HARMANNUS ALLARD HENRIK SCHOMAKER

EMBODIED COMPUTATION FOR EMERGENT GOAL-ORIENTED BEHAVIOR IN SOFT ROBOTS

AMOLF physics of functional complex matter

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

erc European Research Council
Established by the European Commission

NWO

Ph.D. Thesis, Eindhoven University of Technology, January 2026

Embodied computation for emergent goal-oriented behavior in soft robots
Harmannus Allard Henrik Schomaker

ISBN: 978-94-6537-083-5
A catalogue record is available from the Eindhoven University of Technology Library.
A digital version of this thesis can be downloaded from
https://amolf.nl and https://research.tue.nl

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

This work is part of the Dutch Research Council (NWO).

Printing: Ridderprint

Mannus Schomaker: *Embodied computation for emergent goal-oriented behavior in soft robots,* Januari 2026

EMBODIED COMPUTATION FOR EMERGENT GOAL-ORIENTED BEHAVIOR IN SOFT ROBOTS

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus prof. dr. S.K. Lenaerts, voor een commissie aangewezen door het College voor Promoties, in het openbaar te verdedigen op 7 januari 2026 om 16:00 uur

door

Harmannus Allard Henrik Schomaker

geboren te Amsterdam

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

Voorzitter:	prof.dr.ir. P.D. Anderson
Promotor:	dr.ir. J.T.B. Overvelde
Copromotor:	dr. S.R. Eugster
Promotiecommissieleden:	prof.dr. F. Toschi
	dr.ir. Y.B. van de Burgt
	prof.dr. G.C.H.E. De Croon (Technische Universiteit Delft)
	prof.dr. M. Dijkstra (Universiteit Utrecht)
	dr. J. Hughes (École Polytechnique Fédérale de Lausanne)

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

Dedicated to, Dorelies Schomaker, the heart behind all my perseverance and creativity, and to Veronique Schomaker and Annelies Frölke, for their warmth and support.

ABSTRACT

Soft organisms in nature rely on rhythmic patterns of motion that are often distributed across their bodies rather than centrally controlled. Inspired by this principle, this thesis investigates how decentralized coordination and embodied computation can be harnessed in soft robots to generate purposeful, adaptive locomotion. Instead of relying on a central controller, we explore how feedback from local body-environment interactions and minimal computation can produce robust, goal-directed behaviors. Across design, simulation, and experiments, we demonstrate how oscillatory sequences, when physically embodied in the system, can guide locomotion in unstructured and dynamic environments.

We begin by studying limbed soft robots inspired by echinoderms. Each limb operates as a self-contained module with limited sensing, memory, and computation, and through repeated rhythmic actuation, coordinated motion toward a stimulus (phototaxis) emerges. The interaction of limbs with their environment allows the system to dynamically re-coordinate movement in response to changing conditions or damage, without any central controller.

To deepen the understanding of this decentralized coordination, we then study a system of immobile modules that act through expansion and contraction of their connections in a two-dimensional grid. Despite their local limitations, the modules collectively break symmetry to achieve locomotion, similar to peristaltic motion in worms. Simulations and experiments show how physical connections implicitly mediate communication, enabling the system to adapt its sequences to different configurations and dynamic environments. These results highlight how morphology and coordination strategies are fundamentally linked.

Next, we remove electronics entirely and instead realize rhythmic actuation sequences using fluidic circuits. By designing soft relaxation oscillators with directional coupling, we emulate biological central pattern generators in hardware. These circuits allow for reprogrammable sequences and timings, providing a physically embodied control architecture that eliminates the need for electronic coordination.

Finally, we explore how minimal controllers and body morphology can co-evolve. Using simulations of coupled oscillator networks integrated with evolving soft robotic bodies, we demonstrate how even simple networks can, when co-designed with morphology, produce context-dependent behaviors such as running or climbing. This synergy between morphology and control underscores the potential of embodied computation to embed complex behaviors into relatively simple systems.

Altogether, this work advances the vision of soft robots with distributed intelligence. By leveraging rhythmic oscillations, minimal local computation, and morphology, we show how directed, adaptive behaviors can emerge without central coordination. These findings pave the way for autonomous soft robots that exploit their mechanical intelligence to operate effectively in real-world environments.

CONTENTS

1 INTRODUCTION 1

1.1 Soft robots: benefits and control challenges 2

1.2 Alternative approaches to autonomy for soft robots . . . 5

1.3 Example of self-sustained oscillations via the nonlinearity of the soft materials. 10

1.4 Objectives 19

1.5 Outline 21

2 ECHINODERM-INSPIRED AUTONOMY FOR SOFT-LEGGED ROBOTS 25

2.1 Introduction 27

2.2 A soft modular platform to study embodied computations 31

2.3 Analyzing the phototaxis behavior of a four-module configuration 37

2.4 Robustness through embodied computation 38

2.5 Adaptability to morphological changes 40

2.6 Increasing the mechanical intelligence potential 45

2.7 Conclusions 49

S2 SUPPLEMENTARY MATERIAL FOR CHAPTER 2. 51

S2.1 Supplementary Movies S2 52

S2.2 Fabrication of the modules 54

S2.3 Experimental setup 55

S2.4 Experiment protocol 55

S2.5 Details on the gravel surface 57

3 ROBUST PHOTOTAXIS BY HARNESSING IMPLICIT COMMUNICATION IN MODULAR SOFT ROBOTIC SYSTEMS 73

3.1 Introduction 75

3.2 Experimental setup 77

3.3 Phototaxis in experiments 81

3.4 Understanding the observed behavior 83

3.5 Effect of geometry on the long-term behavior of the system 90

3.6 Effect of learning parameters on the long-term dynamics of the system 92

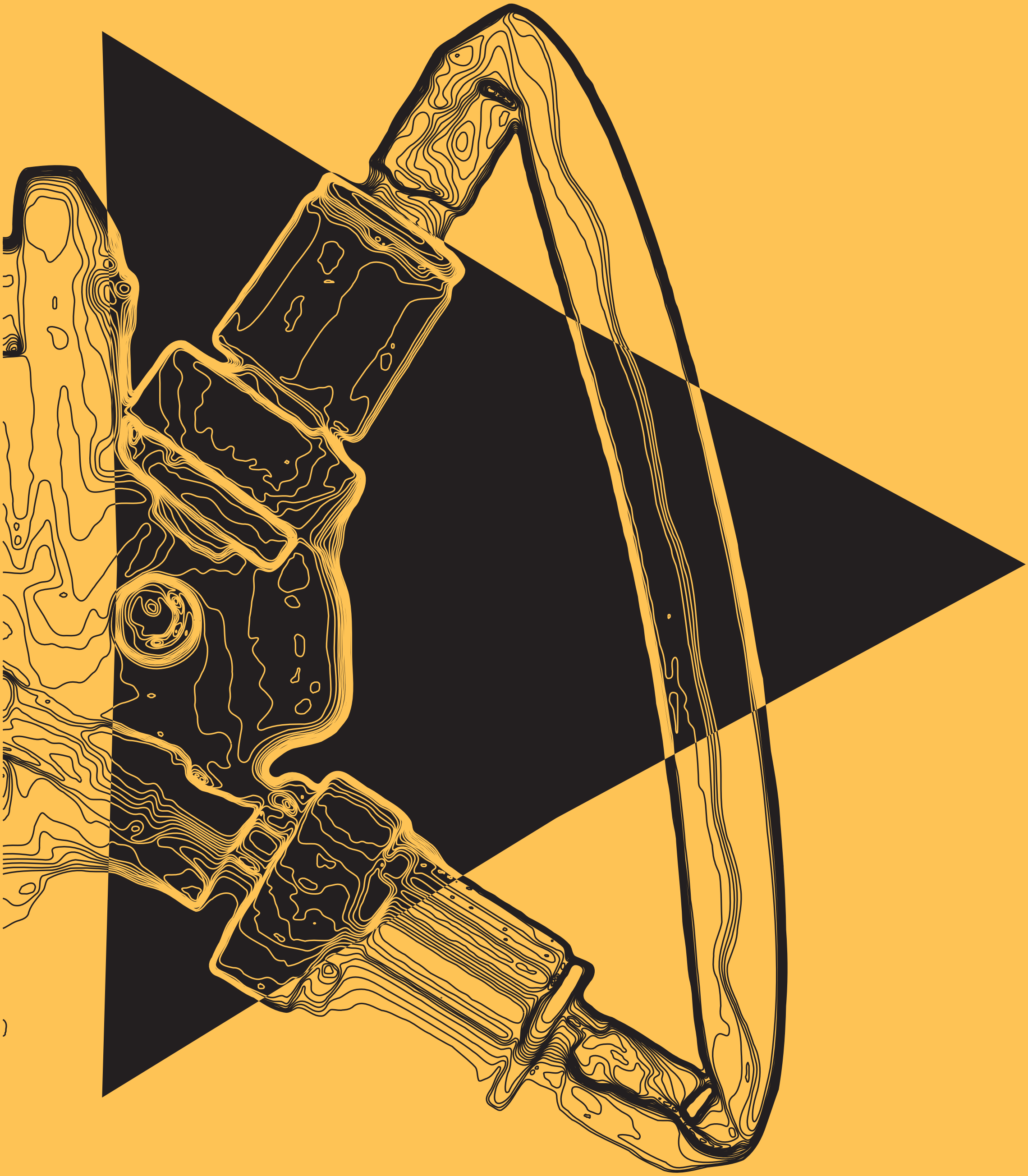
3.7 Experimental validation 97

3.8 Conclusion 107
S3 SUPPLEMENTARY MATERIAL FOR CHAPTER 3. 111
S3.1 Supplementary movies S3 112
S3.2 Methods 114
S3.3 Model 117
4 FLUIDIC OSCILLATORS AS NEURONS FOR SOFT ROBOTS 139
4.1 Introduction 141
4.2 Fluidic relaxation oscillator design 143
4.3 Modeling the fluidic oscillator as a neuron 147
4.4 Oscillator characterization 152
4.5 Oscillator variability 154
4.6 Importance of the recovery variable 157
4.7 Coupling the experimental fluidic oscillators 158
4.8 Phase-locking to a pressure control signal 161
4.9 Mutual coupling of two oscillators 167
4.10 Conclusion 172
4.11 Supplementary movies 4 175
5 ADAPTIVE BEHAVIOR IN MINIMAL OSCILLATOR NETWORKS THROUGH COEVOLUTION OF BODY AND BRAIN 177
5.1 Introduction 179
5.2 Methods 181
5.3 Evolving a body for locomotion 193
5.4 Multiple oscillators 195
5.5 Two different gaits appear 200
5.6 Climbing in a confined space 204
5.7 Environmental feedback 206
5.8 Adaptation to the task 207
5.9 Binary inputs for control 212
5.10 Conclusions 214
5.11 Supplementary movies 5 217
6 CONCLUSIONS AND OUTLOOK 219
6.1 Conclusions 220
6.2 Outlook 223

BIBLIOGRAPHY 227
LIST OF PUBLICATIONS 245
ABOUT THE AUTHOR 255

INTRODUCTION

1.

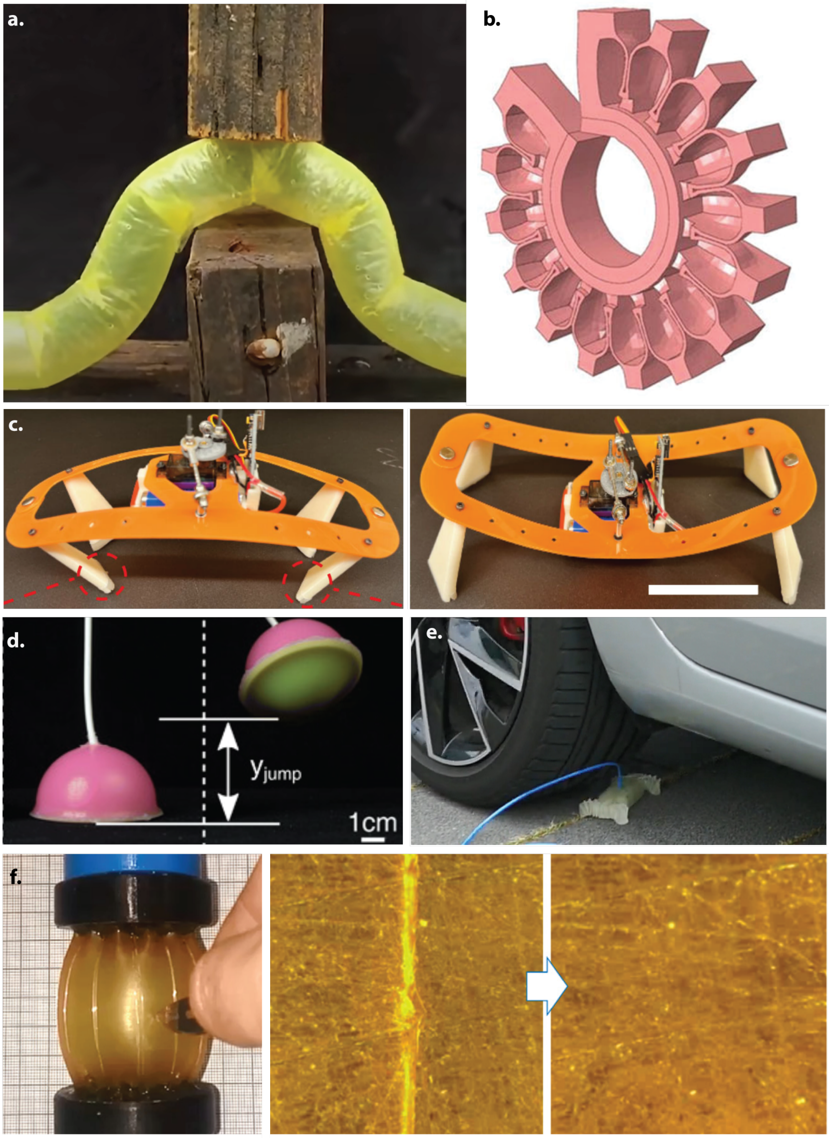


Robots made out of soft, deformable materials (e.g., rubbers or plastics) have started to become an alternative to traditional rigid machines made from metals. Soft robots offer several benefits due to their intrinsic compliance and flexibility. Unlike rigid robots, soft robots can safely interact with fragile objects and humans, as their soft bodies deform upon contact rather than generating large impulsive forces [59, 104]. This makes them ideal for tasks in close proximity to people (for instance, wearable or biomedical robot [70]) and for handling delicate items [30]. Soft robots are also highly adaptive: they can conform to irregular surfaces and squeeze through constrained spaces that would be inaccessible to rigid robots [105]. Their deformability allows locomotion over complex terrain by morphing their shape to grip or brace against the environment. For example, a vine-like soft robot (Fig. 1.1a) can grow and navigate through cluttered environments by extending its body and bending around obstacles [42]. This adaptability translates to versatility in function. Soft robots have been developed for applications ranging from pipeline inspection to search and rescue operations in rubble, where navigation through tight voids is necessary [66].

Soft robotic designs often draw inspiration from the natural world. Many living organisms (such as octopuses, worms, and plants) achieve remarkable movement and manipulation without rigid skeletons. Imitating these, engineers have created soft actuators and structures that mimic biological strategies [59]. Pneumatic network actuators (PneuNets) are a prime example. By embedding air chambers in an elastomer that inflate like muscle tissue, they produce bending and crawling motions reminiscent of octopus arms or inchworms (Fig. 1.1b) [81]. Likewise, the plant kingdom provides ideas for rapid movement using soft materials. The Venus flytrap, for instance, rapidly closes its leaves by storing elastic energy and releasing it using a snap-through instability [33]. This principle has inspired soft robotic mechanisms that achieve fast locomotion through the clever use of material elasticity and geometry (Fig. 1.1c) [137]. Owing to such bioinspired design, soft robots are achieving capabilities previously unheard of for soft materials, like jumping based on snap-through instabilities, as depicted in Fig. 1.1d (analogous to the flytrap) [39]. These results show that softness can be compatible with explosively rapid motions. Engineers have even demonstrated that they can take these compliant materials beyond the jumping capabilities of biology and approach the theoretical limit of what is possible [43], demonstrating how engineering can go beyond the examples provided by nature.

Soft robots also tend to be remarkably resilient. Because they deform under impact rather than shatter, they can survive collisions and other physical stresses that might disable a rigid robot. Researchers demonstrate this resilience by driving cars over small four-legged robots made from silicone or 3D-printed from plastic (TPU) in Fig. 1.1e, and afterward they continue to operate [20, 126]. Furthermore, both physical and chemical approaches have been used to construct self-healing polymers, which give the material the ability to recover from physical damage (Fig. 1.1f) [124]. Such robustness is inherent to soft construction materials, as energy from impacts is dissipated through deformation.

Despite their numerous advantages in interaction, adaptability, and resilience, controlling soft robots remains an immense challenge. Soft structures have virtually infinite degrees of freedom, since any point on a continuum body can, in principle, deform in complex ways. This high-dimensional, continuously deformable state makes it not straightforward to model and control soft robots with conventional techniques [23, 104]. The dynamics of soft materials are nonlinear and often involve coupling between solid deformation and fluidic actuation (in the case of pneumatic or hydraulic soft robots) [64]. As a result, creating accurate mathematical models for planning and feedback control is non-trivial; even simplified models lead to complex, high-order systems. Traditional rigid-body robots typically have a limited number of joints, allowing well-established control algorithms to manage their motion. In contrast, a soft robot's "joint" is essentially distributed along its body. Furthermore, soft actuators (such as air-filled elastomeric chambers) exhibit latency, hysteresis, environment-induced deformations, and other nonlinear behaviors. Sensors that work well on rigid robots (such as encoders) are more challenging to integrate on soft, stretchy substrates, resulting in incomplete state feedback. All these factors contribute to control strategies for soft robots, often relying on heuristics or open-loop control, where precise feedback regulation is challenging to achieve [23].



Caption on next page.

Fig. 1.1: **Soft robots coupled to the environment.** **a**, A soft robot that navigates its environment through growth from [42]. Reprinted with permission from AAAS. **b**, Pneumatic networks for soft robotic actuators, with the interconnected chambers displayed in their inflated state from [81]. Reproduced with permission from Wiley. **c**, Untethered soft robotic crawler that uses an elastic instability to generate a cheetah-like galloping gait for fast locomotion. From [137], Copyright © 2023 IEEE. Reprinted with permission. **d**, A soft jumper that harnesses snap-through instability (analogous to the flytrap). From [39] and reprinted with permission from AAAS. **e**, A fully 3D-printed soft walker with integrated fluidic circuits that still functions after being driven over by a car. From [20], reprinted with permission from AAAS. **f**, Actuator made out of a self-healing polymer that self-heals after being cut with a surgical knife. From [124], reprinted with permission from AAAS.

Researchers are actively exploring advanced techniques, such as machine learning and reduced-order modeling [24], to enhance soft robot control, but it remains a frontier of the field. The challenge is not merely to control a single bending actuator, but to coordinate many continuously deformable degrees of freedom in a way that yields a desired global behavior (like locomotion). In summary, soft robots promise unmatched interaction capabilities and robustness due to their compliance, but their very softness introduces a complexity in control that is currently a significant barrier [24]. Unlocking the full potential of soft robotics will require new paradigms of actuation and control that can handle high-dimensional, nonlinear dynamics.

1.2 ALTERNATIVE APPROACHES TO AUTONOMY FOR SOFT ROBOTS

One promising direction is to directly embed physical intelligence (decentralized intelligence) into the materials or body, thereby mimicking single-cell behavior in sensing and actuating without a central processor, a concept known as embodied intelligence [12]. By leveraging the materials and inherent dynamics of the robot itself to contribute to control, we may cope with complexity in a more distributed fashion.
In nature, living organisms respond, react, and adapt to external stimuli based on their biological intelligence, which enables them to exhibit

goal-directed behavior necessary for survival in the real world. Highly evolved organisms typically interact with their environment via complex neural networks, such as the central nervous system and the brain. Conversely, other life forms, such as fungi, microorganisms, and cephalopods, achieve goal-directed behaviors without relying on nervous systems or centralized commands. Their responses are directly encoded into their physical structure [12].
Empowering soft robots with similar autonomy, enabling them to achieve goal-directed behaviors without human intervention, is a current frontier in soft robotics research. Nature provides valuable insights into self-sensing, self-decision-making, and self-regulation across varying levels of complexity.

Out of equilibrium

A fundamental principle observed in natural systems is that most natural systems are out of equilibrium when subject to external influences [135]. These deviations from equilibrium cause the system to react with distinct responses based on external signals, enabling natural systems to adapt and respond to their environment. Soft robots are a prime candidate for embedding goal-directed behavior through this principle, as their soft bodies inherently deform in response to external inputs. A soft swimming robot constructed from photoresponsive hydrogel illustrates this principle. Exposure to light causes morphological changes that push the robot out of equilibrium, allowing it to steer toward the light stimulus [48]. Leveraging continuously changing or cyclic environmental conditions, these responsive deformations can also facilitate goal-directed actuation. An example is bio-inspired artificial seeds (Fig. 1.2a) capable of digging themselves into the soil by utilizing deformation driven by natural humidity cycles in the environment [69].

Oscillations

In certain special cases, deviations from equilibrium can persist under continuous stimuli, generating sustained oscillations. For example, shining light at the pupil's edge causes the pupil to contract, blocking the incoming light. Due to a 300 ms response delay, it continues to contract

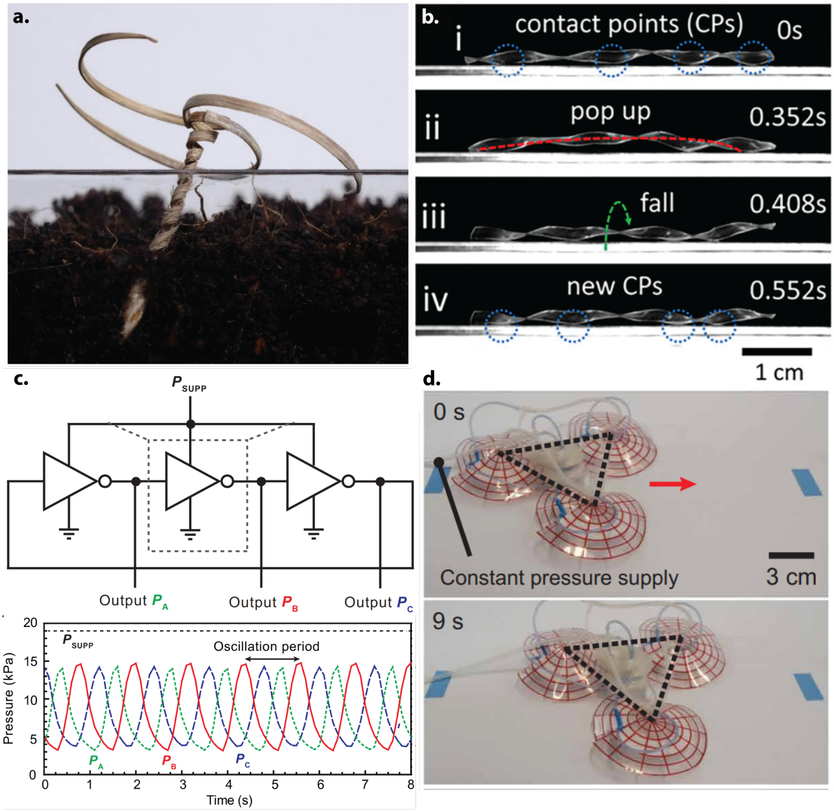


Fig. 1.2: **Oscillation in soft robotics.** **a**, An artificial bio-inspired seed that digs itself into the ground from [69]. Reproduced with permission from Springer Nature. **b**, A twisted soft ribbon that crawls and redirects its crawling when it hits an obstacle from [141]. Published by PNAS, distributed under the terms of the Creative Commons License (CC BY-NC-ND 4.0). **c**, Soft sheet ring oscillator. A schematic of the transistor configuration and the pressure outputs for a constant input pressure [67]. **d**, Snapshots of the walking soft sheet ring oscillator corresponding to **c** [67]. **c** and **c** are reproduced from [67] and reprinted with permission from AAAS.

even after the light is blocked, then expands again once darkness is sensed, perpetuating the cycle [139]. While typically undesirable in engineered systems due to instability, such delay-driven oscillations form the basis for rhythmic motion in many natural systems and soft robots alike [12]. Oscillatory deformations induced by continuous sunlight or heat inputs enable locomotion in small liquid crystal elastomer (LCE)-based soft robots, as depicted in Fig. 1.2b [141, 142]. These robots exploit stimulus-shadowing (light or heat) to maintain oscillations similar to the pupil—a strategy highly effective for individual actuators yet challenging to scale to more complex systems comprising multiple actuators.

