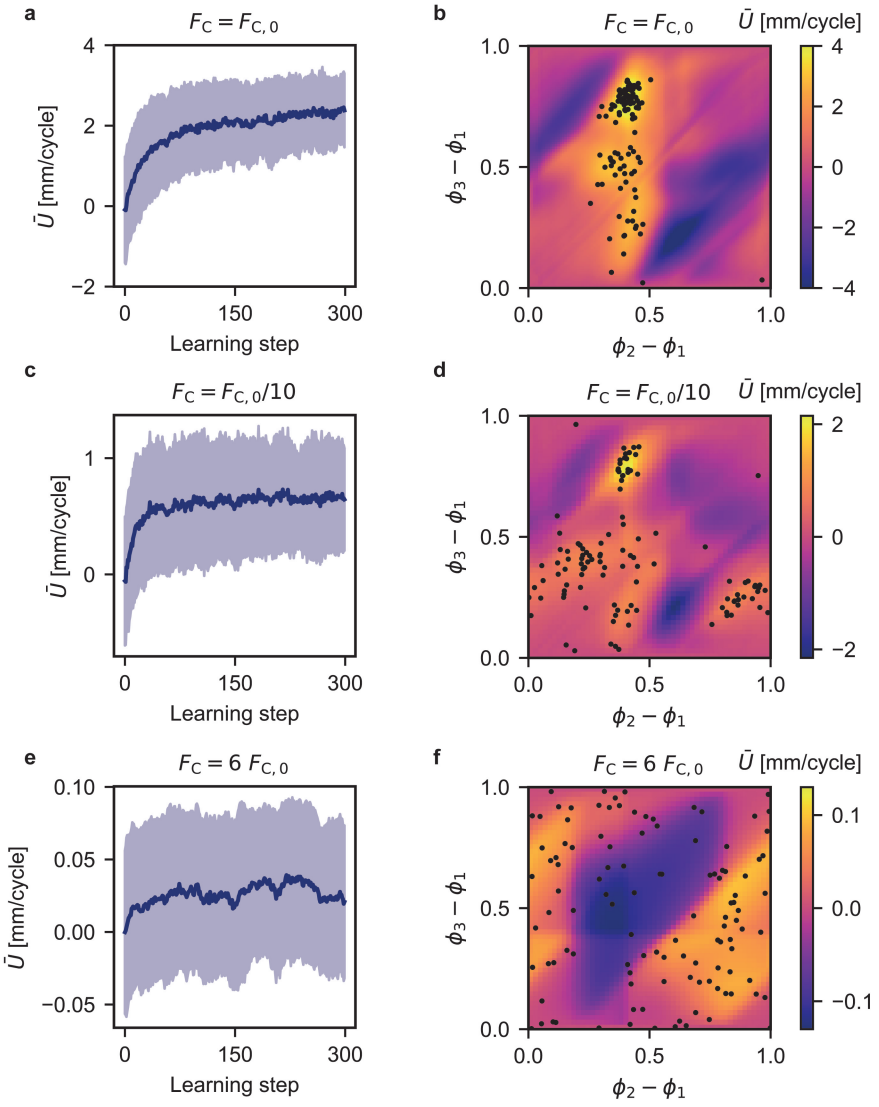

FIGURE 3.17: Effect of friction on the potential velocity and ability to learn of a three active units robot running the Flaky algorithm. We performed 112 simulations for each of three different frictions, where we determine both the average velocity $\bar{U}$ as a function of the learning step (where the shaded area represents the standard deviation), and the optimal phase combinations reached at the end of the last learning step, as indicated by the dots in the contour plot. a-b, Results for the friction similar to the experiments ($F_C = F_{C,0}$) as used throughout our studies. c-d, Results for reduced friction ($F_C = F_{C,0}/10$), qualitatively similar to the robot moving on ice. e-f, Results for increased friction ($F_C = 6\ F_{C,0}$).

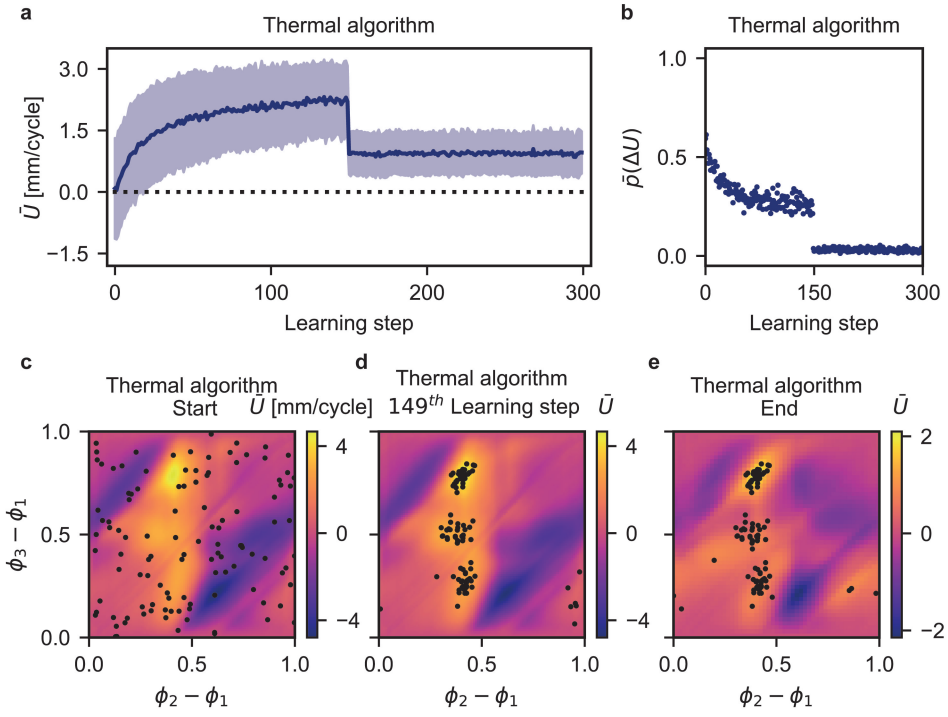

FIGURE 3.18: Results of 112 learning simulations of a three active units robot with sudden and sharp drop in friction at the 150^{th} learning step, for the Thermal algorithm. a-b, Average velocity $\bar{U}$ and average acceptance rate $\bar{p}$, as a function of the learning step. c-e, Optimal phases combinations at the beginning of the learning, just before the friction change, and at the end of the last learning step.

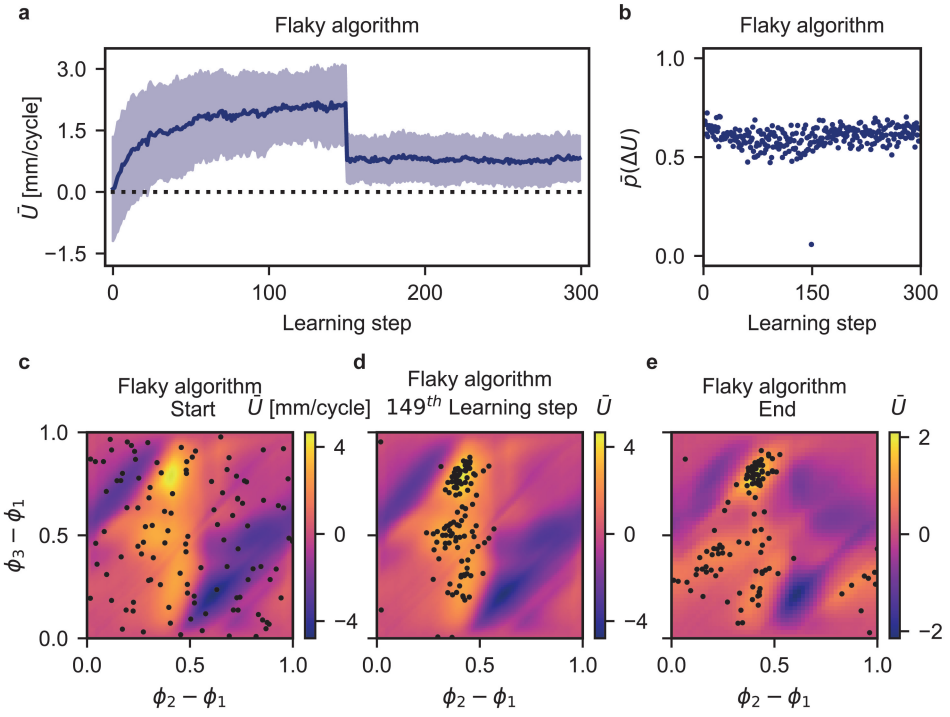

FIGURE 3.19: Results of 112 learning simulations of a three active units robot with sudden and sharp drop in friction at the 150^{th} learning step, for the Flaky algorithm. a-b, Average velocity $\bar{U}$ and average acceptance rate $\bar{p}$, as a function of the learning step. c-e, Optimal phases combinations at the beginning of the learning, just before the friction change, and at the end of the last learning step.

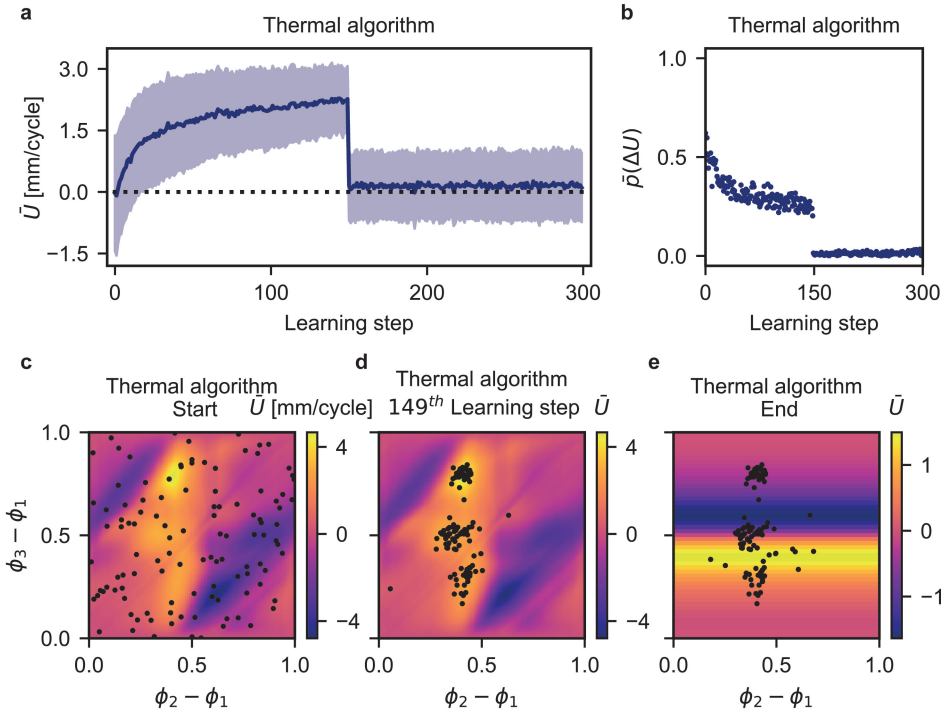

FIGURE 3.20: Results of 112 learning simulations of a three active units robot with sudden damage on the second unit at the 150^{th} learning step, for the Thermal algorithm. a-b, Average velocity $\bar{U}$ and average acceptance rate $\bar{p}$, as a function of the learning step. c-e, Optimal phases combinations at the beginning of the learning, just before the damage is applied, and at the end of last learning step

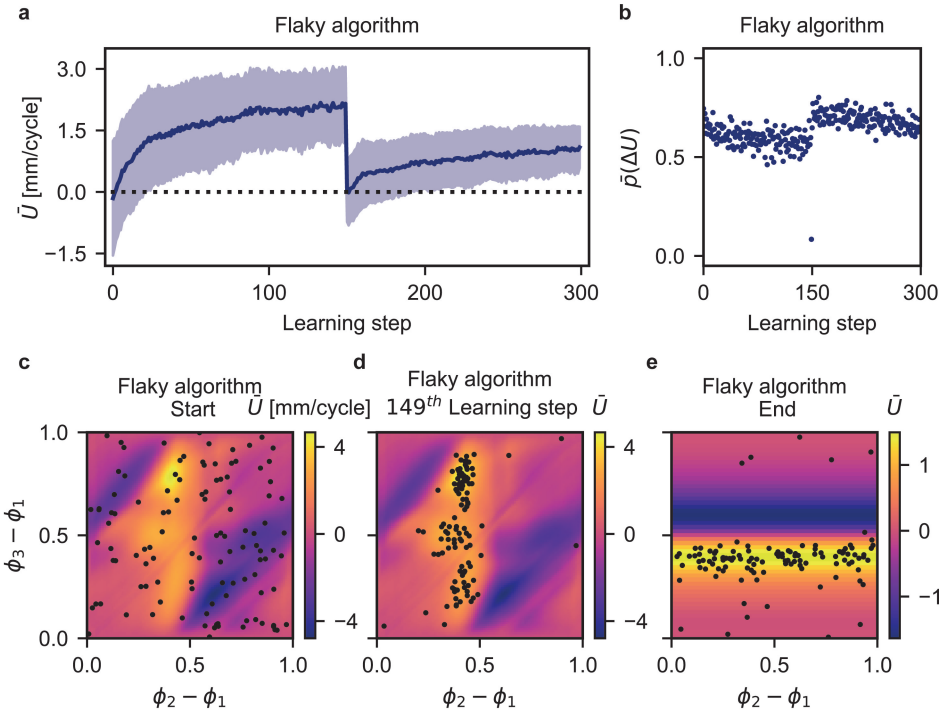

FIGURE 3.21: Results of 112 learning simulations of a three active units robot with sudden damage on the second unit at the 150^{th} learning step, for the Flaky algorithm. a-b, Average velocity $\bar{U}$ and average acceptance rate $\bar{p}$, as a function of the learning step. c-e, Optimal phases combinations at the beginning of the learning, just before the damage is applied, and at the end of last learning step

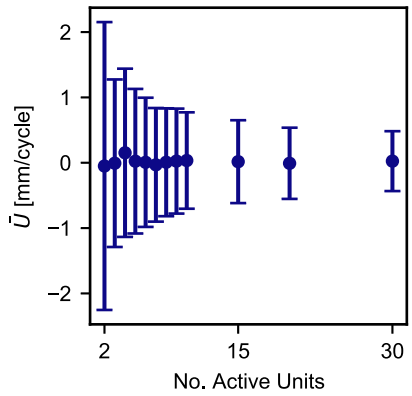

FIGURE 3.22: Sampling the scalable behaviour. Simulated system velocity as a function of the number of active units. The dots represents the average value, and the bars indicate standard deviation values over the 1000 random samples per system size.

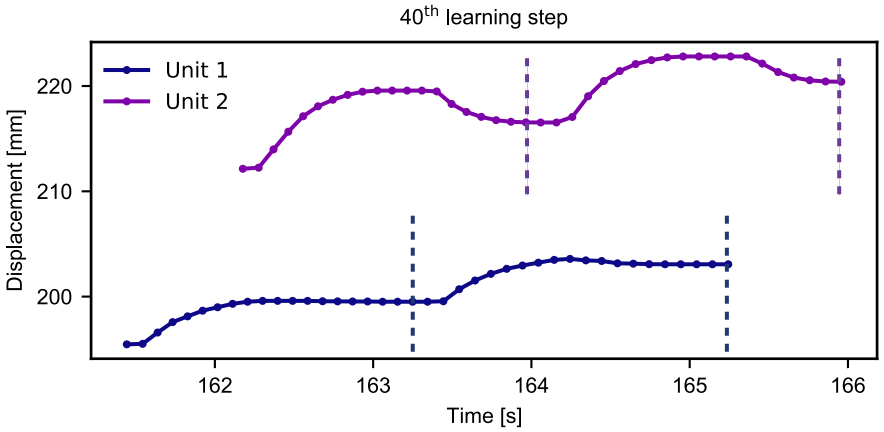

FIGURE 3.23: Measured displacements at the 40^{th} learning step by the two active units in the learning experiment shown in Fig. 1d. The vertical dashed lines mark the beginning and end of the second cycle, that is used to calculate the units' average velocities U_1 and U_2.

CHAPTER 4

Decentralised learning on idealized environments

In preparation as:

Oliveri, G., Carissimo, C., Overvelde, J.T.B.
Reinforcement learning strategies in multi-agent systems

In Chapter 3 we introduced a minimal, decentralised control strategy for robotic systems made of modular independent units moving in 1D. We have shown that a forward locomotion of the complete system can be achieved by having each unit independently adapt its behaviour based on local measurements. Despite the promising results in terms of adaptivity and scalability, the performance of the learning strategies depend on the system's interaction with the environment, making it difficult to understand the qualities and limitations of the used algorithms. In this chapter we develop a simulation framework where the interaction between the system and the environment is analytically described, i.e. representing idealized case scenarios. By defining various environment functions, we characterize the introduced learning strategies in terms of learning rate and ability to find and maintain optima in static and dynamic environments, while also investigating the influence of system size and noise. We find that, while the Thermal algorithm guarantees the highest learning rate and ability to find and maintaining optima in smooth and static environments, it is very sensitive to noise and dynamic circumstances. Differently, the Flaky algorithm is consistently able to improve the system behaviour both in static and dynamic circumstances, even in the case of large noise applied.

4.1 Introduction

In Chapter 3 we have introduced decentralised learning strategies for the control of robotic systems. By embedding sensing and decision making capabilities in each robot part, we were able to achieve an emergent whole-body locomotion as a results of policy specialization of each component. We define this behaviour as emergent since units are individually incapable of locomotion, and require to be physically connected and to cooperate in order to move. We showed that the two introduced learning strategies are successful and improve the overall robot behaviour, and that enhancement to the system's adaptability[1] to changing environments can be achieved by updating how the sampled space is stored in memory. We furthermore showed that the Flaky algorithm is suitable for the control of an increasing number of active units maintaining the ability to improve its performance, and allowed us to define this learning strategy scalable. Nevertheless, since the optimization landscape is the results of the system's physical interaction with the environment, the increase of system size augment the complexity of landscape whose features such as maxima and minima, are impossible to characterize. See e.g. the increase of possible behaviours when going from a two to three active unit system in Fig. 3.2e and Fig. 3.4c-d.

Here, with the aim of defining the limiting factors that this decentralised approach might have, we develop a better understanding of how the two algorithms work. We study the behaviour of independent decision makers on simplified landscapes, which describe the relation between the system and an environment, without the need of solving the equations of motion. In fact, by formulating the landscapes analytically we can fully characterized the complete space of solutions, and asses the performances of the decentralised learning algorithms with respect to different criteria. Note that the idealized cases loos physical meaning, but do highlight specifically the qualities of the learning strategies. First of all, we quantify the learning rate of a given system with respect to the amount of noise (which is rather hard to quantify experimentally), and its number of agents. Specifically, by only focusing on the algorithms, we can focus on understanding how system size plays a role in the learning strategy neglecting physical limitations.[2] Furthermore, as the experimental exploration of the landscape features is cumbersome even for relatively small system sizes, by providing an analytical formulation of

[1] Adaptivity describes the ability of the learning strategy to adapt to mutable circumstances, and maintain the ability to learn and improve the performance of the system. E.g. In Chapter 3 we study the robot adaptive behaviour with respect to variable friction, damage or re-arranging of unit position within the system.

[2] In Chapter 3 the learning strategy is embodied in the units, and we refer to "scalability" as the possibility of physically move with multiple units, and the capability of learning "good enough" behaviours with multiple units. There we could not discern one from another. Here, by leaving out the physical aspect, we can direct our interest to the latter aspect.

the landscape, we can probe the system's ability of finding global optima and ability to maintain it.

4.2 Methods

To fully characterize the performances of our proposed algorithms, we developed a simulation framework where the decision making of N agents is based on the acceptance and rejection of random numbers $\boldsymbol{\phi} = [\phi_1, \phi_2, ..., \phi_N]$. Starting from a random initial state, each agent is initialized with an input $\phi_i \in [0, 2\pi]$, and evaluates the environment function $\Phi(\boldsymbol{\phi})$ which depends on the input of all agents within the system.[3] Then, each agent draws a new input ϕ_i' from a uniform distribution in between $[\phi_i - \Delta s/2, \phi_i + \Delta s/2]$ where Δs is the search bound width or step size, and evaluate the environment function again. Based on the improvement of the environment function, e.g. $\Delta\Phi(\boldsymbol{\phi}) > 0$, each agent decides to accept the new input ϕ_i' and store it in memory, move the search bounds, and draw a new input. In case of rejection, the search bound is maintained around the last-accepted input ϕ_i. It is important to notice that, while agents accept and reject inputs independently, their performances are linked to the choices made by the others, as expressed by the environment function $\Phi(\boldsymbol{\phi})$. In order to take into account the various features a physical landscape can show, in this Chapter we consider multiple environment functions (which represent different global behaviours) to assess the learning strategies' different properties.

Linear environment

First, we start by looking at a linear environment function, which depends on the sum of the inputs ϕ_i,

$$\Phi(\boldsymbol{\phi}) = \sum_{i=1}^{N} \phi_i. \tag{4.1}$$

In particular, given that the function has constant derivative, we can use this function to determine the (linearized) learning rate. This environment does not have maxima, and could e.g. be the case of a system moving towards a light source infinitely far, where $\Phi(\boldsymbol{\phi})$ would represent the measured light intensity. Furthermore, because of its simplicity, this environment function helps us introduce the test cases we will carry out for more articulated environment functions.

[3] In contrast with what done in Chapter 3 where each robotic unit senses the environment independently, here the environment score is the same for all agents. Since agents perform decision independently with no knowledge of the other agents, we refer to this type of learning as decentralised. Nevertheless, in further sections the addition of noise between different agents, makes our definition better fitting the traditional definition of decentralised systems, where independent decisions are based on local measurements.

In addition we explore the effects of noise on the learning capabilities. Following the previous light source example, noise could derive from sensor input which alters the environment score for the same agents' inputs.

Sinusoidal environment

Further, inspired by the two active units landscape observed in Chapter 3, where e.g. the system's maximum speed is obtained with a fixed phase difference between the units, we define a sinusoidal environment function. This is defined as

$$\Phi(\boldsymbol{\phi}) = \frac{1}{N-1} \sum_{i=2}^{N} \sin\left(\phi_i - \phi_{i-1}\right). \tag{4.2}$$

Unlike in Eq. (4.1), Eq. (4.2) has exactly one global maximum such that $\Phi(\boldsymbol{\phi}) = 1$. Therefore, we can use this environment function to determine the ability of the system to reach and maintain maxima in the objective function. This would be for example the case where the units are moving with a maximum speed, where $\Phi(\boldsymbol{\phi})$ represents the velocity.

Dynamic sinusoidal environment

Based on the previous environment function, the dynamic sinusoidal environment function includes time dependency t within the sine argument. The environment function $\Phi(\boldsymbol{\phi}, t)$ is then defined as

$$\Phi(\boldsymbol{\phi}, t) = \frac{1}{N-1} \sum_{i=2}^{N} \sin\left(\phi_i - \phi_{i-1} + \tilde{\omega}\, t\right), \tag{4.3}$$

where $\tilde{\omega}$ is the normalized frequency which describes the change in landscape as a function of time t. This would be for example the case where the system is trying to move in the direction of increasing light intensity, with the light source moving along the horizon. This environment description resemble more closely the experimental conditions, for which the optimal input combination is not constant over time, and the system needs to constantly adapt to the environment. This can be for example seen in Chapter 3, where large performance variations were caused by track changes, or partial failure of the system.

Agents type

To investigate the role of agent behaviour on the overall system performance on various landscapes, we only consider homogeneous systems made of the same type of agents. The agent types we consider here are the Thermal and Flaky agents

introduced in Chapter 3. For the Thermal agents, the acceptance probability of a new tried input $\boldsymbol{\phi}'$, is given by

$$P(\Delta\Phi) = e^{\frac{\Phi(\boldsymbol{\phi}')-\Phi(\boldsymbol{\phi})}{T}} = e^{\frac{\Delta\Phi}{T}}, \tag{4.4}$$

in which T is a 'temperature' that can be used to tune the acceptance probability when $\Phi(\boldsymbol{\phi}') < \Phi(\boldsymbol{\phi})$. Differently, when $\Phi(\boldsymbol{\phi}') \geq \Phi(\boldsymbol{\phi})$ the new input is automatically accepted. In case of acceptance both the input $\boldsymbol{\phi}'$ and the environment score $\Phi(\boldsymbol{\phi}')$ are stored in memory, and used for the next learning step. In case of rejection, the last tried input $\boldsymbol{\phi}'$ is neglected, and both input and environment score of the previous learning step are used as reference in the following step. In this Chapter we will also look at the special thermal agent with $T = 1 \cdot 10^{-10} \approx 0$. Since this agent only accepts better inputs, such that $\Delta\Phi(\boldsymbol{\phi}) = \Phi(\boldsymbol{\phi}') - \Phi(\boldsymbol{\phi}) > 0$, it will behave as a random search algorithm and it will generally represent the upper limit in terms of learning rate. We will refer to this as *Random Search* or *RS* agent.

While in Chapter 3 Flaky agents also consider a temperature T to compute the acceptance probability of a newly tried input, to further simplify the algorithm, we remove it, since it does not alter the its general behaviour. This is because randomness is also built-in through the update of the objective function that does not always correlate to the phase stored in memory. The Flaky agents considered here only accept inputs which improve the environment function ($\Phi(\boldsymbol{\phi}') \geq \Phi(\boldsymbol{\phi})$). The fundamental difference between Random Search and Flaky agents is that the latter stores the environment score $\Phi(\boldsymbol{\phi}')$ even in case of rejection, while the input $\boldsymbol{\phi}'$ is only updated in case of acceptance, maintaining the search space around the last accepted input.

4.3 Learning rate

In order to quantify the systems' learning rate, we first start by considering systems consisting of two agents navigating a linear environment function defined as:

$$\Phi(\phi_1, \phi_2) = \phi_1 + \phi_2. \tag{4.5}$$

Since inputs are updated as

$$\begin{aligned} \phi_1' &= \phi_1 + \epsilon_1 \, \Delta s/2 = \phi_1 + x \qquad x \in \left[-\frac{\Delta s}{2}, \frac{\Delta s}{2}\right] \\ \phi_2' &= \phi_2 + \epsilon_2 \, \Delta s/2 = \phi_1 + y \qquad y \in \left[-\frac{\Delta s}{2}, \frac{\Delta s}{2}\right] \end{aligned} \tag{4.6}$$

where ϵ_1 and ϵ_2 are two independent random numbers within the interval $[-1, 1]$, the improvement between two consecutive learning steps ($\Delta\Phi(\boldsymbol{\phi}) = \Phi(\boldsymbol{\phi}') -$

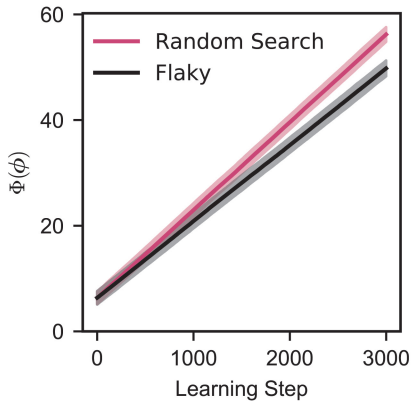

FIGURE 4.1: Average environment score over 100 simulations $\bar{\Phi}(\boldsymbol{\phi})$ as a function of learning steps, for two systems consisting of two agents on linear landscape. Shaded area represents the standard deviation over all simulations results.

$\Phi(\boldsymbol{\phi}) > 0$) occurs either when both agents draw inputs larger than the previous ($x, y > 0$), or when one agent compensate for the unfavourable draw of the other agent ($x + y > 0$). We ran 100 independent learning simulations of $n_{\text{learn}} = 3000$ learning steps, stepsize $\Delta s = 0.1$, with systems consisting of two RS agents and two Flaky agents. In Fig. 4.1a we show the environment score $\bar{\Phi}(\boldsymbol{\phi})$ as a function of learning steps. It is important to notice that here we show the measured score of all learning steps, even the rejected ones. This is to maintain a parallelisms with the experiments, where bad steps affect the overall performance within the given learning steps. As expected, the RS agent systems show a larger improvement than Flaky, since they effectively follows the steepest ascent direction. To quantify the learning capabilities, we define the average learning rate $\tilde{\gamma}$, as

$$\tilde{\gamma} = \sum_{j}^{n_{\text{run}}} \frac{\Phi_j(\boldsymbol{\phi}^{n_{\text{learn}}}) - \Phi_j(\boldsymbol{\phi}^1)}{\Delta s \; n_{\text{run}}}, \tag{4.7}$$

which considers the average variation in score of the individual runs, averaged over all the n_{run}. We measure an average average learning rate $\tilde{\gamma}$ of 0.17 and 0.14 for RS and Flaky respectively.

To understand the difference between the two average learning rates $\tilde{\gamma}$, we need to look at how the update of memory, even in case of rejection, affects the behaviour of Flaky. Considering a simplified case of one Flaky agent on the linear environment, when a new input ϕ^j is drawn, for which $\Phi(\phi^j) < \Phi(\phi^{j-1})$, the worse environment score $\Phi(\phi^j)$ is stored in memory while the input is discarded. On the following step a new input ϕ^{j+1}, such that $\Phi(\phi^j) < \Phi(\phi^{j+1}) < \Phi(\phi^{j-1})$,

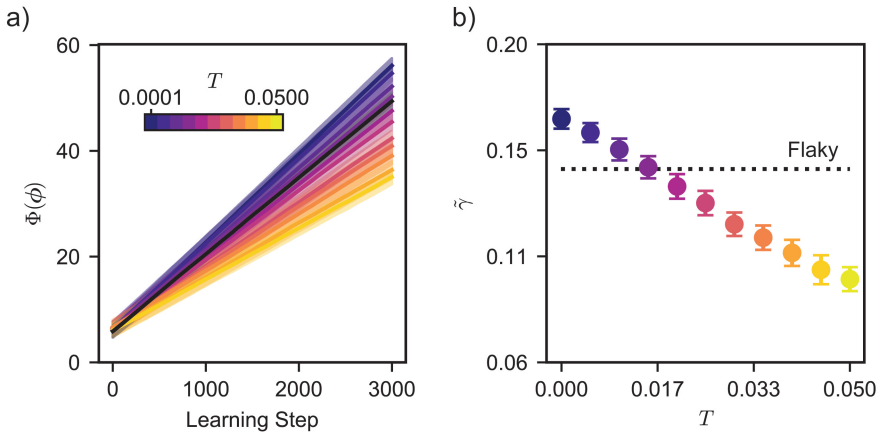

FIGURE 4.2: Temperature study of learning simulations of systems consisting of two Thermal agents on linear landscape. a, Average environment score over 100 simulations $\bar{\Phi}(\phi)$ as a function of learning steps for increasing temperature T for Thermal systems. The black curve represents the system made two Flaky agents. Shaded area represents the standard deviation. b, Average learning rate $\tilde{\gamma}$ for Thermal system with various temperatures T. The dashed line represents the average $\tilde{\gamma}$ for Flaky systems.

will be automatically accepted since the stored environment score is $\Phi(\phi^j)$. This causes the acceptance of a worse score $\Phi(\phi^{j+1})$ from $\Phi(\phi^{j-1})$ in two steps. A RS agent would reject both inputs of these two steps since it only kept $\Phi(\phi^{j-1})$ as reference score. Therefore, Flaky has a slightly lower learning rate than the RS.