Alternatively, self-sustaining oscillations in soft robotic systems can also arise from interactions between fluidic components and the nonlinear mechanical properties of their compliant materials. The hysteresis of a fluidic transistor can produce a fixed sequence of actuation when connected in a ring [67, 101]. In these cases, the transistors themselves cannot sustain the oscillation. Still, the connected network at a system level produces self-sustaining oscillations. Because these ring oscillators (Fig. 1.2c) produce multiple oscillating outputs in a fixed sequence, they can be used to produce locomotion gaits for more complex soft systems. A notable example is the soft sheet ring oscillator system depicted in Fig. 1.2d, which can walk and climb under various system configurations, all driven by a single constant input pressure and without the need for electronics.

Coupling

Lastly, natural systems exhibit high degrees of coupling. Effective coordination of rhythmic muscle activations necessary for adaptive locomotion arises from coupling not only among muscle groups but also between the organism and its environment [52]. Although the rhythmic sequence of soft ring oscillators is inherently fixed, these systems can still be coupled to the environment. In Fig. 1.3a, this is achieved by creating a sensor that switches which actuator is connected to which oscillator when touched [25]. Although the sequence of the oscillators remains unchanged, the resulting behavior of the system switches in response to an environmental input. The drawback of this approach

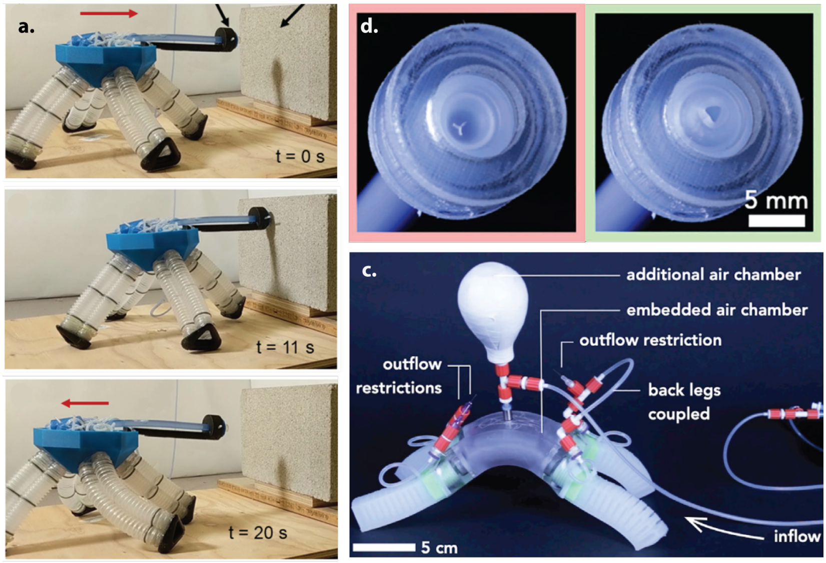


Fig. 1.3: **Soft robots coupled to the environment. a**, A soft robot with a gait that mimics a turtle, driven by a fluidic ring oscillator. A sensor rearranges the connections from the oscillators to the actuators through touch, switching the behavior to cause the robot to walk backwards. From [25] reprinted with permission from AAAS. **b**, a Self-oscillating valve that oscillates for a constant flow input [63]. **c**, Multiple valves depicted in **b** integrated into a walking soft robot [63]. **b** and **c** reprinted from [63], distributed under the terms of the Creative Commons License 4.0 (CC BY-NC-ND 4.0).

is that it requires every change to the system to be pre-programmed with additional fluidic circuitry, limiting its adaptability for greater autonomy in unstructured and unknown environments.

In nature, we find that the networks that drive the rhythmic patterns used for locomotion, swimming, and flying arise from interconnected neurons that can sustain oscillations by themselves when subjected to a stimulus. These self-sustained oscillations at a component level result in highly adaptive patterns when coupled in networks [52]. Fig. 1.3b demonstrates an example of such a self-oscillating component in fluidics. When subjected to a constant input flow, this valve, inspired by a ketchup container, produces an oscillating output caused by the hysteretic buckling of the soft dome with a slit. When multiple of these oscillators are coupled to the same input, they produce a sequence similar to that of the soft ring oscillator (one after the other), which can be used to generate a walking gait for a soft robot (Fig. 1.3c). However, this sequence is no longer fixed, as interactions with the environment can alter the order in which they are activated without the need for additional circuits [63].

1.3 EXAMPLE OF SELF-SUSTAINED OSCILLATIONS VIA THE NONLINEARITY OF THE SOFT MATERIALS.

In summary, the soft materials of soft robots provide both challenges and opportunities for creating systems that can execute goal-oriented behaviors without human intervention. While many traditional control approaches might fail, the complex mechanical responses of the materials can also be harnessed to generate complex but controlled output behaviors. In all of the examples above, we find that the behavior is not generated or controlled by a single element. Instead, it is a constant interplay between all components and the environment that makes a desired behavior emerge. Therefore, to achieve greater autonomy and adaptability in soft robots, we should consider the combination of the robot's sensors, body, environment, and brain as a single, coupled dynamical system, where one cannot be designed without considering the others.

To better illustrate how control can emerge for the nonlinear mechanics of everyday objects, we dive deeper into an example of a self-oscillating

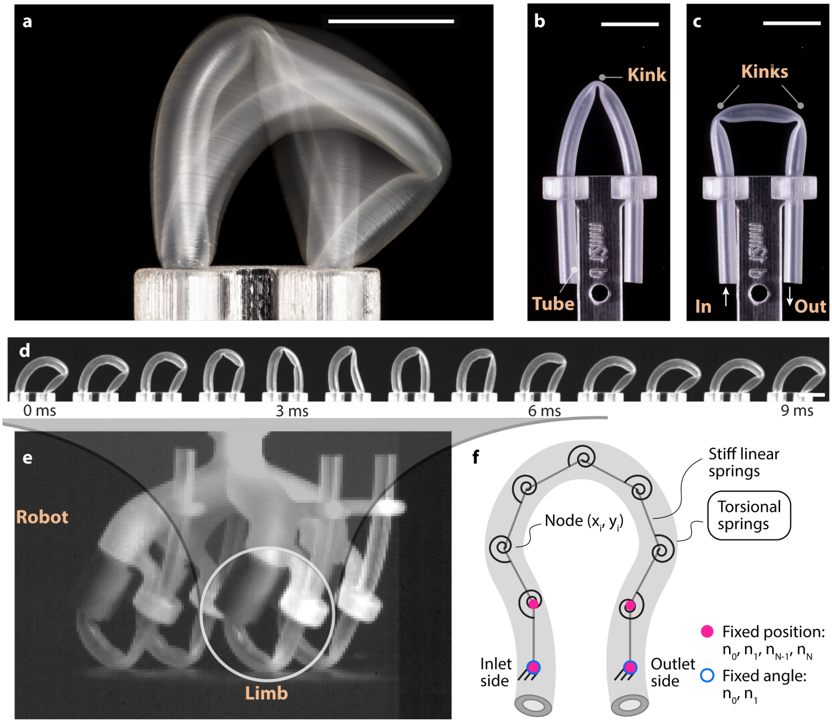


Fig. 1.4: **A soft robotic "limb" can oscillate on its own when airflow is applied through a bent soft tube. a**, With a constant air inflow, the tube spontaneously switches between configurations, as seen in a long-exposure image capturing its motion. Directed behavior arises in a range of morphologies from a similar decentralized nervous architecture, highlighted in orange. **b** and **c** show the two static stable states of the bent tube with one kink vs. two kinks (without airflow). These kinks are localized collapses in the tube's curvature. **d**, High-speed snapshots (times in milliseconds) from one oscillation cycle illustrate how the tube transitions from a one-kink state to a two-kink state and back again. The alternating kink formations cause the tube's free end (tip) to move in a periodic, asymmetric path. **e**, A soft robot configured with four self-oscillating limbs that runs by supplying constant airflow from a single source. **f**, Overview of a minimal mechanical model that explains the self-oscillation.

limb constructed from a single silicone tube[1]. When both ends of the tube are constrained and a constant airflow is applied to one side, the tube exhibits self-oscillating behavior that can be utilized as an actuator for locomotion due to its asymmetric deformations. Fig. 1.4a represents a long-exposure image of the self-oscillating tube. A kink, the local collapse of a tube's inner area, travels down the length of the tube and reinitializes itself, causing the oscillation.

These kinks result from the tube's nonlinear mechanics; once the tube's curvature and bending torque exceed a threshold, a section suddenly collapses inward, forming a kink. Importantly, the bent tube has multiple stable configurations (e.g., one kink vs. two kinks) due to this nonlinear torque–curvature response, as depicted in Fig. 1.4b–c. When a constant airflow is introduced into the tube, the system begins to self-oscillate, continuously and spontaneously alternating between the one-kink and two-kink states. In each cycle, a kink forms or vanishes in an alternating sequence, causing the tube to whip back and forth (Fig. 1.4d). This oscillation is self-sustained, as it requires only a constant flow of air and no external timing or control input. The mechanism can be understood as a limit-cycle oscillation arising from the interplay of flow-induced pressure and the tube's kinked deformations: as one kink collapses and opens, the other forms, producing a repeating pattern of motion. The tube's tip follows an asymmetric closed-loop trajectory (with distinct "forward" and "return" strokes akin to a stance and swing phase of locomoting legs in nature), highlighting that the motion is hysteretic and biased by the directed airflow (the air causes the limb to move faster in one direction). In essence, the airflow through the kinked soft tube exploits the tube's nonlinear buckling behavior to generate spontaneous periodic motion without any active control (Fig. 1.4d).

Assembling four of these limbs in Fig. 1.4e yields a lightweight, fast, and soft robot that requires only a single constant input to function, highlighting how nonlinearities of soft materials can be used to generate complex and adaptive behaviors in soft robots [18]. To explain the self-oscillation mechanism, we consider a minimal mass-spring model composed of a chain of point masses connected with tensile and torsional springs (Fig. 1.4f). The complete mathematical description can be

[1] Alberto Comoretto, Harmannus AH Schomaker, and Johannes TB Overvelde. "Physical synchronization of soft self-oscillating limbs for fast and autonomous locomotion." In: *Science* 388.6747 (2025), pp. 610–615.

found in [18]. We demonstrate that this minimal representation, with only three fundamental assumptions—each reflecting a physical aspect of the system—gives rise to the self-oscillating behavior.

Three assumptions that enable self-oscillation

These three assumptions are based on physical experiments of the silicone tube and can be found in [18]. Firstly, the importance of nonlinear behavior, a nonlinear kink-forming torque response of a tube, is presented in Fig. 1.5a. The torque required to bend the tube increases initially, as expected from a material under stress, until a critical curvature (β_1) is reached. As soon as one of the torsional springs surpasses this angle, the torque drops as a function of the angle (negative stiffness), causing it to collapse (much like a straw buckling). The curvature stabilizes around β_2, where the torque builds up again as the angle increases. As a result, when starting with angles below β_1, the stable state of the system at the start is a dome with a uniform curvature (Fig. 1.5b, left image). But when we drive the left end of the tube towards the right, one of the torsional springs suddenly buckles, and a kink is formed, as depicted in Fig. 1.5b (right image). This assumption captures the tube's tendency to form discrete kinks rather than bend smoothly, and it provides the starting point for the oscillation by allowing sudden state changes (from unkinked to kinked and vice versa).

Secondly, flow resistance is concentrated at kinks: A kink in the tube acts like a valve or constriction, significantly increasing fluid resistance at that specific spot (e.g., a kinked garden hose blocking the water flow). As a consequence, most of the pressure drop from the airflow occurs across the kinked segment of the silicone tube. Physically, when a kink forms, it pinches the tube's cross-section, impeding flow. In the model, we assume that the presence of a kink sharply raises flow resistance, causing the air pressure upstream of the kink to build up (Fig. 1.5c). Conversely, if a kink disappears (or if the tube straightens at that point), the flow experiences significantly less resistance, and the upstream pressure drops. This means that the internal pressure dynamically depends on the kink state and position. It is higher when a kink is present (due to blockage) and lower when the tube is open.

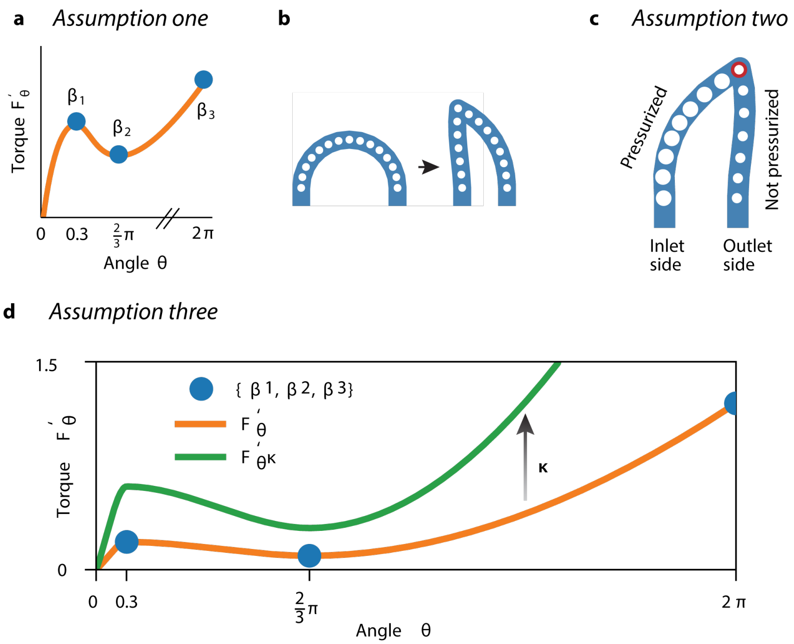


Fig. 1.5: **Minimal Nonlinear Tube Model: Three Key Assumptions. a**, The nonlinear bending stiffness of the torsional springs. **b**, The resulting kink behavior when the two ends of the tube are driven together. **c**, The airflow comes in from one side and is blocked by the kink, producing pressure on the inlet side of the kink. **d**, The pressure inside the tube stiffens the torsional springs by a factor κ.

The kink effectively toggles the pressure in the segments of the tube, linking the fluid dynamics to the tube's deformation state.
Lastly, pressure-dependent stiffening: The tube's internal pressure feeds back into its mechanical response. When the tube is pressurized by airflow, it becomes more resistant to bending. In other words, it requires a higher torque to produce the same curvature. This is caused by the internal pressure that balloons the tube outward, opposing collapse. In the model, this effect is captured by scaling the curvature–torque response of the torsional springs by a factor κ. The orange vs. green curves in Fig. 1.5d illustrate this stiffening of the torsional springs. The green curve (pressurized) lies above the orange curve (unpressurized), indicating that under higher internal pressure, the tube needs more bending force to kink. This assumption reflects the experimental observation that pressurization "inhibits" kinking and increases the torque. It introduces a crucial feedback loop: as pressures of the segments change during the oscillation cycle, it alters the tube segments' preference to kink or unkink in real time. These three ingredients work together to produce self-oscillation.

Self-oscillation in the minimal model

This minimal model reproduces the self-oscillating behavior observed in the real soft limb. The simulation starts with an unkinked tube, as depicted in Fig. 1.5a (top left), with equal curvature throughout the nodes of the chain. In the first phase of the simulation, we drive the first two nodes on the left side of the tube (inlet) toward the right. As soon as the threshold curvature β_1 is reached, one of the torsional springs collapses, reducing the required torque to further bend that torsional spring. As the distance between the two ends of the chain decreases, up to $t' = \text{start}$, the kink is further increased, showing a clear distinction between the angle of the kinked node and the rest of the tube (Fig. 1.5a heatmap and the image of the chain at the top center). At $t' = \text{start}$, the distance between the two ends is fixed, and the simulated air inflow starts, stiffening the upstream (left) section of the chain. In the heatmap, we see that the kink then starts to travel and reinitialize. These global oscillations emerge solely from the three assumptions implemented at a local level.

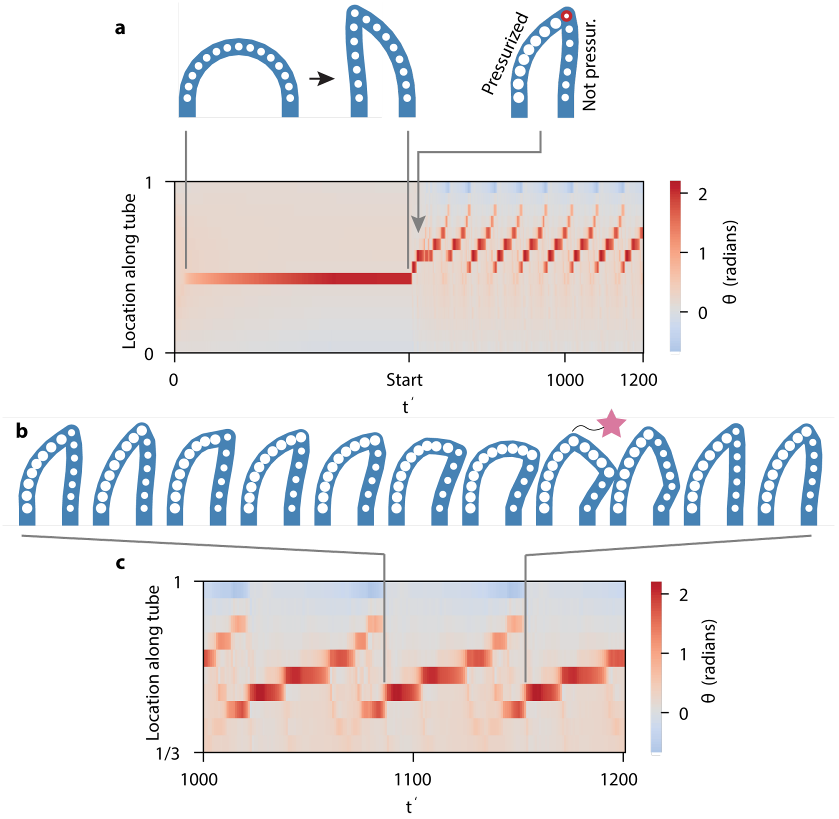


Fig. 1.6: **Self-oscillation in the minimal model.** **a**, For a single oscillating tube, the heatmap shows the spatiotemporal pattern of the deviation of the rest angle of the torsional springs. The heatmap and corresponding images above describe the two stages of the simulation, as the left inlet side of the tube is driven to the right up to $t' =$ start. The heatmap depicts the tube's state over time (red indicating a kink at a given location and time, gray indicating no kink). At $t' =$ start, the start of the air inflow, the pressure starts to build. The tube switches between one-kink and two-kink states in each cycle, and the kink position moves (travels) along the tube before a new kink forms upstream (as indicated by the diagonal streaks in the heatmap). **b** and **c** show a zoomed-in view of the oscillation. Snapshots of a single oscillation in the model are depicted in **b**, with a pink star indicating the formation of the second kink. **c** represents a zoomed view of the same heatmap depicted in **a**.

To interpret the oscillatory behavior, Fig. 1.5b–c shows a zoomed-in view of the oscillations with simulation snapshots. When airflow is applied, the system evolves as follows: a kink forms (Assumption 1), thereby raising the upstream pressure (Assumption 2), By assumption 3, this stiffens the tube upstream, The elevated pressure also pushes the kink open, driving it downstream (Fig. 1.5b, first half of snapshots). As the kink moves closer to the inlet, a new kink eventually forms upstream (assumption 1), marked with a pink star in Fig. 1.5b. The upstream pressure then decreases, lowering stiffness (assumption 3 in reverse), which destabilizes the initial kink, causing it to disappear. The new kink restricts flow again (assumption 2), raising pressure, and the cycle repeats. Thus, kink formation, flow resistance, and pressure feedback create a self-sustained limit-cycle oscillation, a repeating pattern of kink appearance, motion, and disappearance. This collective behavior follows directly from the three assumptions governing local tube dynamics.

Coupling with the environment

Building on the single-limb oscillation mechanism, we now examine how synchronization and adaptation emerge when multiple limbs are integrated into a robot. Figure 1.7a shows that the updated pouch limb undergoes cyclic full-step oscillations, similar to the single-tube but with higher resistance upon kinking, enabling operation at much lower input flows, e.g., a small onboard pump. Mounting two such limbs on a lightweight untethered platform powered by individual pumps and a LiPo battery (Fig. 1.7b) results in a robot that can autonomously locomote without external circuitry or centralized control.

When placed on solid ground, implicit mechanical interactions with the ground surface couple the two independently powered limbs, leading to in-phase synchronization and robust cyclic hopping (Fig. 1.7c, 0–7 s). However, when the robot dives into water, the environmental coupling fundamentally changes: buoyancy forces and hydrodynamic resistance promote anti-phase oscillations between the two limbs. Within seconds, the gait transitions from synchronized in-phase hopping to stable anti-phase swimming (Fig. 1.7c–d).

This behavior illustrates how the identical physical oscillators can produce distinct locomotion modes depending solely on their environment.

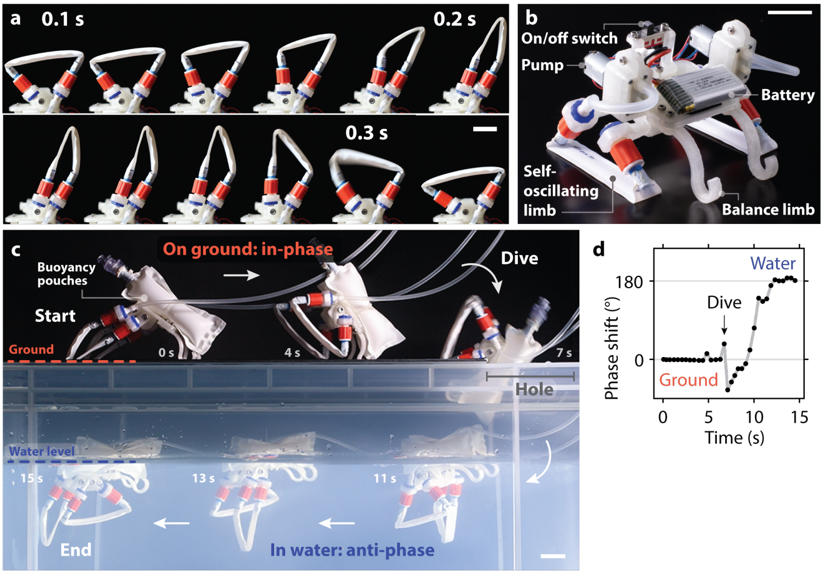


Fig. 1.7: **Coupling-induced synchronization and environmental gait transition.** updated pouch limb **a**, cyclically performs full-step motions **b**, Two self-oscillating pouch limbs mounted on an untethered robot that carries a LiPo battery and two pumps. Overview of the robot's components, including pumps, battery, on/off switch, balance limb, and self-oscillating limbs. **c**, Transition sequence from in-phase hopping on ground (0–7 s) to anti-phase swimming after diving into water (11–15 s), enabled by buoyancy pouches and implicit coupling with the environment. **d**, Phase shift between the two limbs during the dive, switching from ~ 0° (in-phase) on ground to ~ 180° (anti-phase) in water. Scale bars: 1 cm **a**,**b**, 2 cm **c**.

The ground stabilizes an in-phase gait, whereas immersion in water stabilizes an anti-phase gait, highlighting the role of body–environment feedback in determining locomotion strategies. In essence, coupling through the environment allows the robot to autonomously adapt its gait to new conditions, without sensing, computation, or explicit control.

In summary, the nonlinear tube model not only mimics the behavior of a single limb's self-oscillation but also provides insight into the interactions that enable the self-oscillation to persist. The mechanism of self-sustained oscillation, in which alternating kinks are governed by internal pressure and tube elasticity, underpins both the emergence of complex behaviors from everyday soft objects and the utilization of these dynamics in robots that do not require electrical circuits or controllers to operate. Extending beyond the single-limb model, coupling and environmental interactions play a crucial role in shaping collective dynamics. As shown in Fig. 1.7, multiple self-oscillating limbs synchronize through environmental interactions, and the environment itself can induce transitions between distinct gaits, such as in-phase hopping on land and anti-phase swimming in water. These findings highlight how robust and adaptive locomotion can arise naturally from the same minimal physical ingredients, suggesting pathways toward autonomous soft robots where locomotion emerges from local interactions rather than centralized control.

1.4 OBJECTIVES

How can soft robots coordinate complex, goal-directed behaviors without relying on centralized electronics and rigid computation systems? This overarching question guides this thesis, resulting in the demonstration of robust, decentralized control paradigms that leverage the embodied mechanical intelligence of soft robotic systems.

A foundational principle in soft robotics is embedding computational functionality into physical structures themselves, which naturally leads to the resilience and adaptability observed in biological organisms. Yet, a significant gap remains between artificial soft robots and their natural counterparts, particularly in the autonomous coordination of rhythmic behaviors.

> *What constitutes decentralized autonomy in soft robotics, and how does it relate to biological systems?*

Soft robotics inherently offers compliance and adaptability due to its elasticity. Still, achieving truly decentralized, goal-directed behavior requires embedding sensing, computation, and actuation locally, thereby minimizing the computational overhead of central controllers. Inspired by distributed natural systems such as echinoderms, plants, and fungi, this research investigates how minimal localized sensing, memory, and actuation can yield globally coordinated behaviors.

> *How can we design soft robots that achieve complex goal-directed behaviors through localized sensing and minimal processing?*

A core challenge is understanding and characterizing the dynamic behaviors of soft robotic systems that emerge from decentralized interactions and decision-making. Unlike centralized rigid robots with predictable behaviors, decentralized soft robots exhibit multiple coexisting modes as they are affected by their environment. Therefore, understanding the interplay between physical structure, environmental interactions, and resulting emergent behaviors is essential for reliable goal-directed behavior.

> *Can we characterize and understand the emergence of phototaxis in decentralized soft robots, with minimal and local sensing, memory, and actuation?*

Many biologically inspired behaviors, such as animal locomotion, depend on rhythmic oscillatory motions. Achieving autonomous, rhythmic, and repeatable motion without the use of electronics is a critical milestone for soft robotics. We propose embedding rhythmic patterns directly within the robot's physical and fluidic structure, exploring how local couplings between soft oscillators can lead to coordinated rhythmic patterns.

> *Can we create controllable self-sustaining rhythmic patterns and coordination in soft oscillators, through local fluidic coupling and without electronic control systems?*

Finally, inspired by biological evolution, we explore the co-evolution of robot morphology and embedded control strategies for given environments and tasks. Using coupled oscillator networks embedded within evolving soft robot morphologies, we demonstrate how minimal structural complexity can spontaneously yield distinct adaptive behaviors responding dynamically to environmental feedback.

> *How does co-evolutionary design of morphology and embedded computation enhance autonomous adaptability in soft robotics?*

Collectively, this research contributes to the realization of autonomous soft robots capable of exhibiting goal-oriented behaviors that emerge from embodied intelligence, thereby paving the way for soft robotics in complex, real-world applications that do not rely on or offload computation from centralized electronics.

1.5 OUTLINE

In living organisms, directed behavior arises from repeated rhythmic (oscillatory) motions whose sequence and timing are robustly coordinated. This coordination is typically in part or even fully distributed throughout the organism: animals employ central pattern generators within their nervous systems, plants utilize distributed mechanoreceptors, and fungi leverage expansive mycelial networks. Such decentralized orchestration offloads computation from a central brain to the body, allowing behaviors to emerge naturally through interactions between the body and its environment.

This thesis explores alternatives to centralized control inspired by decentralized systems in nature. It identifies sequences and timing of oscillations leading to directed locomotion in soft robots. We aim to embody directed behavior in the physical system so that purposeful actions emerge from local body-environment interactions and feedback. Through an exploratory study spanning design, simulation, and hardware, we demonstrate how soft robotic systems can leverage their embodied mechanical intelligence using embodied computation to achieve complex autonomous behaviors without a centralized processor.

In **Chapter 2**, we draw inspiration from the physiology and decentralized nervous system of echinoderms (e.g., sea urchins, brittle stars, feather stars, and sea cucumbers) to examine how decentralized feedback can facilitate directed locomotion towards a light source (phototaxis) in limbed soft robots. We build a modular system where each limb is a self-contained module that stochastically optimizes its behavior with a feedback loop based on limited sensing, short-term memory, and computation. By harnessing the inherent mechanical intelligence of soft pneumatic actuators, cyclic on-off inputs to a pump at a fixed frequency are converted into complex bending and stepping motions. By physically connecting multiple limbs and letting each limb independently learn the phase of its oscillating motion, coordination between the limbs emerges. We show that, similar to echinoderms, such as sea stars, interactions of the individual limbs with the environment guide the robot toward coordinated movement patterns without relying on comprehensive full-body representations or complex algorithms. The soft robot dynamically re-coordinates its movement in response to changing conditions (changes in actuators and damage) without any central controller. Resilient, whole-body locomotion thus emerges from the interplay of many basic units, each with limited memory and no body awareness, demonstrating a route of adaptable goal-directed movement sequencing in soft robotics through embodied computation. In **Chapter 3**, to gain a better understanding of how this coordination emerges, we build a second modular system of self-contained units. In this system, the modules use the same strategy for sensing and processing, but we limit the actuation, making them immobile on their own. Instead, they expand and contract their connections to the other physically connected modules on a two-dimensional plane. When interconnected in two-dimensional grid configurations, the system as a whole can break the symmetry of the friction to achieve locomotion, similar to earthworms that expand and contract segments. By combining simulations and experiments, we gain an understanding of how this decentralized strategy can follow locally optimal sequences solely from the implicit communication facilitated through their physical connection (as the system moved toward the light, the connected units all increased their light intensity). The simulations also provide insight into how the sequences that the system produces are linked to the potential behaviors of the system and how these change with different configurations and in dynamic, unstructured environments. These results not only demonstrate that robust, directed locomotion in soft robots can emerge entirely from local environmental interactions but also show the profound link between the coordination strategy and the body morphology. They also illustrate the dynamic nature of the learning process as it adapts to changing, partially observable environments.

In **Chapter 4**, we explore control without electronics. While the work mentioned above focuses on reducing the hardware and complexity of algorithms needed to coordinate the sequences of movements starting from random behaviors, it still requires many electronic components to make the individual modules. Therefore, we next aim to embody sequences without relying on electronics by harnessing soft fluidic circuits with integrated magnetic components. By designing a fluidic relaxation oscillator that produces an oscillating output for a fixed input flow, we can encode the rhythmic inflation-deflation cycles into a single component. We implement directional air-driven coupling between the relaxation oscillators to emulate biological central pattern generators, orchestrating the rhythmic motions without electronics. By altering the fluidic coupling between them, we demonstrate rapid and reversible reprogramming of the oscillation sequences and timings. Such physically embodied control paves the way for soft robotic systems equipped with decentralized locomotion primitives, eliminating dependence on complex electronics and centralized controllers.

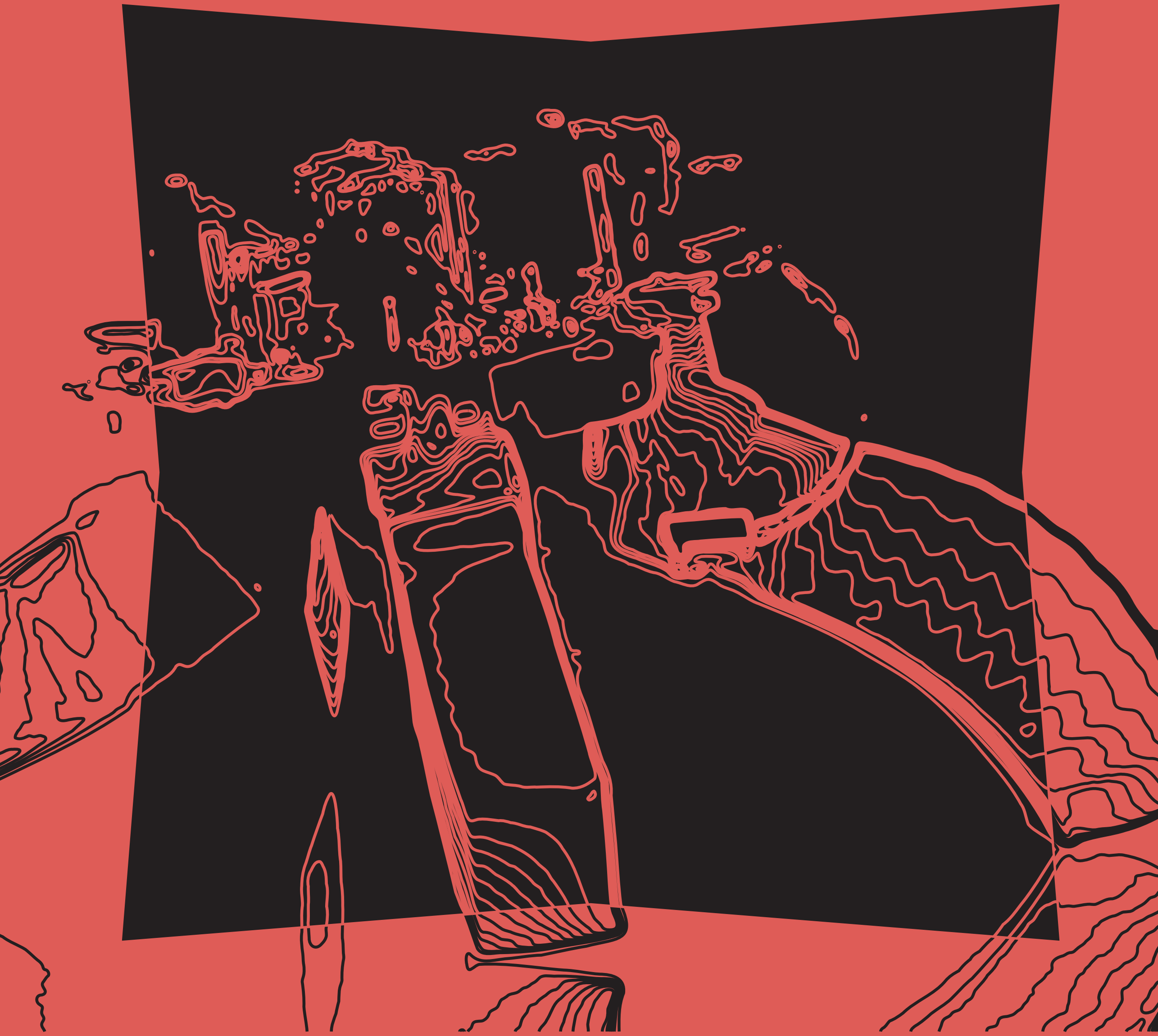
In **Chapter 5**, we explore the synergy between soft minimal controllers and soft bodies. The approaches above start with predetermined morphologies, whereas natural organisms demonstrate how body morphology and embodied computation evolve synergistically over longer time scales. Inspired by this co-evolutionary principle, we simulate coupled oscillator networks as mentioned above, integrated within evolving soft robotic morphologies. We show that oscillator networks with minimal complexity (number of oscillators and number of connections), when co-designed with the body morphology, can enable robots to transition spontaneously between distinct behaviors, such as climbing or running, in response to environmental feedback. These findings emphasize how thoughtful morphological and feedback co-design can embed rich, context-specific behaviors into relatively simple physical structures.

In **Chapter 6**, we conclude on the research objectives, discuss the limitations and achievements, and provide an outlook on possible directions for future work.

Collectively, this work contributes to the broader vision of autonomous soft robots with distributed intelligence, where sophisticated, goal-directed behaviors emerge from the continuous interplay of body and environment rather than explicit centralized command. These results, using embodied computation, pave the way towards soft robots that harness their mechanical intelligence to complete tasks autonomously in real-world scenarios.

ECHINODERM-INSPIRED AUTONOMY FOR SOFT-LEGGED ROBOTS

2.



Abstract. Soft robots harness their built-in mechanical intelligence to respond directly to their environment. However, they typically still depend on predefined sequences to coordinate their limbs, and external centralized hardware is often used for coordination in changing circumstances. In contrast, in nature, invertebrates like echinoderms distribute behavioral control throughout their body. Inspired by this decentralized computation strategy, we present a modular soft robotic system in which each limb independently adjusts the timing of its actuation to achieve phototaxis via entirely local, stochastic feedback with limited memory. Through this embodied computation approach, coordination emerges from the interaction of the body and the environment. We show robust phototaxis for soft robots that have different morphologies, that undergo damage, and that exhibit highly nonlinear leg behavior, all without an internal body representation. These results, therefore, offer a blueprint for designing resilient, autonomous soft robots that exploit the potential intelligence of their soft adaptive bodies.

H.A.H. Schomaker, J. de Vries, J.T.B. Overvelde. "Echinoderm-inspired autonomy for soft-legged robots." *submitted*.

2.1 INTRODUCTION

Living organisms have evolved to exploit mechanical interactions with their environment to achieve function and to delegate computational tasks to their body. Researchers have started to take inspiration from nature to harness such embodied mechanical intelligence (hereafter mechanical intelligence) in the field of soft robotics [51], for example, by developing soft actuators that passively adapt their shape when interacting with their environment. This makes soft devices currently especially suitable for applications that require adaptability, safety, or delicacy [56, 77], such as soft end-effectors for medical and agri-food applications. However, most soft robotic devices to date require external control signals, which limits their autonomy required in applications where the soft robots need to operate independently. As such, we have not seen a soft robot perform and complete an active search task autonomously with all its hardware onboard, which is essential to harness the adaptability of soft robots in the real world. Hence, there is a need for an alternative approach to enable autonomous behavior in soft robots.

The philosophy of soft robotics and its embodied mentality does not end here [51, 104], as nature provides countless examples of achieving autonomous behavior without needing a central brain and using only limited computational power. For example, invertebrates exhibit astounding coordinated behaviors despite missing a skeleton that limits the freedom of motion of the limbs. These creatures utilize what biologists refer to as embodied cognition, which distributes computation from the central brain to other body parts. Besides harnessing their softness, invertebrates employ a localized sensory-motor system to offload computation to their body. For example, an octopus uses its peripheral nervous system to articulate an elbow joint to bring food to its mouth [31, 47], or the camouflage of a catfish, which is largely controlled by organs near the skin [16].

Even though these examples demonstrate the potential of autonomy emerging from embodied cognition, such advanced functionality still seems out of reach. Yet, it does demonstrate key insights that we want to explore further in this work to enable some level of autonomy in soft robotic systems: autonomy does not require full awareness of the body or the environment. This is particularly clear for lower invertebrates (echinoderms like sea urchins, sea cucumbers, starfish, brittle stars, and feather stars) as they do not have a central brain. Yet, they can navigate a multitude of degrees of freedom and environments to exhibit active foraging behaviors [34]. Echinoderms achieve such behavior by primarily performing computations through a minimal radially symmetric nervous system, with a neural ring often centered around the mouth and radial nerves extending to the limbs (depicted in orange in Fig. 2.1a) [15, 144, 145].

By using such embodied computations that trigger muscle behavior, echinoderms show remarkable adaptation and robustness, both to variations in their environment and to (permanent) changes in their body shape. Even with their limited and distributed nervous systems, echinoderms can perform short-term memory learning [34] and display a wide range of locomotion modalities and behaviors through local action planning and information processing [34]. For example, sea stars, sea urchins, and sea cucumbers utilize similar circular nerve net architectures to achieve different gaits such as peristaltic motion, swimming, and limbed locomotion in various species (Fig. 2.1a) [3, 34, 38, 83]. Moreover, similar neural architectures are capable of controlling a diverse range of body plans [32], from worm-like sea cucumbers to limbed brittle stars and spherical sea urchins with hundreds of feet, and even when undergoing damage (Fig. 2.1a) [11, 133]. They also show diversity within species through phenotypic plasticity, where their morphology changes in response to environmental factors and predators [6, 28, 49].

To move towards harnessing such embodied computation to enable echinoderm-like autonomy in soft robots, we implement a distributed short-term memory learning approach in soft-legged robots in the current work. Through a modular platform, we aim to explore how a specific and computationally limited implementation of embodied computation can leverage the existing nonlinear mechanical behavior of various soft actuators (i.e., leverage the mechanical intelligence of the soft actuators) to achieve phototaxis without a central controller. Similar to echinoderms' abilities, our implementation of embodied computation can result in robust behavior for a range of differently assembled body plans, environments, and damage, which we will demonstrate by performing a range of experiments.

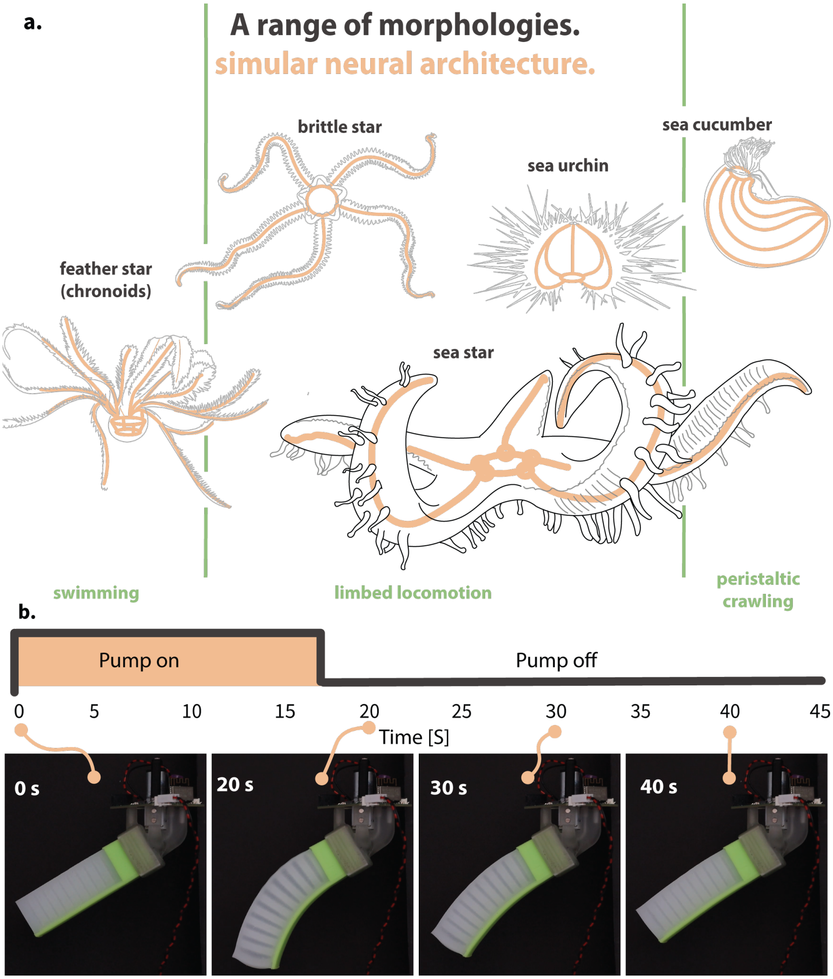


Fig. 2.1: **Inspiration from Echinoderms a**, Echinoderms are classified into five main classes. Directed behavior arises in a range of morphologies from a similar decentralized nervous architecture, highlighted in orange. **b**, Cyclic actuation of soft limbs (PneuNet actuators) for the modular platform.

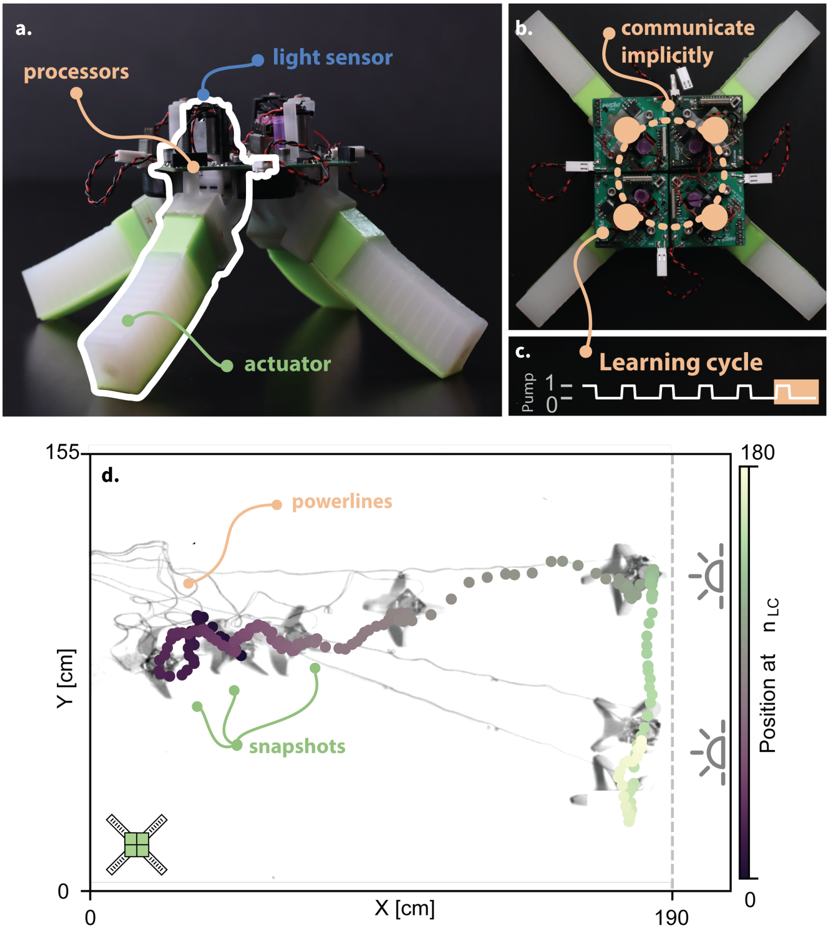


Fig. 2.2: **Modular platform inspired by Echinoderms to study directed locomotion without a central brain. a**. **b**, Each limb of the assembled robot is a self-contained entity capable of actuation, sensing, and computation. **c**, Each limb independently undergoes N_{AC} actuations before adapting its behavior. **d**, Phototaxis observed for a four-module configuration during 180 learning cycles. The modules are initially positioned on the left with a planar light source on the right side of the image. Black and white background images represent snapshots of the experiment taken every 30 learning cycles.

2.2 A SOFT MODULAR PLATFORM TO STUDY EMBODIED COMPUTATIONS

To study how autonomy can emerge from the interaction between embodied computations and mechanical intelligence, in this work, we focus on an active search task with the aim of having soft robots move toward the area of the highest light intensity (phototaxis). To that end, we introduce modules that each consists of a soft actuator, pneumatic pump, light sensor, and processor (Fig. 2.2a). These modules are assembled in a radially symmetric body (Fig. 2.2b), mimicking the morphology of the echinoderms (Fig. 2.1a). In our implementation, all modules are only mechanically connected, so that there is no explicit communication between them, and computation is fully distributed and embodied. Even though no explicit information is exchanged between the modules, coordination could emerge from the implicitly shared information constituted by their physical connection (Fig. 2.2b). As an example, if the assembled system moves in the direction of the light, all modules will sense an increase in the light intensity.

Each soft actuator acts as a limb that transforms a cyclic on-off input signal from the pump into a bending motion (Fig. 2.1b). The deformation of the actuator depends both on its design and its interaction with the environment [143], which together define its mechanical intelligence. Additionally, we use the processor to embody computation in each module. We implement an identical algorithm in each module that aims to increase the light intensity measured by the sensors, where we use a basic stochastic updating rule previously studied in a one-dimensional and two-dimensional framework [87, 108]. In this computationally limited algorithm, the pump oscillates between on and off at a fixed frequency at a specific phase that is kept in the module's short-term memory for four actuation cycles. After these actuation cycles, the module evaluates the difference in the change of the measured light intensity and adapts its phase that is kept in memory accordingly by accepting it or returning to its previous phase. It then randomly perturbs its phase for the next set of actuation cycles. This process is repeated separately in each module, where each complete cycle of evaluating a specific phase is regarded as a learning cycle. This process is repeated then for N_{LC} learning cycles. The full algorithm can be found in the supplementary information.

Algorithm

Each module contains a soft actuator, a microprocessor, an air pump, and a single light sensor. Each module goes through actuation cycles of $t_{AC} \approx 45\,\mathrm{s}$ during which it activates and deactivates its pump to inflate and deflate its soft actuator. The behavior of a module is defined by its phase ϕ_i, which determines the timing of periodic motion at a fixed frequency.

Each module's phase ϕ_i is changed over time depending on a stochastic learning algorithm. All modules independently run a stochastic updating rule to try to achieve phototaxis. Following a module's own clock, each module goes through a number of learning cycles n_{LC} to evaluate and consecutively alter its phase. A learning cycle spans six actuation cycles N_{AC}, during which a module maintains and evaluates a fixed phase ϕ_i' that has at the start of the learning cycle been perturbed by a random amount $\epsilon\,\Delta S$ with respect to the phase ϕ_i that is stored in memory, i.e. $\phi_i' = \phi_i + \epsilon\,\Delta S$. Here, ϵ is sampled uniformly from $[-1, 1]$ and $\Delta S = 0.1$. Each learning cycle ends with a phase-adjustment actuation cycle that implements the "decision" of the unit to accept the new phase ϕ_i', or reject the new phase by restoring the system to the phase ϕ_i kept in memory.