As shown in Chapter 3, the flexibility of accepting worse scores granted by Flaky agents, can be seen as a useful property when dealing with noisy systems (in terms of multiple interacting agents, or varying sensor data over time), for this reason we introduce another system made of Thermal agents with temperature $T > 0$, which shows comparable performances to a Flaky system in terms of learning rate for a linear score function and $N = 2$. To select a comparable system, we vary the temperature T of the system, and ran 100 simulations with the same settings as before. In Fig. 4.2a, we show the average environment score $\bar{\Phi}(\phi)$ as a function of the learning steps, for Thermal systems with increasing temperature T. We notice that at as the temperature increases, the achievable score within the assigned learning steps decreases, since the probability of accepting worse scores increases. This can also be seen by looking at the average learning rate $\tilde{\gamma}$ in Fig. 4.2b. Based on the equivalence of the average learning rate with the Flaky systems, we selected a Thermal agent with $T = 0.016$ and include it for further analysis.

4.3.1 Scalability of the system

We next investigate the performance of systems when increasing the number of agents N. We do this for systems made of the three kind of agents introduced so far, namely RS, Thermal and Flaky. Considering the environment defined in Eq. (4.1), we perform 100 simulations of 3000 learning steps, for increasing system sizes N from 2 to 20. In Fig. 4.3a we show the average learning rates $\tilde{\gamma}$ as a function of the system size, for the three type of agents. As expected, considering the definition of the landscape function in Eq. (4.1), we find an increasing learning rate as the system size N grows. Since each agent can modify its own input with an amount bounded by the step size Δs, the system's effective step size is $N\Delta s$. Furthermore, we notice that as the system size grows, so does the divergence from the linear relationship between N and $\tilde{\gamma}$. To understand this deviation, in Fig. 4.3b we plot the average learning rate $\tilde{\gamma}$ normalized by $\sqrt{N}$ and find a nearly constant relation with the system size N.

Considering that the linear environment score is defined as the sum of the inputs, which are independently and uniformly drawn from an interval bounded by Δs, the distribution of the sums tends to be Gaussian for increasing number of agents. The distribution of the sums, which is also known as the Irwin-Hall distribution [147, 148], has a mean converging to $N/2$ and standard deviation $\sqrt{N/12}$ for $N \to \infty$.

We find that the Thermal system gradually improves its behaviour for increasing N. In fact, while the effective system step size grows with N, the temperature value is constant, so the system becomes more conservative towards input with worse environment scores, see Eq. (4.4). Effectively the system experiences a decreasing temperature T for growing N, such that the Thermal system tends to the behaviour of RS systems. For this specific environment function, we find that the performance in terms of $\tilde{\gamma}$ of Thermal and RS systems are unmatched by Flaky, even for increasing system size.

4.3.2 Noise resistance: environment noise

In this section we further alter the environment function in order to take into account environment noise, and see how the agents behaviour affects the overall system performance. In experiments, environment noise can be thought of as an unpredictable variation in the environment, e.g. robot moving on uneven surfaces, where the same input $\boldsymbol{\phi}$ leads to different scores $\Phi(\boldsymbol{\phi})$. We define an environment Gaussian noise term with standard deviation σ, to be added to the environment score $\Phi(\boldsymbol{\phi})$ at each learning step. The environment score is then defined as

$$\Phi(\boldsymbol{\phi}) = \sum_{i}^{N} \phi_i + e(\sigma). \tag{4.8}$$

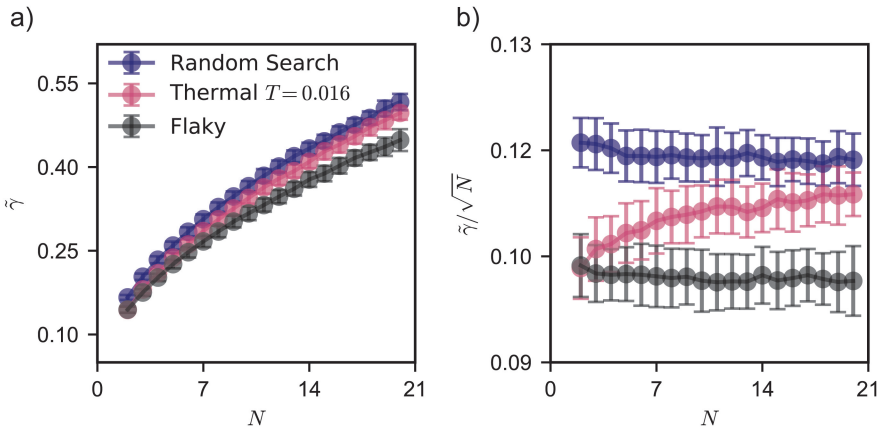

FIGURE 4.3: a, Average learning rate $\tilde{\gamma}$ as a function of system size N, for Random Search, Thermal and Flaky systems. b, Average learning rate $\tilde{\gamma}$ normalized by $\sqrt{N}$ as a function of N, for Random Search, Thermal and Flaky systems.

It is important to notice that as before, each agent will receive the same environment score, which now includes a noise term that is equal for all units. To

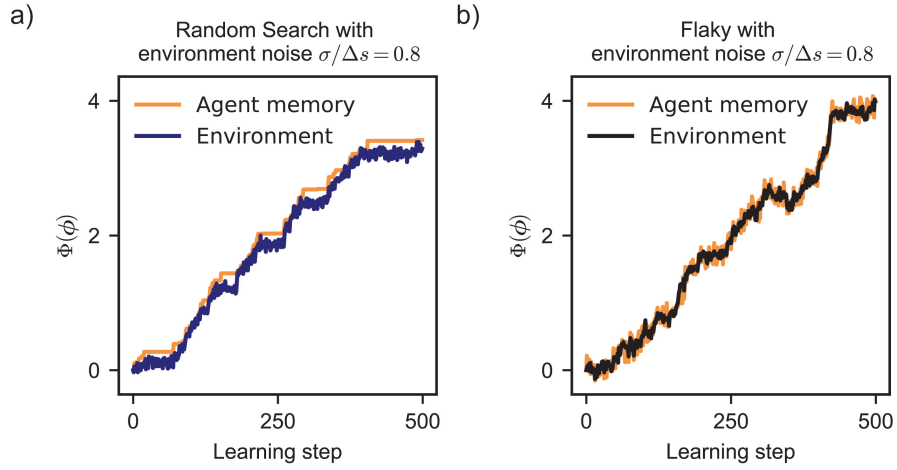

FIGURE 4.4: a-b, Effect of environment noise ($\sigma/\Delta s = 0.8$) on Random Search and Flaky systems of $N = 3$. The agent memory represents the value of environment score stored in memory including noise, which is taken as reference for the acceptance of new scores, while the environment score is the real scores without noise.

illustrate this, in Fig. 4.4a-b we show two simulations for Random Search and Flaky respectively, where a significant amount of environment noise $\sigma/\Delta s = 0.8$ is applied. The introduction of noise causes the agent memory, which is taken as reference for the acceptance of further inputs, to deviate from the effective environment score. Agents will be more likely to accept inputs whose environment score includes a positive noise term, and in contrast this will reduce the chances of accepting effectively better environment scores with a smaller noise term. This has a strong effect on the Random Search behaviour, which keeps fictitious scores in memory for many learning steps, see e.g. the horizontal regions. In contrast, by continuously updating the score kept in memory, Flaky keeps the difference between its memory and the real environment score as low as possible.

Next, we consider 100 simulations of 3000 learning steps, for increasing system size and added noise. In Fig. 4.5 we show the average learning rates $\tilde{\gamma}$, considering the real variation in score at the net of the noise, as a function of the applied noise standard deviation σ for the three system types. As also seen in Fig. 4.3a, we find an increase in learning rate as the system size grows. Importantly, we observe a decreasing trends for all sizes and agent types as the applied noise increases. The fact that higher system sizes perform better than smaller ones for the same amount of noise is expected, since the relative magnitude of applied noise gets consistently smaller as the system size N increases, see Eq. (4.8). Despite the fact that the Flaky systems show a lower absolute learning rate in the absence of noise, if compared with both RS and Thermal systems, it shows a considerable higher $\tilde{\gamma}$ for large applied noise. We can better estimate this by considering the relative difference between the average learning rate of the Flaky systems with respect to the RS and Thermal systems, see Fig. 4.6a-b respectively. We find that for an amount of added noise $\sigma/\Delta s \geq 0.45$, Flaky shows an average learning rate higher than RS for all system sizes. The same can be observed in relation with Thermal for $\sigma/\Delta s \geq 0.6$. The improvements, which is more distinct for smaller system size, reach values up to $30 - 35\%$. In Fig. 4.6c, we show the relative difference between the average learning rate of Thermal and Random Search systems, where we see that the use of temperature T can improve the learning rate of all system sizes for intermediate applied noise.

4.3.3 Noise resistance: agent noise

The environment noise studied in the previous section is not the only type of noise the systems could experience. In fact, while environment noise does not alter the overall decision making of the individual agents, since the score function is the same across the system, adding different noise terms to the environment score each agent receive can lead to agents differently accepting or rejecting their inputs. This is in general what we expect in experimental settings, where knowledge about the environment is fully decentralised. As a consequence, the agent noise can be

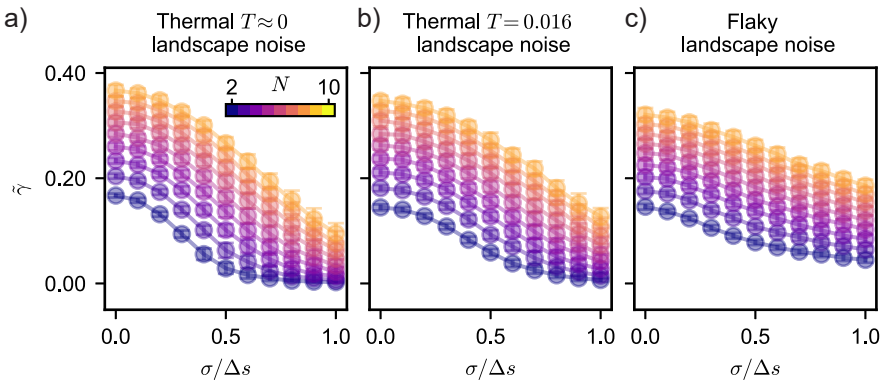

FIGURE 4.5: Average learning rate $\tilde{\gamma}$ as a function of the applied environment noise standard deviation σ, for different system sizes N. a, Average learning rate relative to Random Search systems. b, Average learning rate relative to Thermal systems. c, Average learning rate relative to Flaky systems.

thought of as experimental noise, uneven score across the system or a measuring delay between different agents. We model the agent noise by defining multiple environment scores $\Phi_i(\boldsymbol{\phi})$ as a function of the common real environment score $\Phi(\boldsymbol{\phi})$ (see Eq. (4.2)), as defined as

$$\Phi_i(\boldsymbol{\phi}) = \Phi(\boldsymbol{\phi}) + e_i(\sigma), \qquad \forall i \in [2, .., N] \tag{4.9}$$

where $e_i(\sigma)$ is a Gaussian noise realization of standard deviation σ, different for each agent. We perform 100 simulations of 3000 learning steps, for increasing system size and added noise. In Fig. 4.8 we show the average learning rate $\tilde{\gamma}$ as a function of the applied noise standard deviation σ, for different system sizes. Even thought the magnitude of applied noise is the same as in the previous section, the fact that agents now do not take coordinated decision of their inputs, lead to a sharper decrease in performance with respect to the case of environment noise. An example of qualitative different behaviour can be seen in Fig. 4.7, where we consider two individual runs for Random Search and Flaky systems of $N = 3$, where the applied noise is $\sigma/\Delta s = 0.8$. In both cases all agents keep different scores depending on the noise received, which for increasing noise magnitude cause a deviation between the units and the representation of the environment kept in memory. The decrease in performance can also be seen in Fig. 4.9 where we plot the relative difference in average learning rate $\tilde{\gamma}$ between Flaky and the two other systems. In the case of agent noise, we find that Flaky performs better than RS for an added noise of $\sigma/\Delta s > 0.30$, irrespective of the system size. In relation to the Thermal systems, we find this transition for $\sigma/\Delta s > 0.4$.

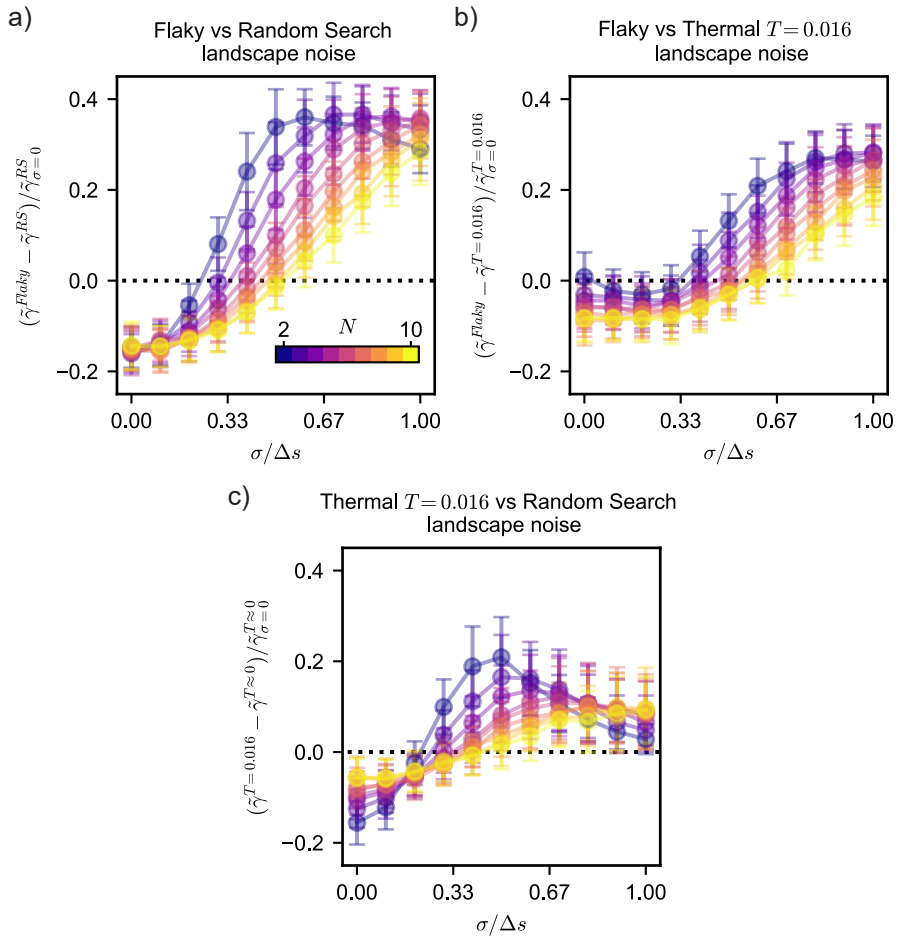

FIGURE 4.6: Relative difference between average learning rate $\tilde{\gamma}$ for system types couples, as a function of the applied environment noise standard deviation σ, for different system sizes N. Results are normalized with respect to the noiseless case. a, Relative difference between Flaky and Random Search. b, Relative difference between Flaky and Thermal. c, Relative difference between Thermal and Random Search.

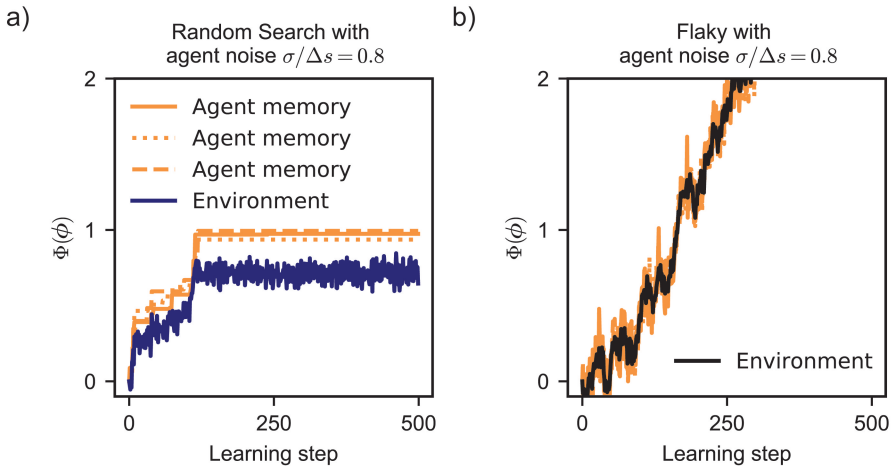

FIGURE 4.7: a-b, Effect of agent noise ($\sigma/\Delta s = 0.8$) on Random Search and Flaky systems of $N = 3$. The agent memories represent the values of environment score stored in memory including the noise terms different for each agent. These are taken as reference for the acceptance of new scores, while the environment score is the real scores without noise.

4.3.4 Optimal Temperature for a given amount of noise

In the previous sections we have shown that in a linear and noiseless environments the performance of a Random Search system is unmatched. This is expected since this type of agent never accepts worse scores. Differently, the introduction of noise requires the system to be more adaptable, and the adoption of a temperature $T > 0$ can help improve the learning rate $\tilde{\gamma}$. While we have so far determined the temperature of the Thermal agents based on simulation with two agents and no noise, here we investigate whether there is an optimal temperature T for given system sizes and amount of noise σ.

We run 100 simulations with the Thermal agent systems of size $N = 2, 5$ and 8 with increasing temperature T, by considering agent noise of standard deviation σ. In Fig. 4.10a we show the average learning rate $\tilde{\gamma}$ as a function of the applied noise for $N = 8$, for various temperature T. We see that for low values of noise, the highest learning rate is achieved with very low temperature T. On the other hand, for larger noise, higher temperature values lead to improved learning rate $\tilde{\gamma}$ if compared with $T = 0.0001$. This can better be seen in Fig. 4.10b, where we plot the average learning rate $\tilde{\gamma}$, normalized by the value of $\tilde{\gamma}$ for $T = 0.0001$, as a function of the temperature T for the different noise level σ. We see that all curves show a maximum value, and especially for the curves with large applied noise, the learning rate $\tilde{\gamma}$ obtained with a temperature $T > 0$ can be a few times higher than in the case of $T = 0.0001$. These maxima represent the optimal temperature values

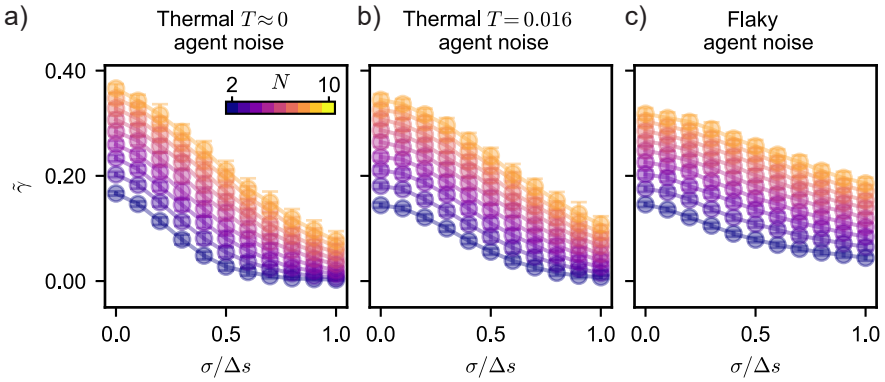

FIGURE 4.8: Average learning rate $\tilde{\gamma}$ as a function of the applied agent noise standard deviation σ, for different system sizes N. a, Average learning rate relative to Random Search systems. b, Average learning rate relative to Thermal systems. c, Average learning rate relative to Flaky systems.

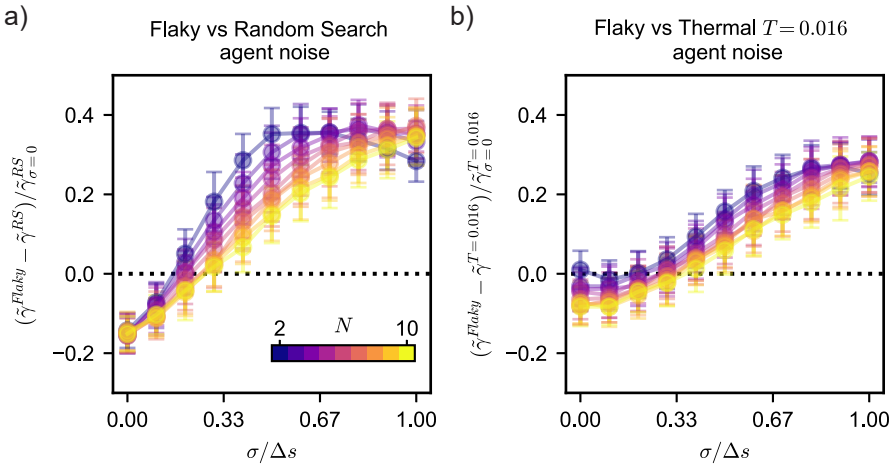

FIGURE 4.9: Relative difference between average learning rate $\tilde{\gamma}$ for system types couples, as a function of the applied agent noise standard deviation σ, for different system sizes N. Results are normalized with respect to the noiseless case. a, Relative difference between Flaky and Random Search. b, Relative difference between Flaky and Thermal.

$T_{\text{opt.}}$ for the given noise level and system size N. The same results for $N = 2, 5$ are shown in Fig. 4.20 and Fig. 4.21.

In Fig. 4.10c we show the optimal temperature values $T_{\text{opt.}}$ as a function of the applied noise, for the three investigated system sizes. We see that for all sizes, the increase in applied noise requires a larger optimal temperature $T_{\text{opt.}}$. This is expected since the increase in temperature can accommodate for larger variation in environment scores. Furthermore, the relation between $T_{\text{opt.}}$ and σ seems to resemble a linear relation, with a decrease in slope for growing N. This means that for a given amount of noise, the optimal temperature for a system made of $N = 8$ agents needs to be smaller than in the case $N = 2$, in order to obtain the highest average learning rate $\tilde{\gamma}$. This is motivated by the fact that for larger systems the relative magnitude of applied noise is smaller, see e.g. Eq. (4.8).

To quantify how the performance of Thermal systems with optimal temperatures compare with the performances of Flaky systems, in Fig. 4.10d, we show the average learning rate relative difference between the two system types. In this case, we consider the learning rates $\tilde{\gamma}$ obtained by using the optimal temperatures $T_{\text{opt.}}$ for given N and amount of noise σ shown in Fig. 4.10c. We find that Flaky systems exhibit an average learning rate $\sim 5\%$ higher than Thermal systems with optimal temperatures for agent noise of $\sigma/\Delta s \geq 0.3$. This is an important result highlighting the flexibility of Flaky agents, which do not need parameter calibration.

To conclude, Fig. 4.10c summarizes the importance of system size N and amount of noise on the choice of temperatures T for Thermal systems. While the relation we found helps to fine-tune the agent properties according to the individual cases, given the current framework and objectives where circumstances are not beforehand known, having parameters which strongly depends on working condition is a severe limitation. For this reason we decided to only consider Random Search and Flaky agents for further studies.

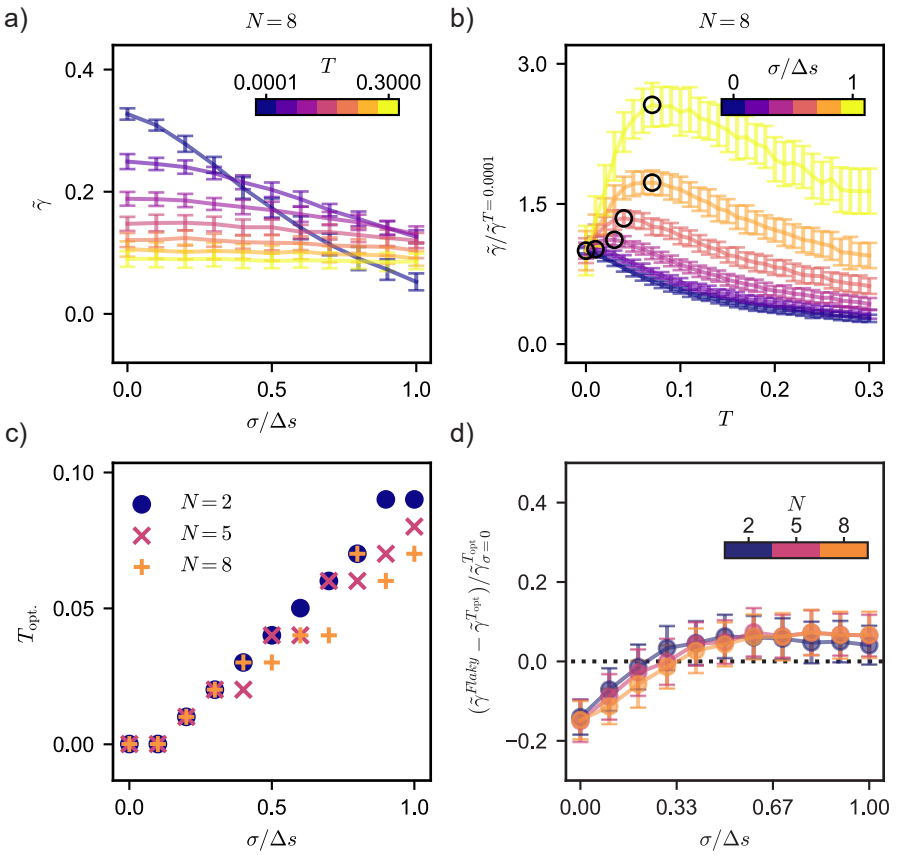

FIGURE 4.10: Optimal temperature selection for Thermal systems. a, Average learning rate as a function of agent noise applied, for different temperature values T. b, Average learning rate $\tilde{\gamma}$, normalized by the learning rate with for $T = 0.0001$, as a function of agents' Temperature T for different agent noise applied. Circles represent the optimal combination of temperature $T_{\text{opt.}}$ and highest learning rate $\tilde{\gamma}$ for the given amount of noise. In a and b some curves are not displayed for simplicity. c, Optimal temperature $T_{\text{opt.}}$ as a function of the applied agent noise, for system sizes $N = 2, 5$ and 8. d, Relative difference between average learning rate $\tilde{\gamma}$ for Flaky and Thermal with optimal temperatures $T_{\text{opt.}}$ as a function of the applied agent noise standard deviation σ, for different system sizes N.

4.4 Finding and maintaining optimal behaviour

The previous environment served well at determining the systems learning rate in a variety of circumstances. Here we consider the sinusoidal environment function shown in Eq. (4.2) for different number of agents N, in an effort to

assess the systems capability of finding and maintaining the optimal score. In this case, the environment function is periodic with respect to the inputs, and it has one maximum for $\phi_{i+1} - \phi_i = \pi/2$. Since the agents' inputs $\boldsymbol{\phi}$ are still drawn uniformly in the interval $[0, 2\pi]$, the starting conditions have a larger impact on the performance of the individual runs, and therefore we decided to run 300 simulations of 3000 learning steps.

In Fig. 4.11a-b we show the median environment scores for RS and Flaky, as a function of the learning steps, for system sizes of $N = 2$ and $N = 10$ respectively. Note that the large spread resulting from the influence of initial conditions is a function of the system size N. This is expected since the environment score is normalized by $N - 1$. However, we also find that the largest initial spread for $N = 2$ quickly reduces within the first ~ 100 learning steps. On the other hand, for larger system sizes N the initially lower spread takes an increasing number of learning steps to reduce. For example, in the case of $N = 10$, it takes ~ 800 learning steps before converging to environment global maximum.

This can be explained by considering that for large N, the probability of drawing inputs which improve the environment score of an arbitrary amount, is lower than in the case for small N. Therefore, the system's effective learning step decreases as N grows, influencing the systems' average learning rate. Given the fact that the sinusoidal environment function is bounded and nonlinear, we compute the average initial learning rate $\tilde{\gamma}$ as shown in Eq. (4.7), by only considering the first 10 learning steps. Furthermore, despite the nonlinear relation between inputs $\boldsymbol{\phi}$ and the environment, we maintain the normalization of $\tilde{\gamma}$ with respect to the step size Δs, since Δs is an agent feature which influence the choice of new inputs. In Fig. 4.11c, we show the average average learning rate $\tilde{\gamma}$ as a function of the system size N. As predicted, we find a decreasing trend for $\tilde{\gamma}$ as a function of N for both systems.

To assess the systems' capability of reaching the optimum value, in Fig. 4.12a we show the average environment score value $\bar{\Phi}_{Eq.}(\boldsymbol{\phi})$ at equilibrium, found by averaging the last 300 learning steps, as a function of the system size N. We find that both Random Search and Flaky reach values $\Phi(\boldsymbol{\phi}) \approx 1$. While Random Search systems seems not to be affected by the increase in size, Flaky shows a decreasing trend as N grows. This is due to the fact that Flaky agents have some probability of accepting worse inputs in two consecutive steps. Differently, RS agents only accept better inputs. For the same reason, for Flaky we also find an increase in spread between individual simulation's equilibrium values.

Besides being able to find the environment maximum, an in important feature of the learning algorithms is to be able to maintain good scores. To assess this property we run 300 simulations of 3000 learning steps, where the initial inputs $\boldsymbol{\phi}$ are seeded on the environment maximum, such that $\phi_{i+1} - \phi_i = \pi/2$. In Fig. 4.12b we show the average environment score value $\bar{\Phi}_{Eq.}(\boldsymbol{\phi})$ at equilibrium, again found by averaging the last 300 learning steps. Here we observe the same behaviour

seen in the case of random initial inputs ϕ. We conclude that on such smooth environment function the Random Search system performs the best in terms of convergence to optimum and ability to maintain it. On the other hand, Flaky systems maintain equilibrium scores close to the environment maximum, which are dependent on the system size N, and not on the initial inputs.