To determine whether the temporary phase ϕ_i' should be accepted, each module compares its performance during the current learning cycle with the previous learning cycle. It does so by comparing two independent light measurements (based on its own light sensor) from a specific actuation cycles (I_1 and I_2) to compute a difference in light intensity $\Delta I_i'$. This value is then compared with the previously stored difference ΔI_i. If $\Delta I_i' \geqslant \Delta I_i$, the new phase is accepted; otherwise it is rejected. As such, at the end of the learning cycle and to prepare for the next learning cycle each module implements a phase shift $\Delta\phi_i$ according to:

$$\Delta\phi_i = \begin{cases} \phi_i - \phi_i' + \epsilon\,\Delta S, & \text{if } \Delta I_i' < \Delta I_i, \\ \epsilon\,\Delta S, \quad \phi_i = \phi_i', & \text{if } \Delta I_i' \geqslant \Delta I_i. \end{cases} \tag{2.1}$$

Note that, according to previous results, we implement the "flaky" algorithm by updating ΔI_i independent of acceptance or rejection [108].

Because each module adjusts its phase independently, relying on local hardware and measurements, there is no explicit communication between the units to synchronize their behavior. Every module processes its own light-intensity feedback and stochastically updates its phase without requiring any simultaneous or centralized coordination. Over time, these local adjustments collectively give rise to emergent phototactic behavior. The pseudo-code for the algorithm can be found below.

Algorithm 1 Stochastic Phase Update for Module i

```
for $\ell = 1$ to $n_{LC}$ do
    Run $N_{AC}$ actuation cycles using fixed phase $\phi_i'$ and
    measure light intensities $I_1, I_2$ to compute $\Delta I_i' = I_2 - I_1$
    if $\Delta I_i' < \Delta I_i$ then
        $\Delta\phi_i \leftarrow \phi_i - \phi_i' + \epsilon\,\Delta S$
    else
        $\phi_i \leftarrow \phi_i'$
        $\Delta\phi_i \leftarrow \epsilon\,\Delta S$
    end if
    where $\epsilon \sim \text{Uniform}(-1, 1)$
    $\Delta I_i \leftarrow \Delta I_i'$
    adjust the last AC by $T_{AC}$
end for
```

Module actuation protocol

Fig. 2.1b and Fig. 2.9a schematically show how each module inflates and deflates its silicone actuators. The actuation cycle (AC) is defined by $T_{AC} = 45\,\text{s}$, during which the air pump remains activated for an α fraction of the cycle (here, $\alpha = 0.25$) and then is switched off for the remainder. Air continuously leaks out through a resistive needle, causing the actuator to deflate when the pump is off.

Every learning cycle (n_{LC}) spans six identical actuation cycles ($N_{AC} = 6$), plus one additional "adjustment" cycle (AC_a). During this adjustment cycle, we alter the time the actuator stays in the deflated state by an amount $\Delta\phi_i$, effectively changing the time that the system remains in its deflated state. This modification shifts the total duration of AC_a and enables the module to begin the next (n_{LC}) with an updated phase ϕ_i. For the experiments with the full-step actuator, we reduced the number of actuation cycles to ($N_{AC} = 4$), as initial tests with the system indicated that the displacement was too large to perform experiments on the experimental setup.

Light intensity measurements, I_1 and I_2, are taken at designated points within the (n_{LC}) to compute ΔI. Spacing the measurements by multiple actuation cycles smooths out noise and simplifies camera tracking. At the end of each learning cycle, every module sends its recorded data to the central computer. Note that there is no direct communication about phase changes among the modules; each module executes its own learning process asynchronously based on local measurements.

Phototaxis experiment

To illustrate the platform, we perform a phototaxis experiment on a flat planar surface where we mount two LED panels on the right side of the rectangular surface (Fig. 2.2d, Fig. S2.1). An assembled robot consisting of four modules is placed on the surface. Note that the assembled system does not have any prior knowledge of its actuator behavior, morphology, and orientation, and starts with random phases. If we then run the experiments for 180 learning cycles, we can observe from Fig. 2.2d that the assembled robot first moves randomly and starts to move away from the light. Yet, at $n_{LC} \approx 20$, the system reverses its direction to move toward the light. At $n_{LC} \approx 110$ we see a steep increase in the displacement towards the light, until the system reaches the end of the canvas at $n_{LC} \approx 135$. Therefore, this initial experiment shows that, in this case, our system learns to coordinate its limbs to achieve phototaxis without a central brain and without explicit communication between the modules. As such, directed behavior at the system level appears in this single experiment to emerge from local sensory feedback without knowing the assembled body plan.

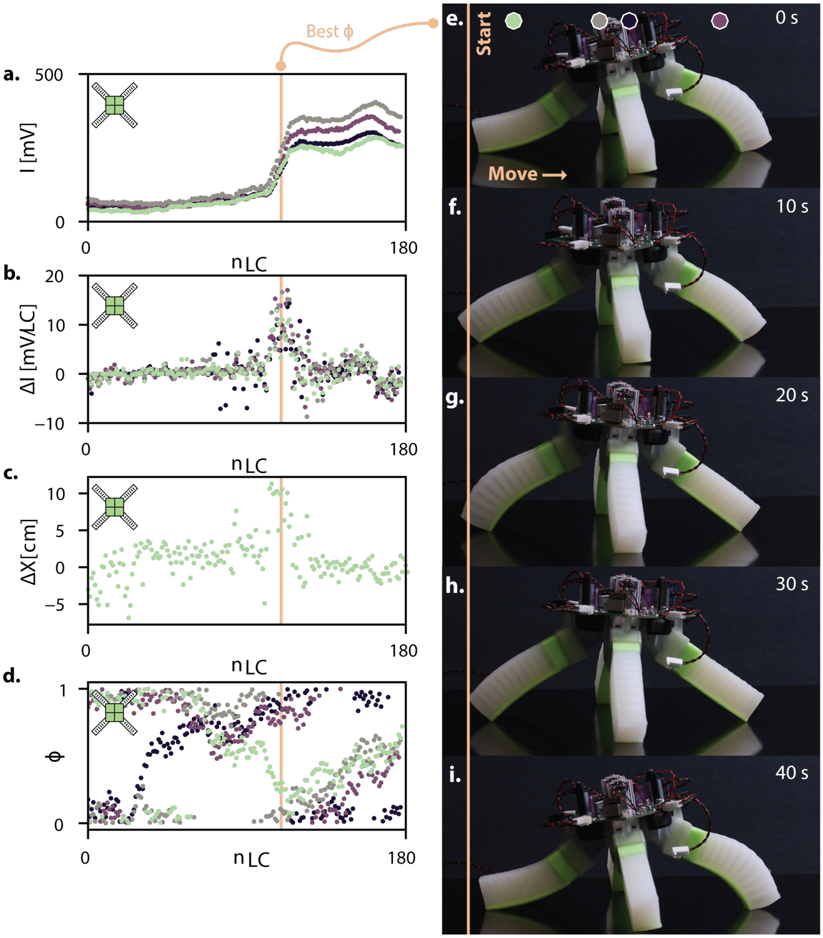


Fig. 2.3: **Observed phototaxis behavior of a four-module system single experiment. a**, Light measurements, I, from the experiment shown in Fig. 2.2d. **b**, Difference in light measurements between learning cycles, ΔI. **c**, Displacement of the assembled robot during a learning cycle, ΔX, obtained from image tracing. **d**, Actuation phases, ϕ_i, for each module. **e-i**, Fastest observed phototaxis behavior of the assembled system during the experiment in Fig. 2.2d.

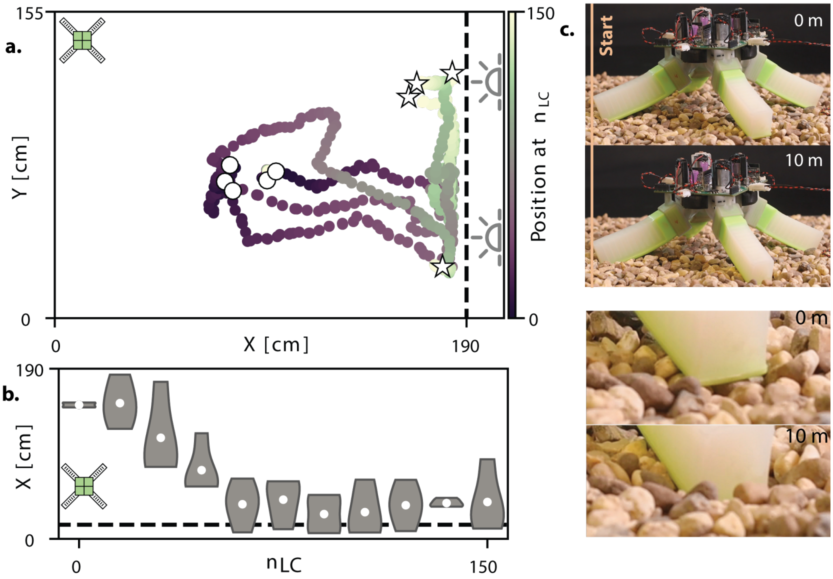


Fig. 2.4: **Observed phototaxis behavior of a four-module system, multiple experiments. a**, Trajectories of five repetitions of the previous experiment, for 150 N_{LC}. Circles and stars indicate the start and end positions, respectively. **b**, Distribution of X positions of the system over time during the repeated experiments. The experiment with a broken module at the start of the experiment is excluded from the distributions. This experiment can be found in Fig. S2.2. The position is measured from the side of the canvas closest to the light. The dotted line marks the distance with the highest light intensity on the canvas. **c**, Similar phototaxis experiment conducted on an irregular surface.

2.3 ANALYZING THE PHOTOTAXIS BEHAVIOR OF A FOUR-MODULE CONFIGURATION

To understand how the assembled system learns to perform phototaxis, we analyze four different metrics from the single experiment of the four-module configuration. The first metric is given by the data from the four light sensors inside the modules, and is shown in Fig. 2.3a. These sensors all measure an initial decrease in light intensity (I), followed by only small increases in the light intensity up to $n_{LC} \approx 110$. The following steep increase in light intensity I indicates an apparent behavior switch and increase in performance. Another behavior switch is observed starting from $n_{LC} \approx 130$, where we observe a plateau with only minor fluctuations in I.

However, it is important to note that the modules do not directly use the absolute value of I for learning. Instead, each module's behavior depends on the difference in light intensity as a result of moving. From this second metric (Fig. 2.3b), we see that ΔI starts negative as the system moves away from the light. It transitions to positive ΔI within $n_{LC} \approx 10$ learning cycles. ΔI slowly increases up to $n_{LC} \approx 122$, where we find the fastest increase. This is followed by a drop in the change in light intensity at $n_{LC} \approx 130$ to around zero as the system reaches the end of the canvas and is not increasing light intensity anymore.

We should mention that the inverse exponential relationship between light intensity and distance means that these results do not reflect the assembled system's actual speed. Therefore, we introduce in Fig. 2.3c the third metric that indicates the global movement speed ΔX of the assembled system (change in X-position per learning cycle). While the overall trend with the individual measured light intensity is the same, we can observe that between $115 \leqslant n_{LC} \leqslant 130$ the movement speed of the system reaches a stable maximum of $\Delta X \approx 10$ cm/cycle, indicating that it is capable of exploiting stable and relatively fast movement in the direction of the light source.

The observed behaviors are also reflected in the final fourth metric in Fig. 2.3d, which indicates the individual phases of each module throughout the experiment. Initially, for $n_{LC} < 20$, the phase order continually switches as the system explores different gaits. From $20 < n_{LC} < 100$, the front limb (the limb closest to the light source) is actuated first, followed by the other three. Beyond $n_{LC} > 100$, a relatively fast gait emerges where the back limb is actuated later than the center two (Fig. 2.3e-i, Movie S2.1), a change that drastically influenced the behavior. For $n_{LC} > 130$ the phases start to quickly change again, as the system is learning how to stop moving to stay close to the light source. These results demonstrate how the system learns to utilize the variable contact friction of its limbs with the ground for locomotion without needing a central brain or a model of its behavior.

To assess the repeatability of this experiment and the learned behavior, we run the experiment five times, each time randomizing the initial phases. The trajectories of all five experiments are depicted in Fig. 2.4a. We found that in four of the experiments, the system learned to move towards the light and reach the location with maximum intensity (Fig. 2.4b). However, in one experiment, the system did not move (2 cm over 150 learning cycles), which can be attributed to a pump failure at the start of the experiment, resulting in one paralyzed limb (Fig. S2.2).

2.4 ROBUSTNESS THROUGH EMBODIED COMPUTATION

Even though one could consider the experiment in Fig. 2.4a with the paralyzed limb a failed experiment, it underscores an interesting dynamic between embodied computation (learning in the modules) and mechanical intelligence (the mechanical behavior enabled by the assembled soft actuators and their interaction with the environment): the mechanical behavior of the system can influence the ability to learn. Apparently, the potential behavior of a four-module system with only three active legs limits the capabilities of the embodied computation. Similarly, when we place a fully functioning four-module system on a thin layer (2 cm) of gravel (see Fig. S2.2 for a detailed explanation of the setup), we observe that the system is also incapable of moving as it starts to dig itself into the gravel (Fig. 2.4c, Movie S2.1). In other words, in both scenarios the system exhibits limited mechanical intelligence as no gait seems to exist that allows it to displace. Hence, the system is not able to learn any directed behavior.

In contrast, Echinoderms are robust to changes in the environment and body morphologies. For example, researchers have shown that echinoderms can adapt to changes in morphology, such as a reduced number of legs [57]. This points towards introducing redundancy as an aid to increase the potential mechanical behavior that our system can exhibit. Therefore, we begin by increasing the number of modules in the system and changing the body architecture by replacing the type of PLA connectors between the modules (Fig. 2.5). From phototaxis experiments, we find that systems with five and six modules learn to move toward the light in a manner comparable to the four-module system (Fig. 2.5a-b).

Interestingly, the fastest observed gaits that emerge for both systems are different (Fig. 2.6a-e). For the five-module system, the two middle limbs inflate, as seen in Fig. 2.6a, followed by the inflation of the two back limbs in Fig. 2.6b, causing the tips of the inflated actuators at the back to creep toward the center of the system. Subsequently, the front limb is actuated to propel the system forward, as it completely lifts the two middle limbs off the ground and causes the back limb tips to stick in place as they deflate. Instead, for the six-module system we find a gait similar to the four-module system, a propagating wave (Fig. 2.6f-j). Initially, the three back limbs are actuated as seen in Fig. 2.6g, followed by the three front limbs in Fig. 2.6i. Note that we do observe a significant reduction in displacement when comparing the six-module system ($\approx$ 2.5 cm/cycle) to the five-module system ($\approx$ 4 cm/cycle), which is likely due to the increased system weight and distance of each actuator to the center of gravity so that it becomes more difficult for each actuator to lift the system.

In addition to these results, which demonstrate how the system harnesses its embodied computation to learn specific gaits that would be challenging to predict and optimize a priori, embodied computation also enables adaptability to unpredictable environmental influences. For example, upon closer inspection of the trajectory of the five-module system (Fig. 2.5a) we observe that at approximately $N_{LC} = 50$ the system briefly loses track of its phototaxis, as it trips over its power line, which causes a relatively quick rotation (Movie S2.2). Since the system's short-term memory only retains the change in light intensity (and not its orientation), the modules do not directly notice this rotation through their light measurements (Fig. S2.3a). After this interaction, the system adjusts its phases within 50 learning cycles N_{LC}, and redirects its displacement towards the light.

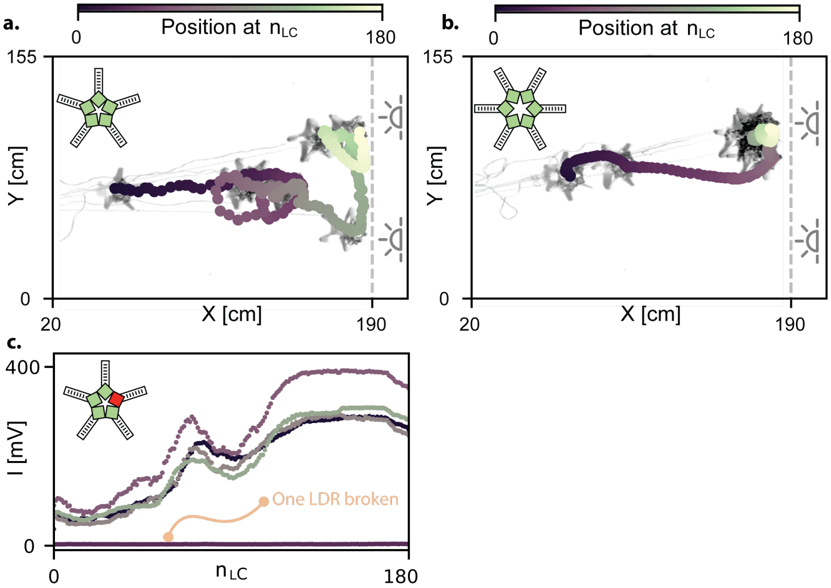


Fig. 2.5: **Observed phototaxis behavior of (damaged) five- and six-module configurations.** Trajectories of experiments with a **a**, five-module configuration **b**, six-module configuration, and **c** five-module configuration with damaged light sensor. Trajectory of the experiment can be found in Fig. S2.3c.

This adaptivity can also be seen in a five-module system where we intentionally damage one of the light sensors, such that it does not give any reading. As shown in Fig. 2.5c and Fig. S2.3c, this damage does not limit the ability of the system to perform phototaxis. Note that the damaged module exhibits random phase changes as a result of the zero light intensity reading, so that the other four modules constantly have to adapt to the damaged module's random behavior (Fig. S2.3d).

2.5 ADAPTABILITY TO MORPHOLOGICAL CHANGES

Based on the evidence of the observed robust behavior in previous experiments, we continue to test the ability of our proposed embodied computation strategy to adapt to morphological variations. We first focus on the six-module system, and deliberately simulate dam-

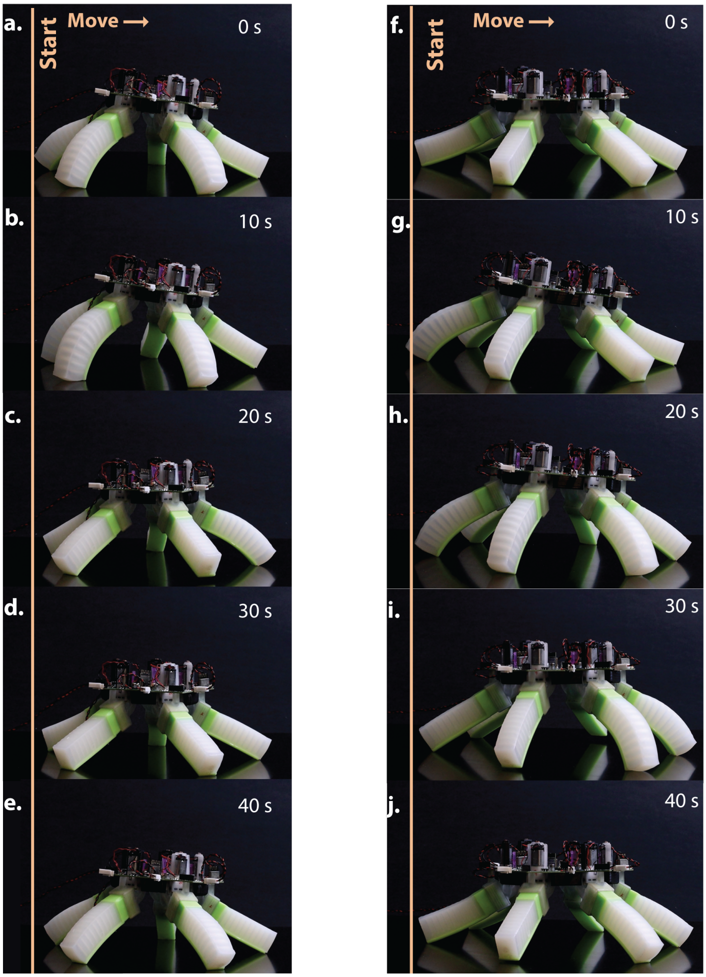


Fig. 2.6: **Observed gaits five- and six-module configurations. a-e**, Fastest observed phototaxis behavior of the assembled system during the experiment in Fig. 2.5a. **f-j**, Fastest observed phototaxis behavior of the assembled system during the experiment in Fig. 2.5b.

age to change its morphology by removing limbs. Fig. 2.7a illustrates how limbs can be detached, rendering modules without actuators non-operational. Yet, as the system does not have a central brain, the embodied computation remains functional, such that the remaining modules try to adapt their behavior to the new morphology.

Fig. 2.7b-c demonstrates that even with limbs missing, the system learns to perform phototaxis in two different tested morphologies. Analyzing the emergent behavior can teach us about the interaction between embodied computation and mechanical intelligence. For example, we see that the configuration of Fig. 2.7b is capable of moving in the direction of the light initially. However, the new morphology seems susceptible to rotation, exemplified by the event that occurs around $n_{LC} \approx 150$, where a relatively quick rotation causes the system to enter a dynamic learning process that overshoots, causing it to rotate back and forth in a semi-circle. Additional information on these experiments is provided in Fig. S2.5.

Keeping the number of active modules that learn constant (i.e., keeping the computational potential identical), we next explore the system's response to more diverse and heterogeneous morphologies. We introduce inverted bending actuators and a longer bending actuator that are 130 % of the original length (Fig. S2.4), and establish four distinct body configurations in Fig. 2.8a-d. Fig. 2.8e shows the trajectories of the learning experiments for these configurations, indicating that all configurations perform phototaxis.

Yet, we also find clear differences between the results of the original four-module configuration shown in Fig. 2.1-2.4. Firstly, we find that the configuration with inverted actuators (Fig. 2.8a) is clearly slower than the original configuration, where a gait emerges (Fig. 2.8f-j) that appears to be reversed (back to front propagation of actuation) compared to the gait observed in Fig. 2.3e-i (Movie S2.3). Secondly, we see that the configurations in Fig. 2.8b and Fig. 2.8d at some point rotate away from the light source. Similar to the results of Fig. 2.7b-c, these results show that different morphologies can be susceptible to rotation. Lastly, the fastest movement occurs in a configuration where half of the actuators are flipped (Fig. 2.8c), potentially using the additional friction from the flipped actuators as anchor points (details in Fig. S2.6 and Movie S2.3). Yet, the same morphologies also demonstrate clear and sudden

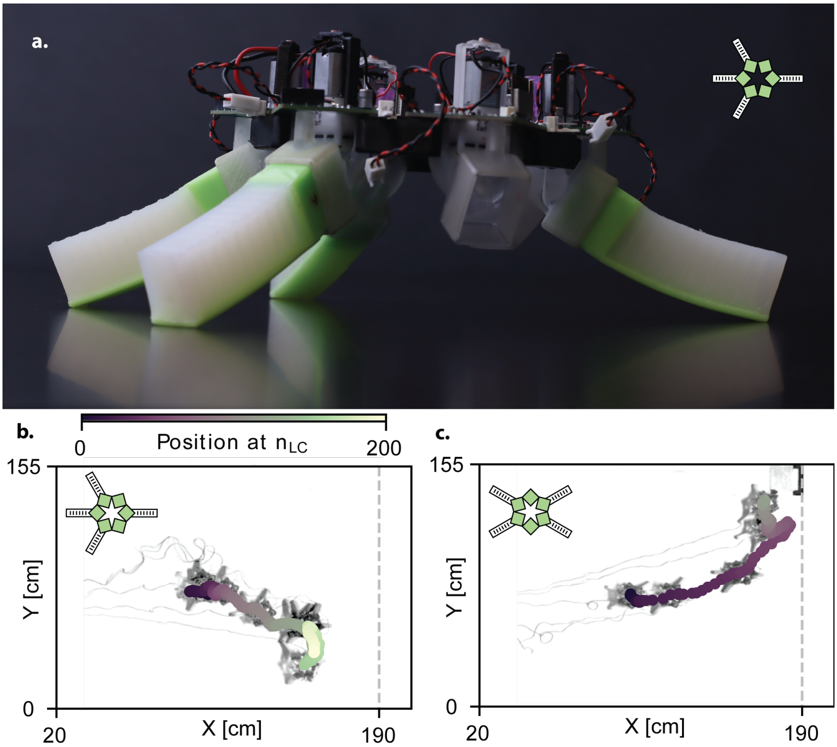


Fig. 2.7: **Adaptability of embodied computation approach to removing limbs.** **a**, Six-module configuration with two limbs removed. **b-c**,

behavior switching as seen by abrupt trajectory changes, indicating more unstable behavior.

Therefore, these results indicate that even though the embodied computation is capable of handling various morphologies, the emerging behavior that the system shows is the result of the interaction between the embodied computation and mechanical intelligence. Designing soft robots that demonstrate robustness to damage requires redundancy in the design of the system's mechanical intelligence, to which the embodied computation can effectively adapt.

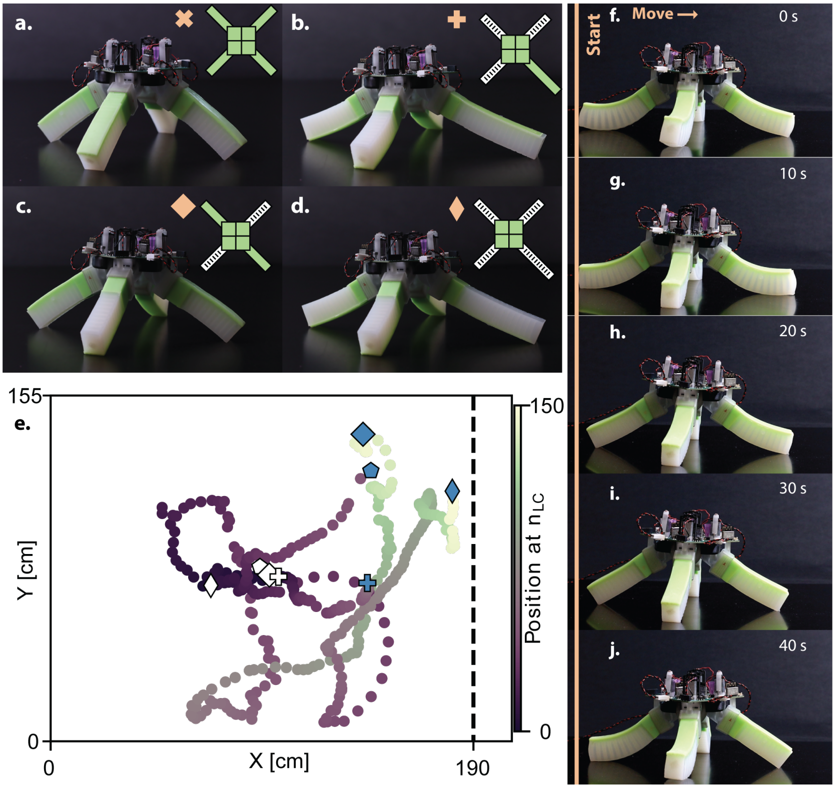


Fig. 2.8: **Adaptability of embodied computation approach to morphological variations. a-d**, Four different body configurations including longer and flipped actuators, and **e** the observed trajectories during phototaxis experiments. **f-j**, Gait during the best phases of **a**.

Changing morphology is not the only way to increase the redundancy embodied in the system. Based on the learned behaviors in the phototaxis experiments performed so far, we find that one of the primary limitations of the PneuNet bending actuators is the reciprocal nature of their inflation-deflation cycle. As a result, several actuators need to interact to enable the friction symmetry to break down at the surface, which is the basis of all the gaits observed so far. Moreover, this also constrains the speed of the system and the variety of terrains it can traverse (Fig. 2.3i). Therefore, we next increase the potential of the mechanical intelligence in each module by leveraging more complex nonlinear behavior in each actuator, allowing it to undergo non-reciprocal (i.e., full-step) motion.

To explore the capabilities of our embodied computation approach, we have developed an actuator that leverages the nonlinear mechanics of thin shells to embed a full-step motion during the inflation and deflation cycles, relying on similar mechanics as [8]. The actuator is shown in Fig. 2.9a, where we show how it goes through four phases during a single actuation cycle. *(1)* the shell starts fully deflated. *(2)* The sequence begins with the inflation of the thinner top half of the shell, which pushes a pillar (i.e., the foot that will be in contact with the surface) downward. *(3)* This is followed by the inflation of the thicker bottom half of the shell that extends the pillar outward. *(4)* In contrast, during deflation, the circumferential stiffness of the shell maintains the inflated state of the thicker bottom side while the thinner top side deflates. *(1)* This is concluded with the collapse of the thicker bottom of the shell to reset the actuator to its initial state. The full-step behavior of this actuator can be seen from the hysteresis of the actuator's tip location as depicted in Fig. 2.9b, which clearly differs from the behavior of the PneuNet. As a result of the shell buckling that underlies the hysteretic behavior, we also find that these full-step actuators are more susceptible to interactions with the environment, which complicates feed-forward control without embodied computation (Fig. S2.7b).

Next, the modules are interconnected using the same connectors as previously used to create radially symmetric morphologies. When performing a phototaxis experiment with a four-module configuration, we also observe the phototactic behavior (Fig. 2.9c), where the traveled distance between consecutive learning cycles appears to be significantly increased compared to experiments with PneuNet actuators. Moreover, as expected, the light measurements appear noisier over time, likely due to the actuators' increased stroke and tilting that move the sensors out of plane (Fig. 2.9d). Repeated experiments shown in Fig. S2.7d-h reveal that trajectories vary widely and often appear erratic. Still, in all trials, the system can handle the increased (erratic) movement and achieve movement towards the light source (Fig. S2.7d-h).

The full-step nonlinear motion also opens up the opportunity to traverse more diverse terrains, as the motion is now characterized by the lifting of the limbs instead of the friction-dominated moments we found with the PneuNet actuators. Fig. 2.9e shows how a four-module system comprised of the full-step actuators can learn to achieve phototaxis on an uneven terrain comprising a thin layer of gravel, which was not possible for a four-module system comprised of PneuNet actuators (Fig. 2.3i). All three repetitions of the experiment can be found in Fig. S2.8. When comparing the fastest gait of a four-module assembly of full-step actuators on a flat surface and on gravel, we find a similar gait (Fig. 2.10f-g) in which the system initially lifts off the ground due to inflation of all actuators, after which it propels forward by deflating the two front modules.

These observations underscore the profound synergy between the system's 'body' and brain,' emphasizing that modifications of the body can enhance task performance. Yet, they must not interfere with the predictability of system behavior. Even though phototaxis can still be achieved for these systems comprising nonlinear and non-reciprocal behavior, current shell actuators seem to push the complexity of the system to its limits, and we expect that at some point, short-term variations become too large for the embodied computation to keep up.

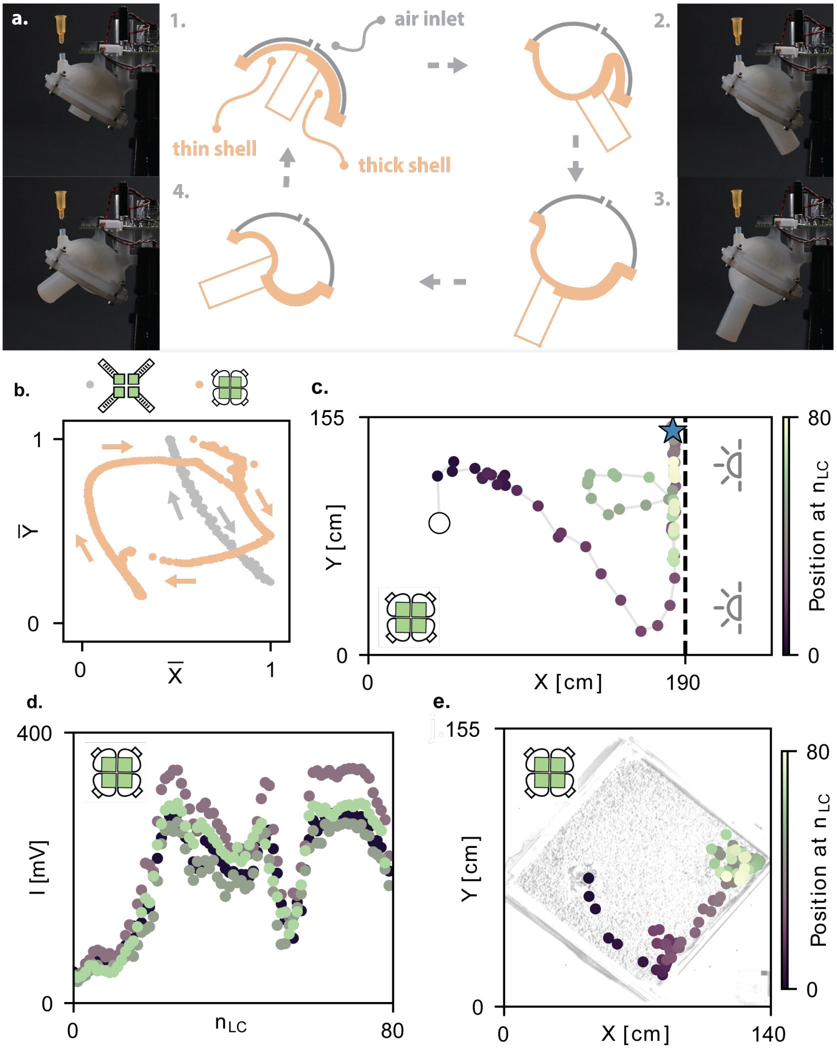
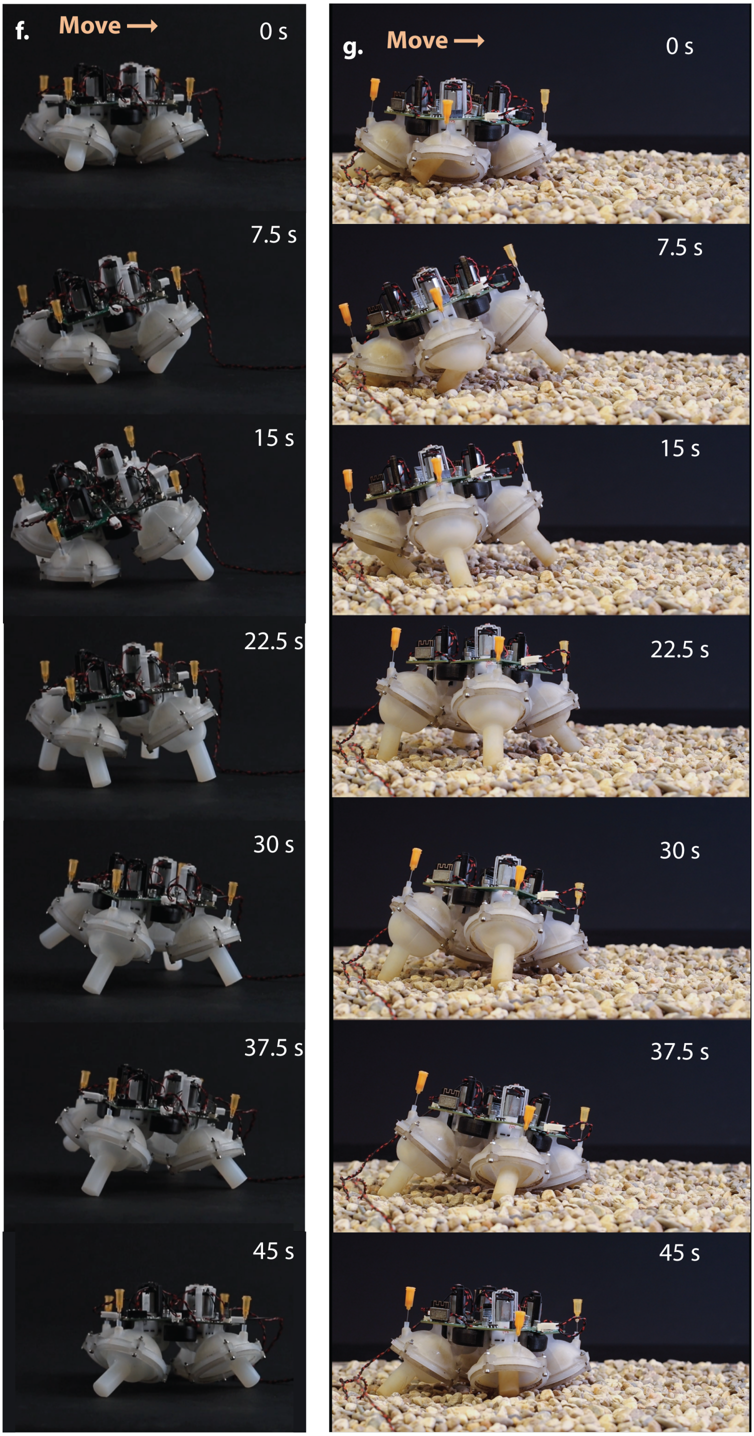


Fig. 2.9: **Phototaxis capabilities of four-module systems comprising nonlinear full-step actuators. a**, Experiments and schematic of the inflation-deflation cycle of the "full-step actuator." **b**, Reciprocal behavior of the PneuNet actuator and non-reciprocal behavior of the full-step actuator, illustrated by the trajectory of the actuator's tip normalized to the minimum and maximum positions. **c**, Trajectory of a phototaxis experiment for a four-module configuration with full-step actuator, for 80 N_{LC}. **d**, Light measurements in each module obtained during the same experiments. **e**, The same phototaxis experiment on an irregular gravel surface. More details on the gravel setup are provided in Fig. S2.2.



Caption on next page.

Fig. 2.10: **Observed gaits of four-module systems comprising nonlinear full-step actuators. f**, Fastest observed phototaxis behavior during the experiment in **c**. **g**, Fastest observed phototaxis behavior during the experiment in **e**.

2.7 CONCLUSIONS

In this work we demonstrated how "embodied computation" can leverage the potential of the mechanical intelligence present in soft-limbed robots to achieve a dedicated goal (i.e., phototaxis). In contrast to most approaches to control soft robots that, e.g., depend on predefined or manual sequencing or centralized model representations of the robot's behavior, our developed modular soft robotic platform can achieve emergent directed locomotion without a central brain, a-priory system knowledge, or explicit communication.

We demonstrate that such an embodied computation approach can provide robustness against changes in body morphology, actuator behavior, dynamic environments, or damage. This is illustrated by the various gaits learned in a range of different circumstances (Fig. 2.6a-e, 2.6f-j, 2.8f-j, and 2.10f and g), without requiring any user interference. While we mostly focused on online learning in this work, our platform can also be used to find gaits for a wide variety of soft robots. For example, the same embodied computation strategy can even be put on inflating cubes where the locomotion modality changes to peristaltic-like crawling (Fig. S2.11).

Even though we were inspired by the capabilities of echinoderms that originate from their decentralized architecture, the implementation of the learning strategy we embodied differs from the exact computations performed by echinoderms. As such, our approach only represents one of the many possible ways to implement embodied computation, and future research is needed to explore alternative approaches that can lead to different and perhaps more robust behavior. Similar to how natural systems have also co-evolved their body and brain to achieve useful emergent behavior, key in future explorations will be the co-design of both the mechanical intelligence (i.e., body) and embodied computation (i.e., brain). For example, in our system we observed two situations for which the system stopped performing phototaxis. The first case is rather obvious, as it seemed to be the result of the system not being capable of locomotion at all, e.g., due to damage as shown in Fig. 2.4a or due to interactions with the environment as shown in Fig. 2.4c. The second case is less obvious and is the result of the interaction between body and brain, where the system got stuck in a dynamic limit cycle due to a delay in sensory observation of the environment (e.g., Fig. 2.7b). Note that in nature, similar unwanted emergent behavior can be observed, such as death spirals in army ants.

We believe that the implications of our research reach beyond the existing platform, promising significant insights on how to incorporate more autonomous behavior in the fields of soft robotics, swarm intelligence, and micro- and nano-robotics [10, 80, 89]. The fact that our algorithm is fully decentralized and only requires a few lines of code makes it especially suitable in those applications where computational resources are limited by size, cost, and weight, or where behaviors and interactions are difficult or impossible to model *a priori*. Future research could also explore hierarchical approaches similar to the "sub-brain architecture" of the octopus and further diversify the tasks the system can perform. As such, we believe our insights on the interaction between embodied computation and mechanical intelligence could lead to a new generation of soft robots that are not only inspired by nature, but are also capable of matching its versatility, robustness, and autonomy.

SUPPLEMENTRY MATERIAL FOR CHAPTER 2.

S2.

This Chapter provides additional details to support the *Chapter 2*. It includes comprehensive descriptions of the fabrication methods, experimental setup, and protocols used in this study, as well as information on the gravel surface employed in the experiments. Supplementary figures and captions for Movies S2.1–S2.5 are also provided.

Other Supplementary Material for this Chapter includes the following:

S2.1 SUPPLEMENTARY MOVIES S2

Movie S2.1. Introduction of the system architecture and individual modules. During a phototaxis experiment, a four-module configuration harnesses its embodied computation to learn to move towards a light source. We also show the fastest-moving walking gait that was observed for the four-module configuration. The assembled robot does not locomote on a granular terrain.

Movie S2.2. During a phototaxis experiment, a five-module configuration learns to move towards a light source. We also show the fastest-moving walking gait that was observed for the five-module configuration. A phototaxis experiment for a six-module configuration shows similar results, yet the fastest-moving gait is different from that for a five-module configuration.

Movie S2.3. An alternative four-module configuration where the actuators are flipped upside down also achieves phototaxis, yet learns its own unique walking gait.

Movie S2.4. To increase the potential mechanical intelligence, we introduce a full-step actuator module. These actuators are more susceptible to changing their behavior based on environmental influences, as demonstrated by manually interacting with the actuator. The same embodied computation implementation also allows a configuration with four of these full-step modules to learn how to perform phototaxis, even on granular terrains.

Movie S2.5. The same embodied computation approach can also be used on different platforms, such as inflatable cube actuators that we introduce here. Phototaxis experiments with a four-module configuration of these inflatable cubes show that phototaxis is learned over time.

Link to the Movies.

S2.2 FABRICATION OF THE MODULES

Each robot module is composed of several custom structural parts: A 3D-printed body printed from VeroClear (Stratasys), on a PolyJet 3D printer (Eden260VS, Stratasys), custom PCB (see Fig. S2.11), airpump (YYP032, Huizhou Yingyi Motor Co.), LDR light sensors (NSL-19M51, Luna Optoelectronics) with a 3D-printed PLA holder (printed on an Ultimaker 3 with BASF Ultrafuse PLA filament, color black, 2.85 mm), various soft actuator (see below) and a needle that is used as a fluidic resistance that leaks air from the actuator to atmosphere.

In each module the algorithm runs on a microcontroller (ESP8266), placed on a rectangular custom PCB. The PCB is connected to the LDR sensors that are mounted on a PLA holder above the PCB. Each PCB is connected to its own 3.2 v air pump, which is press-fit into the 3D-printed body. A 5 mm silicone tube (3 mm inner diameter, 5 mm outer diameter, from Rubbermagazijn) and silicone grease seal the interface between the pump and its enclosure. The pump is secured by the PCB and its LDR holder, bolted into place with M3×6 inbus screws. A resistive needle (Metcal) is also press-fitted into the Veroclear Body enclosure, sealed with another short silicone tube (0.1 mm inner diameter, 1 mm outer diameter) and silicone grease.

Soft PneuNet actuators are cast using two-part silicone (Smooth-On DS20 in opaque white and Elite Double in green) injected into custom Veroclear 3D-printed molds sprayed with Ease release 200 (Smooth-On). The fabrication follows a two-stage process: first casting the white elastomer for the actuator chambers, letting it cure, and then pouring the green silicone to form a stiffer layer on the bottom of the actuator. Finally, the water-soluble inner mold (BVOH printed on an Ultimaker) is dissolved by flushing with pulsating water pressure. A detailed description of the manufacturing process is given in previous work [99]. The PneuNet actuators are press-fitted in their body, making them airtight using silicone grease.

Full-step hysteretic actuators are cast using two-part silicone (Smooth-On DS20 in opaque white) injected into a custom Veroclear 3D-printed mold sprayed with Ease release 200 (Smooth-On). Next, the silicone shell is placed in the body, and an airtight seal is created by bolting down the holder ring (see Fig. S2.7).

Individual modules are linked together by employing 3D-printed PLA (BASF Ultrafuse PLA filament) connectors with embedded magnets to facilitate attachment. A different type of connector is used for each morphology of the assembled system.

S2.3 EXPERIMENTAL SETUP

We follow the same overall procedure as previously reported [108]. Experiments take place on a 195 cm × 155 cm white multiplex platform, surrounded by a black multiplex frame (Fig. S9b). Once powered, each module is wirelessly initialized via a master-slave protocol using an ESP32 configured in a one-to-many network with the ESP-NOW library. After this one-time synchronization, every module operates autonomously and transmits local measurements at the end of each learning cycle. These data packets are collated by the ESP32 master, which then writes them to a text file on a connected laboratory computer. While this logging process helps us visualize the results, it does not influence the phase updates happening within each module's PCB. A camera (GoPro Hero 4) is positioned 1.11 m above the center of the white platform and records overhead images at the end of each learning cycle. The camera is controlled wirelessly from the same computer that gathers the sensor data. To accurately determine the modules' positions, we first correct for optical distortion and camera perspective using the OpenCV Python library. Four ArUco markers placed at the corners of the platform serve as reference points, enabling us to map pixel distances to physical dimensions. Next, we track a single marker (approximately 4 cm wide) in the center of the robot assembly.

S2.4 EXPERIMENT PROTOCOL

The experimental procedure ensures the reproducibility and reliability of the experiments while minimizing errors due to misalignment of components or system failures. The protocol follows a structured sequence of assembly, testing, and data collection, which is detailed below. Such an approach has proven to be essential in this study, given the relatively long time (up to 17.5 h) that each phototaxis experiment takes.

Assembly of the Experimental Configuration. The setup of the experiment involves assembling the modules and their associated components according to the predefined configuration. This includes assembling modules, sensors, and structural components, ensuring that all elements are correctly positioned and securely fastened. Particular attention is given to the alignment of the full-step actuator membranes in their holder, as misalignment in their membrane holders can introduce unintended rotational bias in a predefined direction.

Pre-Experimental Testing Outside the Experimental Area. Before introducing the system into the experimental arena, a preliminary test phase is conducted in a separate environment to verify the proper functioning of all components. This includes checking the operation of the system's actuators (not leaking), sensors (working and collecting data), the PCB (operating correctly and connecting to the central computer), and the inflation-deflation cycle functions (no broken pump, leakage, or clogged leakage needle). Any malfunctioning components are identified and corrected at this stage to prevent failures during the main experiment.

Test Phase in the Experimental Arena. Once the preliminary testing is completed, the assembled robot is introduced into the designated experimental area for an initial functional test of around 20 learning cycles. This phase ensures that the assembled configuration operates as expected in the controlled environment. Special attention is given to experiments that include the full-step actuator modules, as improper alignment of the membranes within their holders is found to induce excessive rotation in a predefined direction, affecting the accuracy of the experiment. If such misalignments are detected, necessary adjustments are made before proceeding.

Execution and Data Collection. After confirming the stability of the setup, the main experimental phase starts. In the case of repeated experiments, the experiment is repeated multiple times in succession, with data recorded for each iteration. At the start of each experiment, the robot is placed in a random orientation at a specific location in the arena. The experiment is only excluded from the reported dataset if a significant component failure or a crash of the experimental setup occurs. Each valid repetition is documented in the article. In case of a component failure, we return to step one of the experimental procedure. If the component can be replaced without disassembling the modules, we continue directly and restart the experiment.

Post-Experiment Video Selection and Recording. Following the experiment, the phases that resulted in the highest displacement towards the light are selected for a repeated experiment where the movement is recorded. To maintain consistency in the videos and for lighting purposes, this process is conducted outside the experimental area by placing the system on a table and running a script that repeats the selected phases for ten actuation cycles. It is important that no modifications are made to the configuration, as variations between pumps, actuators, and other system components can introduce behavioral differences in the generated phases. This ensures that the recorded video accurately reflects the experimental conditions.

By following this protocol, we aim to maintain the robustness and reproducibility of the experimental results, minimizing errors due to misalignment or inconsistent component performance.

S2.5 DETAILS ON THE GRAVEL SURFACE

To conduct experiments on an irregular surface, we designed a dedicated gravel setup. A 1 m × 1 m multiplex wooden plate is used as the base, surrounded by 20 mm × 20 mm aluminum extrusions to form a containment box. The interior is lined with a 1 mm thick plastic sheet to prevent gravel loss. As a support layer, a steel mesh grid (as shown in Fig. S2.2d) is placed at the bottom. On top of the mesh, we distribute a 2 cm thick layer of gravel with grain sizes ranging from 6 to 10 mm, sourced from Cobo Garden (see Fig. S2.2b). This gravel setup is then placed in the same experimental setup for the other experiments. We align the gravel box with the light gradient so the system is tasked to move along the diagonal in order to maximize the distance towards the light within the box.