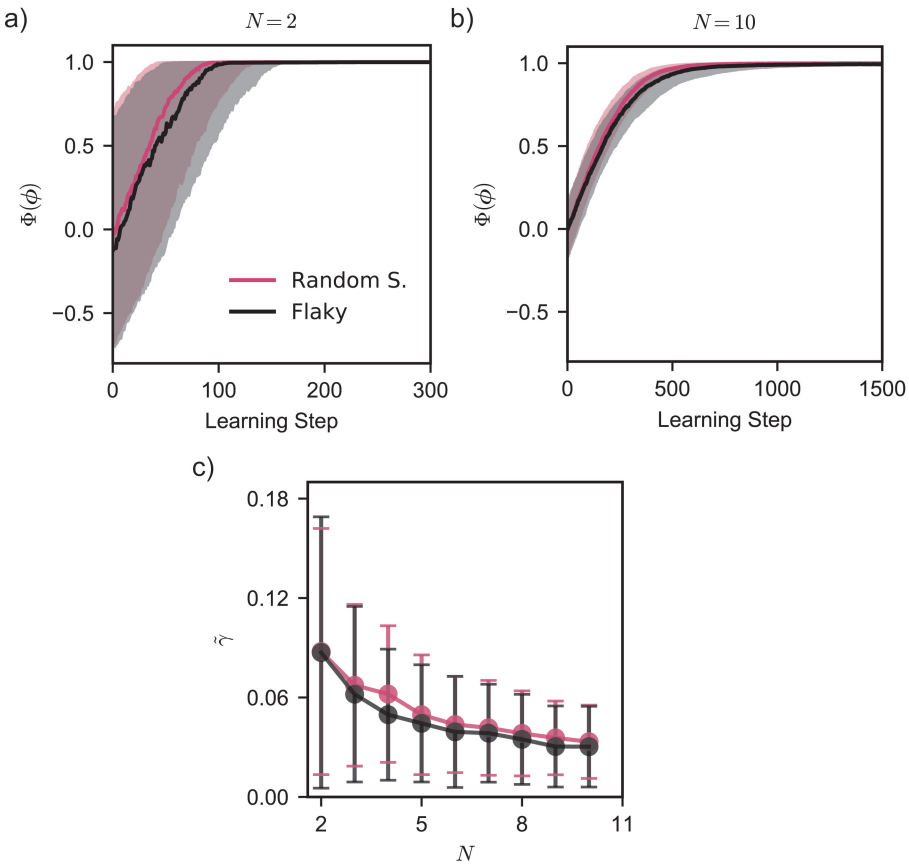

FIGURE 4.11: Learning behaviour on sinusoidal environment. a-b, Median environment score over 300 simulations $\bar{\Phi}(\phi)$ as a function of learning steps, for Random Search and Flaky systems for $N = 2$ and $N = 10$ respectively. Shaded area represents central distribution of results between the 25^{th} and 75^{th} percentiles of all simulations results. c, Average learning rate $\tilde{\gamma}$ as a function of system size N for Random Search and Flaky systems.

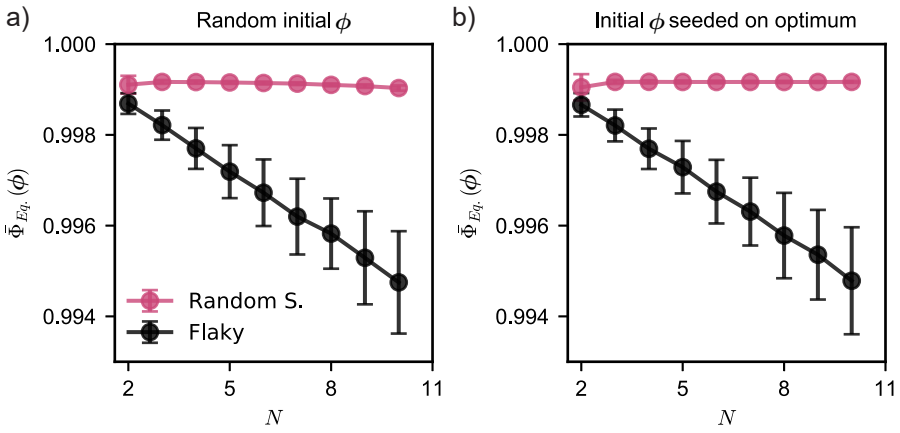

FIGURE 4.12: a-b, Environment score equilibrium values as a function of the system size N for different starting conditions. The equilibrium values are calculated by averaging the last 300 learning steps. The central points represent the median values over the individual runs, while the extremes represent the 25^{th} and 75^{th} percentiles.

4.4.1 Noise resistance: agent noise

In this section we consider the sinusoidal environment function with different noise terms between different agents. This is the type of noise deriving from local measurements we expect in experiments. Also in this case we model the agent noise by defining multiple environment scores $\Phi_i(\phi)$, function of the real environment score $\Phi(\phi)$ shown in Eq. (4.2) and a noise term $e_i(\sigma)$, where σ is the noise standard deviation.

In Fig. 4.13a-b we show the the average environment score value $\bar{\Phi}_{Eq.}(\phi)$ at equilibrium, found by averaging the last 300 learning steps, for Random Search and Flaky systems, when simulations are initialized with random initial inputs ϕ. Note that these results show the effective score the system would exhibit for the tried inputs in absence of noise. In both cases we find that the systems' equilibrium scores start to deviate from the optimal score, showing a decreasing trend for increasing noise. Importantly, the system performance for small N are less sensitive to noise. This was not the case for the agent noise in the linear environment, where small systems were more affected by noise. This can be explained by two factors. First, to have the sinusoidal environment score bounded, we normalize it by $N-1$. This maintains the ratio between score and noise constant for different system sizes N. Second, as also shown by the decrease in learning rate for larger N in Fig. 4.11c, the average score variation between two consecutive steps is smaller for larger system sizes, hence a fixed amount of noise has a larger effect on large systems. For the Random Search systems we see that a

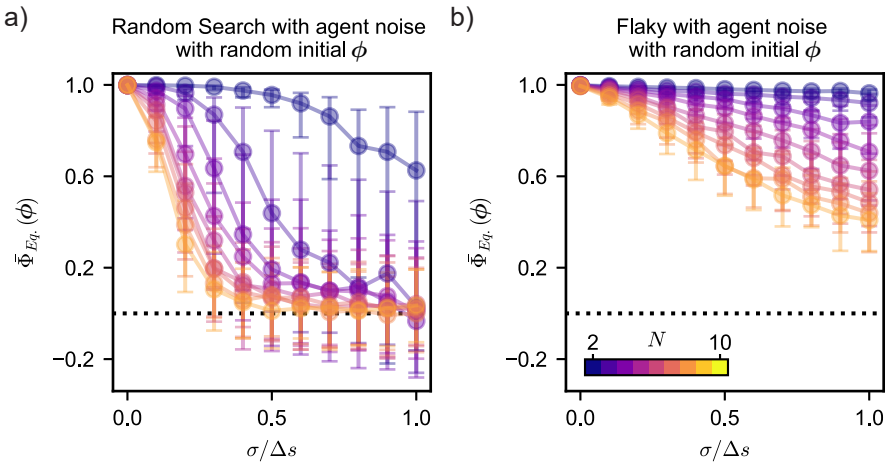

FIGURE 4.13: a-b, Environment score equilibrium values as a function of the applied agent noise and different system sizes for Random Search and Flaky respectively, with random initial ϕ. The equilibrium values are calculated by averaging the last 300 learning steps. The central points represent the median values over all runs, while the extremes represent the 25^{th} and 75^{th} percentiles.

noise $\sigma/\Delta s \geq 0.4$ is enough to prevent systems of $N \geq 4$ to improve their score, i.e. $\bar{\Phi}_{Eq.}(\phi) \leq 0.3$. This is not the case for Flaky systems, where even in the worst case scenario of $N = 10$ and $\sigma/\Delta s = 1$, the median environment score $\bar{\Phi}_{Eq.}(\phi)$ at equilibrium is around 0.5.

It is noteworthy how the distribution of results is rather skewed for Random Search (see percentiles in Fig. 4.13), whereas for Flaky the results are centrally distributed around the median values. The difference between the two cases is caused by the effect of random noise draws on the agents behaviours. In fact, while Flaky agents keep updating the score in memory at every learning step, Random Search agents update the score kept in memory only in case of the new input acceptance. As a consequence, positive noise realization leads to the acceptance of inputs which are not necessarily associated with good scores. The acceptance of unrealistic scores comprising large noise, prevents the system from accepting better scores with smaller noises, leading to a large spread in results.

To quantify the ability of the systems to maintain optimal scores in the presence of noise, we run 300 simulations of 3000 learning steps starting from the optimal input ϕ. In Fig. 4.14 we show the environment score values at equilibrium, as a function of the applied noise and system size N. Interestingly, the addition of noise causes a drift towards worse environment scores even for Random Search starting from optimal scores. This is due to the acceptance of effectively worse

scores caused by the addition of positive noise realizations. Nevertheless, the equilibrium environment score values for Random Search are very close to the environment local maximum, as these are bounded by the step size Δs and the amount of noise applied. For the Flaky systems we find a decreasing linear relation between applied noise $\sigma/\Delta s$ and $\bar{\Phi}_{Eq.}(\phi)$, and e.g. in the worst case scenario of applied noise of $\sigma/\Delta s = 1$ and $N = 10$, the equilibrium environment score value is $\bar{\Phi}_{Eq.}(\phi) \approx 0.7$. These results are in qualitative agreement with the results without the addition of agent noise discussed in the previous section.

By comparing the results for different starting configurations in the presence of noise for both Random Search and Flaky in Fig. 4.13 and Fig. 4.14, we see that the initial starting configuration has an impact on the equilibrium environment scores. While this effect is mild for Flaky systems, for Random Search systems it is very pronounced. In fact, Random search is only able to reach good environment scores from a random initial configuration for limited amount of noise and sizes (e.g. $\sigma/\Delta s \leq 0.2$ and $N < 4$), or when starting from the environment optimum.

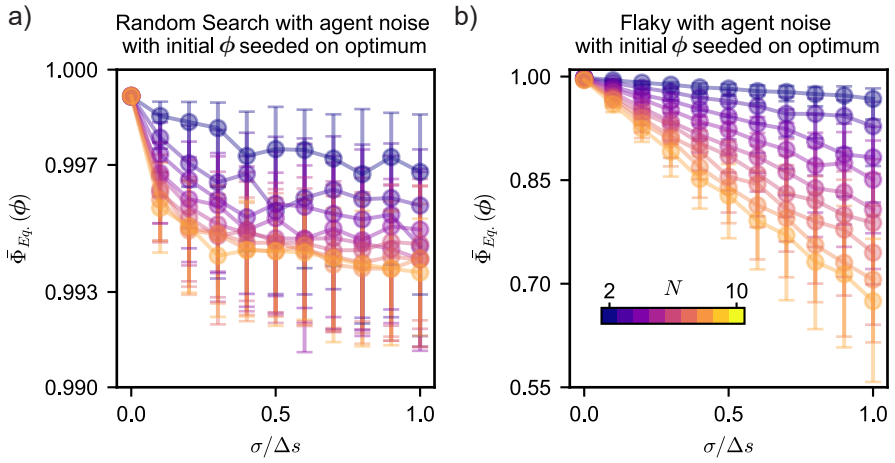

FIGURE 4.14: a-b, Environment score equilibrium values as a function of the applied agent noise and different system sizes for Random Search and Flaky respectively, with initial ϕ seeded on environment optimum. The equilibrium values are calculated by averaging the last 300 learning steps. The central points represent the median values over all runs, while the extremes represent the 25^{th} and 75^{th} percentiles.

In this section we have shown that on the smooth sinusoidal environment function, the Random Search systems perform best at reaching and maintaining the environment optimum. While the introduction of noise do not affect the system capability of maintaining optimal scores, its effectiveness at finding them is largely influenced. This causes Random Search system to stop learning, especially when N is large. Flaky system do perform well on smooth sinusoidal environments,

but most importantly exhibit a good resistance to noise, both at finding and maintaining optimal scores. While in this section we only discuss the effects of agent noise on the learning capabilities of the systems, similar results for the case of environment noise can be found in Fig. 4.22 and Fig. 4.23.

4.5 Finding and maintaining optimal behaviour in a dynamic environment

In this section we move on to the dynamic environment function defined in Eq. 4.3. Here we are interested in assessing the performance of the systems at finding the environment optimum, and their ability to maintain good scores in dynamic circumstances. This environment description in fact, resembles more closely the dynamic experimental conditions where optimal input combinations vary over time, and the system needs to continuously adapt its behaviour.

We consider the environment to vary with frequency $\tilde{\omega} = \omega/\omega_0$, where $\omega_0 = \pi/1000$ such that the dynamic environment has a period of 2000 learning steps. We start by considering Random Search and Flaky systems of sizes N up to 10, and run 300 simulation of 3000 learning steps in which $\tilde{\omega} = 1$. In Fig. 4.15a we show the median environment score $\bar{\Phi}(\boldsymbol{\phi}, t)$ for $N = 2, 6$ and 10. In the case of $N = 2$, we see that both systems can reach on average the environment maximum within ~ 100 learning steps, similarly to the static case shown in Fig. 4.11a. Once the environment optimum is reached, Flaky can continuously adapts its inputs to the changing environment, maintaining an equilibrium score very close to the maximum. On the other hand, when Random Search system reaches the optimum, the average system score begins fluctuating as a function of time. This means that the system is not able to further improve its score, and the score variations as a function of learning step are the result of the changing environment. In fact, once the Random Search system reaches the environment optimum, the optimal score in memory is used as reference for the acceptance of new phases. As the learning run continues and the landscape is changing, the inputs saved in memory are used to draw new inputs which will produce worse scores then rejected.

We can better assess the systems' adaptivity to the environment by looking at the average score at equilibrium $\bar{\Phi}_{Eq.}(\boldsymbol{\phi}, t)$, calculated by considering the last 2000 learning steps, in Fig. 4.15c. Here in fact, we see that the Random Search equilibrium score is around zero, where the small deviation from it can be attributed to the individual spread. On the other hand, Flaky can consistently maintain scores around the environment maximum up to $N = 4$, after which we see a sudden sharp decay of $\bar{\Phi}_{Eq.}(\boldsymbol{\phi}, t)$ as a function of N. Interestingly, by looking at the average score as a function of the learning steps in Fig. 4.15a, we see that the Flaky systems equilibrate at scores which are lower than the maximum reached at the beginning of the learning experiments. Overall, moving from a static to

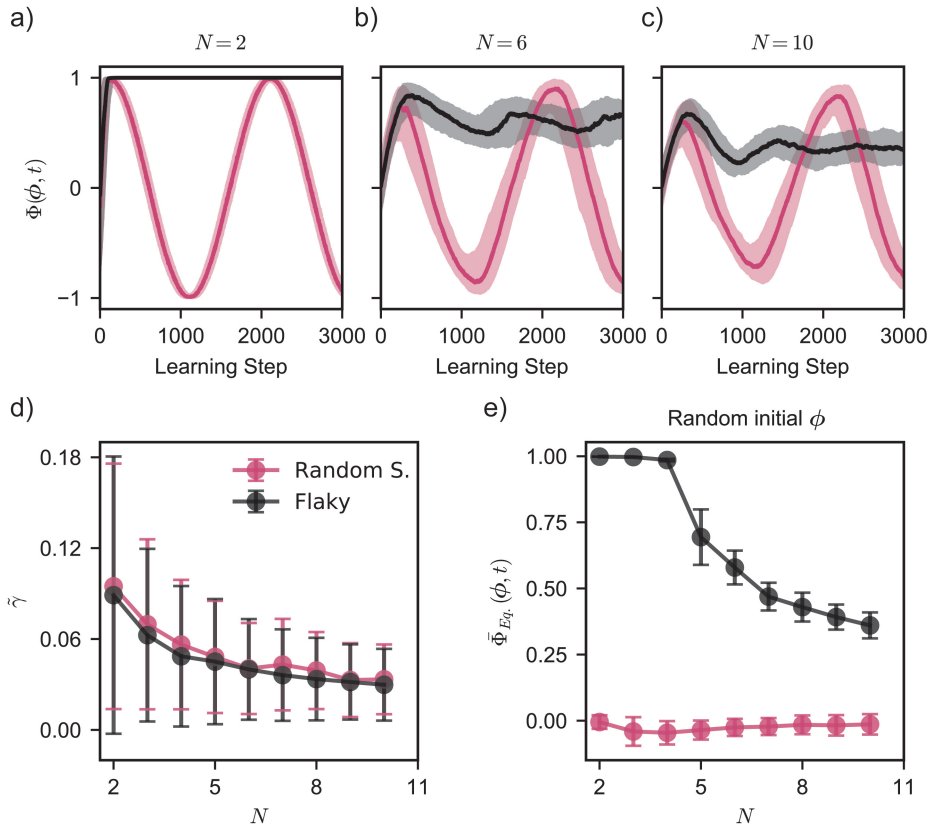

FIGURE 4.15: Learning behaviour on dynamic sinusoidal environment with random initial ϕ. a-c, Median environment score over 300 simulations $\bar{\Phi}(\phi)$ as a function of learning steps, for Random Search and Flaky systems for $N = 2, 6$ and $N = 10$ respectively. Shaded area represents central distribution of results between the 25^{th} and 75^{th} percentiles of all simulations results. d, Average learning rate $\tilde{\gamma}$ as a function of system size N for Random Search and Flaky systems. e, Average environment score at equilibrium, obtained by averaging the last 2000 learning steps, as function of system size N for Random Search and Flaky.

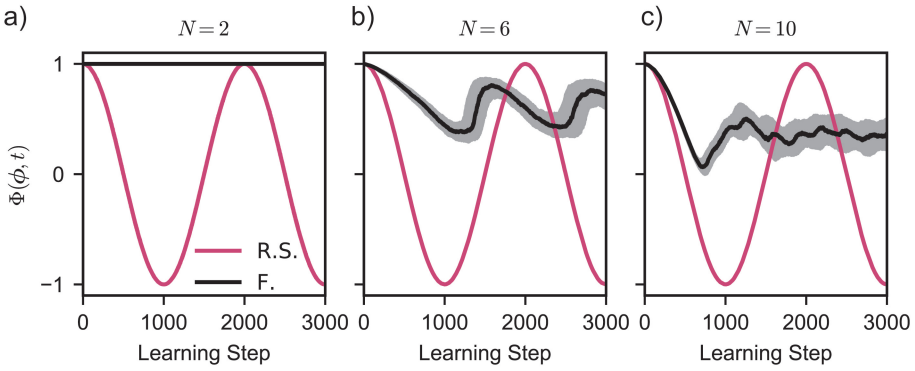

FIGURE 4.16: Learning behaviour on dynamic sinusoidal environment with initial ϕ seeded on environment optimum. a-c, Median environment score over 300 simulations $\bar{\Phi}(\phi)$ as a function of learning steps, for Random Search and Flaky systems for $N = 2, 6$ and $N = 10$ respectively. Shaded area represents central distribution of results between the 25^{th} and 75^{th} percentiles of all simulations results.

a dynamic environment does alter both systems' behaviours, while leaving the initial learning rate unchanged, see Fig. 4.15b.

To assess whether the starting inputs have an effect on the average equilibrium values, we then run 300 simulations of 3000 learning steps by setting the initial inputs at $\phi_{i+1} - \phi_i = \pi/2$, such that the systems start from the environment maximum. In Fig. 4.16 we show the median environment score $\bar{\Phi}(\phi, t)$ as a function of the learning step for $N = 2, 6$ and 10. While Flaky systems behave similarly to the random initial input case, Random Search system keep rejecting new inputs since their score is lower than the optimal score kept in memory, leading to a fluctuating score over time. By considering the average environment score at equilibrium, we found the same behaviour as for random initial ϕ shown in Fig. 4.15d. Furthermore, by running 300 simulations of 3000 learning steps with the system starting from the environment minimum, we found that and also in this case the behaviour at equilibrium is unchanged. This allow us to conclude that the equilibrium average scores do not depends on the initial inputs ϕ.

4.5.1 Rate of change dependency

So far we have seen that the introduction of a dynamic term prevents the Random Search system to maintain optimal score once it has found it, and in fact completely stops it from working. Differently, Flaky systems are able to find and maintain optimal score for system sizes $N < 5$, exhibiting a fluctuating behaviour around equilibrium scores that tends to zero as N grows. To understand how the rate

of change $\tilde{\omega}$ affects the learning capabilities and average scores at equilibrium for Flaky, we ran 300 simulations of 3000 learning steps for increasing rate of change up to a value $\tilde{\omega} = 5$. In Fig. 4.17a we show the median environment score $\bar{\Phi}(\boldsymbol{\phi}, t)$ for the Flaky systems of size $N = 3, 6$ and 10, for different rates $\tilde{\omega}$. For $N = 3$, for few of the $\tilde{\omega}$ values studied, Flaky systems can consistently find and maintain optimal environment scores. For $\tilde{\omega} \geq 2.5$ the systems show decreasing performances. Considering higher N results in gradually decreasing scores for a given $\tilde{\omega}$. This can be better seen by considering the equilibrium environment score in Fig. 4.17b.

In the case of $N = 3$ and $\tilde{\omega} \geq 5$ we notice that the equilibrium scores $\bar{\Phi}_{Eq.}(\boldsymbol{\phi})$ are much lower than the maximum reached scores. This is similar to what we observe in Fig. 4.15a for i.e. $N = 10$. To understand this, in Fig. 4.19 we show the distribution of all runs on the environment score, for different learning steps and $\tilde{\omega} = 1, 2.6$ and 5. Initially, the environment score $\Phi(\boldsymbol{\phi}, t)$ is the same for the three cases since $t = 0$ (learning steps). We can see the initial random distribution of the 300 optimizations. As the number of learning steps increases, we see a clustering of the individual runs towards the inputs associated with the environment score maximum for the given learning step. We notice three different optimization behaviours. For the case $\tilde{\omega} = 1$, the environment changes are slow enough for the individual runs to alter their inputs, and are able to follow the optimal maximum. This behaviour leads to equilibrium value very close to the environment maximum.

In the case of $\tilde{\omega} = 2.6$, the environment changes are too rapid for all the individual runs to follow, this can be seen in the streamlined distribution of runs in the direction of the environment change. While some some runs are able to follow, others are left behind, and as the environment is changing rapidly, they will harness the environment periodicity to find the optimum after $\sim 1/\tilde{\omega}$. Importantly, starting from random initial inputs is actually beneficial in finding the dynamic environment maximum, while 'being left behind' by the environment forces the individual runs to go through all environment scores before approaching the maximum. This explain why the equilibrium values are lower than the maximum values reached during the initial learning.

Finally, in the case of $\tilde{\omega} = 5$, the environment is changing so rapidly that individual runs can not really optimize their inputs. While they can locally improve their score, their performance mostly depend on the environment. This is somewhat similar to what happens to Random Search once they have found the environment maximum.

The behaviours shown for $\tilde{\omega} = 1$ and $\tilde{\omega} = 5$ are effectively two different phases describing the converging behaviour of the Flaky learning strategy. Going from small to large $\tilde{\omega}$ values, causes the system to experience a phase transition from a converging to a non-converging behaviour. This can also be seen by plotting

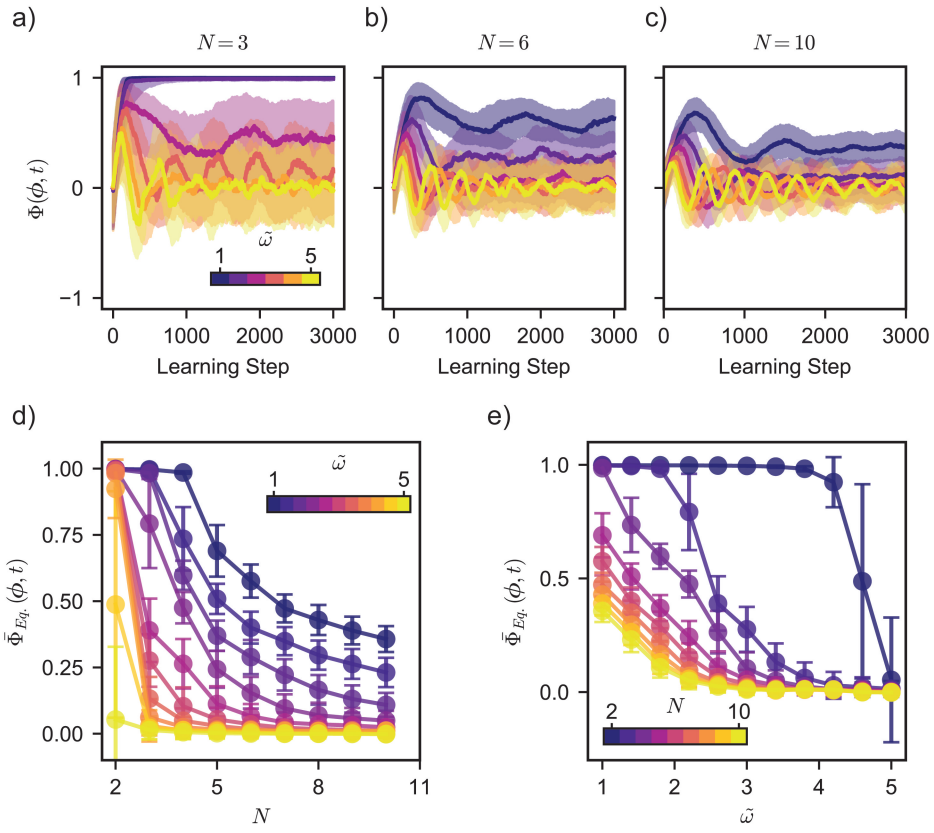

FIGURE 4.17: Learning behaviour of Flaky system on dynamic sinusoidal environment with various $\tilde{\omega}$ with random initial ϕ. a-c, Median environment score over 300 simulations $\bar{\Phi}(\phi)$ as a function of learning steps for $N = 3, 6$ and $N = 10$ respectively, for different rates of change $\tilde{\omega}$. Shaded area represents central distribution of results between the 25^{th} and 75^{th} percentiles of all simulations results. d, Average environment score at equilibrium, obtained by averaging the last 2000 learning steps, as function of system size N different rates of change $\tilde{\omega}$. e, Average environment score at equilibrium as function of different rates of change $\tilde{\omega}$ for different system sizes N.

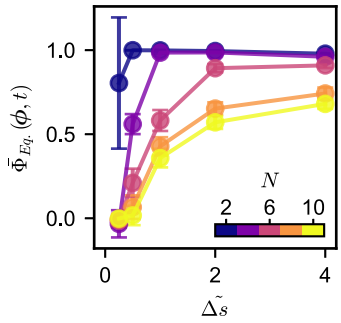

FIGURE 4.18: Learning behaviour on dynamic sinusoidal environment with $\tilde{\omega} = 1$. Average environment score at equilibrium for Flaky systems as function of different step sizes $\tilde{\Delta s}$ and system sizes N.

the average environment score $\bar{\Phi}_{Eq.}(\phi, t)$ at equilibrium as a function of $\tilde{\omega}$, see Fig. 4.17c.

The ability to follow a moving optimum is influenced by the step size Δs, since it determines the maximum variation in ϕ between two learning steps. For a $\tilde{\omega} = 1$, we run 300 simulations of 3000 learning steps with normalized step size $\tilde{\Delta s} = \Delta s / \Delta s_0$ ranging between $1/4$ and 4, where $\Delta s_0 = 0.1$ is the step size used so far for all studies. In Fig. 4.18 we show the average environment score at equilibrium as a function of the normalized step size. As expected, an increase in step size leads to a higher mobility on the environment for all N, which results in the ability of maintaining the optimal scores. Furthermore, we see a stronger dependency of $\tilde{\Delta s}$ on $\bar{\Phi}_{Eq.}(\phi, t)$ as the system size grows. This means that in order to maintain comparable performances in dynamic environments, larger systems needs to use larger step sizes.

4.6 Conclusions

In this chapter we have discussed the development of a numerical framework aimed at simulating the behaviour of multi agent systems on simplified landscapes, where the systems' interaction with the environment is arbitrary chosen. We consider agents based on the algorithms introduced in Chapter 3, and define various environment scores to assess their performances in terms of learning rate, convergence and ability to maintain optimal behaviour in static and dynamic environments, and resistance to noise.

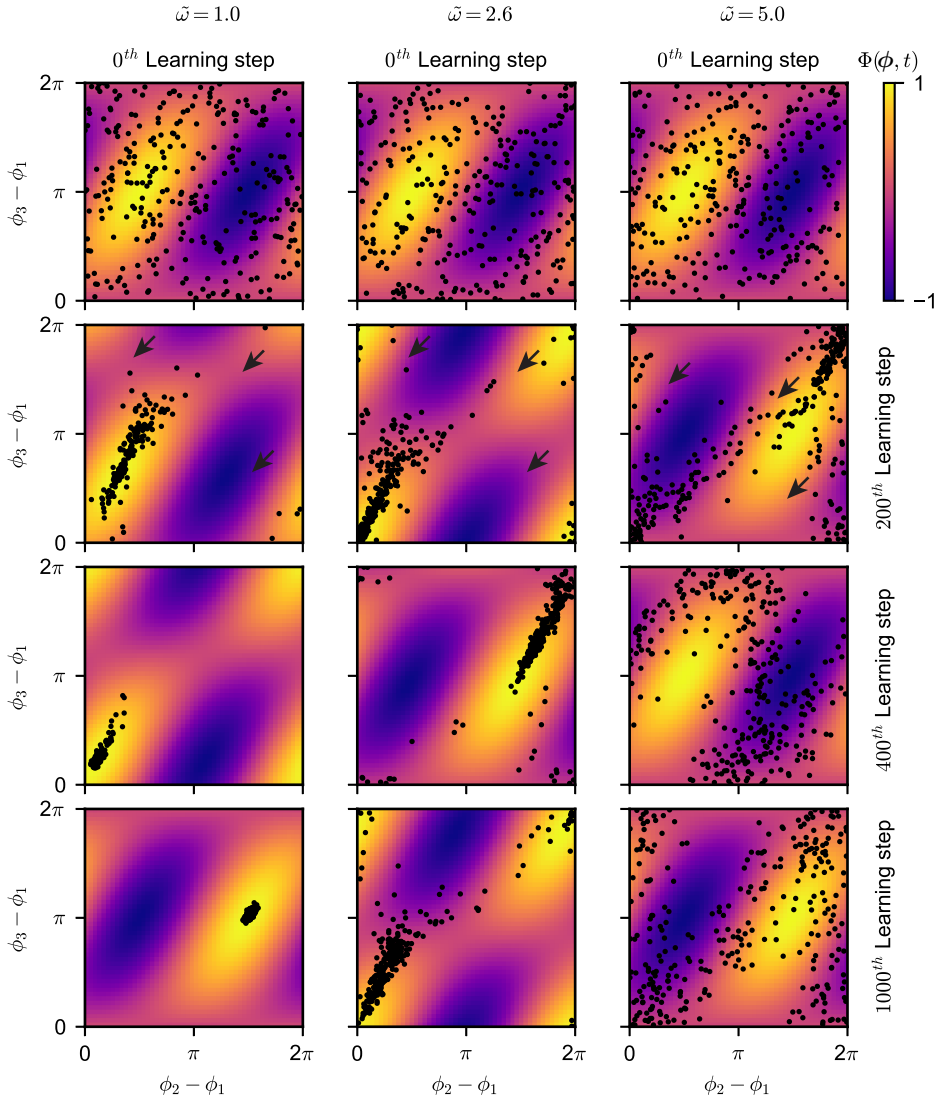

FIGURE 4.19: Learning behaviour of Flaky system of $N = 3$ on dynamic sinusoidal environment with random initial ϕ and $\tilde{\omega} = 1, 2.6$ and $\tilde{\omega} = 5$. The first rows show the simulation batches at the starting configurations, where the 300 runs (dots) are uniformly distributed on the environment. The following rows present the evolution of the simulated runs at the learning steps 200^{th}, 400^{th}, and 1000^{th} respectively. The direction of change of the landscape is indicated by the arrows in the second row.