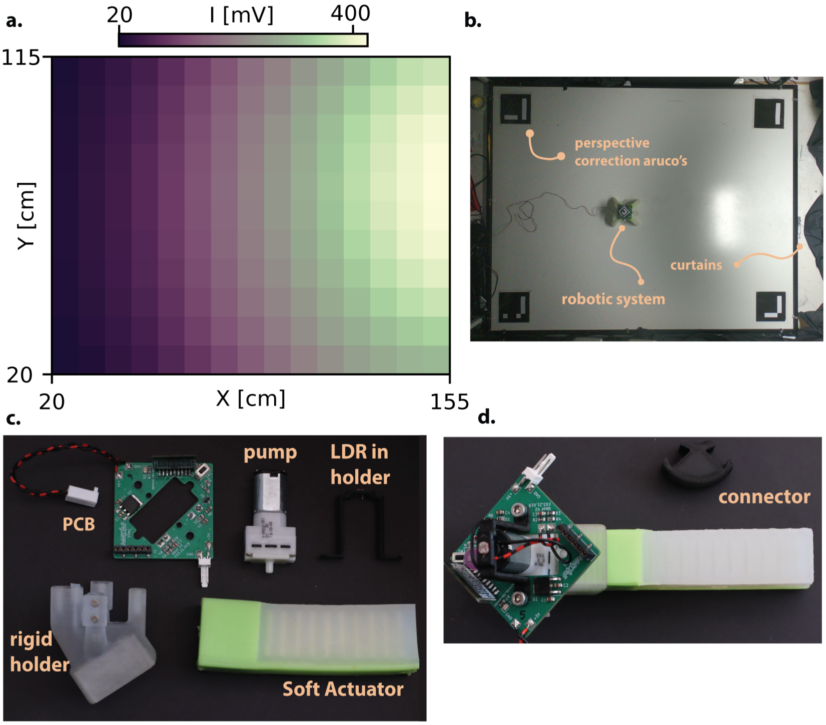
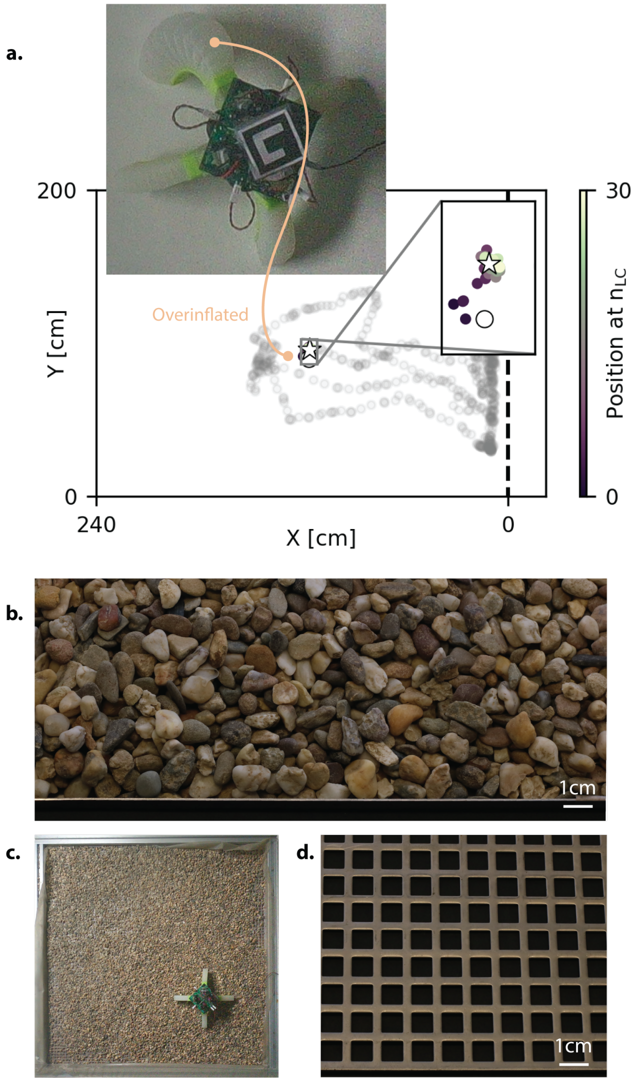


Fig. S2.1: **Experimental setup.** **a**, Light measurements of the experimental arena [108]. **b**, Photo taken during a phototaxis experiment using the GoPro Hero 4 that is used for data collection. The four ArUco markers in the corners of the arena are used to correct for the perspective of the camera and to convert the pixel distances to cm. The ArUco marker on top of the robot is used to track its position over time. The cable attached to the robotic modules is connected to a power source. The cable is only used to provide power, since experimental data used to interpret the behavior of the modules is transmitted over WiFi. Black curtains surrounding the arena are used to block any parasitic light from entering the experimental arena, keeping the lighting conditions consistent throughout the experiment. **c**, Components of the module used for the initial experiments that use PneuNet actuators. **d**, Assembled module, and a 3D-printed connector piece from PLA (using an Ultimaker) used in the four-module assembly. Similar connectors with different angles are used to configure different numbers of modules in the radially symmetric circle configuration.



Caption on next page.

Fig. S2.2: **Additional information corresponding to Fig. 2.3 and Fig. 2.4. a**, Experimental result of the failed phototaxis experiment that was left out of the batch experiment repetitions in Fig. 2.4a-b. During the startup of this experiment, one of the modules restarted, causing the pump to be on for two cycles without intermediate deflation. As a result, the actuator pressure reached significantly higher levels, likely causing the actuator to reach the ballooning stage of its pressure-volume curve. This resulted in an actuator that was not able to deflate sufficiently during its deflation stage and instead remained inflated throughout the entire experiment. Due to this, the two closest actuators lifted off the ground, such that the system was not able to move for any of the tested phases. **b**, photo of the gravel used in the experiments.**c**, Still images from a learning experiment in the gravel pit with a four-module configuration as depicted in Fig. 2.4c. From these results, we find that the system is not able to find phases that displace its center of mass. The reciprocal motion, as a result of the same inflation and deflation, slowly digs the system into the gravel instead of moving it. **d**, The steel rooster is used as the base to layer the gravel on.

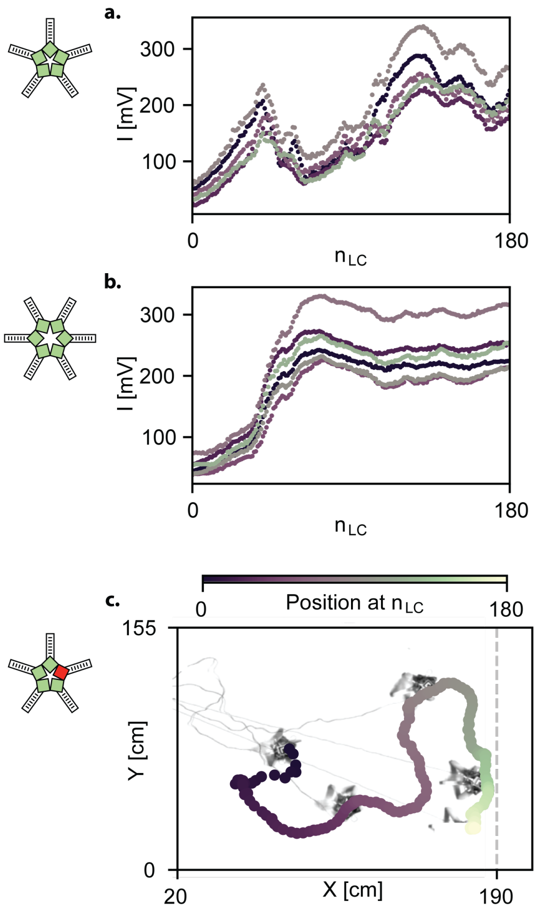


Fig. S2.3: **Additional information corresponding to Fig. 2.5 and Fig. 2.6. a**, Light measurements of the phototaxis experiment for a five-module experiment, corresponding to the trajectory depicted in Fig. 2.5a. **b**, Light measurements of the phototaxis experiment for a six-module experiment, corresponding to the trajectory depicted in Fig. 2.5b. **c**, Trajectory of the phototaxis experiment with the five-module configuration with one broken light sensor, corresponding to Fig. 2.5c

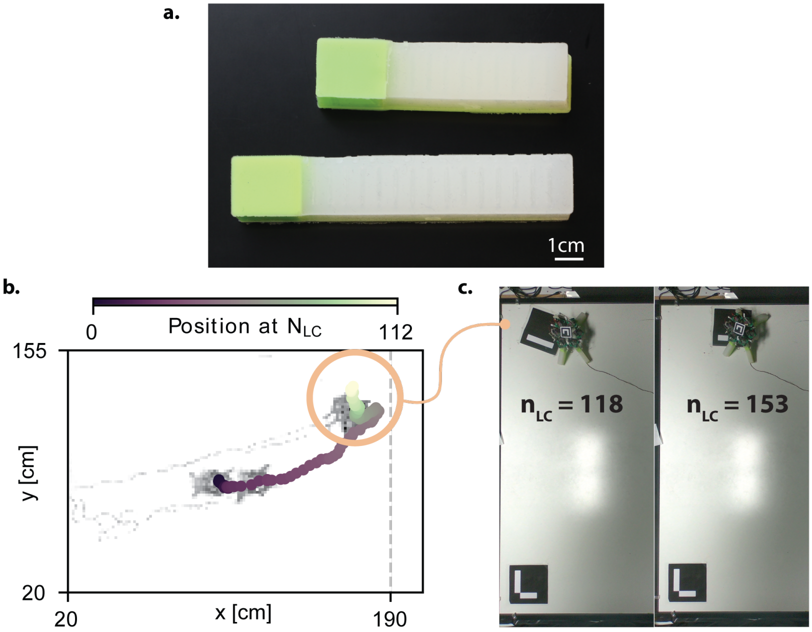
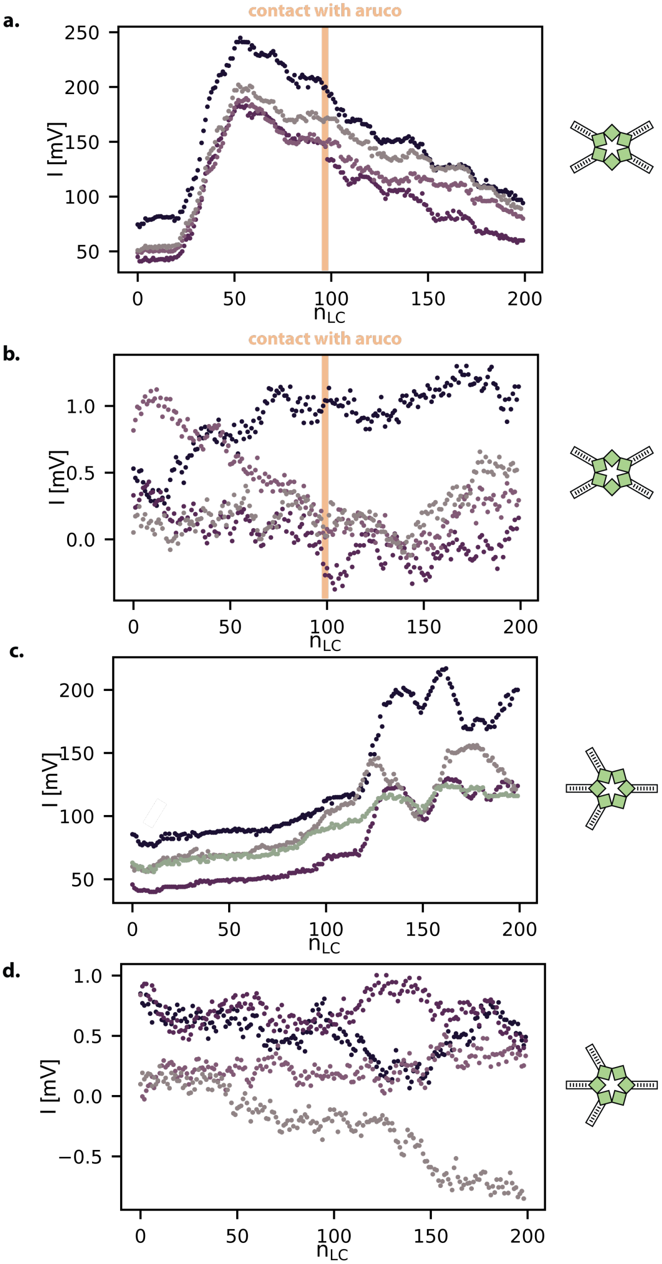


Fig. S2.4: **Additional information corresponding to Fig. 2.7 and Fig. 2.8. a**, Length difference of the longer (used in Fig. 2.7) and shorter (used in Fig. 2.1-2.7) actuators. **b**, Highlight of the location during the phototaxis experiment of Fig. 2.7c where the robot pulls the ArUco marker from the experimental arena. **c**, Demonstration of how the ArUco is pulled from the experimental canvas. As shown, one of the limbs drags the ArUco along. The detached ArUco has little friction with the canvas, making the limb on top of the ArUco slip. As a result, the system with only three effective limbs remaining is unable to direct its motion. **g**, Phases of the same experiment as depicted in f.



Caption on next page.

Fig. S2.5: **Additional information corresponding to Fig. 2.7 and Fig. 2.8 part 2. a**, Light measurements from the phototaxis experiment depicted in Fig. 2.7c. The orange line indicates the first contact with the ArUco. **b**, Phases of the same experiment show that there is a transition in the behavior before and after the system hits the ArUco. During the first half of the experiment, we observed little change as the system moved to the light. After the system pulls the ArUco from the ground, we observe how the system is still varying its phases. Still, as the resulting behaviors are disturbed by the ArUco, the system is not able to steer the phases in any particular direction, leaving the phase differences relatively equal for the remainder of the experiment. **c**, Light intensity for the modules during the phototaxis experiment depicted in Fig. 2.8a. **d**, Phases of the same experiment as depicted in f.

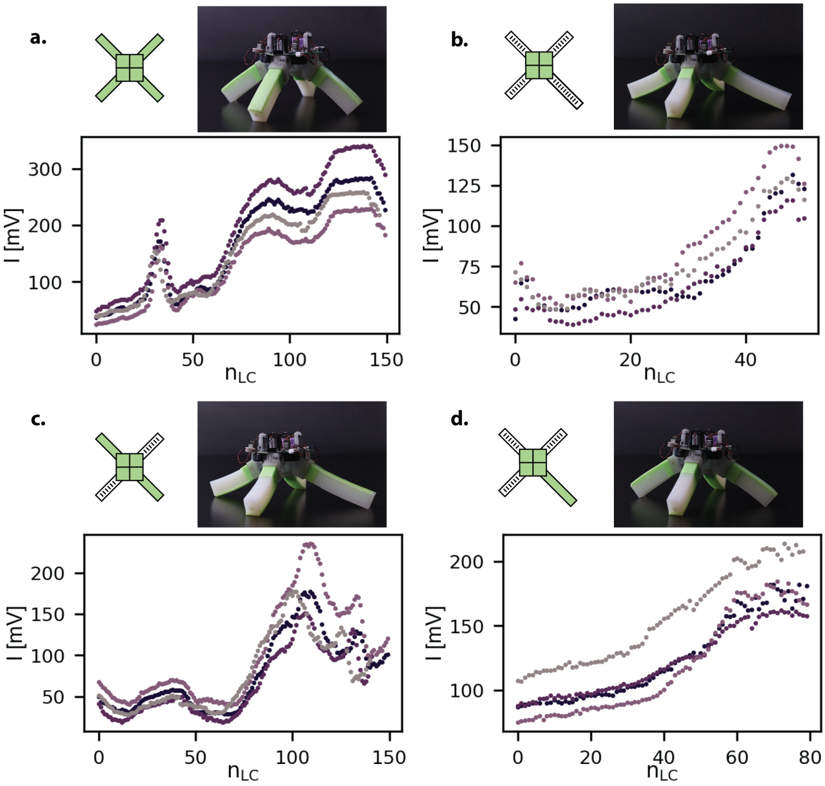
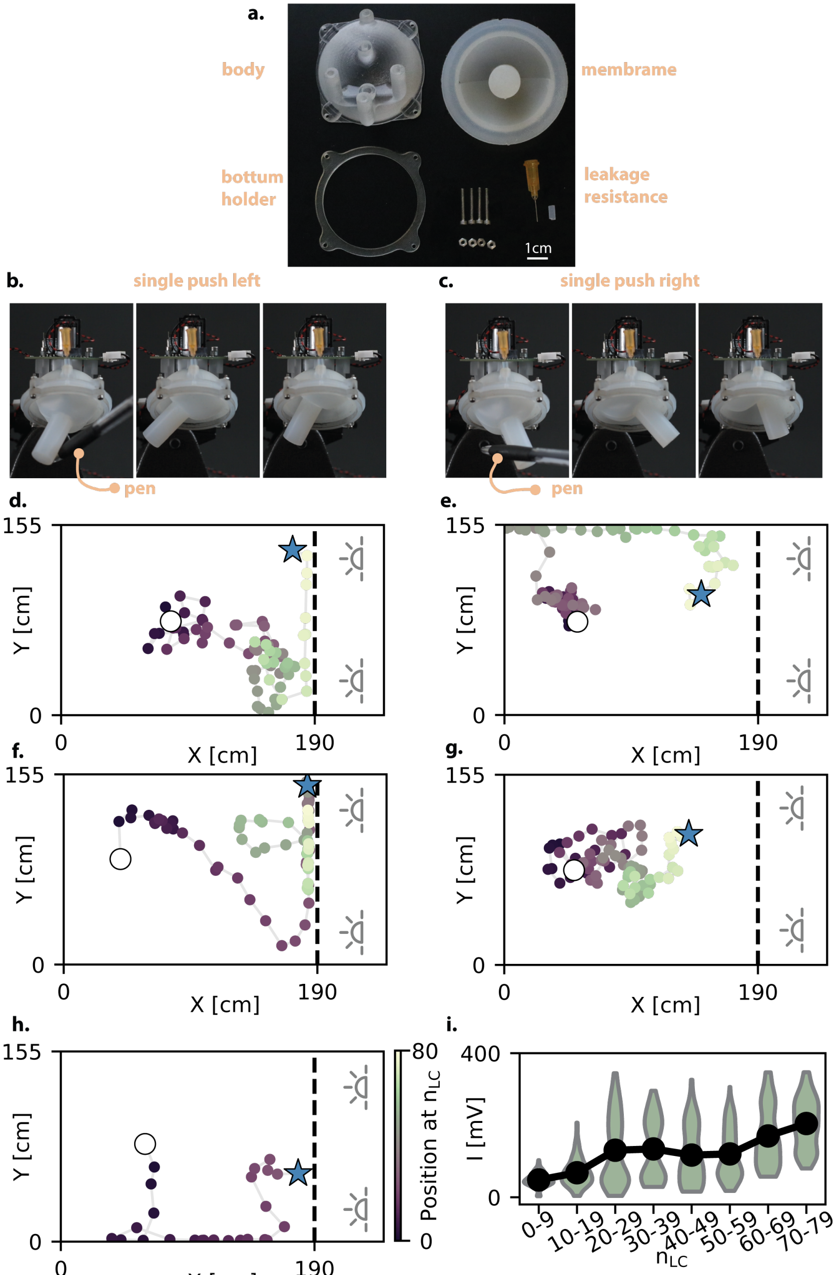


Fig. S2.6: **Additional information corresponding to Fig. 2.8. a-d**, Light intensity data of the phototaxis experiment for the four-module systems with various actuator lengths and arrangements. Note that during the experiment in **b** we only obtained 50 learning cycles both for the data transmitted from the modules and for the photos of the GoPro, as the central lab computer that collects and stores the data experienced a crash four hours after the experiment was started. The actual modules remained functional, and continued the experiment without collecting any data. A similar problem occurred for the experiment in **d**, where we were only able to collect data for 80 cycles. Note that for the study presented in this article, these were the only two experiments where such a crash occurred. For both experiments only the available data is presented in Fig. 2.7d.



Caption on next page.

Fig. S2.7: **Additional information corresponding to Fig. 2.9 and Fig. 2.10. a**, Module components for the full-step actuator. The body and bottom holder are printed on a Stratasys with Veroclear. The flow resistor (Metcal needle gauge 22) constantly leaks air to the atmosphere so that the module deflates when the pump is off. **b** An interaction with the environment (for the demonstration represented by an interaction with a pen) can change the trajectory of the tip during deflation. **d-h**, Trajectories of all five repetitions of the phototaxis experiments with the full-step actuators in a four-module configuration. The tracking of the modules is done with the same algorithm as for the experiments in Fig. 2.9. Note that as the modules in these experiments tilt the system more out of the plane, they caused reflections from the light source, making the ArUco undetectable. In those cases, we manually detect the corners in the image. **i**, Distribution of the light intensity measurements from the four modules of all five experiments, the average light intensity is depicted with a black line. As the learning cycles increase, we see how the distribution moves upwards, indicating the average increase in light intensities as the system moves towards the light.

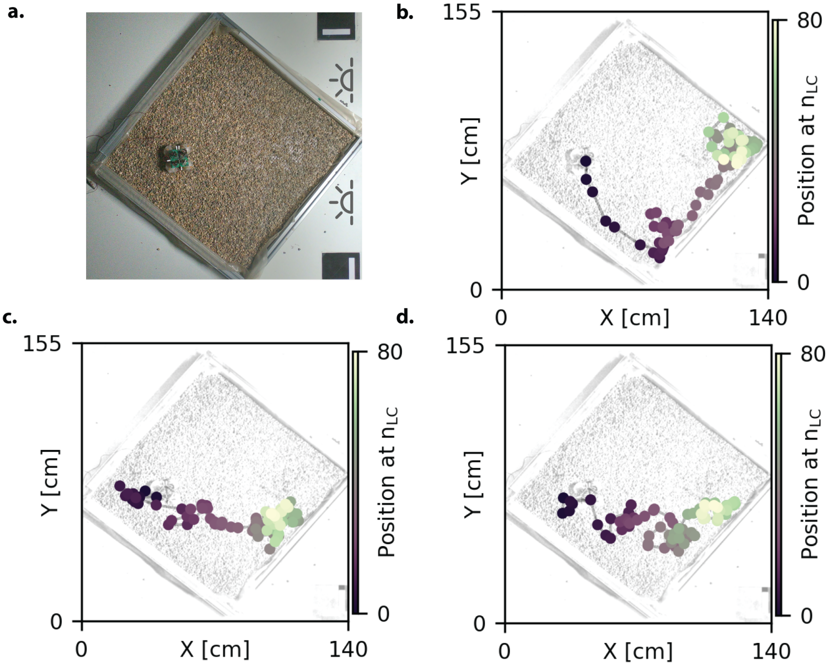


Fig. S2.8: **Additional information corresponding to Fig. 2.9 and Fig. 2.10 part 2. a**, Overview of the gravel experiment, as seen from the GoPro Hero 4 camera that is used to track the robot. The full-step four-module configuration is placed on the gravel for size reference. **b-d**, All three conducted phototaxis experiments on the gravel surface, for the four-module dome configuration.

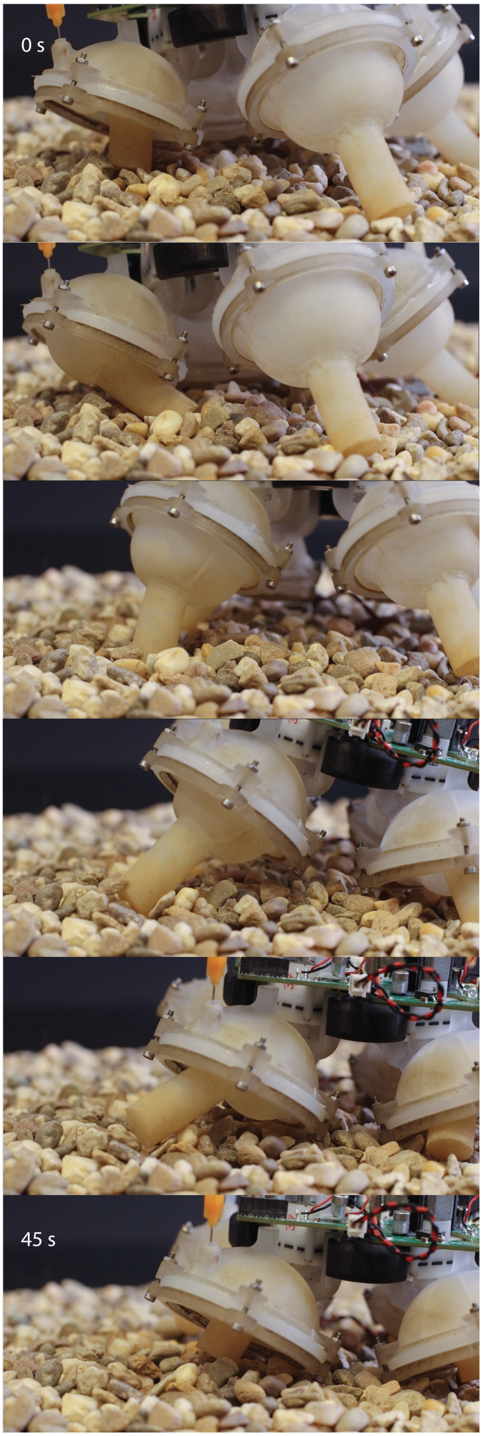


Fig. S2.9: **Additional information corresponding to Fig. 2.9 and Fig. 2.10 part 3. e**, A single actuation step (of the phases found during the experiment).