We find that in smooth and noiseless environments the performances of Random Search systems are unmatched. This system in fact exhibits the highest learning rate towards the environment optimum, while also being able to maintain it. On the other hand, since Random Search only accept better scores, the introduction of a limited amount of noise prevents this system from learning. To this extent, the introduction of a temperature T in case of the Thermal agents, reduces the influence of noise on the learning performance. Nevertheless, for the larger amount of noise applied in this study, Flaky systems are able to maintain higher learning rates while also maintaining optimal inputs. This is due to how the memory is updated.

In dynamic environments, Random Search systems are able to improve their score until the moment they reach the landscape optimum. After that, by continuously rejecting new tried inputs, their score only depends on the environment variations. Differently, Flaky systems continuously adjust their score to the environment variations as long as the environment variations are not too rapid. For highly dynamic environments also the Flaky systems fail at improving their scores. In fact, for increasing environment variations rates, we find that the system experience a phase transition between a converging to a non converging behaviour. We find that this phase transition depends on the rate at which the environment is changing, the system size and the agent step sizes. While in the decentralised system settings discussed here the first two parameters are unknown to the individual agent, by acting on their step size, we believe that agent could be able to control the converging behaviour of the complete system in highly mutable environments.

Even though the simplified landscapes discussed in this chapter are just a few of all the possible interactions the system could experience with the environment, how the agents with this inherent dynamics respond to different situations, can potentially be understood by the results obtained here. For example, reconnecting to the experimental track changes discussed in Chapter 3, instead of a continuous change, a landscape could suddenly change in time, due to e.g. damage. While this would bring the Random Search Systems to a complete stop, it would only take two steps for the Flaky system to adjust to the new landscape morphology, and to start adapting its behaviour. Furthermore, going to a more general case, a landscape with multiple optima is another interesting test case. In this situation we expect that all systems would converge to local optima, and while Random Search would not be able to escape from it, Thermal would be able to escape given a high enough temperature T. Flaky systems would be able to escape within a time scale proportional to the optima score barriers and step size. This makes the Flaky systems autonomously adapt to the environment.

4.7 Supplementary figures

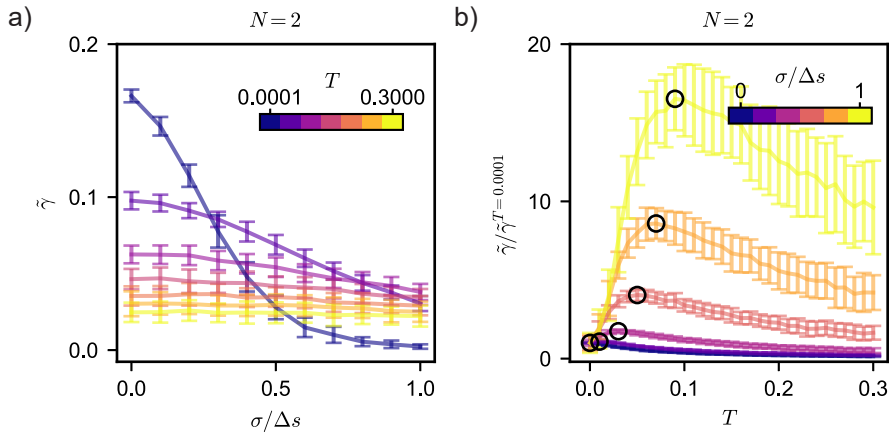

FIGURE 4.20: Optimal temperature selection for Thermal systems of $N = 2$. a, Average learning rate as a function of agent noise applied, for different temperature values T. b, Average learning rate $\tilde{\gamma}$, normalized by the learning rate with for $T = 0.0001$, as a function of agents' Temperature T for different agent noise applied. Circles represent the optimal combination of temperature $T_{\text{opt.}}$ and highest learning rate $\tilde{\gamma}$ for the given amount of noise. Some curves are not displayed for simplicity.

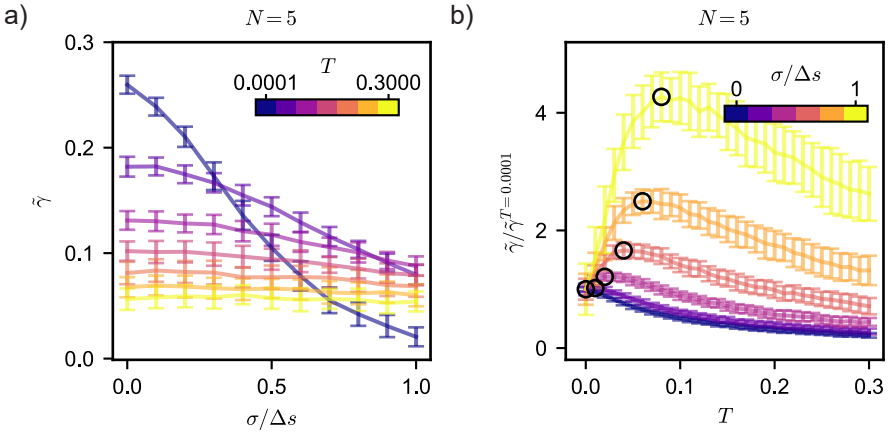

FIGURE 4.21: Optimal temperature selection for Thermal systems of $N = 5$. a, Average learning rate as a function of agent noise applied, for different temperature values T. b, Average learning rate $\tilde{\gamma}$, normalized by the learning rate with for $T = 0.0001$, as a function of agents' Temperature T for different agent noise applied. Circles represent the optimal combination of temperature $T_{\text{opt.}}$ and highest learning rate $\tilde{\gamma}$ for the given amount of noise. Some curves are not displayed for simplicity.

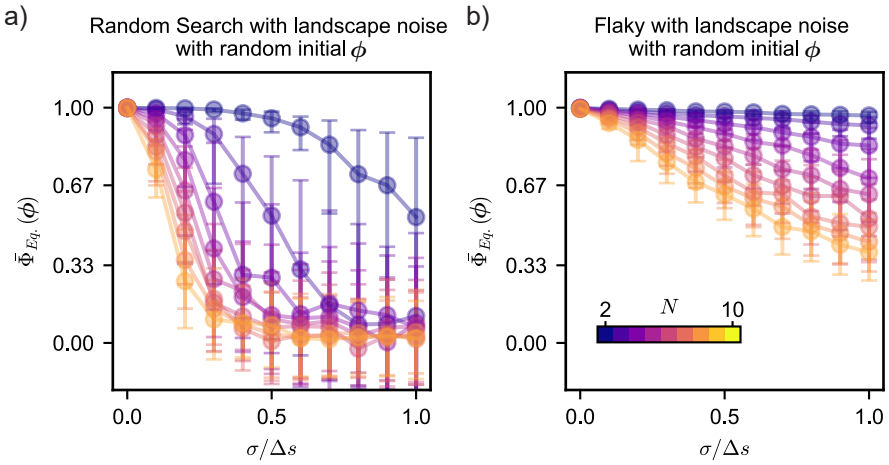

FIGURE 4.22: a-b, Environment score equilibrium values as a function of the applied environment noise and different system sizes for Random Search and Flaky respectively, with random initial ϕ. The equilibrium values are calculated by averaging the last 300 learning steps. The central points represent the median values over all runs, while the extremes represent the 25^{th} and 75^{th} percentiles.

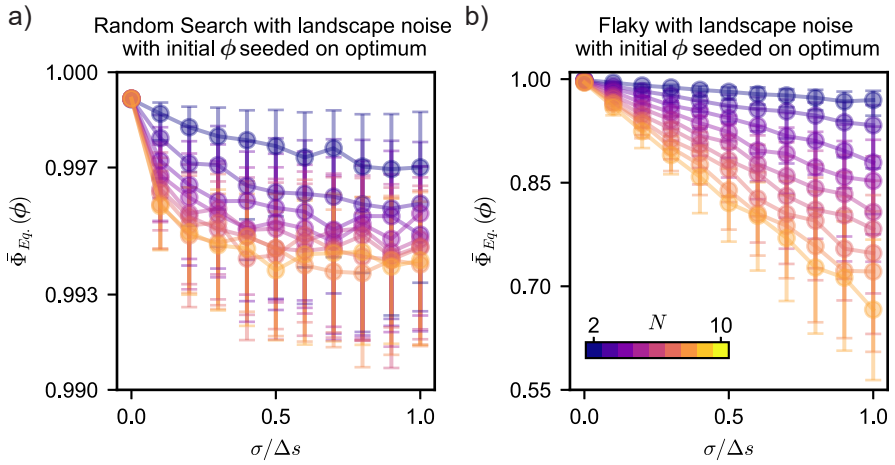

FIGURE 4.23: a-b, Environment score equilibrium values as a function of the applied environment noise and different system sizes for Random Search and Flaky respectively, with initial ϕ seeded on environment optimum. The equilibrium values are calculated by averaging the last 300 learning steps. The central points represent the median values over all runs, while the extremes represent the 25^{th} and 75^{th} percentiles.

CHAPTER 5

2D Robotic matter

In preparation as:

Oliveri, G., Schomaker, M., Overvelde, J.T.B.
Autonomous 2D robotic matter

In Chapter 3 we have introduced decentralised learning strategies for robotic systems consisting of multiple units. We have shown that a forward locomotion of the complete system can be reached, by having each unit independently adapting its behaviour based on local measurements. Despite the interesting results in terms of adaptivity and resistance to damage, the architecture of the assembled robots and the range of possible behaviours was limited to linear motion on a circular track. In this chapter, we develop a simulation framework based on mass-spring systems, and explore the performance of the Flaky learning strategy when applied to a qualitative model of a potential robot, consisting of units that can be placed and assembled in a square lattice. In spite of the increased complexity deriving from locomotion on a plane and possible rotation that the system can undergo, we find that our decentralised learning strategy is potentially suitable for the control of robotic systems consisting of different number of units, various planar arrangements, considerable damage, and has the ability of navigating complex environments with obstacles.

5.1 Introduction

In Chapter 3 we introduced a decentralised control strategy for robotic systems made of modular independent units. By means of experimental and numerical work we showed how complex behaviour can emerge from simple unit-to-unit interactions. While we looked at the locomotion of a one dimensional arrangement of active units, and show promising results in terms of scalability, adaptation to dynamic environments and damage, the system's behaviours were limited to a linear motion. In this chapter we apply the same decentralised control strategy, and extend it to robotic units that deform and can move on a plane. This allow us to investigate the role of system connectivity and shape, in-plane scalability, resistance to damage and impervious working circumstances, thereby increasing the complexity of the behaviours that needs to be learned and maintained. We do this by developing a simplified simulation framework based on two-dimensional mass-spring systems, which may serve as benchmark for the development of the actual experimental setup, whose implementation is left outside the scope of this thesis.

We consider an arbitrarily shaped bidimensional system made of active unit cells which can be connected in a square lattice. Each unit can independently interact with neighbouring units by expanding and contracting cyclically with actuation phase ϕ_i. We investigate the maximization of a local reward which leads to a global preferred behaviour, e.g reaching the highest speed in a certain direction, which can be achieved by having each unit independently alter ϕ_i, such that an overall locomotion pattern can be established. This is done by having each unit run the Flaky algorithm introduced in Chapter 3, which as also shown in Chapter 4, performs well in dynamic situations. To allow for complete modularity, the decision making within units is independent from the others, and it is only based on local measurements.

In Chapter 3 and Chapter 4 we showed that the locomotion behaviour the system is aiming to achieve is strongly influenced by the number of units belonging to the system, which modify the landscape of possible phase configurations by e.g. adding and removing maxima and minima. Moving to a bidimensional description introduces new complexity which does not necessarily derive from the increase in system size. In fact, while in the one-dimensional case the interaction between the system and the environment can be considered static, unless it is purposely made dynamic by e.g. damaging some units or having friction that depends on location, in the two-dimensional case, the system can rotate as a result of the exploration of different ϕ_i, altering the system's interaction with the environment. The performance of possible behaviours are then not only dependent on the current system state, but also on its history, effectively making the environment highly mutable. By studying the performance of our decentralised learning strategy on bidimensional systems showcasing e.g. more complex situations, it allow us to

further understand what the limitations of our decentralised control strategy are, and help us define critical factors to improve autonomous robot behaviours. Most important is for us to gain additional evidence that the Flaky algorithm can be used in more complex cases. Therefore, the model used here does not have to be precise as long as it introduces the additional complexity rising from 1D to 2D motion.

5.2 Bridging 1D to 2D

In order to extend our study on the decentralised control strategy from one to two-dimensional arrangement of active units in simulations, some of the assumptions made for our previous model need to be revised. In Chapter 3 we model the one-dimensional modular robot with a mass-spring system, where each active unit is represented by a mass and an adjacent active spring. While this description could be extended to a two-dimensional square lattice, i.e. by representing each active unit with a mass and two orthogonal springs connecting to the neighbouring units, the mass rotation and shear between units would not be taken into account. Additionally, in order to include the torsional stiffness units will experience between each other in experiment, we model each active unit with four masses (or nodes) connected by springs arranged on a square layout as shown in Fig. 5.1a. Each unit also includes two diagonal active springs which provide the actuation resulting into a bidimensional unit expansion.

Multiple units can be assembled together to form a single-body system by sharing the overlapping masses. For example, for the five active unit system shown in Fig. 5.1a, which we refer to as star configuration, the central unit is connected to the neighbouring four units by sharing all of its own nodes.

The actuation of the two active springs belonging to the same unit is synchronous, and similar to the one-dimensional implementation in Chapter 3. It is modelled by having springs with time-varying preferred spring length[1] $l(t)$, see Sec. 5.10.1. Furthermore, to describe the frictional behaviour between masses and environment, we adapt the friction model used in Chapter 3 to the 2D case, see Sec. 5.10.1 for a detailed description. Note that according to this model, the masses belonging to the same unit are now weakly coupled (via soft springs), and can move independently from each other. This causes that masses experience e.g. friction forces, according to their individual relative velocity with respect to the environment. While this implementation is different than real experimental conditions, if we e.g. adapt the experimental platform described in Chapter 3

[1] In the 1D case, $l(t)$ was modelled after the inflation and deflation cycles of the pneumatic actuator used in experiments. Despite the fact that the units in the 2D experimental setup will actuate by using servo motors and different actuation as a function of time can potentially be used, in order to qualitatively reproduce the previous results, we maintain a similar description of $l(t)$ to the one used in Chapter 3.

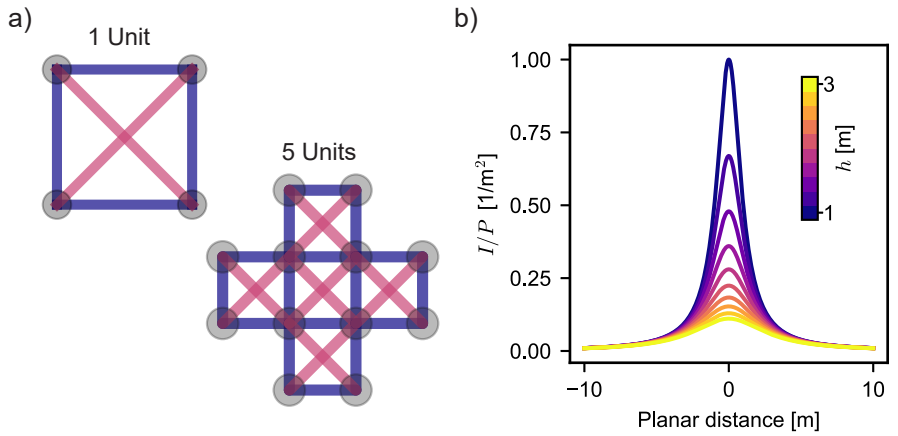

FIGURE 5.1: a, Model schematic of one individual unit, and a five active unit system in a start shape configuration. One unit consists of four masses and six springs. When multiple units are assembled together, masses in common are shared. b, Measurable light intensity as a function of the planar distance of detection, for different light source heights with respect to the horizontal plane.

to allow for bi-dimensional assembly, this model serves as a benchmark to asses whether the learning strategy is suitable for such problems.

The transition from 1D to 2D also requires the redefinition of how the environment sensing is used to compute the score function each unit is optimizing. In Chapter 3, the units try to maximize the variation in traveled distance within a learning step, this was experimentally achieved by measuring the displacement with an optical motion sensor. Since the one-dimensional system is placed on a circular track, the motion of the system is confined to the track circumference, and the displacement can be either positive or negative according the chosen winding. In the two-dimensional case, to be able to account for change in direction, due to e.g. rotation of the system, we need to consider an absolute measurements of distance. From a practical point of view, the optical motion sensors are not able to capture units' rotation, and the units will then have problems in determining the orientation. For this reason, we envision to replace the motion sensor with a light sensor that is able to measure light intensity, and is therefore able to asses the approaching to a single light source. As a consequence, while the 1D system aims at maximizing the change in distance (per unit cycle) in a fixed direction, the 2D system aims at maximising the change in light intensity per unit cycle. Both definitions are equivalent to the search for the highest velocity. The new score function to maximize can then be written as

$$\Delta I = I' - I, \tag{5.1}$$

where I' and I are the brightness measurements of the current step and the step saved in memory respectively. Importantly, since here we only consider the Flaky algorithm, where the new score is always stored in memory, the values I' and I are the brightness measurements of two consecutive learning steps.

Considering a light source of power P, the light intensity measured at a distance r from the light source is given by

$$I = \frac{P}{4\pi r^2}, \tag{5.2}$$

where the denominator represent the surface of the sphere of radius r and centre on the light source. In Fig. 5.1b we show the measurable light intensity, as a function of the linear distance from the source on the plane, for different light source heights h.

The nonlinear relation between planar distance and light intensity requires the modification of the Monte Carlo optimization scheme used in Chapter 3, where the Flaky algorithm uses a temperature T to tune the acceptance probability of steps which do not improve the saved score. As shown in Chapter 4, the value of T is highly susceptible to the environmental conditions and system size. In the two-dimensional case, having a fixed temperature would allow different behaviours as a function of the absolute distance to the light source. As an example, when the system is far from the light source, for a disadvantageous displacement (for which $\Delta I < 0$), the decrease in brightness will be small and it can potentially be accepted. On the other hand, when close to the light source, the same disadvantageous displacement will have greater change in brightness and the step will be surely rejected. Since we use the light sensor as a measure of the traveled distance, which is the quantity we are interested in maximizing, we only consider a Flaky algorithm with no temperature.

Finally, contrarily to the 1D simulation setup discussed in Chapter 3, where each learning step consisted of a new simulation with the assigned ϕ_i, in the 2D case we implemented a continuous learning simulation framework. According to this implementation each learning experiment is one simulation consisting of an assigned number of learning steps, and phase changes are implemented by altering the duration of actuation cycles with respect a common time reference. See Sec. 5.10.1 for further details. This new implementation effectively replicates the experimental settings discussed in Chapter 3, and is needed to perform simulations with varying objective (e.g. resulting from system rotations during learning) and diverse environments.

5.3 Validation of 1D results

Before proceeding with more complex environments and system shapes comprising many active units, we first qualitatively validate the current 2D model with

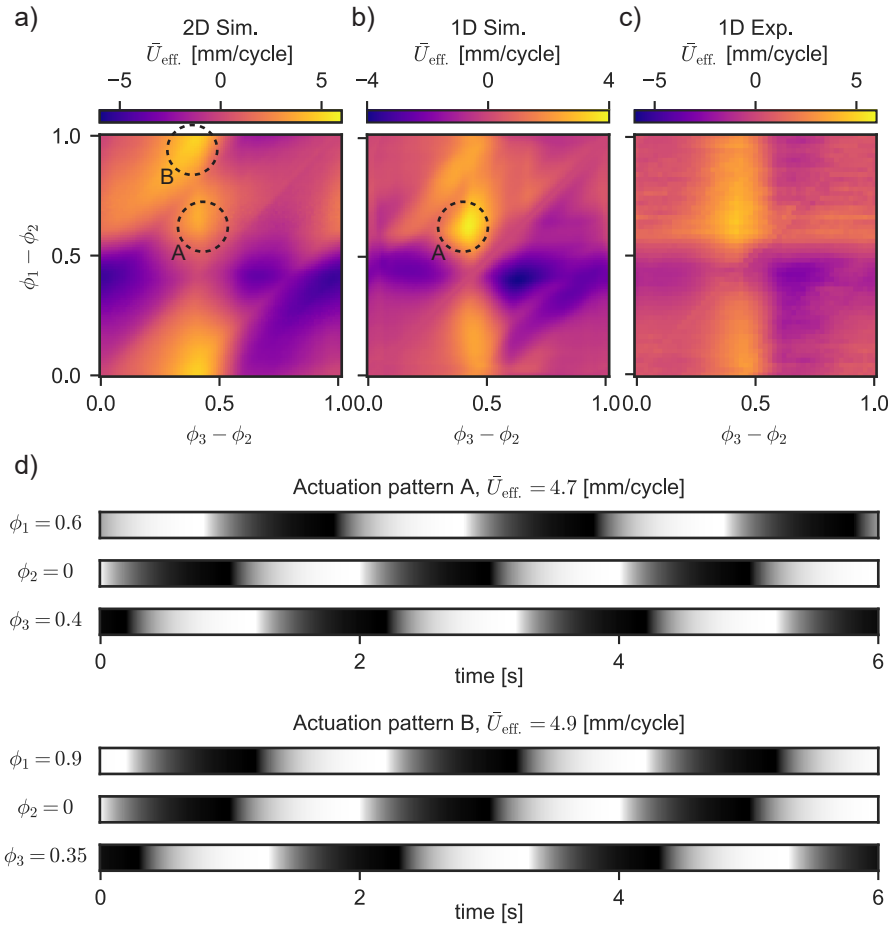

FIGURE 5.2: a-c, System average effective velocity as a function of phase combinations for 2D model, 1D model and 1D experiment respectively. d, Representation of optimal phase combinations relative to point A and B in Fig. 5.2.

the model and experiments presented in Chapter 3. We do this by considering the performance of a three active unit system for all combination of phases ϕ_i. Since the system now moves on a 2D plane, we will measure its performance by considering the average effective system velocity $\bar{U}_{\text{eff.}}$ in the direction of the light. This is computed by averaging the euclidean distance variation ($U_{\text{eff.},i}$) between the individual units and the light source during a learning step. In this specific case, we place the system oriented radially with respect to the light source, and while the system actuates in 2D, because of its symmetry, it will only move along

the radial direction, moving towards the light source. Note that we can only make a qualitative comparison, since the underlying actuation and deformation of the 2D system is slightly different.

In Fig. 5.2a-b we show the average system velocity $\bar{U}_{\text{eff.}}$ for the 2D and 1D model respectively, and in Fig. 5.2c we show the experimental results of the 1D case. Note that here we are adopting a symmetric notation for the system phases on the axis, as we expect a symmetric response between forward and backward locomotion. Apart from a qualitative similarity, we find that the maxima (and minima) found in the 1D model and the experiments are present in the 2D model results, see point A in Fig. 5.2a-b.

Furthermore we also find the emergence of another maximum (and minimum) for $\phi_3 - \phi_2 \approx 0.35$ and $\phi_1 - \phi_2 \approx 0.9$ ($\phi_3 - \phi_2 \approx 0.9$ and $\phi_1 - \phi_2 = 0.4$), see point B in Fig. 5.2a. We believe that the new maximum can be associated with the combination of larger actuation stroke (preferred spring length $l(t)$), and the increased stiffness of the units which can now more easily propagate their actuation by actuating together, see actuation pattern of B. In the 1D experiments this locomotion strategy would not work because of the high compliance of the actuators, which bend when compressed. In general we find a very good qualitative agreement, and the difference in absolute values can be related to the choice of actuator stroke, springs stiffness, and viscous term, see Sec. 5.10.3.

5.4 Learning to move towards a light source

To verify whether the proposed decentralised learning strategy based on local light intensity measurement can actually achieve the locomotion of 2D systems towards a light source, we perform learning simulations where the light source is placed at different distances from the system starting point. We consider the bidimensional star shape configuration shown in Fig. 5.1a, and run 40 simulations of 300 learning steps for three different light source locations. Specifically, we place the system at the origin of our coordinate system, and consider the light source at increasing distances along the positive x axis, at $0, 2$ and 5 m.

To display the qualitative behaviour, in Fig. 5.3 we show the trajectories of two simulation runs in the case of a light source placed at 2 m, where the depicted system positions are relative to the 50^{th}, 200^{th} and 300^{th} learning steps. In one case the system move directly towards the light source, while in the second case the system first moves perpendicular to the direction of the light, after which it makes a sharp turn towards it. We also notice the alternation of relatively straight trajectories with sharp bends. This can be associated with the exploitation of relatively good phase combinations in the straight part, and exploration of different phase combination in the bends. The transition between exploitation and exploration can also be seen by the overshooting trajectories, especially visible in the proximity of the light source. Here in fact, good locomotion patterns lead to

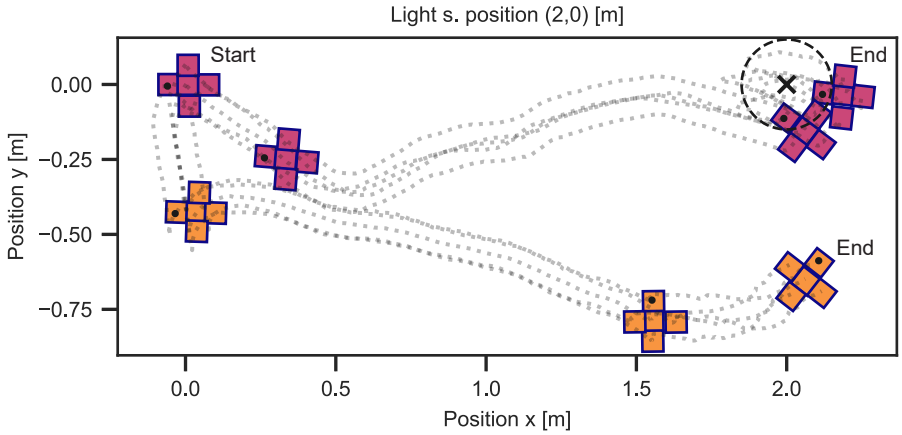

FIGURE 5.3: Learning experiment trajectories for two simulated runs in a start shape configuration, with light source placed at $(2, 0)$ m. The screenshots represent the position of the systems at the start of the run, at 50^{th} and 200^{th} learning steps and at the end of the run. The dashed circle represent the circular area of $r = 0.15$ m around the light source.

greater speeds towards the light, which cause the system to surpass the target and increase the relative distance to it, causing the system to look for different patterns to move in the opposite direction. As consequence, the systems roam around the light source. In the other case, although the system does not get as close to the light source as the first example, it considerably reduce its distance to the target, and the overall behaviour alternating straight paths with bends is similar. Both examples show how the complexity of the task execution has considerably increased moving from the one-dimensional to the two-dimensional description.

Considering all the 40 simulation runs, in Fig. 5.4a we show the system's centre position during the learning experiments for the different light source distances. We find that all simulation runs are able to learn to move towards the light, while also displaying some displacement along the y direction. The trajectory spread along the vertical direction can be associated with the exploration of sub-optimal locomotion patterns, mostly occurring at the beginning of the learning experiment. After the initial learning steps in fact, the trajectories seem to straighten toward the direction of the light source as also shown in Fig. 5.3.

In the case of a light source placed at 2 m, the system is able to approach the target and roam around it within the assigned learning steps. In fact, we find that 48% of our simulations reach the proximity of the light source (systems' centre distance from the light source < 0.15 m) within the assigned learning steps.

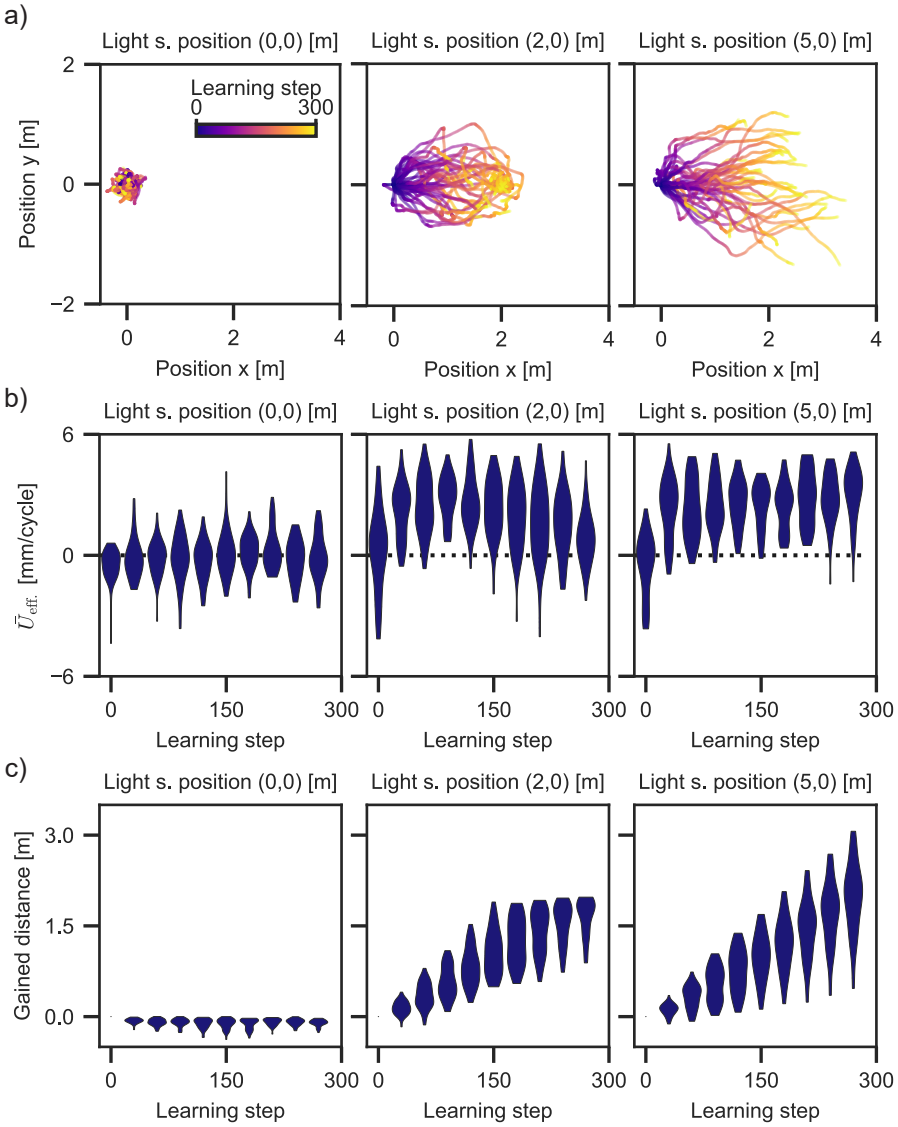

FIGURE 5.4: Star shape configurations results with light source position at $0, 2$ and 5 m along the positive x axis, where each light source distance comprise 40 runs. a, System trajectories as a function of the learning step. b, Distribution of system average effective velocity $\bar{U}_{\text{eff,}}$ as a function of the learning steps. c, Distribution of system gained distance as a function of the learning steps.

This percentage goes up to 70% if we consider a threshold distance of 0.4 m. As mentioned before, once the system approach the light and surpass it, it will exhibit a negative effective velocity $\bar{U}_{\text{eff.}}$ (see the distribution across negative values around ≈ 150 learning steps in Fig. 5.4b). As a result, the system will need to learn different locomotion patterns towards the opposite direction. The fact that the system change direction of motion can be seen by the non-diverging gained euclidean distance in Fig. 5.4c.

This is an important difference between the 1D and 2D systems. In fact, while the 1D system aims at maximizing the velocity in a fixed direction searching for a fixed actuation pattern, the 2D system aims at maximizing the velocity in the direction of the light. Since the relative position of the system with respect to the light source constantly changing, the optimal actuation patterns are changing as well.

The two remaining cases, for light source placed at 0 and 5 m, represent the two general behaviours that the system needs to be able to achieve. For 0 m, the system's optimal behaviour consist of maintaining its position, roaming around the light source. For the case in which the light source is placed at 5 m, the systems can not reach the target within the assigned learning steps, and the optimal behaviour consist in moving as fast as possible in the direction of the light. For these, we find the reach of a positive effective velocity $\bar{U}_{\text{eff.}} \approx 3$ [mm/cycle] and and monotonic increase in gained distance.

In general we find that ~ 40 learning steps are needed to achieve and maintain a positive $\bar{U}_{\text{eff.}}$ for all light source distances greater than zero, see Fig. 5.4b-c. Importantly, we see that the learning behaviour is not effected by the absolute value of the light intensity, making this measure suitable for our purpose.

5.5 Learning to move in various shape configurations

In the previous section we have shown that our decentralised control strategy, based on light intensity measurements, is suitable for the control of 2D modular systems. So far we only considered a double-symmetric star shape system consisting of five units. Here instead we look at a larger range of systems with various shapes. Specifically, we consider all possible system configurations polyomino shapes up to order 5 [149], where adjacent units share at least one edge. We only consider unique shapes which can not be obtained by rotation or translation of other shapes, and consider symmetric configurations only once. As discussed in Sec. 5.3, in-line configurations can only produce locomotion along the longitudinal axis since they can not break symmetry across the transversal direction. For this reason, we remove the linear polyominoes from the ensemble of all possible configurations. In Fig. 5.5a we show the 15 unique shapes that are left given mentioned considerations.

We run 100 simulations of 600 learning steps per individual shape, by placing the system on the reference system's origin with a random rotation. The light source is positioned at $(2, 0)$ m. We find that all of the considered shapes can improve their score by approaching the light source, and find no long term influence of the initial rotation. Nevertheless, we do find differences in performance between individual shapes. To assess the property of convergence to the light source, in Fig. 5.5b we show the median number of steps needed to reach the light source location for the first time within a radius of $r = 0.15$ m. While we see that most of the shapes can approach the light source in a comparable number of steps, where the best performing shape is number 2, the shapes $0, 4$ and 11 stands out for their higher values. We believe that the better convergence can be related to systems asymmetry, which may have limited number of optimal locomotion patterns restricted to a particular orientation of the system with respect to the light source.

As also illustrated in the previous section, when the system reaches the target it often overshoot and moves away from it. In Fig. 5.5c we show the number of steps spent within the target radius with respect to the number of steps need to reach it. We see that most of the results leave the target region after entering it. To quantify which shapes are better at maintaining a good position with respect to the light source, in Fig. 5.5d we show the median number of steps spent within the target region as a function of the geometry index. Also here we find that the best performing shape is number 2, while the worst are shapes number $0, 4$ and 11.

In order to assess the general behaviour of these shapes, in Fig. 5.6a we show the positions during the learning experiments for geometries $0, 2$ and 11 respectively. We see that the overall worse performance of geometries 0 and 11 can be seen by ample motion on the plane with larger trajectories spread. Especially in the case of geometry 11, the trajectories show very sharp bends. We believe this can be associated with the need of adjusting the locomotion pattern to different orientation of the system with respect to the light source.

In Fig. 5.6b we show the distribution of average effective velocities of the three systems. While geometry 0 improve its effective velocity at the beginning of the learning experiment, it quickly converges to lower average equilibrium values of ~ 1 mm/cycle only after ~ 120 learning steps. This low scores can not be associated with the roaming around the light source, since as shown in Fig. 5.5b this geometry takes on average 400 learning steps to do so. Differently, shape 2 rapidly converge to high effective velocities and maintains them on average, until it reach the target region at $\sim 250^{\text{th}}$ learning step.

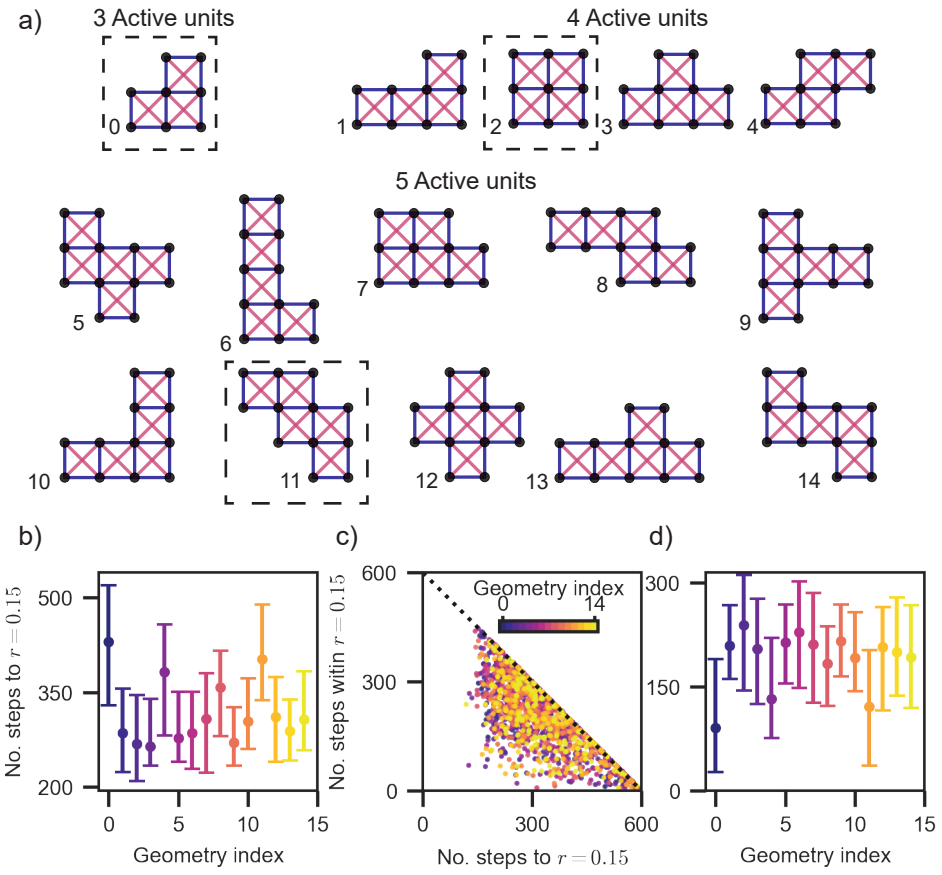

FIGURE 5.5: a, Ensemble of all unique system shapes up to 5 active units, where the number represent the geometry index use for reference. b, Median number of steps required to reach the circular area around the light source with $r < 0.15$ m, as a function of the geometry index. The errorbar represent the 25^{th} and 75^{th} quantiles respectively. c, Number of steps spent within the circular area around the light source with $r < 0.15$ m, with respect to the number of steps needed to reach it for all geometries. d, Median number of steps spent within the circular area around the light source with $r < 0.15$ m as a function of the geometry index.

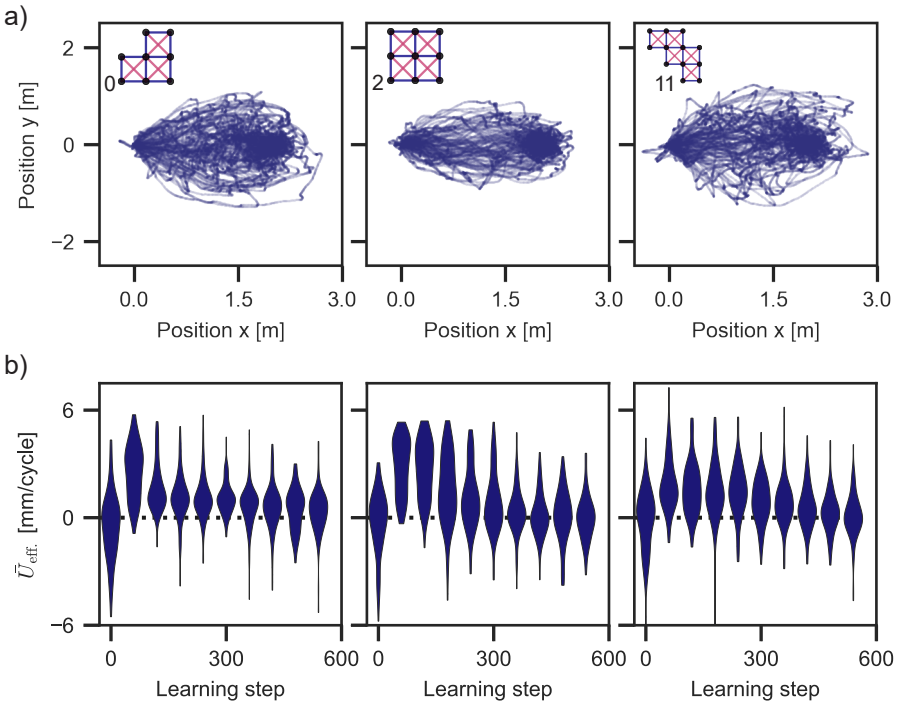

FIGURE 5.6: a, Selected system's trajectories for 100 runs with random initial orientation. Distribution of selected system average effective velocity $\bar{U}_{\text{eff,}}$ as a function of the learning steps.

5.6 Adaptability to obstacles

In the previous section we have shown that the proposed learning strategy is able to adapt to different arrangements of active units. We have shown that the adoption of a particular shape can influence the traveled distance by changing the range of possible behaviours, and that different shapes can converge or maintain their position close to the light differently than others. In this section we investigate the adaptability of our proposed learning strategy to sudden variation in the environment. As in Chapter 3 we have focused on the adaptability of the 1D system to variations of the track and friction changes, here we look at an impervious landscape where obstacles are interposed between the system and the light source.

Since the system's shape did not affect the overall behaviour of the learning algorithm, since all shapes were able to approach the light source, for simplicity we consider the star shape configuration shown in Fig. 5.1a, placed on the reference

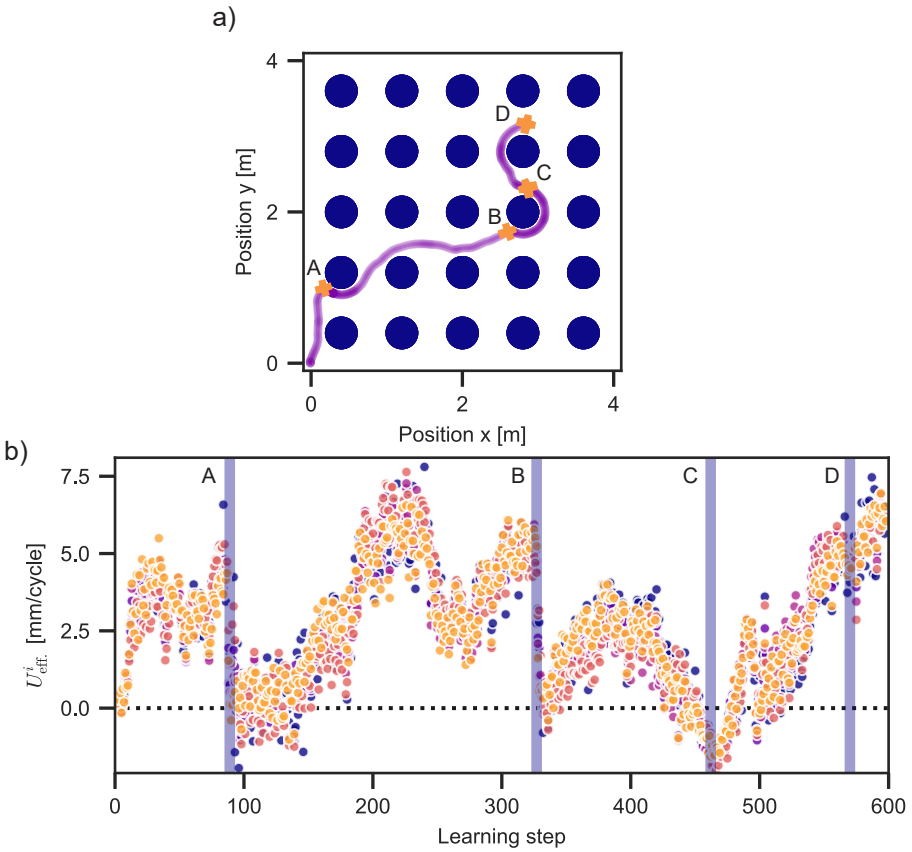

FIGURE 5.7: a, Trajectory of start shape configuration with circular obstacles interposed between starting position and light source. b, Individual units measured effective velocity U^i_{eff} as a function of the learning step. Shaded band are located and $89^{\text{th}}, 327^{\text{th}}, 462^{\text{th}}$, and 570^{th} learning steps respectively, whose position in space are shown in Fig. 5.7a at point A, B, C and D respectively.

system's origin, with a light source placed at $(5, 5)$ m. Between the light source and the system's starting position, we interpose a square array of circular obstacles of radius 0.2 m. Importantly, the obstacles do not generate shadows, but define regions of the space where the system can not reach by exerting a reaction force on the system, see Sec. 5.10.1 for details on the implementation. In Fig. 5.7a we show the system trajectory and four screenshots of a learning experiment consisting of 600 learning steps. We notice that after an initial forward locomotion along the vertical direction, the first contact with one obstacle occurs. This causes a sharp drop in the average effective velocity since no forward locomotion is possible

anymore. This is shown by the first vertical shaded band in Fig. 5.7b. After trying different locomotion strategies, which incur into negative effective velocities, the system manages to get passed the obstacle and move towards the light. Later on along the learning process, approximately between the 220^{th} and 280^{th} learning steps, the system experience further contacts with the two adjacent obstacles, without coming to a compete stop this time.

Note that the type of locomotion around obstacles shown between the 327^{th} and 462^{th} learning steps, is maintained as long as it exhibit a good average effective velocity. When the circular locomotion causes the system to reach negative velocities, hence moving away from the light source, the system keeps adjusting its behaviour reaching a different locomotion pattern, as it can be seen in the trajectory after the 462^{th} learning step. This is a clear indication that the system is highly adaptable to the environment, continuously adjusting the locomotion patterns according to the situation.

In Fig. 5.8a we show the distribution of gained distances for 100 similar runs. Here we find a monotonic increase in gained distance towards the light, meaning that despite the interaction with obstacles, the system can proceed moving towards the target location. The encounters between the system and obstacles lead to a general decrease in performance of the system decreasing its speed, which can also be seen in the darker trajectories of all simulation runs in Fig. 5.8b. To conclude, the adaptive behaviour of our learning strategy, when applied to the star shape configuration, can also be seen by noticing that the system average direction is along the direction of the light source (at a $45°$ angle respect to the x axis).

5.7 System scalability

The ultimate prerogative of decentralised robotic system is the possibility of deploying a growing number of units, while being able to maintain the intended functionality. In this section we discuss the scalability for 2D systems, and investigate how the increasing number of active units N, affect the system learning capabilities and overall behaviour in terms of effective velocity and type of motion. For systems made of square lattices of $N = n \times n$ active units, we run 100 simulations of 500 learning steps with a light source placed at $(20, 0)$ m, and n ranging from 2 to 7. As also shown before, the consideration of different number of unit leads to modification of the landscape of possible behaviours the system can achieve. For this reason, to assess the performance of our system we will look at the average effective velocity $\bar{U}_{\text{eff.}}$ and determine an equilibrium effective velocity $\bar{U}_{\text{eff.}}^{Eq.}$ the systems converge to.

In Fig. 5.9a-c we show the average effective velocity $\bar{U}_{\text{eff.}}$ as a function of the learning steps for $n = 2, 5$ and 7 respectively. We find that on average all system sizes can improve their velocity towards the light within the first ~ 50 learning steps, and that the maximum reached velocities decrease at the increase of N.

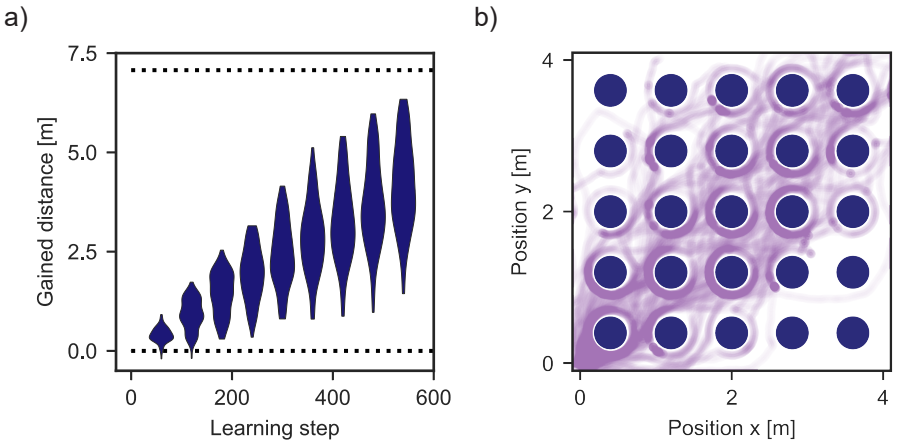

FIGURE 5.8: a, Distribution of gained distance for 100 runs for star shape configuration with obstacles as a function of the learning step. The top dashed line represents the light source distance from the system starting point. b, Center position trajectories of start shape configurations with obstacles.

We also find that the spread between individual runs, as shown by the average curve standard deviation, decrease with the increase in system size N, this is in qualitative agreement with the 1D results shown in Chapter 3, highlighting the lower impact of single units' behaviour on the overall locomotion for large N. To quantify the equilibrium effective velocity $\bar{U}^{Eq.}_{\text{eff.}}$, we fit the average behaviour to the exponential function:

$$y = \bar{U}^{Eq.}_{\text{eff.}} - \left(\bar{U}^{Eq.}_{\text{eff.}} - \bar{U}_0\right) e^{rx}, \tag{5.3}$$

and in Fig. 5.9d we show the relation between $\bar{U}^{Eq.}_{\text{eff.}}$ and system size N. We find a decreasing trend as the system size grows. While the equilibrium effective velocity $\bar{U}^{Eq.}_{\text{eff.}}$ decreases, interestingly the number of learning steps needed to reach 80% of it also decreases (Fig. 5.9e).

By considering the variation in effective velocity $\bar{U}_{\text{eff.}}$ within the first 10 learning steps, similary to Chapter 4, we can calculate the average learning rate as

$$\bar{\gamma} = \frac{1}{\Delta s \; n_{runs}} \sum_{j=1}^{n_{runs}} \frac{\bar{U}^{10}_{\text{eff.,j}} - \bar{U}^{1}_{\text{eff.,j}}}{9}. \tag{5.4}$$

In Fig. 5.9f we show the dependency of $\bar{\gamma}$ on the system size, and find a decrease for increasing N. Most importantly, the trend of the normalized learning rate is in very good agreement with the scalability results shown in Fig. 4.11 of Chapter 4.

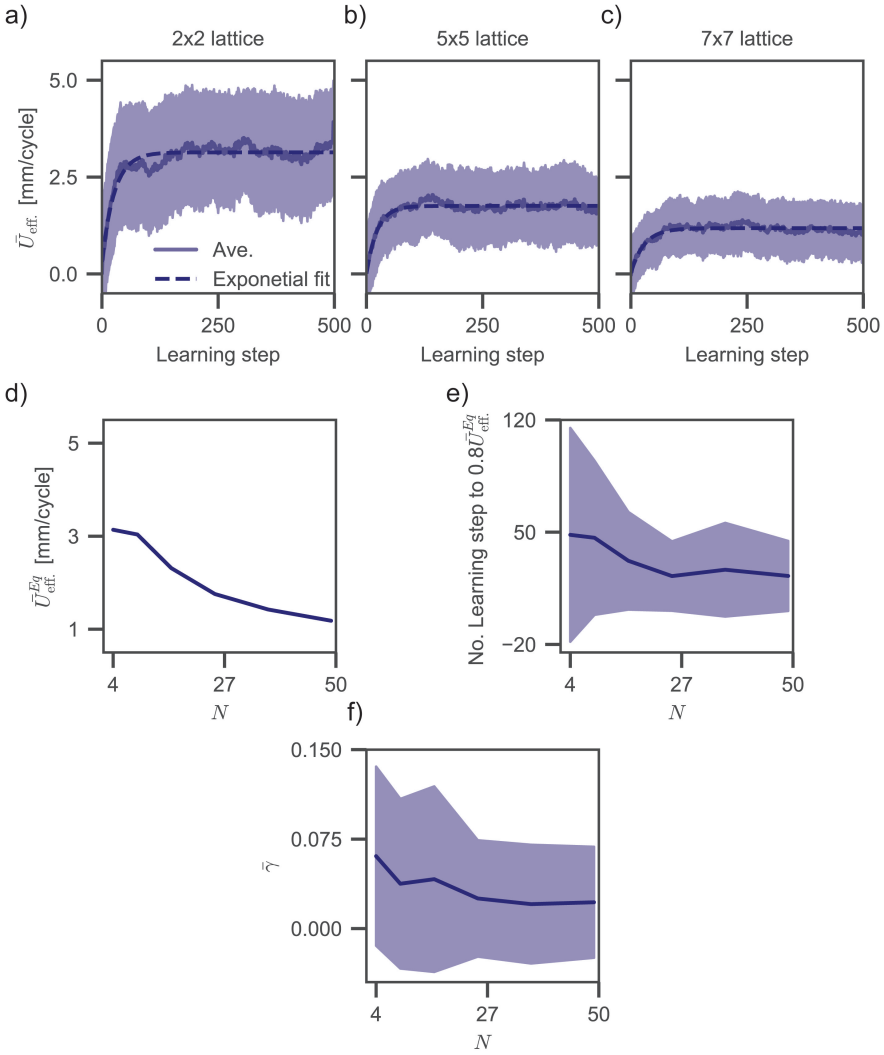

FIGURE 5.9: a-c, Average effective velocity $\bar{U}_{\text{eff.}}$ as a function of the learning step for square system of $N = 4, 25$ and $N = 49$ active units. The solid lines represent the average curve over 100 runs, the shaded area the results within the standard deviation, whereas the dashed line represent the exponential fit to the average curve. d, Equilibrium effective velocity $\bar{U}_{\text{eff.}}^{Eq.}$ as a function of system size N. The values are obtained from the exponential fits where line thickness represent the standard deviation on the parameters. e, Average number of learning steps needed to reach $0.8\bar{U}_{\text{eff.}}^{Eq.}$ as a function of the system size N. f, Average normalize learning rate, considering the first 10 learning steps, as a function of the system size N.

To conclude, we believe that the minor reduction in average learning rate in comparison with the reduction of the equilibrium velocity at the increase in N, causes the improvement in the average number of learning steps to reach $0.8\,\bar{U}_{\text{eff.}}^{Eq.}$.

Initializing the simulations from good solutions

The results shown so far are for simulations where the initial configurations ϕ_i were randomly assigned. Even for relatively small systems, i.e. $N > 3$, computing the possible behaviours as a function of all phases combinations ϕ_i becomes computationally expensive, therefore we do not have knowledge on the location and the amount of optima the system presents. This was in fact one of the motivations highlighted in Chapter 4. However, based on physical reasoning, we can make an educated guess on the phase combinations which would produce forward locomotion with significant velocity. By doing so we are able to determine whether the system converges to local optima, or if it at least is able to maintain good solutions.

In Chapter 3 we found that the one-dimensional optimal locomotion is obtained with phase differences $\Delta\phi = \phi_2 - \phi_1 = 0.4$ for two active units, and $\phi_2 - \phi_1 = 0.4$, $\phi_3 - \phi_1 = 0.8$ for three active units. In Sec. 5.3 we showed that the same phase combination produce an optimal locomotion in the two-dimensional model as well. This locomotion mode in fact features the least actuation interference between adjacent units in the direction of the locomotion. This causes one unit to push onto the next, when the second one is not actuating.

Here we also consider systems made of square lattices of $N = n \times n$ units, and assign initial phase difference $\phi_j - \phi_i = 0.4$ between units belonging to consecutive columns. This is displayed in Fig. 5.10a, where the peristaltic actuation wave moves from the right (closer to the light source) to the left. In Fig. 5.10b we show the average effective velocity $\bar{U}_{\text{eff.}}$ as a function of the system size N, and find an increase of velocity. While this might seem counter intuitive based on the previous learning experiment results, the increase in velocity can be explained by the increase of effective stroke of the system due to its increased dimension along the vertical axis. This can be seen Fig. 5.10c, where a system of $N = 64$ active units shows a greater expansion than a $N = 9$ system in Fig. 5.10a. Furthermore, given our model definition where units share the masses in common, considering larger 2D system leads to a decrease of the average weight of each active unit.

Nevertheless, it is important to realize that these velocities are specific to these square configurations, with the assigned peristaltic locomotion pattern, and most importantly, for the given system orientation with respect to the light. While we do not have certainty that the suggested locomotion pattern is the system's absolute maximum, we have verified that it grants locomotion speeds higher than those obtained when optimizing from a random initial phase combinations. We can then assess the system learning behaviour starting from a good solution, instead of from random phase combinations as we did so far.

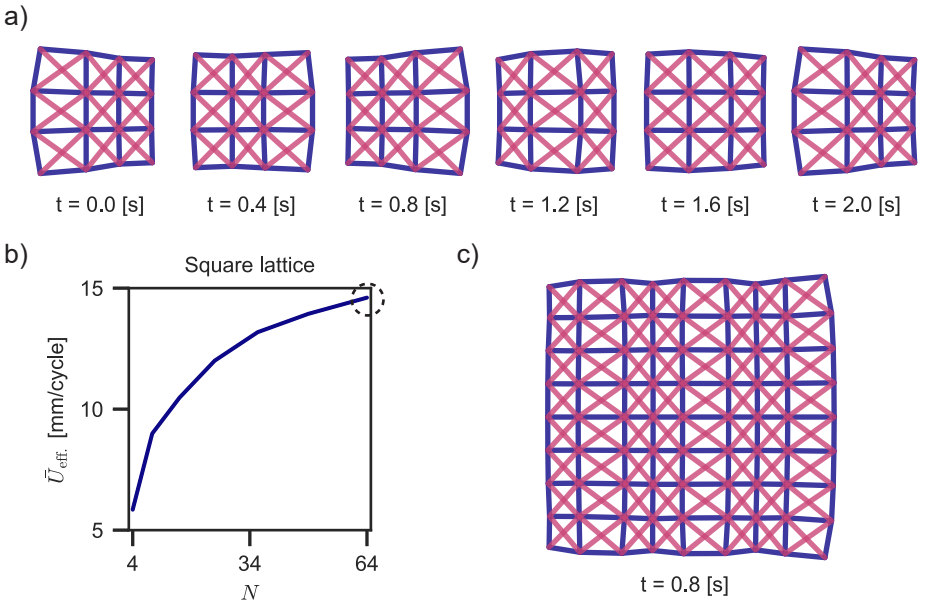

FIGURE 5.10: a, Schematic of system actuation with peristaltic phases combination, such that $\phi_j - \phi_i = 0.4$ between units belonging to consecutive columns, on a $N = 9$ active units system. b, Average effective system velocity as a function of system size N for peristaltic phases combination and horizontal alignment with respect to the light source. c, Schematic of system actuation with peristaltic phases combination on a $N = 64$ active units system.

We next run 100 learning experiments on systems made of square lattices of $N = n \times n$ active units, with n ranging from 2 to 7, by seeding the initial phases to the phases of the peristaltic locomotion pattern. In Fig. 5.11a-c we show the average effective velocity $\bar{U}_{\text{eff.}}$ as a function of the learning steps for $n = 2, 5$ and 7 respectively. We notice a sharp drop in velocity in the first learning steps, which seems to be more pronounced for small N. In Fig. 5.11d we show the velocity equilibrium values as obtained from the exponential fit, and compare them with the equilibrium values when starting with random phases as also shown in Fig. 5.9. Apart from a small variation for $N = 4$, we find almost a perfect match between the two cases. This is an indication that the average velocity equilibrium values $\bar{U}_{\text{eff.}}^{Eq.}$ are not affected by the initial phase combinations.