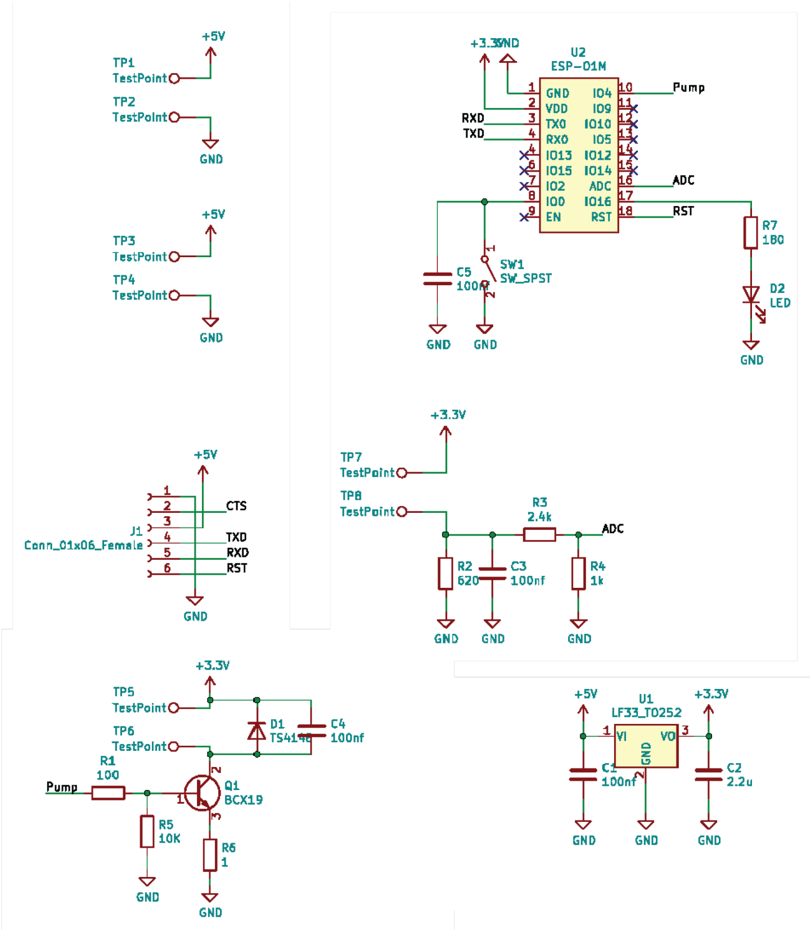
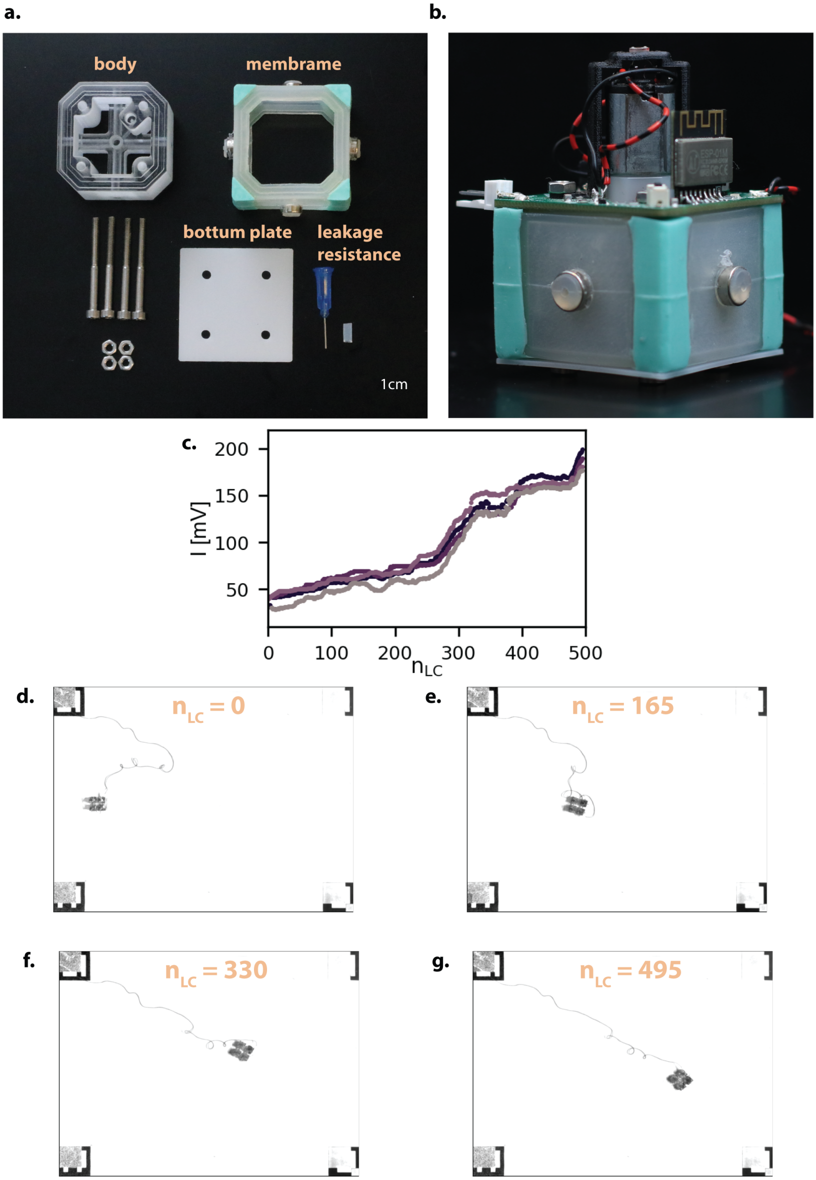


Fig. S2.10: **Electrical schematic diagram of the PCB design used throughout the Chapter.** The schematics show the ESP module that is used for processing the pump circuit and the circuit that establishes the power to the module and power distribution to the other modules.

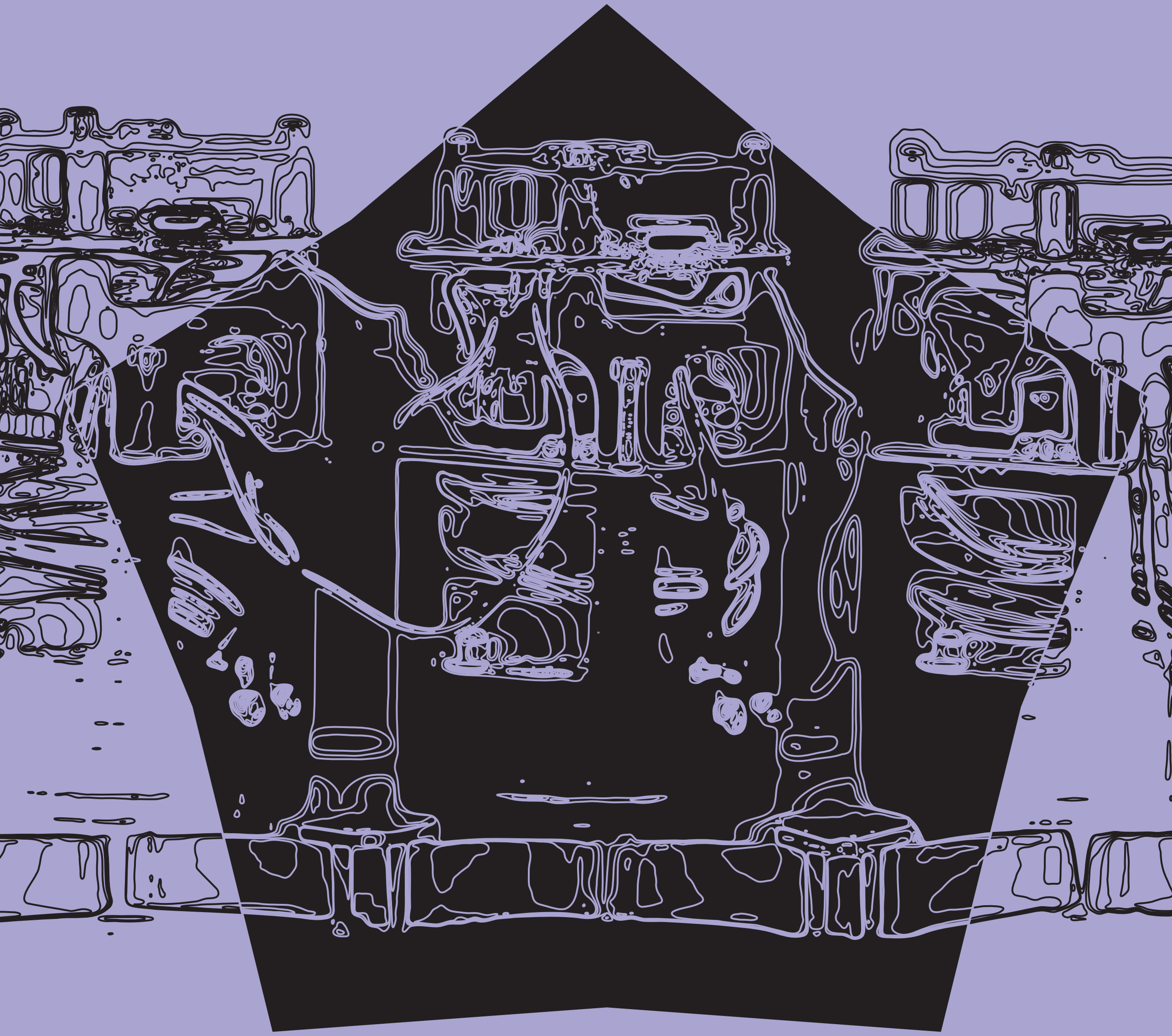


Caption on next page.

Fig. S2.11: **Phototaxis experiment with inflating cubes. a**, Components of the inflating cube modules. The body is 3D printed on a Stratasys with Veroclear. The body contains four channels that distribute the air from the pump to the four sides of the cube. The membrane is a two-step cast using DS20 (white) and Elite Double 32 (Turquoise). The difference in stiffness between the DS20 and Elite Double 32 causes the sides to expand while the corners remain in place, resulting in a large stroke of the connection points for relatively small volume change. For this reason, the current system can have a faster inflation-deflation cycle of 4 s compared to the 45 s of the PneuNet and Full-step actuators used in this study. Four magnets are placed into the sides of the module during casting, to allow easy connection to the other modules. The bottom plate is laser-cut from POM and holds the membrane in place from the bottom side of the body, ensuring an airtight seal. On the top, the PCB has the same sealing effect. The flow resistor (needle with gauge 20) has the same purpose as for the other module designs. **b**, Assembled module. Note that the module still uses the same PCB, light sensor, and pump compared to the other modules with PneuNet and Full-step actuators, only its body has changed. **c**, Light measurements for an experiment with a four-module configuration. **d-g**, Snapshots of the experiment with the four-module configuration that correspond to the data in **c**.

ROBUST PHOTOTAXIS BY HARNESSING IMPLICIT COMMUNICATION IN MODULAR SOFT ROBOTIC SYSTEMS

3.



Abstract.
In robotics, achieving adaptivity in complex environments is challenging. Traditional robotic systems use stiff materials and computationally expensive centralized controllers, while nature often favors soft materials and embodied intelligence. Inspired by nature's distributed intelligence, this study explores a decentralized approach for robust behavior in soft robotic systems without knowledge of their shape or environment. We demonstrate that only a few basic rules implemented in identical modules that shape the soft robotic system can enable whole-body phototaxis, navigating on a surface towards a light source, without explicit communication between modules or prior system knowledge. Our results reveal the method's effectiveness in generating robust and adaptive behavior in dynamic and challenging environments. Moreover, our approach's simplicity makes it possible to illustrate and understand the underlying mechanism of the observed behavior, paying particular attention to the geometry of the assembled system and the effect of learning parameters. Consequently, our findings offer insights into the development of adaptive, autonomous robotic systems with minimal computational power, paving the way for robust and useful behavior in soft and microscale robots, as well as robotic matter, that operate in real-world environments.

H.A.H. Schomaker, S. Picella, A. Küng Garcia, L.C. van Laake, J.T.B. Overvelde. "Robust Phototaxis by Harnessing Implicit Communication in Modular Soft Robotic Systems." *Advanced Functional Materials*, 2024.

3.1 INTRODUCTION

Unlike living systems that are comprised primarily of soft tissues that adjust their shape during interactions with the environment, robotic systems generally use stiff materials [51]. As a result, these conventional robotic systems exhibit well-defined motion and excel in precision and speed within controlled environments. In contrast, natural systems have evolved a focus toward robustness to survive in our complex world. These natural systems have inspired researchers to create soft robots that deform when interacting with the environment, increasing the potential for improved performance in more complex and unstructured real-world environments [104].

Inspired by the many invertebrates that exist in nature (e.g., sea stars, octopuses, sea urchins), we aim to explore how distributing the robot's "brain" over the body could result in more robust behavior [46, 58, 88]. For example, sea urchins have hundreds of tube feet, spines, and pedicellaria with locally integrated perception and motor control. They can sense and process information independently while still leading to whole-body locomotion, and robust behavior emerges without a central brain and only a limited central nervous system [118]. There has been a growing effort to understand such distributed systems in biology, in part due to their potential applications in autonomous robotic systems [46, 58, 91]. In the field of robotics, distributed control is mostly studied in modular and swarm robotics, which utilizes synergy and redundancy to improve the system's adaptability, functionality, reliability, and robustness [74, 119, 134]. Importantly, these systems typically have reduced complexity of their controllers since behavior emerges from interactions between agents [37, 84]. These systems show that similar to out-of-equilibrium systems in nature, global behavior can emerge from non-reciprocal interactions [36, 120]. More application-driven robotics examples include adaptation to mechanical stimuli [97, 106], construction [5] and locomotion [68, 103, 107, 112].

Previous research has found inspiration in emergent natural systems like amoeboids [112] and collective migration phenomena in cell biology [68], to create more resource-efficient and adaptive systems (able to achieve complex tasks with minimal computational resources) [98]. These systems often rely on coupled oscillators to adapt their behavior based on external stimuli [55]. While this continuing line of research shows that autonomy and adaptivity can be achieved with minimal computational requirements, it remains intrinsically dependent on sensitive parameters that have to be a priori optimized (e.g., through evolutionary algorithms) to produce their desired emergent properties. Instead, we focus on embodying a basic and stochastic learning algorithm throughout the robot that allows it to adjust to its environment and situation within limits specified by the robot's design and functional capabilities. This embodied computation is realized with active and identical modules constituting a single robot body.

Importantly, rather than optimizing the physical system for the specific task, we focus on a physical platform with minimal functionalities in order to understand to global behavior that arises from the local interaction. Therefore, we focus on the general principles involved in decentralized whole-body phototaxis in the hope of making the outcomes more generally applicable to other physical platforms.

To constrain the information in the system to be strictly local, we exploit a form of implicit communication facilitated by sensing changes to a shared environment, similar to the implicit communication of stigmergy in nature [9, 13, 132]. Our approach demonstrates that a decentralized system can optimize its behavior without explicit communication between its modules in various situations and environments.

Our previous research on a distributed stochastic learning algorithm showed great promise for emergent decentralized locomotion control in a one-dimensional framework [88] (the robot was constrained to a circular track), illustrating the value of a short-term memory algorithm in changing environments, as well as demonstrating the system's ability to resist damage [88]. Here, we move from a one-dimensional to a two-dimensional domain (the robot can freely move on a surface), which considerably increases the system and task complexity, e.g., due to competition. Importantly, even though we use exactly the same algorithm as in previous work, the fact that rotation is not directly controlled (only indirectly through competition) results in a dynamic learning process that is more complicated than in one dimension. From a physics perspective, transitioning from a one-dimensional to a two-dimensional transforms the state space from fixed to dynamic. In a one-dimensional setting, the phase space is static, and the optimization problem is relatively straightforward because the system's orientation does not change. Conversely, in a two-dimensional environment, the optimization problem becomes dynamically evolving. The system's orientation, and consequently its interaction with the environment, changes over time. This introduction of rotational dynamics that individual units cannot directly control enables a much richer set of behaviors and emergent phenomena that were not possible in the one-dimensional case. In this novel setting with increased complexity, we focus on three fundamental questions: *(i)* Does the modular robot exhibit robust learning behavior? *(ii)* How general is the implemented distributed stochastic approach, and how does geometry or configuration impact the system's ability to perform a task? *(iii)* How adaptable is the system? Can it robustly adapt to the environment and damage?

3.2 EXPERIMENTAL SETUP

We start by introducing the modular robotic platform that we developed (Fig. 3.1a-b) to try to answer these questions. Our robotic platform consists of physically connected modules that can move on a plane and can be assembled in square lattice configurations. Each robotic module in the lattice is identical and has its own microprocessor to control a stepper motor, measure the light intensity, and run a stochastic algorithm to change its behavior. Importantly, we aim to employ a stochastic algorithm in each unit such that the system as a whole performs phototaxis (i.e., moves to a light source). An individual unit is programmed to periodically expand and contract its four connection points every 2 s (Fig. 3.1a), where the only parameter that each module can tune is the phase of actuation ϕ_i, corresponding to the phase of unit i. Note that an individual unit cannot displace itself if it is not connected to other units. By physically connecting multiple units (Fig. 3.1c) by their soft elastomeric arms, the system as a whole starts to move and rotate depending on the phase differences between the units (Movie S3.1). As each unit can only sense the light intensity from its own sensor, it is unaware of what the other connected units are doing. Each unit can only change ϕ_i and operates on this one-dimensional search space. However, the corresponding collective behavior changes when the connected units change their phase or when changes occur in the

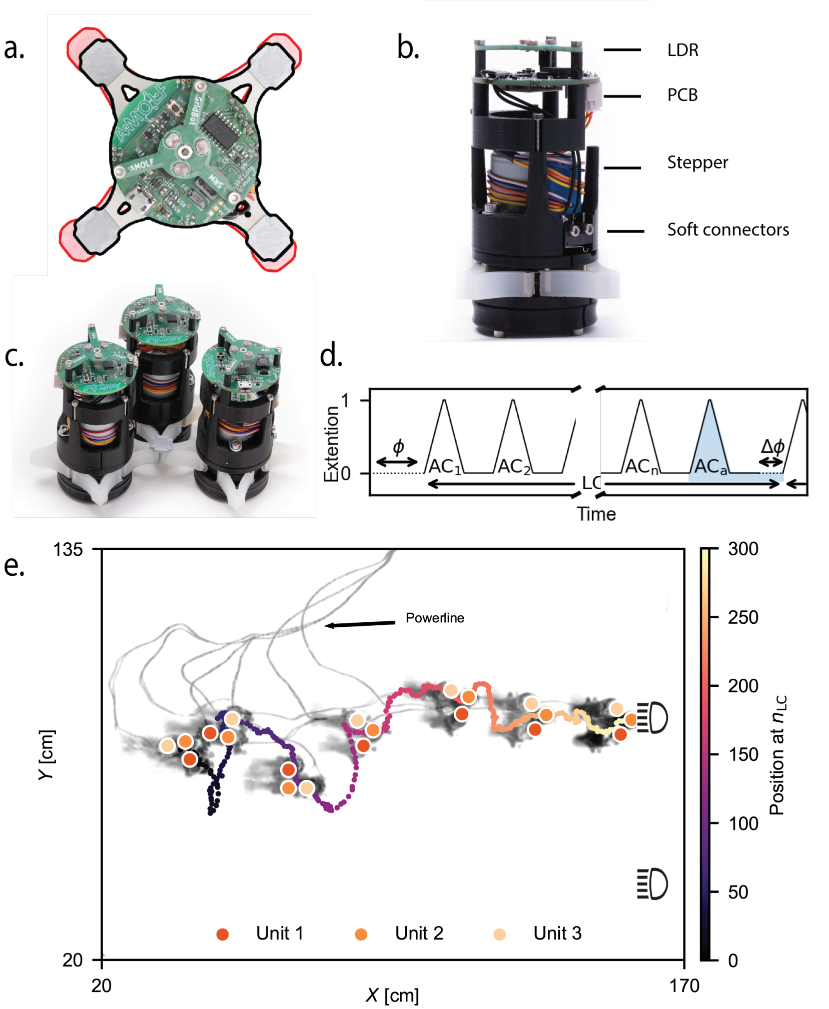


Fig. 3.1: **Robotic unit design and learning experiments for an assembled system. a**, Realization of a unit and its corresponding extended (red outline) and contracted state (black outline). **b**, Side view of a robotic unit. **c**, Robot assembled from three units. **d**, Schematic representation of a learning cycle LC that occurs in each unit. **e**, Evolution of a single learning experiment with a three-unit system (**d**) during $N_{LC} = 300$, given by overlapping images of the experiments at every 50 n_{LC}. The colored line indicates the position of the center of mass at every learning step n_{LC}. For this experiment, the robot starts on the left, and a planar light source is placed on the right.

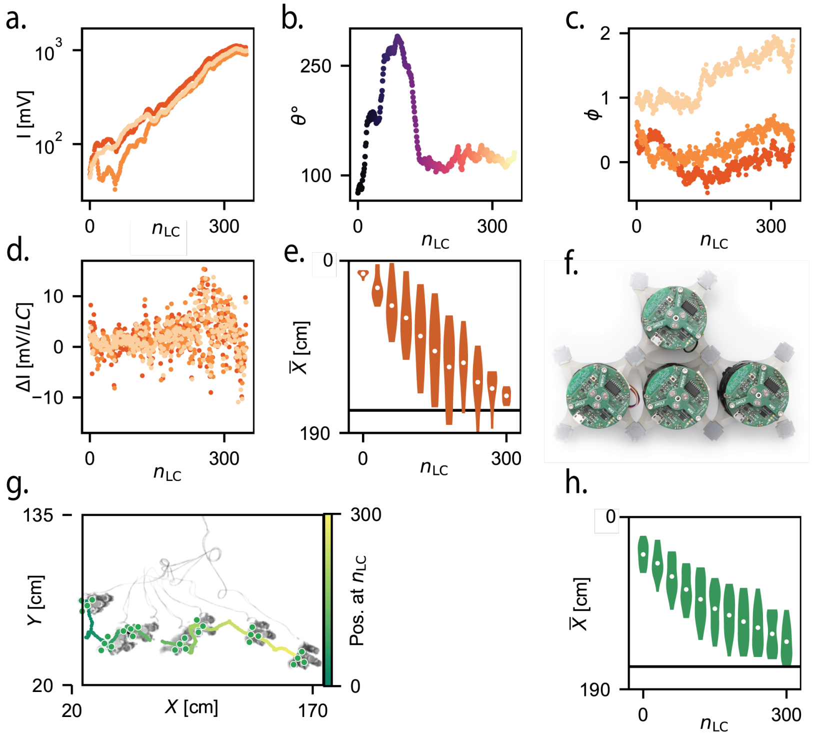


Fig. 3.2: **Learning experiments for an assembled system in two different configurations.** **a**, The measured light intensity I in each unit. **b**, Orientation in degrees (θ°) of the three-unit system during the single experiment. **c**, Absolute phase ϕ_i for every learning cycle. **d**, The measured difference in light intensity ΔI. **e**, Distribution of the average X positions of the assembled robot for ten experiments with the three-unit configuration. **f**, Robot assembled in a configuration with four units. **g**, Evolution of a single learning experiment with the four-unit system during 300 learning cycles, under similar experimental conditions as in the three-unit experiment (Fig. 3.1e). **h**, Distribution of the X positions for ten experiments with the four-unit configuration.

environment. Due to the change in collective behavior, each unit must continuously re-evaluate its performance in its one-dimensional search space. In a way, these units, therefore, can also change the environment for the other units in the assembly.