The fact that, in the long term, the system does not benefit from starting off a good set of phases, can be related to the combination of two phenomena. Firstly, the Flaky algorithm used in this chapter has preference on the exploration of new phases rather than maintaining similar values. While this grants adaptability, it comes at the cost of not maintaining good scores for long, as also shown in

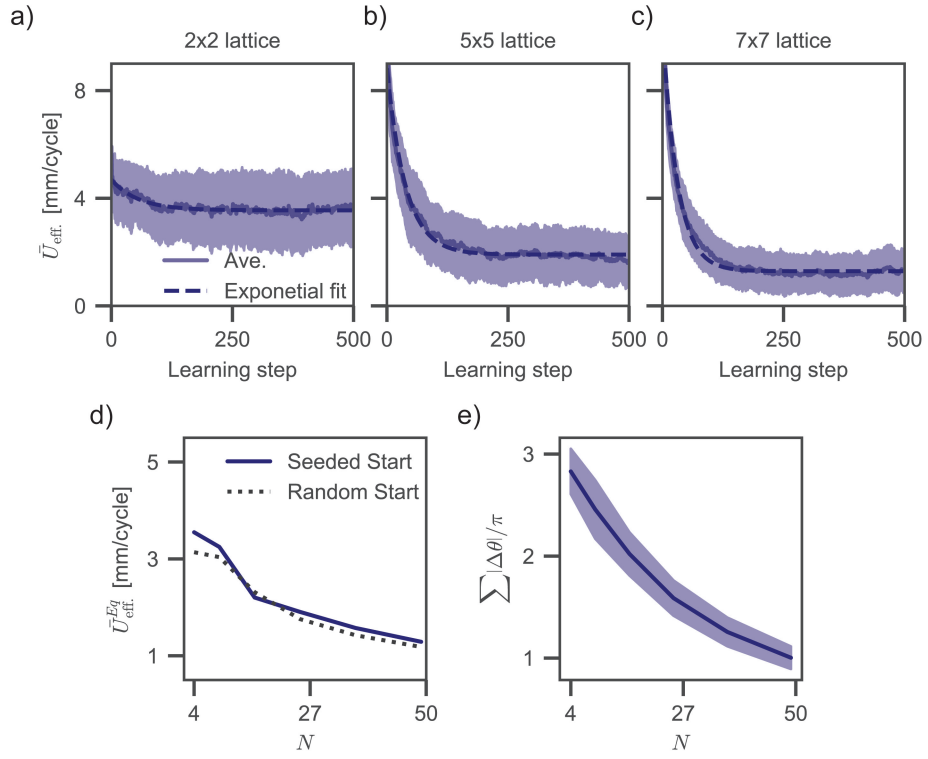

FIGURE 5.11: a-c, Average effective velocity $\bar{U}_{\text{eff.}}$ as a function of the learning step for square system of $N = 4, 25$ and $N = 49$ active units, with initial peristaltic phases combination. The solid lines represent the average curve over 100 runs, the shaded area the results within the standard deviation, whereas the dashed line represent the exponential fit to the average curve. d, Equilibrium effective velocity $\bar{U}_{\text{eff.}}^{Eq.}$ as a function of system size N for peristaltic and random initial phases combinations. The values are obtained from the exponential fits where line thickness represent the standard deviation on the parameters. e, Average total angle variation over 100 runs as as a function of the system size N, where the shaded areaa represent the standard deviation values.

Chapter 3 and Chapter 4. A second effect can be related to the system rotation during the learning. As mentioned earlier in fact, the seeded phase combinations allow for great velocity in the case of perfect alignment of the system with respect to the light source. System rotation occurring during learning, alter the interaction between the system (with rather good phases) and the environment. This effect is further strengthen by the explorative nature of Flaky, which continuosly try to correct for the past steps.

To quantify the amount of rotation the systems experience, we measure the total angle variation during a learning experiment, calculated as

$$\sum_{k=1}^{n_{\text{learn}-1}} |\theta_{k+1} - \theta_k| \, \frac{1}{\pi}, \tag{5.5}$$

where the θ_k is the rotation of the system with respect to the horizontal direction at learning step k. Importantly, since we have proven that the system is able to learn despite its orientation and shape, the absolute variation in angle is proportional to the change in the system's interaction with the environment. In fact, the larger the angle variation between two learning steps, the more different the performances of the system are for the same phase combinations.

In Fig. 5.11e we show the total angle variation as a function of the system size. We see that small systems, e.g. $N = 2 \times 2 = 4$, experience approximately three times as much rotation as a $N = 7 \times 7 = 49$ system. This means that large systems, by rotating less during a learning experiment because of physical reasons, operate in a less dynamic environment, where optimal phases combinations change not as rapidly as for the small systems. Reconnecting to the results discussed in Chapter 4, we can say that small systems experience large $\tilde{\omega}$ whereas large systems experience a smaller $\tilde{\omega}$. This would furthermore explain the sharper drop in effective velocity for small N shown in Fig. 5.11a-c. To conclude, as the system size N grows and the complexity to find optimal locomotion rises, the physical stability of the systems increases pushing the system towards stable regimes in terms of variation of the environment.

5.8 Resistance to damage

The benefit of using a decentralised system based on modular elements becomes evident when the redundancy of the building blocks grants the complete system resistance to damage. This section aims at quantifying how much our learning strategy is affected by damage, represented by the amount of inactive units. Here we consider systems made of square lattices of active units, and perform knock-out experiments, where a percentage of units are not able to actuate, and are effectively dead weight to be carried. We run 100 simulations of 500 learning steps with a light source placed at $(20, 0)$ m, where the location of inactive units is randomly assigned, and consider system sizes of $N = 16, 25$ and 36.

In Fig. 5.12a-c and Fig. 5.15 we show the average effective velocity for the considered system sizes and increasing percentage of inactive units. We notice that all systems can withstand some amount of damage without compromising the learning capabilities as shown by the positive effective velocity. Surprisingly, it seems that some amount of damage can actually help the systems achieve higher velocities, see e.g. Fig. 5.12a-b. To better quantify these findings, and compare

between different system sizes, in Fig. 5.12d we show the equilibrium effective velocity $\bar{U}_{\text{eff.}}^{Eq.}$ obtained from the exponential fits, as a function of the percentage of inactive units. We see that the three considered lattices can cope with a percentage of inactive units up to $\sim 45\%$ without lowering their equilibrium effective velocity $\bar{U}_{\text{eff.}}^{Eq.}$. Interestingly, besides the $N = 36$ also the systems with $N = 25$ can benefit from having some inactive units, since they reach $\bar{U}_{\text{eff.}}^{Eq.}$ higher values compared to the completely intact case, Fig. 5.15d-e and Fig. 5.12a-b.

We believe this is the results of a trade off between three effects. First, inactive units needs to be dragged, lowering the effective velocity of the system. Second,

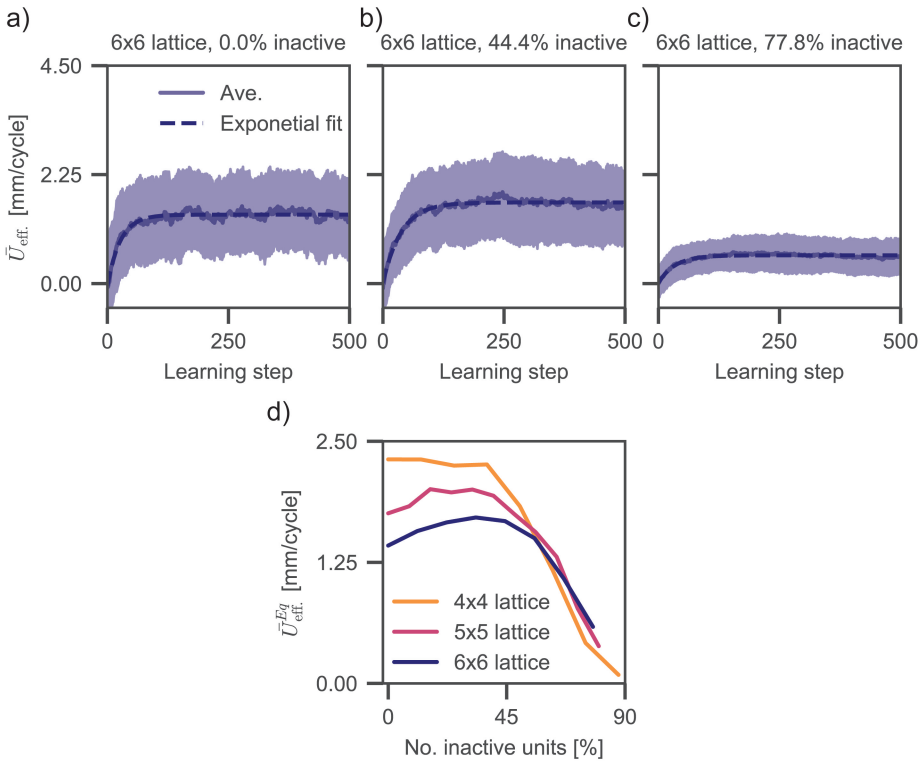

FIGURE 5.12: a-c, Average effective velocity $\bar{U}_{\text{eff.}}$ as a function of the learning step for square system of $N = 36$ units with percentage of inactive units equal to $0, 44.4$ and 77.8% respectively. The solid lines represent the average curve over 100 runs, the shaded areaa is the standard deviation, whereas the dashed line represent the exponential fit to the average curve. d, Average effective velocity $\bar{U}_{\text{eff.}}$ as a function of the percentage of inactive units, for systems of $N = 16, 25$ and $N = 36$ respectively.

as discussed in previous section, larger systems experience less rotation, making the system's interaction with the environment less mutable. This depends on the total number of units, despite of their activity level. Last, inactive units reduce the complexity of the optimization since less units are updating their phases, making it simpler to explore the realm of possible phase combinations, facilitating the reach of good locomotion pattern.

5.9 Conclusions

In this chapter we considered the decentralised control strategy developed in Chapter 3, and applied it to bidimensional systems by developing a simulation framework based on simplified two-dimensional mass-spring systems. With in mind the aim of benchmarking the experimental counterpart for further implementation, we indentify the limitations emerging from bridging the 1D case to 2D, and perform a series of explorative numerical experiments.

We showed that in spite of the significantly increased complexity of the task, caused by the motion of the system in 2D, all considered systems can improve their behaviour moving towards the light source. By investigating systems of different shape, number of units and amount of damage, while we observed some differences in term of type of motion and maximum speed, we found that all systems maintain the capability to learn. If on one hand the explorative nature of the Flaky algorithm favours the adaptivity of the system to the environment, by e.g allowing it to move around obstacles, on the other it affects the exploitation of good locomotion patterns. This effect is particularly pronounced for large systems. We believe this can be potentially improved by updating the current algorithm to include a variable step size. In fact, as a large step size is beneficial for adapting the system behaviour, since it can quickly reach and explore different phases, a smaller step size guarantees a slower and steady improvement as also shown in Chapter 3. In order to obtain a scalable and adaptable system, a balance between the two needs be found during operation. This could be for example done by varying the step size as a function of the variation of score between the last few learning steps.

5.10 Supplementary information

5.10.1 Model implementation

We model the bidimensional arrangement of a robotic metamaterial with a 2D mass-spring system, where each mass experiences friction with the environment. Each robotic unit is modelled with four masses m connected by springs as shown in Fig. 5.1a. We consider passive springs connecting the masses horizontally and vertically, and active springs connecting masses diagonally. Both types of spring have a stiffness k_s. When multiple units are assembled together, the additional overlapping masses are removed.

We define a global reference frame, where the initial position of the masses is $\mathbf{r_{ini}} = [\mathbf{r_{ini,1}}, ..., \mathbf{r_{ini,n}}]^T$, where $\mathbf{r_{ini,1}} = [r_{ini,1x}, r_{ini,1y}]^T$. The masses positions at the following step is found as

$$\mathbf{r} = \mathbf{r_{ini}} + \mathbf{u} = [\mathbf{r_1}, ..., \mathbf{r_n}]^T = [\mathbf{r_{ini,1}} + \mathbf{u_1}, ..., \mathbf{r_{ini,n}} + \mathbf{u_n}]^T , \tag{5.6}$$

where e.g. $\mathbf{u_i} = [u_{ix}, u_{iy}]^T$ is the displacement of the mass m_i. The force balance of our system can be written as:

$$\mathbf{F}_m(\ddot{\mathbf{u}}) + \mathbf{F}_f(\dot{\mathbf{u}}) + \mathbf{F}_k(\mathbf{u}) = \mathbf{F}_e, \tag{5.7}$$

where $\mathbf{F}_m(\ddot{\mathbf{u}})$ are the inertial forces, $\mathbf{F}_f(\dot{\mathbf{u}})$ the dissipating forces, $\mathbf{F}_k(\mathbf{u})$ represents the elastic forces of the springs, and $\mathbf{F}_e$ are the external forces. Since each mass has two degrees of freedom, each term in Eq. (5.7) contains $2n$ elements, where n is the number of masses taken in consideration.

The dissipating term $\mathbf{F}_f(\dot{\mathbf{u}})$ comprises the effects of the Coulomb friction, and viscous friction. Both terms are function of the velocity of the masses $\dot{\mathbf{u}}$. For example, considering a mass m_i, the dissipating term is

$$\mathbf{F}_{f,i}(\dot{\mathbf{u}}_i) = \mathbf{F}_{f\mathrm{C},i}(\dot{\mathbf{u}}_i) + \mathbf{F}_{f\mathrm{V},i}(\dot{\mathbf{u}}_i) \tag{5.8}$$

$$\mathbf{F}_{f,i}(\dot{\mathbf{u}}_i) = -\left[(\mu_{\mathrm{Brk}} m_i g)\tanh\left(\frac{\|\dot{\mathbf{u}}_i\|}{v_{\mathrm{C}}}\right) + c\,\|\dot{\mathbf{u}}_i\|\right]\frac{\dot{\mathbf{u}}_i}{\|\dot{\mathbf{u}}_i\|}, \tag{5.9}$$

in which μ_{Brk} is the breakaway friction coefficient between each mass and the surface, g is the gravitational constant, c is the viscous damping coefficient, and v_{C} is the Coulomb velocity threshold which scales the speeds at which the friction barrier is broken. Note that with $\|\dot{\mathbf{u}}_i\|$ we indicate the magnitude of the displacement vector $\dot{\mathbf{u}}_i$. According to this model, the Coulomb friction is assumed to be purely in the direction of the velocity $\frac{\dot{\mathbf{u}}_i}{\|\dot{\mathbf{u}}_i\|}$, which is a simplification of the real frictional behaviour we would have experimentally.

The elastic forces of the springs $\mathbf{F}_k(\mathbf{u})$ depends on the distance between masses connected by springs. E.g. considering two masses m_i and m_j, the elastic force

acting on m_i, depends on the relative position of the two masses and the spring rest length L_{ij}, as:

$$\mathbf{F}_{k,ij}(\mathbf{u}) = (\|\mathbf{r}_j - \mathbf{r}_i\| - L_{ij})\, k_s \frac{\mathbf{r}_j - \mathbf{r}_i}{\|\mathbf{r}_j - \mathbf{r}_i\|}. \tag{5.10}$$

Then, considering all masses $p \in P$ that share a spring with the mass m_i, the total elastic force on the mass m_i is

$$\mathbf{F}_{k,i}(\mathbf{u}) = \sum_{p \in P} \mathbf{F}_{k,ip} \tag{5.11}$$

Note that the preferred spring length L_{ij} between two masses in Eq. (5.10) is constant for normal springs such that $L_{ij} = l_0$. Differently, in the case of active springs, L_{ij} is time-dependent, which is used to model the actuation of units. Specifically, assuming that the masses m_i and m_j are connected by an active spring,

$$L_{ij}(t) = l_0\sqrt{2} + l_{ij}(t), \tag{5.12}$$

where the $\sqrt{2}$ comes from the fact that active springs are placed along the units' diagonals, and

$$l_{ij}(t) = \begin{cases} L_{ij,\text{off}} + (IR - L_{\text{max}})(1 - e^{(-t+\phi)/(RC)}) & \text{for } 0 \leq t \leq t_{on} \\ L_{\text{max}}\, e^{(-t+\phi)/(RC)} & \text{for } t_{\text{on}} \leq t \leq \lambda, \end{cases} \tag{5.13}$$

where L_{max} is the maximum variable spring length, $L_{ij,\text{off}}$ is the variable spring length at the end of the previous actuation cycle, and I, C and R are the parameters obtained from the electrical-pneumatic analogy discussed in Chapter 3. With respect the description used in Chapter 3, we additionally add the dependency on the following parameters: ϕ which is actuation phase of each unit, and λ represent the duration of the actuation cycle such that $\alpha = t_{on}/\lambda = 0.4$ is kept constant. Importantly, the variable spring length defined Eq. (5.12) is equal for active springs belonging to the same unit.

Simulations setup

We transform Eq. (5.7) into matrix formulation

$$\mathbf{M}\, \ddot{\mathbf{u}}(t) + \mathbf{F}_f\, \dot{\mathbf{u}}(t) + \mathbf{K}\, \mathbf{u}(t) = \mathbf{F}_e, \tag{5.14}$$

and since our model allows for large deformation, we numerically solve it by turning it to the state-space representation such that

$$\mathbf{y}(t) = \begin{bmatrix} \dot{\mathbf{u}}(t) \\ \mathbf{u}(t) \end{bmatrix}, \tag{5.15}$$

then, the first derivative of Eq. (5.15) with respect to time can be written as

$$\dot{\mathbf{y}}(t) = \begin{bmatrix} \ddot{\mathbf{u}}(t) \\ \dot{\mathbf{u}}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{M^{-1}} \left(\mathbf{F}_e - \mathbf{F}_f \, \dot{\mathbf{u}}(t) - \mathbf{K} \, \mathbf{u}(t) \right) \\ \dot{\mathbf{u}}(t) \end{bmatrix}, \tag{5.16}$$

which allow us to describe the system with first-order differential equations. We then use an explicit solver with 5^{th} order Runge-Kutta integration scheme.

To perform learning simulations, a system formed of N units is initialized and all the parameters are set to their default values. This include the unit size, associated with the constant rest length of all springs $l_0 = 0.06$ m, the maximum variable spring length $L_{\text{max}} = 0.06$ m of the active springs such that $L_{ij}(t) = l_0 + l_{ij}(t) = 0.12$ m, stiffness of the springs $k_s = 100$ N/m and mass of the nodes $m_i = 0.073$ kg. Furthermore, for the calculation of friction forces and the actuation, we set $\mu_{\text{Brk}} = 0.24$, $c = 4$, $R = 5$, $I = 1$, $C = 0.05$. Unless explicitly specified, the system is placed on the origin of the reference system by considering its overall geometrical centroid, while the light source is placed at different position and constant height of 1 m with respect to the horizontal plane. A random phase ϕ_i is assigned to each active unit, which will be used to change the preferred spring length of the two active springs connecting diagonally the unit's nodes. We assign a number of learning steps in which each unit is optimizing its phase ϕ_i to reduce the distance to the light source.

A learning step consists of $t \approx 6$ s in which three cycles of elongation-shortening of the active springs in performed. The first cycle has a variable duration, and it is included to implement a unit's phase change from ϕ_i to ϕ_i' by modifying the value of λ in Eq. (5.13). The phase variation $\Delta\phi = \phi_i' - \phi_i$ is used to determine the duration of the first cycle λ_{change}, such that

$$\lambda_{\text{change}} = \begin{cases} \lambda & \text{for } \Delta\phi = 0 \\ \lambda + \Delta\phi & \text{for } \frac{\lambda}{2} \leq \Delta\phi \leq -\frac{\lambda}{2} \\ 2\lambda - \Delta\phi & \text{for } \Delta\phi > \frac{\lambda}{2} \\ 2\lambda + \Delta\phi & \text{for } \Delta\phi < -\frac{\lambda}{2}. \end{cases} \tag{5.17}$$

In Fig. 5.13 we show a schematic of the phase change within a learning step. The blue curve represent an initial phase $\phi_i = 0$, whereas the pink solid line represent the new phase $\phi_i' = 0.2$ the unit wants to achieve. In this case, the phase change is done by extending the first cycle such that the second cycle effectively exhibit the new phase ϕ_i'. The variation in duration of the first cycle with respect the others only depends on the variation between ϕ_i and ϕ_i', which is bounded by the small step size $\Delta s = 0.1$. We expect this effect to be negligible. The second cycle with a duration of $t = 2$ s is included to extend the total duration of a learning step. The third cycle, which also has a duration of $t = 2$ s, it used to measured the travelled distance associated with the tried phase. While the second cycle could be removed, its adoption increase the chance of measuring travelled

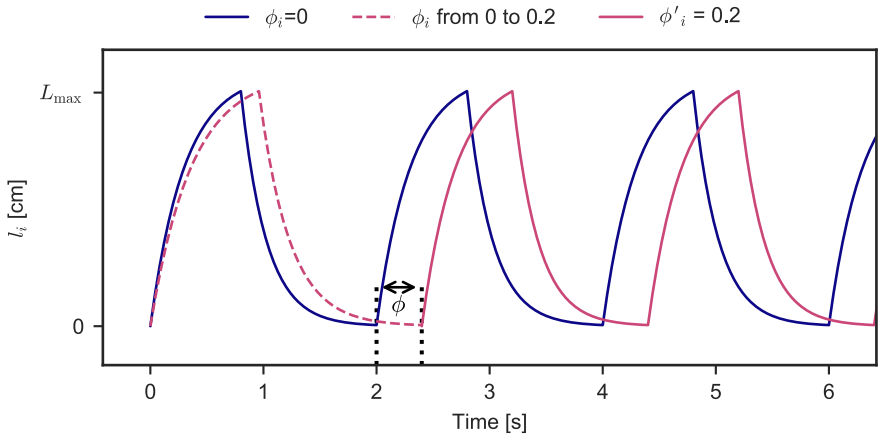

FIGURE 5.13: Schematic of the actuation function as expressed by the time-dependent part $l_{ij}(t)$ of the preferred spring length $L_{ij}(t) = l_0\sqrt{2} + l_{ij}(t)$ as a function of time. The dashed line represent the extended adjusted actuation cycle used to perform a pahse variation from $\phi_i = 0$ to $\phi_i = 0.2$.

distances in the third cycle when other units, which are out of sync, are still on the same learning step.

The travelled distance measured in the third cycle is calculated from the average change in position of the four masses belonging to one unit. This average displacement value associated with one learning step is used to compute the variation in brightness the centre of the unit would measure according to Eq. (5.1) and Eq. (5.2). This information is used to determine the acceptance of the tried phase ϕ_i according to the Flaky algorithm.

5.10.2 Obstacles implementation

The adaptability section of this chapter discussed the ability of the system of learning locomotion patterns towards a light source, when equally spaced obstacles are interposed between them. The obstacles are modeled as circular region of the space where the system nodes can not be. To keep the added computational cost minimum, the interaction of the system with the obstacles is implemented by applying a constant reaction force, radially oriented from the obstacles centre, on the system's nodes which have a distance $d_i \leq r_o$, where r_o is the obstacles radius. The reaction force applied on the i nodes is expressed by

$$F_i = \begin{cases} 0 & \text{for } \|r_i\| > r_o \\ \eta \frac{r_i}{\|r_i\|} & \text{for } \|r_i\| \leq r_o, \end{cases} \tag{5.18}$$

where $\eta = 0.5$ was arbitrary chosen to be higher than the total force nodes experience during learning experiments without obstacles. This prevent nodes to actually overlap with the obstacles.

To avoid that the condition expressed by equation (5.18) is checked for all the obstacles and all the nodes at every simulation timestep, the space is divided in square portions of dimensions $l_{\text{ref}} \times l_{\text{ref}}$, with an obstacles placed at it centre ($r_o < l_{\text{ref}}$). As a consequence equation (5.18) is checked only for the nodes belonging to a specific space portion.

5.10.3 Model parameter sensitivity

To determine the model sensitivity the the choice of the numerical parameters, we considered a system made of three active units in a line configurations oriented radially with respect to the light source, and compute the average system effective velocity $\bar{U}_{\text{eff.}}$ for all phases combinations. Apart from the control parameter we vary, we keep all parameters set to the values defined in Section 5.10.1. In Fig. 5.14a we show the dependency of α on the overall system performance. We see that by changing the value from $\alpha = 0.4$ to $\alpha = 0.2$ causes the merging of the two optima into one, and that the system velocity is maximized for $\alpha = 0.3$. Importantly, in the latter case the optimal phases combination is found for $\phi_1 - \phi - 2 \approx 0.7$ and $\phi_3 - \phi_2 \approx 0.3$. This can be intuitively understood by considering the fact that units are working in sync exhibiting peristaltic actuation across the system without interference. This is not the case for $\alpha = 0.4$, where some interference between consecutive units is present, see Fig. 5.2b.

In Fig. 5.14b we show the influence of the spring stiffness k_s on the system effective velocity. We notice that the increase in stiffness causes an enhancement of the system velocity around the optima, i.e. these regions get sharper, without altering the phase combinations which define them. Finally, in Fig. 5.14c, we show the effect of the viscous friction term c. Going from $c = 2$ to $c = 6$ causes the system to go from an underdamped configuration, where units oscillation are large and with higher frequency, to an overdamped case, where the total system effective velocity is low because of dissipation.

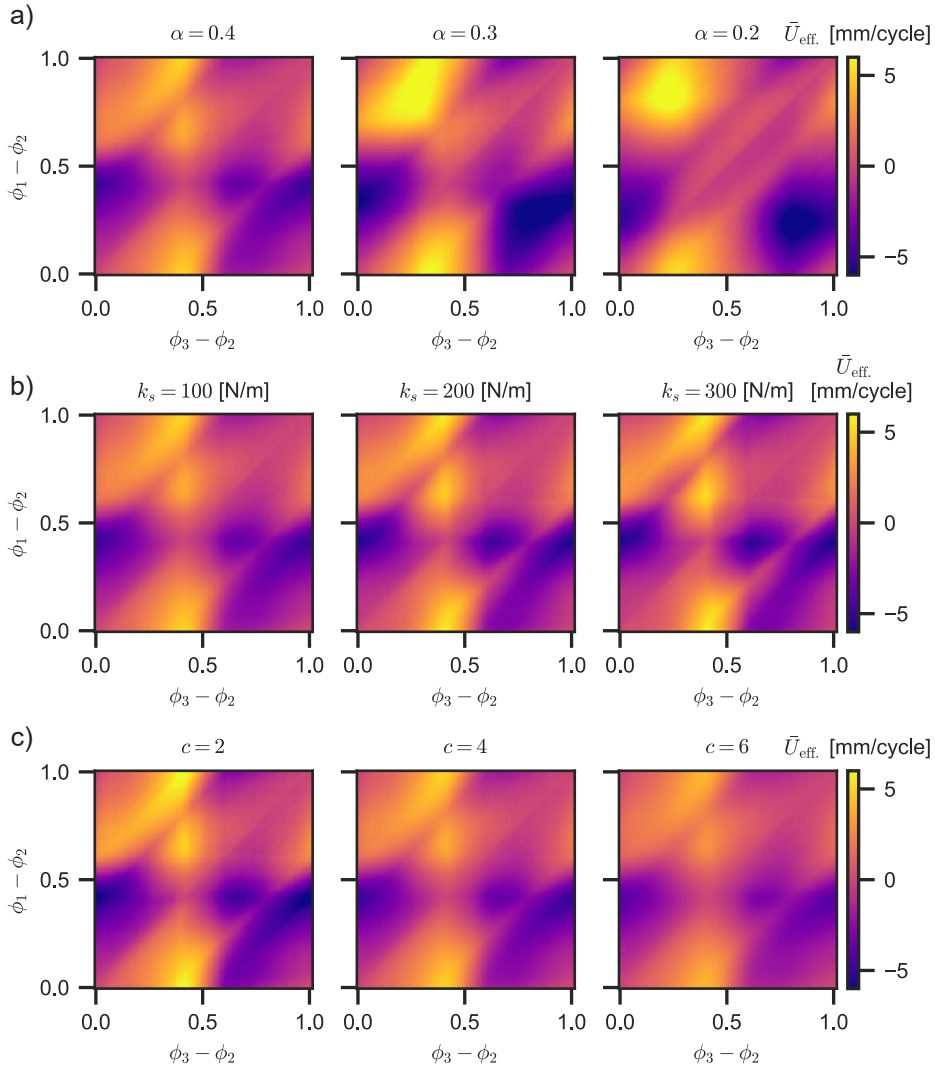

FIGURE 5.14: Effect of numerical parameters on the average effective velocity $\bar{U}_{\text{eff.}}$, as a function the phases combination for a three active unit system in a line configurations. a, Sensitivity analysis results for α. b, Sensitivity analysis results for spring stiffness k_s. Sensitivity analysis results for viscous damping coefficient c.

5.10.4 Supplementary figures

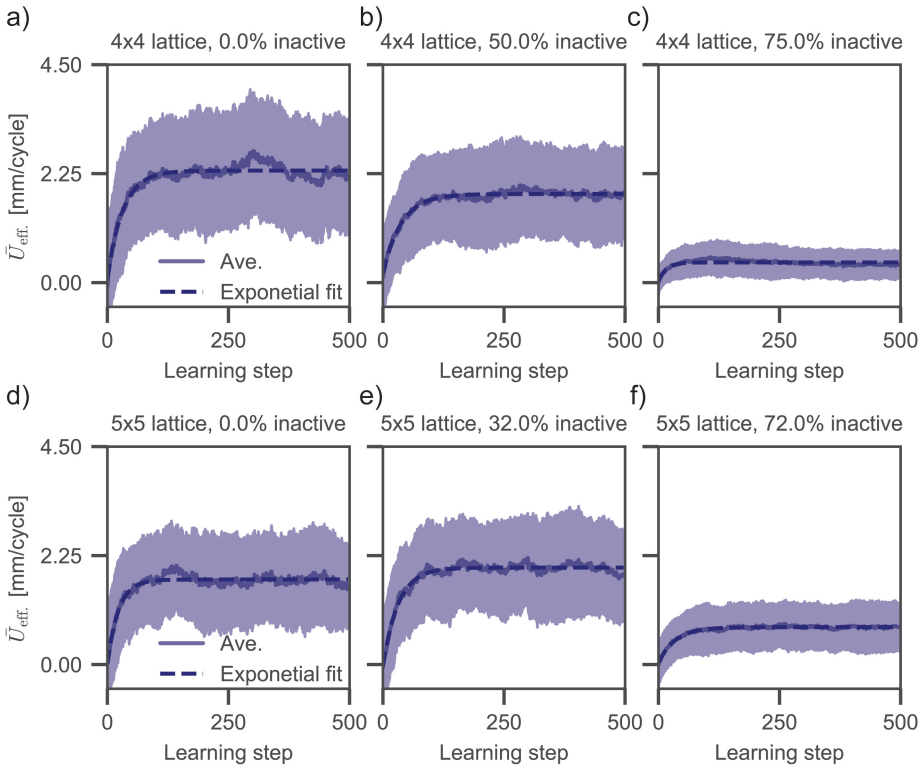

FIGURE 5.15: Average effective velocity $\bar{U}_{\text{eff.}}$ as a function of the learning step for square system of $N = 16$ (a-c) and $N = 25$ units (d-f) with increasing percentage of inactive units. The solid lines represent the average curve over 100 runs, the shaded area the results within the standard deviation, whereas the dashed line represent the exponential fit to the average curve.

CHAPTER 6

Conclusions and Outlook

The work discussed in this thesis is devoted to designing and understanding optimal behaviour in mechanical and robotic metamaterials. Whether these are made of active or passive unit cells, we develop optimization strategies for determining optimal cellular features which allow for targeted material properties and behaviours. In this chapter we summarize the main contribution of this thesis, include some recommendations and provide an outlook.

6.1 Conclusions

In Chapter 2 we focus on mechanical metamaterials, where a straightforward design strategy for mechanical metamaterial with tailored behaviours is lacking. In fact, while harnessing instabilities is a common practise to obtain materials with novel properties, their tedious numerical description constrained researchers to a trial and error design process. To provide a general solution, which can also be extended to other fields, we implemented a Monte Carlo-based inverse design algorithm capable of dealing with disjoint optimization problems. Here, we show that we can design mechanical metamaterials with tuned buckling properties within a wide range of values. By tuning higher order modes, we also show that unpredictable material behaviours, which emerge from the interaction of multiple modes, can be controlled and avoided. Importantly, while we mainly focus on the design of mechanical metamaterials with targeted buckling behaviour, our algorithm can easily be extended to a range of problems not limited to mechanics. For example, we adapted this algorithm to inversely design overlay targets for metrology applications [150], where the discrete nature of the problem made other alternative design techniques unfeasible.

Bridging these concepts to the realm active metamaterials, we are interested in determining if such algorithms could also be used to learn behaviours 'on the go'. To verify this, in Chapter 3 we develop a modular robotic metamaterial platform, where individual units can adapt their behaviour based on local sensing, in order to achieve a global forward locomotion along a circular track. We propose two learning algorithms, namely Thermal and Flaky, to be ran at the unit level. We show that while both algorithms allow for the complete system to start to locomote, only the Flaky algorithm makes the system robust to changes in the environments and damage, maintaining the capability of learning even for greater system sizes.

While the development of a metamatateriał platform allow us to test the learning strategies in real-life circumstances, it also poses some constrains. First of all, within the settings discussed in Chapter 3, the learning strategy is embodied in the robotic platform, and it is not possible to assess the learning capability of the learning strategy, discerning the effects imposed by the platform. For example, the range of possible behaviours that can be learnt by the learning strategy is dictated by the physical locomotion possibilities of the platform. Furthermore, the use of systems consisting of many units creates a range of possibilities too vast to be explored, and makes it impossible to assess the learning capabilities. For these reasons, in an effort to understand what the limits of the learning capabilities are, in Chapter 4 we developed a numerical platform where we only consider the learning processes of the units, translating them to the evaluation of simplified environment functions. While the definition of such environment functions loose physical meaning, it helps us highlight the qualities and limitations of the algorithms. Here we find that, while the Thermal algorithm exhibits a modest resistance to noise as results of the effect of its characteristic temperature, we show that this temperature value depends on the system size and noise, which are information not available to the unit. This makes the Thermal learning strategy not suitable for our purposes. We also find that the Flaky algorithm is scalable, presents a good resistance to noise and can adapt to dynamic environments. Importantly, within the investigated settings, we find the adaptation to dynamic environments to be critical. In fact, we discover that the Flaky algorithm undergoes a phase transition from converging to a non-converging behaviour with respect to the optima. We find that the transition between these two behaviours depends on the system size, the rate at which the environment is changing, and the algorithm step size.

Next, to maintain the focus on the development of autonomous and self-improving materials in real-life situations, in Chapter 5 we have a peek at more complex circumstances deriving from the motion of our system on a plane. Here we develop a simplified, qualitative model of robotic metamaterial where units are placed and assembled in a square lattice. We apply the Flaky decentralised learning strategy to the units. With the goal of benchmarking future experiments, we look at the learning capabilities in a variety of circumstances such as obstacles avoidance, locomotion in various shapes, resistance to damage and increase in size. While we observe differences in the type of motion and maximum speed, despite the considerable increase in complexity, we find that in all cases the ability of learning is maintained. Furthermore we find that the great adaptivity of the Flaky algorithms which allow e.g. the systems to navigate around obstacles, comes at the cost of not being able to exploit good locomotion patterns.

We believe the work discussed in this thesis emphasizes the importance of cellular design for novel metamaterials, where unit-to-unit interaction can coalesce to form novel and complex behaviours. Specifically in the robotic metamaterials prospect, the results and discussions presented in this thesis set the ground for a

paradigm shift where metamaterials can improve themselves during operation, offering a leap forward for the development of thriving ubiquitous autonomous systems.

6.2 Recommendations

The inverse design algorithm we propose in Chapter 2 is a general algorithm to tackle disjoint optimization problems. In this case, the advantage of a simple implementation comes with the drawback of high computational cost. A possible improvement can be made by considering multiple material-void swaps per optimization step, in order to accelerate the material clustering of the initial configurations. This effect can be gradually reduced towards the end of optimization, where smaller geometrical changes are required. Furthermore, while we found that the Ising-inspired subroutine works well for our purposes, different candidate selection subroutine can be chosen to tackle different optimization problems. This is additionally made possible by the modular nature of this algorithm, which decouple the optimization routine from the candidate selection subroutine.

Within the context of robotic metamaterials, in this thesis we present only a few decentralised learning algorithms, and test them on numerical and experimental platforms. In Chapter 4 we find that highly dynamic circumstances prevent the Flaky algorithm from learning. We show that this effect depends on the system size, the rate at which the environment is changing, and the algorithm step size. Within the current decentralised settings, only the last information is available to the units. As a consequence, we believe that an interesting direction to explore is the development of an algorithm with adaptable step size and increased memory. In fact, while a large step size allow for prompt variations in behaviours, a small step size help exploiting good behaviours. By adopting an algorithm with adaptable step size, the system would be able to further tune its behaviour finding an optimal balance between exploration and exploitation. This is especially needed when the system is moving in 2D, where system's rotation during continuous learning alters the landscape of possible behaviours, effectively creating a dynamic environment.

Importantly, in this work we are interested in defining the minimal requirements that allow robotic metamaterials to learn a certain behaviour, and have purposely considered distributed sensing and decision making, with no communication between units. This is to explore essential requirements for algorithms that can be translated to different material systems. We believe this choice could be revised without going towards fully centralised systems. In fact, local communication between neighbouring unit could be implemented, where e.g. the sensed information is averaged across a few units, before being processed by the learning algorithm.

6.3 Outlook

In this section we highlight three key elements of this work, and emphasize how their contribution will shape materials in the future.

Inverse design or continuous learning?

As the human progress advances at an ever-growing pace, the need of promptly customizing solutions to a variety of different problems and constrains becomes essential. This calls for the development of fundamental scientific understanding, and high-tech engineering applications. We believe that automated design processes are both a tool and a solution to these needs. While we can inversely design optimal solutions for specific problems, by embracing the fact that circumstances are in general not known beforehand, the development of continuously optimizing systems can be itself the solution to families of problems. For example, while in this thesis we focused on learning locomotion by automatically taking the robotic metamaterial's configuration or environment circumstances, one could imagine other applications such as a robotic metamaterial that learns to morph its shape around various objects. This is a fundamental aspect in transitioning from mechanical to robotic metamaterials, where the optimal behaviour is adapted to the dynamic circumstances.

Algorithm design: performance or adaptation?

The development of automated design routines goes hand in hand with the development of novel algorithms, and their performances has been of prime importance for the computer science and engineering communities. Unfortunately, striving for efficiency has also lead to a shift in focus away from the algorithms purposes, reducing their improvement to mere numerical exercises. This led to the creation of algorithms with tens, if not more, parameters specific to a particular application, favouring fragmentation between fields. Differently, in order to be able to focus rapidly on the growing number of challenges, we need to develop simpler algorithms, even if this comes at the cost of efficiency and precision. In a way, a similar trend is observed in the field of soft robotics, where high speed and precise limited degree of freedom robots are being challenged by soft robots with lower accuracy, but higher adaptivity. This simplifications will nourish collaborations, help maintain a comprehensive view on the matter, and allow for a steady coordinated scientific development. Furthermore, in an effort to find generic approaches, great inspiration can be found by crossing boundaries with other fields such as physics and biology, where extensively studied phenomena can provide the fundamental twist in finding solutions to novel challenges.

Materials with robotic behaviour

We believe that progressively, all materials will become 'smarter', by assuming e.g. shapes which grants them novel features. Sharing the same drive for improvement, some metamaterials will eventually turn to robotic metamaterials, where the optimal behaviour is adapted to the dynamic circumstances. Materials will be designed to continuously improve themselves. But how will these materials look? In this thesis we developed a macroscopic platform to study the potential learning algorithms. Future steps will look into down scaling these systems, e.g. by making dedicated electronic circuits that directly integrate the algorithms without the need for code. That brings up the question whether the learning behaviour can be encoded via other means such as chemical processes that sense and activate materials (e.g. liquid crystal elastomers). We envision robotic metamaterial systems consisting of hundreds of independent building blocks, which due to their reduced size, can have a perception of the environment at a higher resolution. This will result in enhanced system's adaptivity and increased embodied intelligence. The 'only' question to answer is: how to incorporate objectives at these scales, and in a decentralised approach?

Definitions

In this section we define few of the recurrently used terminology.

Optimal behaviour

Within the work on robotic metamaterials, and under the assumption that information on the working condition is partially available to the individual units, optimal behaviour is intended in a heuristic term, where "good enough" performance is optimal. Such that the task is executed, even if not with the best performance.

Adaptivity and Robust behaviour

Considering the general operational settings of the robotic metamatateriał, where circumstances are dynamic, optimal behaviours change over time. Then, adaptivity describes the ability of the learning strategy to adapt to mutable circumstances, and maintain the ability to learn and improve the performance of the system. If a learning strategy is adaptive with respect to e.g. variable friction or damage, it exhibits a robust behaviour with respect these factors.

Scalability

Scalability can refer to the robotic platform and to the learning strategy. A robotic platform can be said scalable if it can physically move when more units are connected together. The learning strategy can be scalable, if it maintains the ability to learn optimal behaviours when multiple agents contribute to the overall behaviour.

Bibliography

1. Bose, J. C. On the rotation of plane of polarisation of electric wave by a twisted structure. *Proceedings of the Royal Society of London* **63,** 146–152 (1898).
2. Landau, L. D. & Lifshitz, E. M. *Course of Theoretical Physics Vol 7: Theory and Elasticity* (Pergamon press, 1959).
3. Stavroulakis, G. Auxetic behaviour: appearance and engineering applications. *physica status solidi (b)* **242,** 710–720 (2005).
4. Robert, F. An isotropic three-dimensional structure with Poisson's ratio=- 1. *Journal of elasticity* **15,** 427–430 (1985).
5. Kolpakov, A. Determination of the average characteristics of elastic frameworks. *Journal of Applied Mathematics and Mechanics* **49,** 739–745 (1985).
6. Lakes, R. Foam Structures with a Negative Poisson's Ratio. *Science* **235,** 1038–1040 (1987).
7. Gibson, L. J. & Ashby, M. F. *Cellular solids: structure and properties* (Cambridge university press, 1999).
8. Milton, G. W. Composite materials with Poisson's ratios close to—1. *Journal of the Mechanics and Physics of Solids* **40,** 1105–1137 (1992).
9. Friis, E., Lakes, R. & Park, J. Negative Poisson's ratio polymeric and metallic foams. *Journal of Materials Science* **23,** 4406–4414 (1988).
10. Evans, K. E., Nkansah, M., Hutchinson, I. & Rogers, S. Molecular network design. *Nature* **353,** 124–124 (1991).
11. Hengsbach, S. & Lantada, A. D. Direct laser writing of auxetic structures: present capabilities and challenges. *Smart materials and structures* **23,** 085033 (2014).
12. Bückmann, T. *et al.* Tailored 3D mechanical metamaterials made by dip-in direct-laser-writing optical lithography. *Advanced Materials* **24,** 2710–2714 (2012).
13. Carneiro, V., Meireles, J. & Puga, H. Auxetic materials—A review. *Materials Science-Poland* **31,** 561–571 (2013).
14. Kolken, H. M. & Zadpoor, A. Auxetic mechanical metamaterials. *RSC advances* **7,** 5111–5129 (2017).
15. Prawoto, Y. Seeing auxetic materials from the mechanics point of view: a structural review on the negative Poisson's ratio. *Computational Materials Science* **58,** 140–153 (2012).

16. Alderson, A. A triumph of lateral thought. *Chemistry & Industry* **17,** 384–391 (1999).

17. Schwerdtfeger, J., Heinl, P., Singer, R. & Körner, C. Auxetic cellular structures through selective electron-beam melting. *physica status solidi (b)* **247,** 269–272 (2010).

18. Sanami, M., Ravirala, N., Alderson, K. & Alderson, A. Auxetic materials for sports applications. *Procedia Engineering* **72,** 453–458 (2014).

19. Saxena, K. K., Das, R. & Calius, E. P. Three decades of auxetics research-materials with negative Poisson's ratio: a review. *Advanced Engineering Materials* **18,** 1847–1870 (2016).

20. Scarpa, F. Auxetic materials for bioprostheses [In the Spotlight]. *IEEE Signal Processing Magazine* **25,** 128–126 (2008).

21. Duncan, O., Foster, L., Senior, T., Alderson, A. & Allen, T. Quasi-static characterisation and impact testing of auxetic foam for sports safety applications. *Smart Materials and Structures* **25,** 054014 (2016).

22. Bertoldi, K., Vitelli, V., Christensen, J. & Van Hecke, M. Flexible mechanical metamaterials. *Nature Reviews Materials* **2,** 1–11 (2017).

23. Timoshenko, S. P. & Gere, J. M. *Theory of elastic stability* (Courier Corporation, 2009).

24. Bažant, Z. P. Structural stability. *International Journal of Solids and Structures* **37,** 55–67 (2000).

25. Reis, P. M. A perspective on the revival of structural (in) stability with novel opportunities for function: from buckliphobia to buckliphilia. *Journal of Applied Mechanics* **82** (2015).

26. Bertoldi, K. & Boyce, M. C. Mechanically triggered transformations of phononic band gaps in periodic elastomeric structures. *Phys. Rev. B* **77,** 052105 (5 Feb. 2008).

27. Bertoldi, K., Reis, P. M., Willshaw, S. & Mullin, T. Negative Poisson's ratio behavior induced by an elastic instability. *Advanced materials* **22,** 361–366 (2010).

28. Kang, S. H. *et al.* Complex ordered patterns in mechanical instability induced geometrically frustrated triangular cellular structures. *Physical review letters* **112,** 098701 (2014).

29. Florijn, B., Coulais, C. & van Hecke, M. Programmable mechanical metamaterials: the role of geometry. *Soft matter* **12,** 8736–8743 (2016).

30. Overvelde, J. T. & Bertoldi, K. Relating pore shape to the non-linear response of periodic elastomeric structures. *Journal of the Mechanics and Physics of Solids* **64,** 351–366 (2014).

31. Coulais, C., Sabbadini, A., Vink, F. & van Hecke, M. Multi-step self-guided pathways for shape-changing metamaterials. *Nature* **561,** 512–515 (2018).
32. Mullin, T., Willshaw, S. & Box, F. Pattern switching in soft cellular solids under compression. *Soft Matter* **9,** 4951–4955 (2013).
33. Overvelde, J. T., Shan, S. & Bertoldi, K. Compaction through buckling in 2D periodic, soft and porous structures: effect of pore shape. *Advanced Materials* **24,** 2337–2342 (2012).
34. Shim, J. *et al.* Harnessing instabilities for design of soft reconfigurable auxetic/chiral materials. *Soft Matter* **9,** 8198–8202 (2013).
35. Coulais, C., Sounas, D. & Alù, A. Static non-reciprocity in mechanical metamaterials. *Nature* **542,** 461–464 (2017).
36. Ion, A., Wall, L., Kovacs, R. & Baudisch, P. *Digital mechanical metamaterials* in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (2017), 977–988.
37. Coulais, C., Teomy, E., De Reus, K., Shokef, Y. & Van Hecke, M. Combinatorial design of textured mechanical metamaterials. *Nature* **535,** 529 (2016).
38. Johnson, C. G., Jain, U., Hazel, A. L., Pihler-Puzović, D. & Mullin, T. On the buckling of an elastic holey column. *Proc. R. Soc. A* **473,** 20170477 (2017).
39. Bertoldi, K., M., R. P., Stephen, W. & Tom, M. Negative Poisson's Ratio Behavior Induced by an Elastic Instability. *Advanced Materials* **22,** 361–366 (2010).
40. Florijn, B., Coulais, C. & Van Hecke, M. Programmable mechanical metamaterials. *Physical Review Letters* **113** (2014).
41. Ronellenfitsch, H., Stoop, N., Yu, J., Forrow, A. & Dunkel, J. Inverse design of discrete mechanical metamaterials. *Physical Review Materials* **3,** 095201 (2019).
42. Ferrari, F. & Sigmund, O. Revisiting topology optimization with buckling constraints. *Structural and Multidisciplinary Optimization* **59,** 1401–1415 (2019).
43. Siddhant, K., Tan, S., Li, Z. & Kochmann, D. M. Inverse-designed spinodoid metamaterials. *NPJ Computational Materials* **6** (2020).
44. Clausen, A., Wang, F., Jensen, J. S., Sigmund, O. & Lewis, J. A. Topology optimized architectures with programmable Poisson's ratio over large deformations. *Advanced Materials* **27,** 5523–5527 (2015).
45. Ion, A. *et al. Metamaterial mechanisms* in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (2016), 529–539.
46. Abelson, H. *et al.* Amorphous computing. *Communications of the ACM* **43,** 74–82 (2000).

47. McEvoy, M. A. & Correll, N. Materials that couple sensing, actuation, computation, and communication. *Science* **347** (2015).

48. Hughes, D., Heckman, C. & Correll, N. Materials that make robots smart. *The International Journal of Robotics Research* **38,** 1338–1351 (2019).

49. Brandenbourger, M., Locsin, X., Lerner, E. & Coulais, C. Non-reciprocal robotic metamaterials. *Nature communications* **10,** 1–8 (2019).

50. Scheibner, C. *et al.* Odd elasticity. *Nature Physics* **16,** 475–480 (2020).

51. Ghatak, A., Brandenbourger, M., van Wezel, J. & Coulais, C. Observation of non-Hermitian topology and its bulk–edge correspondence in an active mechanical metamaterial. *Proceedings of the National Academy of Sciences* (2020).

52. Nash, L. M. *et al.* Topological mechanics of gyroscopic metamaterials. *Proceedings of the National Academy of Sciences* **112,** 14495–14500 (2015).

53. Smith, D. R., Pendry, J. B. & Wiltshire, M. C. K. Metamaterials and Negative Refractive Index. *Science* **305,** 788–792 (2004).

54. Pendry, J. B. Negative refraction makes a perfect lens. *Physical Review Letters* **85,** 3966–3969 (2000).

55. Zheng, X. *et al.* Ultralight, Ultrastiff Mechanical Metamaterials. *Science* **344,** 1373–1377 (2014).

56. Ergin, T., Stenger, N., Brenner, P., Pendry, J. B. & Wegener, M. Three-dimensional invisibility cloak at optical wavelengths. *Science (New York, N.Y.)* **328,** 337–9 (2010).

57. Bückmann, T., Kadic, M., Schittny, R. & Wegener, M. Mechanical cloak design by direct lattice transformation. *Proceedings of the National Academy of Sciences* **112,** 4930–4934 (2015).

58. Wei, Z. Y., Guo, Z. V., Dudte, L., Liang, H. Y. & Mahadevan, L. Geometric mechanics of periodic pleated origami. *Physical Review Letters* **110** (2013).

59. Overvelde, J. T. *et al.* A three-dimensional actuated origami-inspired transformable metamaterial with multiple degrees of freedom. *Nature communications* **7,** 10929 (2016).

60. Holmes, D. P. & Crosby, A. J. Snapping surfaces. *Advanced Materials* **19,** 3589–3593 (2007).

61. Li, J. *et al.* Switching periodic membranes via pattern transformation and shape memory effect. *Soft Matter* **8,** 10322 (2012).

62. Wang, P., Shim, J. & Bertoldi, K. Effects of geometric and material nonlinearities on tunable band gaps and low-frequency directionality of phononic crystals. *Physical Review B - Condensed Matter and Materials Physics* **88** (2013).

63. Yang, W.-P. & Chen, L.-W. The tunable acoustic band gaps of two-dimensional phononic crystals with a dielectric elastomer cylindrical actuator. *Smart materials and structures* **17,** 015011 (2007).

64. Terwagne, D., Brojan, M. & Reis, P. M. Smart morphable surfaces for aerodynamic drag control. *Advanced Materials* **26,** 6608–6611 (2014).

65. Lin, P.-C. & Yang, S. Mechanically switchable wetting on wrinkled elastomers with dual-scale roughness. *Soft Matter* **5,** 1011–1018 (2009).

66. Kim, J., Yoon, J. & Hayward, R. C. Dynamic display of biomolecular patterns through an elastic creasing instability of stimuli-responsive hydrogels. *Nature materials* **9,** 159–64 (2010).

67. Schulz, M. & Pellegrino, S. Equilibrium paths of mechanical systems with unilateral constraints I. Theory. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* **456,** 2223–2242 (2000).

68. Cook, R. D., Malkus, D. S., Plesha, M. E. & Witt, R. J. *Concepts and Applications of Finite Element Analysis* (John Wiley & Sons, 2007).

69. Shim, J. *et al.* Harnessing instabilities for design of soft reconfigurable auxetic/chiral materials. *Soft Matter* **9,** 8198–8202 (34 2013).

70. Sicong, S. *et al.* Harnessing Multiple Folding Mechanisms in Soft Periodic Structures for Tunable Control of Elastic Waves. *Advanced Functional Materials* **24,** 4935–4942 (2014).

71. Bendsøe, M. P. & Kikuchi, N. Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering* **71,** 197–224 (1988).

72. Sigmund, O. & Maute, K. Topology optimization approaches. *Structural and Multidisciplinary Optimization* **48,** 1031–1055 (Dec. 2013).

73. Challis, V., Roberts, A. & Wilkins, A. Design of three dimensional isotropic microstructures for maximized stiffness and conductivity. *International Journal of Solids and Structures* **45,** 4130–4146 (2008).

74. Neves, M. M., Rodrigues, H. & Guedes, J. M. Generalized topology design of structures with a buckling load criterion. *Structural optimization* **10,** 71–78 (Oct. 1995).

75. Dobson, D. C. & Cox, S. J. Maximizing Band Gaps in Two-Dimensional Photonic Crystals. *SIAM Journal on Applied Mathematics* **59,** 2108–2120 (1999).

76. Sigmund, O. & Søndergaard Jensen, J. Systematic design of phononic band–gap materials and structures by topology optimization. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* **361,** 1001–1019 (2003).

77. Jensen, J. S. & Sigmund, O. Systematic design of photonic crystal structures using topology optimization: Low-loss waveguide bends. *Applied Physics Letters* **84,** 2022–2024 (2004).

78. Sigmund, O. & Hougaard, K. Geometric Properties of Optimal Photonic Crystals. *Phys. Rev. Lett.* **100,** 153904 (15 Apr. 2008).

79. Halkjær, S., Sigmund, O. & Jensen, J. S. Inverse design of phononic crystals by topology optimization. *Zeitschrift für Kristallographie-Crystalline Materials* **220,** 895–905 (2005).

80. Sigmund, O. & Torquato, S. Composites with extremal thermal expansion coefficients. *Applied Physics Letters* **69,** 3203–3205 (1996).

81. Deng, H., Cheng, L., Liang, X., Hayduke, D. & To, A. C. Topology optimization for energy dissipation design of lattice structures through snap-through behavior. *Computer Methods in Applied Mechanics and Engineering* **358,** 112641 (2020).

82. Borrvall, T. & Petersson, J. Topology optimization of fluids in Stokes flow. *International journal for numerical methods in fluids* **41,** 77–107 (2003).

83. Gao, X. & Ma, H. Topology optimization of continuum structures under buckling constraints. *Computers & Structures* **157,** 142–152 (2015).

84. Thomsen, C. R., Wang, F. & Sigmund, O. Buckling strength topology optimization of 2D periodic materials based on linearized bifurcation analysis. *Computer Methods in Applied Mechanics and Engineering* **339,** 115–136 (2018).

85. Sigmund, O. On the usefulness of non-gradient approaches in topology optimization. *Structural and Multidisciplinary Optimization* **43,** 589–596 (2011).

86. Shim, P. Y. & Manoochehri, S. Generating optimal configurations in structural design using simulated annealing. *International journal for numerical methods in engineering* **40,** 1053–1069 (1997).

87. Baumann, B. & Kost, B. Structure assembling by stochastic topology optimization. *Computers & Structures* **83,** 2175–2184 (2005).

88. Rocks, J. W. *et al.* Designing allostery-inspired response in mechanical networks. *Proceedings of the National Academy of Sciences* **114,** 2520–2525 (2017).

89. Goodrich, C. P., Liu, A. J. & Nagel, S. R. The principle of independent bond-level response: Tuning by pruning to exploit disorder for global behavior. *Physical review letters* **114,** 225501 (2015).

90. Hinkley, D. V. & Cox, D. *Theoretical statistics* (Chapman and Hall/CRC, 1979).

91. Pihler-Puzović, D., Hazel, A. & Mullin, T. Buckling of a holey column. *Soft matter* **12,** 7112–7118 (2016).

92. Seyranian, A. P., Lund, E. & Olhoff, N. Multiple eigenvalues in structural optimization problems. *Structural optimization* **8,** 207–227 (1994).

93. Bendsøe, M. P. Optimal shape design as a material distribution problem. *Structural optimization* **1,** 193–202 (1989).

94. Sigmund, M. P. B. *Topology Optimization Theory, Methods, and Applications* (Springer, 2004).

95. Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. Optimization by simulated annealing. *Science* **220,** 671–680 (1983).

96. Nourani, Y. & Andresen, B. A comparison of simulated annealing cooling strategies. *Journal of Physics A: Mathematical and General* **31,** 8373 (1998).

97. Olhoff, N., Bendsøe, M. P. & Rasmussen, J. On CAD-integrated structural topology and design optimization. *Computer Methods in Applied Mechanics and Engineering* **89,** 259–279 (1991).

98. Sigmund, O. A 99 line topology optimization code written in Matlab. *Structural and multidisciplinary optimization* **21,** 120–127 (2001).

99. Sigmund, O. & Petersson, J. Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural optimization* **16,** 68–75 (1998).

100. Diaz, A. & Sigmund, O. Checkerboard patterns in layout optimization. *Structural optimization* **10,** 40–45 (1995).

101. Sigmund, O. *Design of material structures using topology optimization* PhD thesis (Technical University of Denmark Denmark, 1994).

102. Poulsen, T. A. A simple scheme to prevent checkerboard patterns and one-node connected hinges in topology optimization. *Structural and Multidisciplinary Optimization* **24,** 396–399 (2002).

103. Rouhi, M., Rais-Rohani, M. & Williams, T. N. Element exchange method for topology optimization. *Structural and Multidisciplinary Optimization* **42,** 215–231 (2010).

104. Ising, E. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik* **31,** 253–258 (1925).

105. Brush, S. G. History of the Lenz-Ising model. *Reviews of modern physics* **39,** 883 (1967).

106. Budiansky, B. Theory of buckling and post-buckling behavior of elastic structures. *Advances in applied mechanics.* **14,** 1–65 (1974).

107. Johnson, S. C. Hierarchical clustering schemes. *Psychometrika* **32,** 241–254 (1967).

108. Hohlfeld, E. & Mahadevan, L. Unfolding the sulcus. *Physical review letters* **106,** 105702 (2011).

109. Hwangbo, J. *et al.* Learning agile and dynamic motor skills for legged robots. *Science Robotics* **4,** eaau5872 (2019).

110. Kwiatkowski, R. & Lipson, H. Task-agnostic self-modeling machines. *Science Robotics* **4,** eaau9354 (2019).

111. Rus, D. & Tolley, M. T. Design, fabrication and control of soft robots. *Nature* **521,** 467–475 (2015).

112. Cully, A., Clune, J., Tarapore, D. & Mouret, J.-B. Robots that can adapt like animals. *Nature* **521,** 503–507 (2015).

113. Haarnoja, T. *et al.* Learning to walk via deep reinforcement learning. *arXiv preprint arXiv:1812.11103* (2018).

114. Ficuciello, F., Migliozzi, A., Laudante, G., Falco, P. & Siciliano, B. Vision-based grasp learning of an anthropomorphic hand-arm system in a synergy-based control framework. *Science Robotics* **4,** eaao4900 (2019).

115. Mahler, J. *et al.* Learning ambidextrous robot grasping policies. *Science Robotics* **4,** eaau4984 (2019).

116. Fazeli, N. *et al.* See, feel, act: Hierarchical learning for complex manipulation skills with multisensory fusion. *Science Robotics* **4,** eaav3123 (2019).

117. Wehner, M. *et al.* An integrated design and fabrication strategy for entirely soft, autonomous robots. *Nature* **536,** 451–455 (2016).

118. Majidi, C. Soft-Matter Engineering for Soft Robotics. *Advanced Materials Technologies* **4,** 1800477 (2019).

119. Tolley, M. T. *et al.* A resilient, untethered soft robot. *Soft Robotics* **1,** 213–223 (2014).

120. Chen, T., Bilal, O. R., Shea, K. & Daraio, C. Harnessing bistability for directional propulsion of soft, untethered robots. *Proceedings of the National Academy of Sciences* **115,** 5698–5702 (2018).

121. Howard, D. *et al.* Evolving embodied intelligence from materials to machines. *Nature Machine Intelligence* **1,** 12–19 (2019).

122. Laschi, C. & Cianchetti, M. Soft robotics: new perspectives for robot bodyware and control. *Frontiers in Bioengineering and Biotechnology* **2,** 3 (2014).

123. Mengüç, Y., Correll, N., Kramer, R. & Paik, J. Will robots be bodies with brains or brains with bodies? *Science Robotics* **2,** eaar4527 (2017).

124. Shepherd, R. F. *et al.* Multigait soft robot. *Proceedings of the National Academy of Sciences* **108,** 20400–20403 (2011).

125. Williamson, R. & Chrachri, A. Cephalopod neural networks. *Neurosignals* **13,** 87–98 (2004).

126. Slavkov, I. *et al.* Morphogenesis in robot swarms. *Science Robotics* **3,** eaau9178 (2018).

127. Li, S. *et al.* Particle robotics based on statistical mechanics of loosely coupled components. *Nature* **567,** 361 (2019).

128. Brambilla, M., Ferrante, E., Birattari, M. & Dorigo, M. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence* **7,** 1–41 (2013).

129. Rubenstein, M., Cornejo, A. & Nagpal, R. Programmable self-assembly in a thousand-robot swarm. *Science* **345,** 795–799 (2014).

130. Peleg, O., Peters, J. M., Salcedo, M. K. & Mahadevan, L. Collective mechanical adaptation of honeybee swarms. *Nature Physics* **14,** 1193 (2018).

131. Romanishin, J. W., Gilpin, K., Claici, S. & Rus, D. *3D M-Blocks: Self-reconfiguring robots capable of locomotion via pivoting in three dimensions* in *2015 IEEE International Conference on Robotics and Automation (ICRA)* (2015), 1925–1932.

132. Augugliaro, F. *et al.* The flight assembled architecture installation: Cooperative construction with flying machines. *IEEE Control Systems Magazine* **34,** 46–64 (2014).

133. Shimizu, M., Ishiguro, A. & Kawakatsu, T. *Slimebot: A modular robot that exploits emergent phenomena* in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation* (2005), 2982–2987.

134. Leottau, D. L., Ruiz-del-Solar, J. & Babuška, R. Decentralized reinforcement learning of robot behaviors. *Artificial Intelligence* **256,** 130–159 (2018).

135. Christensen, D. J., Schultz, U. P. & Stoy, K. A distributed and morphology-independent strategy for adaptive locomotion in self-reconfigurable modular robots. *Robotics and Autonomous Systems* **61,** 1021–1035 (2013).

136. Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics* **21,** 1087–1092 (1953).

137. Seok, S. *et al.* Meshworm: a peristaltic soft robot with antagonistic nickel titanium coil actuators. *IEEE/ASME Transactions on Mechatronics* **18,** 1485–1497 (2012).

138. Boxerbaum, A. S., Shaw, K. M., Chiel, H. J. & Quinn, R. D. Continuous wave peristaltic motion in a robot. *The International Journal of Robotics Research* **31,** 302–318 (2012).

139. Kotikian, A. *et al.* Untethered soft robotic matter with passive control of shape morphing and propulsion. *Science Robotics* **4,** 7044 (2019).

140. Miskin, M. Z. *et al.* Graphene-based bimorphs for micron-sized, autonomous origami machines. *Proceedings of the National Academy of Sciences* **115,** 466–470 (2018).

141. Ceylan, H., Giltinan, J., Kozielski, K. & Sitti, M. Mobile microrobots for bioengineering applications. *Lab on a Chip* **17,** 1705–1724 (2017).

142. Medina-Sánchez, M., Magdanz, V., Guix, M., Fomin, V. M. & Schmidt, O. G. Swimming microrobots: Soft, reconfigurable, and smart. *Advanced Functional Materials* **28,** 1707228 (2018).

143. Rothemund, P. *et al.* A soft, bistable valve for autonomous control of soft actuators. *Science Robotics* **3** (2018).

144. Zhang, H., Zeng, H., Priimagi, A. & Ikkala, O. Programmable responsive hydrogels inspired by classical conditioning algorithm. *Nature communications* **10,** 1–8 (2019).

145. Oh, K. W., Lee, K., Ahn, B. & Furlani, E. P. Design of pressure-driven microfluidic networks using electric circuit analogy. *Lab on a Chip* **12,** 515–545 (2012).

146. Armstrong, B. & de Wit, C. *Friction Modeling and Compensation, The Control Handbook* (CRC Press, 1995).

147. Irwin, J. O. On the frequency distribution of the means of samples from a population having any law of frequency with finite moments, with special reference to Pearson's Type II. *Biometrika,* 225–239 (1927).

148. Hall, P. The distribution of means for samples of size n drawn from a population in which the variate takes values between 0 and 1, all such values being equally probable. *Biometrika,* 240–245 (1927).

149. Golomb, S. W. *Polyominoes: puzzles, patterns, problems, and packings* (Princeton University Press, 1996).

150. Rohrich, R. *et al.* Uncertainty estimation and design optimization of 2D diffraction-based overlay metrology targets. *ACS Photonics* **7,** 2765–2777 (2020).

Summary

Humankind has been morphing materials to their needs since its first appearance, and the adoption of different types of material became a reference measure of our evolution. While the importance of shape in materials has been vastly harnessed – e.g. a metal wheel as opposed to a metal knife – the idea that geometry alone could help surpass the limits of known laws in physics is relatively new. Metamaterials in fact, exhibit properties which were thought not to be achievable beforehand, where examples range from negative refraction index to bandgap materials. In the field of mechanics, this idea has led to metamaterials that show auxetic behaviour, tuneable mechanical responses and tailored acoustic properties. While these materials show complex behaviours, their geometry is especially simple, often composed of repeating identical unit cells, and usually designed through a trial-and-error process.

With the aim of being able to reprogram the behaviour of such materials, by carefully designing the arrangements of different types of building blocks, researchers were able to obtain materials capable of performing elemental computations, transforming a specific input actuation into a desired output response, creating materials capable of performing tasks. Furthermore, motivated by the idea that materials could be made of responsive building blocks whose features go beyond the mere geometrical mass distribution, the concept of robotic metamaterials emerged, broadening the definition of what is traditionally thought of as a material. For example, one could imagine a robotic material constantly adapting its shape to a varying surface, or robotic fabric changing its thermal properties based on local temperature measurements. Despite the recent developments in both mechanical and robotic metamaterials, defining what the geometrical and behavioural features of the unit cell should look like in order to get a desired function remain a difficult problem to solve. This is especially challenging in the latter case, where these features need to continuously adapt during operation.

In this thesis we focus on exploring different pathways to achieve metamaterials with specific and optimal properties. We do so by exploring reinforcement learning strategies to design shape and behaviour in elemental building blocks. In the first part of the thesis, we focus on the inverse design of mechanical metamaterials. The leveraging of instabilities, which is the key ingredient in many mechanical metamaterials, has shown to be a constraining factor in the automated design of such metamaterials. In this part of the thesis we use a Monte Carlo-based strategy to design metamaterials with targeted buckling properties, by modifying the unit cell topology. By complementing simulations with experiments, we show that we can inversely design and manufacture metamaterials with predefined buckling loads, that can be tuned within a wide range of values. We furthermore

show that by controlling the critical load of higher buckling modes, we can tune their interaction to avoid unstable and unpredictable metamaterial behaviours.

In the second part of the thesis we look at robotic metamaterials made of active unit cells, each containing sensing, actuation and decision-making capabilities. We are interested in exploring certain emerging properties in a dynamic environment, and define what the minimal requirements are, such that a robotic metamaterial can adapt to its surrounding. In this chapter, besides developing a 1D experimental robotic platform, we introduce two decentralized algorithms that are executed in parallel at the unit level, such that the complete robotic material is able to reach and maintain global locomotion in a predefined direction. Complementing these experiments with simulations, we show that while both algorithms are able to operate in static environments with no noise, the representation of memory becomes especially important to allow learning of optimal locomotion strategies under uncertainty, e.g. when dealing with sensor noise, damages, and changes in the environment.

To better understand the workings of the proposed algorithms and probe their limits with respect to various circumstances, we continue this study by removing the complexities arising from the system's physical interaction with the environment. To achieve this, we developed a simulation framework in which we specifically look at the performances of the strategies in terms of scalability, learning rate, permanence time around optima, and adaptability to changing environments. In an effort to emulate experimental settings, we also investigate and quantify the amount of noise the system can experience without effecting its learning capabilities.

Finally, we extend the application of the decentralized learning strategies into bi-dimensional robotic metamaterials. Here, we reintroduce the system's physical interaction with the environment, and with further experimental validation in mind, developed a simulation framework to study the emergence of locomotion in larger and arbitrary-shaped systems. Despite the increase in complexity arising from bridging to 2D, we show that the proposed learning strategy is capable of learning locomotion in a predefined direction, is able to circumvent obstacles and it is scalable to arbitrary shapes.

In conclusion, in this thesis we investigate how the features of elemental building blocks affect the properties and behaviour of metamaterials. We developed numerical models and experimental platforms to help blur the boundaries between materials and robots, in an effort to create reactive and autonomous systems that behave optimal in applications encountered in everyday life.

List of Publications

On work presented in this thesis:

Oliveri, G., Overvelde, J.T.B., *Inverse design of mechanical metamaterials that harness instabilities.* 2020, Advanced Functional Materials, 1909033.
(Chapter 2)

Oliveri, G., C. van Laake, L., Carissimo, C., Miette, C., Overvelde, J.T.B. *Continuous learning of emergent behavior in robotic matter.* In Review.
(Chapter 3)

Oliveri, G., Carissimo, C., Overvelde, J.T.B. *Reinforcement learning strategies in multi-agent systems.* In preparation.
(Chapter 4)

Oliveri, G., Schomaker, M., Overvelde, J.T.B. *Autonomous 2D robotic matter.* In preparation.
(Chapter 5)

On other work:

Röhrich, R., **Oliveri, G.**, Kovaios, S., Tenner, V.T., den Boef, A.J., Overvelde, J.T.B., Koenderink, A.F., *Uncertainty estimation and design optimization of 2D diffraction-based overlay metrology targets.* 2020, ACS Photonics.

Mulla, Y., **Oliveri, G.**, Overvelde, J. T. B., Koenderink, G. H., *Crack initiation in viscoelastic materials.* 2018, Physical Review Letters, 120, 268002.

Acknowledgments

This book is the tip of the iceberg, the visible and tangible outcome of my last four years as a PhD student. To understand how this book has made it to you, we need to look below the surface, and see what else this iceberg is made of. The last four years have been an incredible journey through science, and through myself. It has been a mix of pride and frustration, sense of accomplishment and scientific loneliness, joy and anger, sleepless nights of celebrations and sleepless nights of racing thoughts. While I can only appreciate one aspect by also knowing its counterpart, each of these aspects taught me something, helped me identifying myself, and understand better what my values are. Each and every moment contributed to this thesis, and I'm grateful for them. Although this thesis is the results of my scientific contributions and the ones of my collaborators, this was ultimately accomplished by the invaluable support, love and laughs of the many people who surrounded me closely during these years. Without them this thesis would not exists.

First, I would like to thank my supervisor Bas Overvelde. I'm very proud to be your first PhD student and that I could help starting your lab at AMOLF. Witnessing the group growth, and how you developed the many research lines currently active, has been of great inspiration for me. Thank you for your patience, and for always been there to discuss approaches and possible directions. I really appreciated your guidance in turning my optimisms into feasible plans with well defined objectives. I admire your positivity, and how quickly you come up with new ideas, and a straightforward action plan to accomplish them. Working with you has been a tremendous learning experience, thank you. Next, I would like to thank all the present and past members of the Soft Robotic Matter group. Each and everyone of you created the right atmosphere for discussing, learning and improving, while also making every moment fun and enjoyable. I'm already missing the coffee breaks and group meetings together. A big thank you also goes to the Designer Matter people. It has been very interesting to find a common ground between different research lines, and bring a new perspectives into our own researches. The poster sessions and presentation rounds have served perfectly at that, while also helping at adjusting the level to different audiences. Thank you Martin, Wim, Huib and Bas for creating this familiar environment, promoting collaborations and triggering discussions. This also applies to AMOLF as a whole. Working there has been a great privilege, and I feel honored that I could meet and interact with such creative and highly motivated people, while enjoying the feeling of belonging to the same family. A special thank you goes to Paula, Gijsje, Femius, Bela, Pieter Rein, and Said.

I also want to thank the committee members who dedicated time to read this thesis, assess its quality, and provide suggestions for improvements. Your time and effort has been gratefully appreciated, and your comments made this thesis a better one, thank you. I am looking forward discussing with you during the defense ceremony.

The next person I would like to thank is Agustin, my paranymph, partner in crime and PhD twin for the past 4 years. Starting a PhD just few months apart made us share a lot. You are a great companion, but most importantly a great friend. The time spent with you is fun and enriching, whether we are debugging code or getting ready for the next party. We've encouraged each other to be better scientist, helped each other to become better people, but also supported each other when things weren't going well. With the excuse of going to conferences we've had the greatest trips and lived amazing experiences, and even if you often complain about my sicilian driving style, I'm looking forward our next trip together. Dear man of surprises, dear Luuk, being around you the past years has been fantastic: your knowledge and immeasurable interests on basically anything is just amazing. You are an unstoppable source of energy and passion, thank you for changing the days for the better, and share your endless optimism. Maziar, behind your serious and impeccable mustache hides a smart and fun person, thank you for making the lunch breaks a moment not to miss. Mannus, working together has been great, and because of you I even enjoyed my own work more. Part of the results shown here is the outcome of our fantastic work together. Thank you for being such an interesting and open minded person, always ready to listen different opinions and brainstorm. You will continue working in this direction for you PhD, and I'm certain you will do great. Cesare, working with you has been enriching and fun, your abstraction capabilities are simply impressive. Thank you for offering me a glimpse on the dark side of theoretical computer science. I would also thank all the others students I supervised during my PhD: Mark, Leah, Clara, Joost, and Sarah. If you have learnt $1/10$ of what I've learnt from you, I did a great job. Thank you for your hard work and for making me a better mentor. Niels, my man! You have been a companion since day 0, your help has been invaluable! During pandemic you made true every PhD student's dream: running experiment from home! Thanks for all the help, the creative solutions, the dirty fixes, and all the time we spent together.

Jenny, this PhD would not have been the same without you. When Clyde moved me to your office it felt like being promoted. You are smart, powerful, fun and kind. Going to work felt like going to a party. We cheered for our achievements, shared our frustrations and laughed as nothing has ever happened. Thank you for being such a great friend, and sharing the dream of a Mediterranean office with view on the sea! And of course, the Mediterranean office was complete only when Javi stepped in. Dear Javi, since you arrived the office got to a whole new level. You are brilliant, hard working but always ready to share the latest

funny viral video the internet has to offer. Thank you for being such a kind person, always ready to stop whatever you are doing in order to help and support the people around you. I do think that you were sent to redeem us from the chaos to wisdom: you did a great job. I'm sure that your talent, determination, and passion for science will soon bring you to have your own research group in your beloved Madrid. I also want to thank Bruna for her being so genuine and fun, and of course all the deserts she has been preparing in the past years.

Lorenzone! My other paranymph! Since we became neighbours life has become more fun! Thank you for showing how great science can be combined with great fun! The spontaneous afternoon boat rides, SUP or frisbee sessions made the last year exciting and so refreshing. Your excitement towards new things is just amazing, and you always end up involving people in your crazy ideas. Thank you for all the cheerful moments spent together, for dreaming about starting our own company together with Ale, for listening and the always wise advice. It's your turn Giulia. First, thank you for introducing me (together with Nicola) to a whole new level of partying. In the past years you have become more and more my angelo custode: checking on me, always ready to listen and give advice. With your "aperitivo" approach, you managed to remind me and everyone else around you, to chill a bit more. You are great companion and friend, this journey wouldn't not be the same without you. Judith, thank you for all the laughs and all the mocking to Giulia. I'm looking forward more parties together. Alessandro, gordibueno italiano outside, sweetheart inside. I don't remember how we met, but as far as I'm concerned I've always known you. Thank you for being such a fun, kind, approachable and authentic friend. Without you all the parties would have been boring, and all the dances lame. Thank you for all the amazing moments we spent together, can't wait to live more. Dear Marco Saltini, since you left Amsterdam the sarcasm level of the city dropped dangerously. "ACTUALLY..", thank you for bringing your immeasurable knowledge everywhere you go, thank you for eating all the chips and candies at parties, thank you for your unique imitations, and by far, the most incredible acting skills in "de nezerlands". Thank you Federica for your modesty, for showing me how to fully take advantage of your free time, while also being a serial world-class traveler! Vanessa, you are the real intelleattuale impegnato, thank you for transmitting your ideals and passions in anything you do. The pride you show when talking about Italy, and Calabria is simply amusing. Being around you is always fun, and the "occhi a Pampinedda" inevitable. Domenico, you are bomb of energy, thank you for all great fun together!

Dear Lucie, the Drama Queen! Thank you for being such an amazing person, and for being so true, and show your real feelings. Thank you for always ending up dancing when you have a good news to share, and thank you for been there when I needed a hug! You are smart and determined, you will achieve whatever goal you set for yourself. Lucas, your knowledge, passion and dedication towards food is just enviable, thank you for that! Nasim, thank you for showing your

strength by showing your vulnerabilities. You are pragmatic, determined, and with a big smile always on, thank you for being a role model. Susan, thank you for the contagious humor your spread around AMOLF.

A special thanks goes to the Sicilian family at AMOLF. Andrea, my indentical twin, thank you for all the pizza evening we had together, and for all the endless discussions at the coffee corner. We will one day succeed at opening an IIT centre in Catania. Loreta, thank you for careful analysis on life, for being smart, funny and very Sicilian. You let me feel home everyday. You are doing great in science, but if you ever get bored, your panelle will bring you fame and glory! Silvia thank you for always being ready to share a laugh while running up and down the lab.

Dear Lukas, a big thank you goes to you for all the amazing time we spent together. Whether it's cooking, finding the optimal parameter settings for coffee-brewing, programming Arduino, or baking bread. You are an invaluable friend, thank you for always being by my side. I'm looking forward new adventures together. Carolyn, thank you for accompanying me from day 0 through this years with a big smile, with your kindness, and home made muffins. Thank you for the gluhwein parties, all the parties without gluhwein, and for keeping up the effort of going to the gym despite all the excuses I could come up with. Thank you Laura for bringing parties to a whole new level 4 years ago, and thank your for keeping showing up despite you moved to Berlin a while ago! Please, don't forget the NINININI! Gracias Roberto por ensenarme el maxicano de barrio mas bajo que se pueda. Thank you for helping me finding my limits and being such a good friend. Whether we are out cycling, drinking a coffee or at party, the time spent with you is always the best, thank you! Said, thank you for all the nice chats, the encouragements and for sharing your singular and interesting perspectives. My dear Iarik, we started together and finished together, thank you for being such a great confidant: talking and opening up to you has been so natural. Thank you for helping making up my mind, and for always being so kind. Hans, you are the first person I met during my PhD, thank you for loving to gather people together and organise many activities. When you are excited about something, everyone else does too: thank you for that! The bike-camping combo with Arno, Gede and Harshall was memorable, thank you for organizing it! Thank you for mocking me about my limited time with respect to the number of things I want to do, I started to act upon it! Arno, thank you for never stop surprising me! You are a smart and caring person, who will always do his best to help, thank you! Harshall, Harshall, Harshall, thank you for being such a cheerful friend, bumping in to you is always fun and a mood booster. Sven, thank you for being so talented, pragmatic and fun. Your advice is as dear as an older brother's, and the laughs with you memorable. Dear Eitan, thank you for being such a true, hilarious and interesting person to talk to you. Your chill attitude makes everyone around you relax.

Dear Ruslan, thank you for being unstoppable like a train. Thank you for being so interesting, kind and fun. Writing a paper together was a great pleasure, but

partying with you has been even a greater one! Isabelle, thank you for yours puns during lunch, and thank your for always giving the 200% when working on PV activities! Annemarie, thank you for being so genuinely interested in people, for being so spontaneous, and for all the fun we had by working in the PV, it would not have been the same without you!

A special thanks go to the cool kids who 'knew more' because they were in their 3rd year of PhD when I starred: Michele, Jacopo, Parisa, Agata and Nicola. You guys have been the inspiration on how to do a PhD. Michele, thank you for your afternoon poetry, your radiating passion for science, and for always finding the silver lining. You are amazing. Agata, thank you for being so authentic, smart and fun! Parisa, thank you for being a good friend full of advice, for being always ready to party hard, and of course, for leaving me your desk in Jenny's office. Thank you Jacopo for being such a passionate, interesting and fun person. A special thank you to Nicola, who taught me to drink cocktails from a bucket. Also, thank you for having this halo around you that makes me feel like I'm suddenly in Italy.

Great part of the fun at AMOLF was because of the PV. I'm pretty sure I'm the only PV member who had to take an oath to join, right Floortje? As PV leader you have been simply great, tough at times ;) , but always keeping the morale UP. You made us a team, and no matter how much more things were left to do, we were all in for it because of you, thank you! Thank you also for being an amazing friend. Always there for a laugh, hear a complain or just some random talk. Stopping by at your office is one my favourite activities. Of course this not only because of you, but also because of your amazing office mates: Teresa, Henriette and Karelia. Thank you for cheering up even the gloomiest day!

Dear Niels, Dion and Hinco, thank you for taking all the brainfarts, and turning them in actual working things. Your help is invaluable, your friendship even more. Thank you Dico and Marco for all the help with the Nworm project, it was really fun to work with you. A big thank you goes to the whole Design Department, where you ask for 2 and they give you 5, thank you Iliya, Dirk-Jan, Henk-Jan and Ricardo. I would also like to thank the whole Precision Manufacturing group, and especially Jan, Tom and Wessel. I'm constantly amazed by your work, and meeting you in the workshop is always a pleasure. Thank you for always been approachable, kind, and ready to cut through 5cm thick stainless steel plates without asking questions!

Thank you Cristina and Luca for all the positive vibes you give, for the food appreciations events, and for the invaluable advice on bread making! Dear Lennard, thank you for organising parties at my house, they always turn out to be great! Steven, thank you for being everywhere, and thank you for always carrying around you wonderful babies! They will grow party animals like you! Clyde, AMOLF without you would simply collapse. Thank you for always be there to solve problems, arrange stuff, and most importantly, thank you for being the

most fun person to bump into. GiovanniGiuzio TVB. Thank you for knowing everything about anything, and have a strong opinion about it. You are the only person who got me shouting at AMOLF, and I love you for that. Dear Ruben, thank you for being such a nice and interesting friend, always bringing a different perspective to the table. Thank you for being a loyal gym companion, with all the mandatory beer duties afterwards. Thank you Anne for your determination, passion for physics and for your stunning presentations. Thank you Marloes for your positivity and energy, you bring the good mood around AMOLF. Christian, thank you for being such a fun and interesting person. It's always a pleasure to drink an espresso with you! Thank you Giada for being the way you are. Dear Weiyi, I admire your passion, dedication, energy and humbleness, you are great! Thank you Rene for being such a kind and interesting person, bumping into you is always fun! Thank you Hedrien, Amitesh, Jiangnan, Moritz & Laura, Mario & Olga, Eliane, Magda, Tom, Verena, Ad, Marga, Juliette, Fotis, Manuel, Maraike, Jesse, Robin, Jente, Roel.

Dear Petra, thank you for being so friendly and caring. You are full of new ideas, and always ready to get inputs from others. Petra and Erny, thanks for organising all the events and lab tours, but also for being the catalyst of many new connections! Grace and Oana, thank you for being so nice and kind. I spent 90% of my PhD time in front of the PC, and this would not have been possible without the support from the ICT department, and especially without Wiebe, Richard, Rutger and Fatih. I enjoyed coming to your office to have some problems fixed, but also just to have a chat, thank you for that.

Thank you Bram for being fun, knowing ALL the statistics, and being such an interesting friend to be around with. I want to thank you Anej for being such a great friend, and always be there to share fun or tough moments. You are amazing, kind and true. Thank you Crys for all the laughs, and for the honest views from the outside. You are a great friend and companion on many funny adventures. I'm looking forward more craziness with you. Dear Sudip, thank you for all the mocking and your English lessons. Dear Gustavo, thank you for all the fun moments lived together, for the Mexican lessons and for your true friendship. Dear Dimitris, thank you for showing what the real tourist lifestyle is about, you are an inspiration. Thank you for all the laughs, for all the moments around food we had together, and for being a true friend. Dafni, thank you for being able to tame Dimitris, and for being so interesting, nice and fun. Dear Daniel, thank for spreading your wisdom and calmness around you. Thank you Cristina for simply showing how it's done. Thank you Roelof for all the nice and stimulating conversations you always bring to the table. Thank you Samantha for being so unique, interesting y muy chistosa. Thank you Marco for telling me that doing a PhD would be cool, thank you Roberta for believing in me!

Even if we are often far, your support has been encouraging, and your friendship always present. Thank you Ario for all our discussions about the PhD, we

will hopefully heal soon. Thank you for being a caring listener, for opening up about your frustrations and for seamlessly change the topic of discussion into absurdity. Thank you Barby for always being there to analyze the situation, give precious advice, and guard the temple. Grande Mina, thank you for showing how dedication and hard work lead to great achievements. Thank you Fede for being such a mentor and a fun friend, for showing your weaknesses without fear, and for sending 14 minutes audio messages. Dear Dan, thank you for all the fun adventures together, I'm looking forward having more. Thank you Paolo to show me that great things can only be achieved with great leisure. Thank you also to Alessia, Raffaele, Gabbro and Serena, Flavio, Dario and Chiara, OG, Mazzola, Alessandro, Fede Re, PeppeLonghi, Fede and Andreina. Being with you feels like time has never passed.

Un agradecimiento especial va para la banda de latinos de Wageningen. Gracias a todos por aceptar mi español sin juzgar, y por siempre traer el fuego a las fiestas. Carlos y Sylvain, gracias por todo el apoyo en la fase final de mi escritura, gracias por las buenas comidas y por las pequeñas reuniones, ustedes son muy divertidos. Dear Melba, gracias por siempre estar de buen humor, contar chismes y ser muy divertida. Diana, por se la mas nerd, y a Melisa por la buenas platicas.

Aunque están lejos, su apoyo y cariños me motivaron mucho e hicieron que todo se haya sentido posible. Salvador e Yvonne, gracias por escucharme cuando necesitaba hablar y gracias por ayudarme a que se me olvidaran las preocupaciones. Gracias por tratarme como a un hijo, ustedes son como mis papas. Sofí y Juan David, gracias por ser mis compañeritos de estudio, hacer chistes y traer la buena onda. El tiempo pasado con ustedes siempre es pura diversión. Gracias Don Rogelio por ensenarme las bases del ajedrez en las noches y gracias a la señora Olga por la diversión cocinando juntos. Gracias a todos por estar orgullosos de mí.

Cari mamma e papa', anche se sopratutto negli ultimi anni siamo stati fisicamente distanti, vi ho sentiti piu' vicini che mai. Al mio fianco avete combattuto il mio cattivo umore e il nervosismo con tenacia, sempre sapendo quello di cui avevo piu' bisogno: una chiamata in piu', a volte una chiamata in meno, o anche una semplice presa in giro. Grazie papino per avermi insegnato a dire sempre quello che penso, e battermi per quello io credo sia giusto. Grazie mammina per dimostrare che la pazienza ed il duro lavoro pagano sempre. Grazie Silvia per far si' che sia sempre una festa quando siamo insieme, e grazie Peppe per il tuo umorismo e positivita' che sempre trasmetti intorno a te. Sono veramente fortunato ad avervi accanto.

This journey would not be possible without you, Andrea. Thank you for the unconditional support through these years. You were there in the toughest moments, but also contributed in shaping the happiest ones. Thank you for helping me putting my life in perspective, for helping understanding my strengths and weakness, and guiding me on how to do better. The passion and talent you

bring in anything you do is simply disarming, it is for me a constant inspiration and a powerful driving force. Thank you for making things fun, for making my life a continuous surprise, for lighting me up, and for being the wonderful life partner you are. This thesis is for you.

About the Author

Giorgio Oliveri was born in 1989 in Catania. After obtaining the scientific high school diploma in 2008, uncertain whether to study physics or engineering, he eventually enrolls in a Bachelor of Science in Mechanical Engineering at the University of Catania. After enjoying the rather theoretical studies, for his bachelor thesis project he takes on an assignment in automation, where he designs and builds a mechanical testing machine for soft materials. Feeling the need for a jump towards the new, where discoveries are not only made thorough books, in 2013 he moves to The Netherlands. There, he enrolls in a Master of Science in Mechanical Engineering at the University of Twente, selecting the R&D track in mechanics. On the side, being convinced that great things can only be achieved in cooperation with others, Giorgio applies and gets accepted in the Master Honours Programme 'Change Leaders' of the University of Twente, where he works on obtaining knowledge, attitude and skills related to leadership.

Curious to have a taste of what working in the high-tech industry is, and eager to discover the Dutch pride from within, Giorgio obtains an internship project at ASML B.V., where he works on finite element modelling of air bearings. Giorgio loves challenges, and in 2015 together with his team, wins the TNO Student Challenge: Dynamics Structure. Thinking about what the future has to offer him, Giorgio decides to find out what doing research is like. With the seed of doubt still in his mind, he decides to work with physicists for his master thesis, and focuses on wave propagation on soft-stiff granular mixtures, a joint experimental research project between the Multiscale Mechanics group at the University of Twente, and the Daniels Lab at the North Carolina State University. Awarded with the Twente Mobility Fund scholarship, Giorgio has the wonderful opportunity of spending three month in the US under the supervision of prof. Karen Daniels, and six months under the supervision of prof. Stefan Luding and dr. Vanessa Magnanimo in Enschede. In June 2016 Giorgio defends his master thesis, and gets awarded with the *Cum Laude* distinction.

Looking for PhD positions, Giorgio gets particularly interested in metamaterials. Understanding how shape could grant novel features to traditional materials appeared to be an exiting direction to follow. At that time dr.ir. Bas Overvelde had just started looking for the first PhD student to join his brand new group at AMOLF. This was also the ultimate chance to blend in with physicists. During his PhD career, Giorgio enjoyed working on a variety of numerical and experimental projects, presented his work at multiple international conferences, and mentored 7 students with great passion. In his free time Giorgio enjoys traveling, cycling, preparing good food and talking to people.