We implement the same stochastic learning algorithm that we previously studied in one-dimensional experiments, where the robotic system was constrained to a circular track to limit its movement to a single dimension [88]. In this learning algorithm, each unit performs random experiments by perturbing its phase and only conditionally accepts these phases by comparing the difference in light intensity with its previous step. Specifically, units individually undergo learning cycles (n_{LC}) as shown in Fig. 3.1d, each consisting of a total of N_{AC} regular actuation cycles (AC) with a constant phase. Each actuation cycle has a duration of $t_{AC} = 2\,\mathrm{s}$ and consists of an extension of the soft connector arms with a duration of $0.6\,\mathrm{s}$, followed by contraction of the arms with the same duration, where it remains in its contracted state for the remainder of the actuation cycle. At the end of every learning cycle, the unit perturbs its phase ϕ_i to explore a new phase ϕ_i'. The perturbation is performed according to $\phi_i' = \phi_i^m + \epsilon\Delta S$. Here, ϕ_i' is the phase that is evaluated the next n_{AC}, ϕ_i^m is the previous phase kept in memory, ϵ represents a uniformly distributed random variable between the interval $[-1, 1]$, and the parameter $\Delta S = 0.1$ describes the maximum difference between two phases of consecutive learning cycles.

To implement the phase change, each learning cycle is followed by a phase adjustment actuation cycle (AC_a). Only the last part of the cycle is altered during the phase adjustment cycle as the unit implements a new phase for the next step. The amount of adjustment conditionally depends on the acceptance or rejection of the current phase. If the current phase is accepted, phase difference $\Delta\phi_i$ between the current and the next learning cycle is given by ΔS. In case of rejection, we perturbed the phase in memory to find a new phase for the next learning cycle, resulting in a $\Delta\phi_i$ that is a combination of the difference between these two phases and new perturbation ΔS for the next learning cycle. This results in a new $\phi_i^{'}$ by extending or shortening t_{cycle} according to:

$$\Delta\phi_i = \begin{cases} \phi_i^m - \phi_i^{'} + \epsilon\Delta S, & \text{for } \Delta I_i^{'} < \Delta I_i, \\ \epsilon\Delta S, \phi_i = \phi_i^{'} & \text{for } \Delta I_i^{'} \geqslant \Delta I_i. \end{cases} \tag{3.1}$$

Here, each unit approximates the quality of its current $\phi_i^{'}$ by evaluating the difference $\Delta I_i^{'}$ between two independent light measurements $I_{1,i}$ and $I_{2,i}$ measured during AC_1 and AC_4, respectively. Therefore, the units evaluate the rate of change in light intensity rather than only looking at the absolute light intensity. The units accept the current phase $\phi_i^{'}$ if the difference in light intensity $\Delta I_i^{'}$ is larger then ΔI_i stored in memory. Finally, in previous work, we established that to achieve adaptive behavior ΔI_i should change to $\Delta I_i^{'}$ after every learning cycle [88]. Note that every unit independently changes its ϕ_i. Therefore, the units are not synchronized, resulting in asynchronous timing between the units for measurements and phase adjustment. More details on the learning are provided in the method section (Fig. S1).

3.3 PHOTOTAXIS IN EXPERIMENTS

To determine if a system of coupled units can perform phototaxis while moving on a surface, we first conduct experiments with the smallest two-dimensional configuration (i.e., with units that do not only lie on a line) as shown in Fig. 3.1c. We place the assembled system on the left side of a flat rectangular surface with two LED panels on the right side (Fig. 3.1e) and observe the system's behavior during $N_{LC} = 300$ learning cycles (Movie S3.2). Even though the units operate entirely independently and asynchronously (the only electronic connection is the power supply), Fig. 3.1e shows how the assembled units rotate and move while getting closer to the light source on the right. This is clearly demonstrated in Fig. 3.2a, where we show that the average light intensity I increases for all units during the experiment.

Looking closer at the behavior of the assembly in Fig. 3.1e, we can roughly distinguish four regimes. *(i)* During approximately the first $n_{LC} = 50$ learning cycles, the measured ΔI does not seem to increase for all units (Fig. 3.2a). Yet, we do observe a change of orientation $\Theta°$ of the assembled system. This orientation is represented in degrees and measured with respect to the experimental canvas (Fig. 3.2b). *(ii)* Between approximately $50 \leqslant n_{LC} \leqslant 150$ learning cycles, the units start to move towards the light source while also undergoing significant rotation. *(iii)* We find a straight trajectory for the last $n_{LC} = 50$ learning cycles, along with a stabilization of the orientation of the system (Fig. 3.2b). While the phases of the individual units still vary over time, the phase differences between the units stabilize (Fig. 3.2c). It appears that the assembly has reached stable behavior. *(iv)* Around $n_{LC} = 300$, the units reach a peak of their change in light intensity ΔI as presented in (Fig. 3.2d). This indicates that they arrived at the position of highest light intensity, as also revealed by the drop in ΔI shown in Fig. 3.2a. In general, Fig. 3.2a-d demonstrates the emergence of robust global phototaxis of the system. To investigate the robustness of the observed phototaxic behavior, we repeat the experiment of Fig. 3.2a-d ten times, where we start from initially random phases and random orientations. Fig. 3.2e shows the position distribution for all experiments for progressing learning cycles. Note that in this figure, the position of the light source is at $x = 165$, and a black line indicates the place with the highest light intensity I as measured in separate experiments (Fig. S3.2c). Fig. 3.2e demonstrates that the distributions converge to the location with the highest light intensity, thereby indicating that all assemblies move towards the point of highest light intensity and thus achieve robust phototaxis.

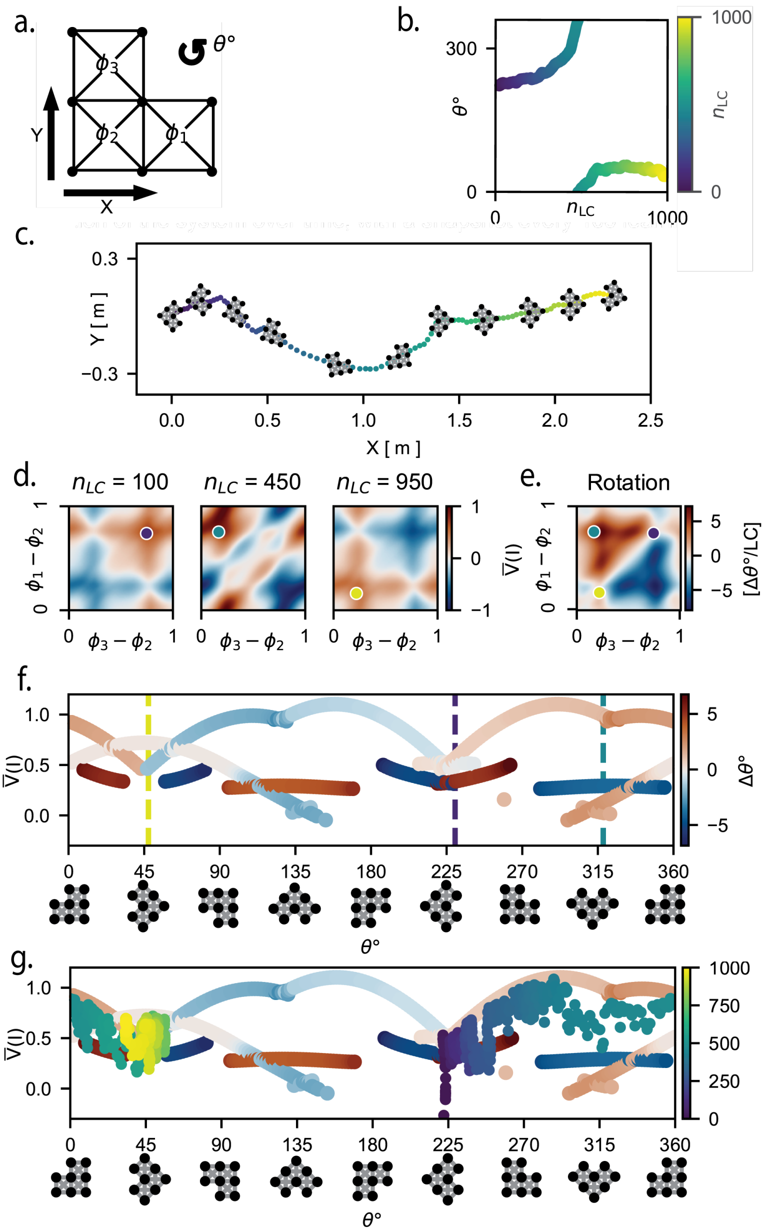
Next, to explore if phototaxis is also achievable for a robot with a different geometry, we assemble a robot by adding an additional unit (Fig. 3.2f). We place the assembly on the left side of the test setup to repeat the same experiment as before and observe the first $N_{LC} = 300$ learning cycles in Fig. 3.2g. Similar to the three-unit system, we find robust directional motion towards the light source, with considerable rotation during the first part of the experiment and convergence to stable behavior for the second part.

Subsequently, we repeat the experiment 10 times (Fig. 3.2h) with random initial phases. Similarly, we find a displacement towards the light source for all ten experiments. Interestingly, we find slower average velocities for the four-unit system compared to the three-unit system and a smaller variance between experiments. These results emphasize the importance of previously raised questions on how general the implemented distributed stochastic approach is and what the impact is of the geometry or geometry on the system's ability to perform a task.

3.4 UNDERSTANDING THE OBSERVED BEHAVIOR

Before answering how geometry affects the ability to perform phototaxis, we first return to the first question raised in the introduction and try to better understand how the system can achieve its behavior without explicit communication between modules. To get a better visualization and understanding of the underlying dynamic behavior from both the perspective of the individual units and the assembled system, we implement a coarse-grained mass-spring model that qualitatively captures the system's behavior.

To reduce the computational requirement for the model, we build single units using four masses and four springs, as shown in Fig. 3.3a. Two additional diagonal springs actively drive the extension and contraction of the unit by varying their rest length periodically in time. The behavior over time is found by numerical integration of the equations of motion using RK45. Within the numerical integration, we perform a discrete event model to capture the decentralized nature of the learning behavior in experiments. A block diagram of the event model can be found in Fig. S1. Further information on the model can be found in the method section. Fig. 3.3a presents the model representation of the three-unit configuration as depicted in Fig. 3.1c. Similar to the experiments performed in Fig. 3.1, we place a planar light source on the right side of the two-dimensional plane and model the light source as a two-dimensional scalar light intensity field, where the intensity has an inverse square relation with the distance to the light source $I \propto \frac{1}{d^2}$. Simulating the three-unit assembly, we find that the system starts to move in the direction of the light within $n_{LC} \approx 10$ learning cycles (Fig. 3.3b-c). In agreement with the experiments, the assembled robot initially behaves in a regime dominated by large rotations that occur while learning to move toward the light, followed by a more consistent orientation and movement.



Caption on next page.

Fig. 3.3: **Course-grained mass-spring model to numerically study the qualitative behavior of the system. a**, Mass-spring representation of a three-unit system as depicted in Fig. 3.1c, with the active springs in the diagonal of the squares and passive springs around the edges. **b**, Orientation over n_{LC} for a single simulation of the three-unit system for a thousand learning cycles. **c**, Trajectory of the same simulation, with a planar light source placed on the right side of the two-dimensional plane. The system's state is depicted every hundred n_{LC}. **d**, Evolution of the potential velocity mapped to the phase space from a system perspective measured in $\overline{V}(I)$ [mm/LC] at three-time instances during the simulation of **c**. The color map indicates the velocity in the direction of the light for all possible phase combinations of the system. The colored dots indicate the phase combination during the simulation of **c** colored by n_{LC}. **e**, The rotation ($\Delta\theta$) of the system for all phase combinations. The dots indicate the phases at the n_{LC} snapshots of **d**. **f**, a projection of the local optima of the $\overline{V}(I)$ [mm/LC] in the phase space for all system orientations. Obtained by rotating the reference frame of Fig. S3a and numerically extracting the local optima for each orientation. The dashed vertical lines indicate the slices as represented in **d**. The color of the optima indicates the change in angle $\Delta\theta$ for the phases that correspond to the local optima as can be found in **e**. **g**, The same projection as represented in **f** with the optimal $\overline{V}(I)$ [mm/LC] for each orientation represented in an increased size compared to the local optima. The trajectory of the single simulation (**c**) is mapped onto this projection to visualize the long-term phototaxic behavior.

System Perspective

To find an accurate representation of how the system with the learning algorithm performs phototaxis, we evaluate the system displacement towards the light for all possible phase combinations as presented in Fig. S3a. Now, a change of reference frame is applied to obtain the search space for a specific system orientation during the simulation. Fig. 3.3d presents this rotated search space for three different snapshots of the simulation presented in Fig. 3.3b-c, at different learning cycles. Furthermore, the respective rotations of these three phase combinations are represented in Fig. 3.3e. We can make four main observations from the perspective of this system. First, the search space drastically changes due to the system's rotation, meaning it must continuously reevaluate its behavior. Second, we see that the maximal achievable displacement towards the light changes over time due to the rotation of the system, from $\overline{V} \approx 1.0$ [mm/cycle] at 450 n_{LC} to $\overline{V} \approx 0.6$ [mm/cycle] at 950 n_{LC}. Third, the assembly finds the area close to optimal phototaxis in the phase space, demonstrated by these three orientations. Note that due to the stochastic nature of the optimization, the behavior inherently fluctuates rather than fixates on the optimal position in the phase space. Fourth, depending on the phases, the assembly unintentionally (without being observed by the light intensity sensor) changes its orientation. These changes in the search space indicate that the assembly can not sustain a specific, and maybe faster, behavior over extended periods due to the inevitable rotation that coincides with the displacement towards the light source. These results raise the question of whether the system can stabilize its behavior over a longer time scale and what the effect of the stabilization is on the phototaxic behavior.

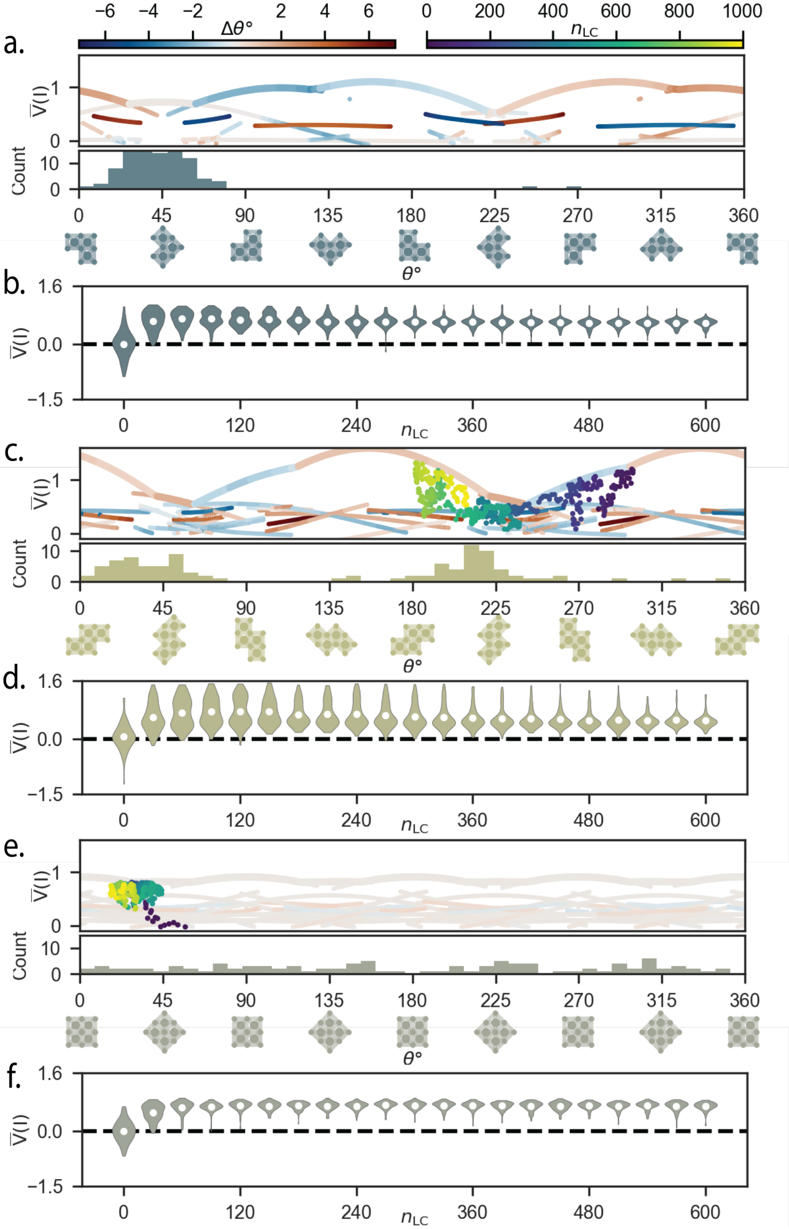
Orientation Perspective

Of course, as each unit can only sense a single light intensity and there is no inter-unit communication, neither the system nor the units have any notion of their orientation. However, Fig. 3.3d reveals that the maximum speed heavily depends on the system's orientation. In fact, the system stabilizes at an orientation for which the maximum velocity in the direction of the light is lower than what this specific assembly can potentially achieve at different orientations (Fig. 3.3d). This difference in maximum directional velocity for different orientations can be more clearly seen by looking at the fastest velocities for any given orientation, as shown in Fig. 3.3f. This figure depicts the amplitude of phototaxis (displacement towards the light) for all the local optima in each orientation, obtained by rotating the global reference frame of Fig. S3a and numerically finding the local optima for each orientation. The color indicates how much the system rotates during a learning cycle. A more detailed view of how this projection is created can be found in Movie S3.3 Note that Fig. 3.3f solely resembles the potential (locally) optimal displacement of the assembly without considering any behavior related to the learning, but it helps us to visualize the potential optimal behavior the system could exhibit. From this perspective, we find a reflective symmetry between 45 ° and 225 ° and the existence of only two local optima for which the assembly doesn't rotate with peaks at 45 ° and 225 °.

To better understand how the system moves through this three-dimensional search space (consisting of two $\Delta\phi$ and the θ Fig. 3.3d-e) and how close the behavior is to a (local) optimum, we project the displacement and orientation of the single simulation of the three-unit system into Fig. 3.3g. We observe that the assembly seems to find the general areas of optimal phototaxis and tracks them over time as its orientation changes. Stability occurs when the units reach an orientation for which no rotation occurs, in this case, at approximately 45 °. Interestingly, the assembly spends most of its learning cycles on the two optima with minimal rotation at the respective 45 ° and 225 ° angles. Moreover, the assembly seems to move away from one to the other and stabilize its orientation around 45 °. These results could indicate stable orientations that emerge on longer timescales, dominated by the geometry of the assembly.

Long-term stability

With this improved understanding of the system's behavior from different perspectives, we explore the dynamic stability of the system over longer timescales. We repeat the simulation performed in Fig. 3.3g a hundred times for different initial conditions and starting orientations of the assembly. In Fig. 3.4a we present the distribution of the orientations of the assembly after at $n_{LC} = 1000$. In line with the results presented in Fig. 3.3g, almost all simulations converge towards 45 ° (with only two outliers around 250 °). Therefore, we conclude that the 45 ° angle serves as an "attractor" for the three unit configurations. In contrast, the 225 ° angle is an "unstable fixed-point" for the long-term behavior. Although stable in the short term, the stochastic nature of the algorithm will eventually push the system away from the 225 ° angle.



Caption on next page.

Fig. 3.4: **Effect of geometry on the long-term behavior of the system.** Orientation perspective for a (**a**) three-unit robot and two four-unit robots with (**c**) skew and (**e**) square shape. The distribution underneath the orientation perspective indicates the final orientation after a thousand N_{LC} for a hundred simulations. The corresponding evolution of the velocity $\overline{V}(I)$ [mm/LC] for the first 600 learning cycles are shown in **b**, **d** and **f**.

To further investigate the influence of these long-term dynamics on phototaxic behavior, in Fig. 3.4b, we show a distribution of the average displacement of the system for all hundred simulations while moving towards the light source. Considering the analysis from Fig. 3.3, we can identify the same three distinct regimes in the assembly's behavior as we have observed in experiments in Fig. 3.1, but with additional insights. Firstly, for approximately $0 \leqslant n_{LC} \leqslant 50$ learning cycles, the system quickly changes phases to improve its velocity in the direction of the light ($\overline{V}(I)$). Next, between approximately $50 \leqslant n_{LC} \leqslant 250$, the system reaches a peak in $\overline{V}(I)$ due to changing orientation, where higher velocities are still possible as the system is still changing its orientation. Note that although these orientations allow for phase combinations with a high $\overline{V}(I)$, these phases also result in large rotations $\Delta\theta$ and are therefore unstable and can't be maintained for longer time periods. Finally, for $n_{LC} > 250$, the variance of the distribution becomes smaller, and the $\overline{V}(I)$ converge across all simulations. This convergence occurs because more simulations reach the $45\,^{\circ}$ angle, approaching the same stable orientation and phase combination.

Unit Perspective

It is important to consider that so far, we have visualized the potential behavior from the assembled system perspective. Yet, the individual units do not directly communicate. To understand the dynamics from the perspective of the individual units, we use the mapped search space as shown in Fig. S3.3a of the three-unit system to visualize the one-dimensional search space of every unit over time in Fig. S3.4. Each unit operates in a different one-dimensional search space, depending on its position in the assembly. Note that the units do not see the

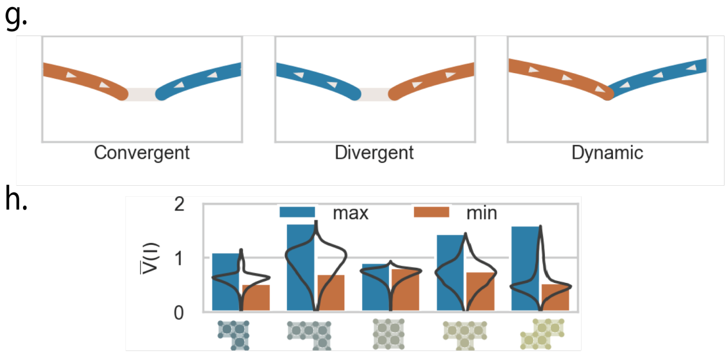


Fig. 3.5: **Overview of the effect of geometry on the long-term behavior of the system. a**, Schematic representation of the different types of local optima in the orientation projection. **b**, Overview of the simulation results obtained for all robot shapes up to four units. The distribution (black line) of the average velocity ($\overline{V}(I)$ [mm/LC]) for 100 simulations at the 600th learning cycle, compared to the maximum and minimum velocity of the global optimal behavior.

full search space of the whole system and can only probe the local one-dimensional search space by varying their phase. Importantly, the search space's mapping from phase to behavior fluctuates over time due to the rotation of the assembly and the continuously changing behavior of the surrounding units. Fig. S3.4 demonstrates how all three units can adapt their phase ϕ_i to maintain the desired phototaxis behavior of the assembly. Surprisingly, the individual units stay close to the optimal behavior in their constantly changing search space.

3.5 Effect of geometry on the long-term behavior of the system

With this more explicit understanding of the long-term dynamics for a three-unit configuration, we now focus on getting a better idea of how the system's geometry influences long-term behavior. We, therefore, perform the same analysis for two distinct four-unit configurations. In Fig. 3.4c-f, we show the learning behavior of the two four-unit configurations (skew- and square-shape) that, together with the three-unit system, can describe the main similarities and differences between geometries. First, Fig. 3.4c describes the trajectory and global optima of the skew-shape configuration with one-fold rotational symmetry. Due to the one-fold rotational symmetry of the configuration, we find a repeating pattern in the global optima projection. If we follow the optima that cause counter-clockwise rotation (in blue) to the left and the optima that cause clockwise rotation (in red) to the right, we find two intersections at approximately 50° and 230°. These intersections lead to the preferential orientation of the system in these two orientations, as seen from the distribution in Fig. 3.4c. However, where the three-unit configuration has one stable optimum without rotation that all simulations converge towards, this four-unit geometry constantly transitions from rotation in one direction to the other, as there is no global optimum with zero rotation. This differs from the "convergent" behavior, as the phases must be "dynamically" adjusted to maintain phototaxis. This could also lead to a wider distribution of velocities, as observed when running multiple simulations (Fig. 3.4d).

In contrast to the skew-shape that does not have stable convergent orientations, the optimal behavior of the square-shape is characterized by relatively little rotation (Fig. 3.4e). The combination between this observation and the four-fold symmetry of the unit leads to a nearly flat projection of the local optima. Apparently, for any given orientation of the square geometry, an optimal phase combination exists that leads to a relatively stable and fast phototaxis. As a result, the square-shape also doesn't exhibit the initial overshoot in velocity that is the result of stabilization of orientation (Fig. 3.4f) that is observed for both the three-unit (Fig. 3.4b) and skew-shape (Fig. 3.4d).

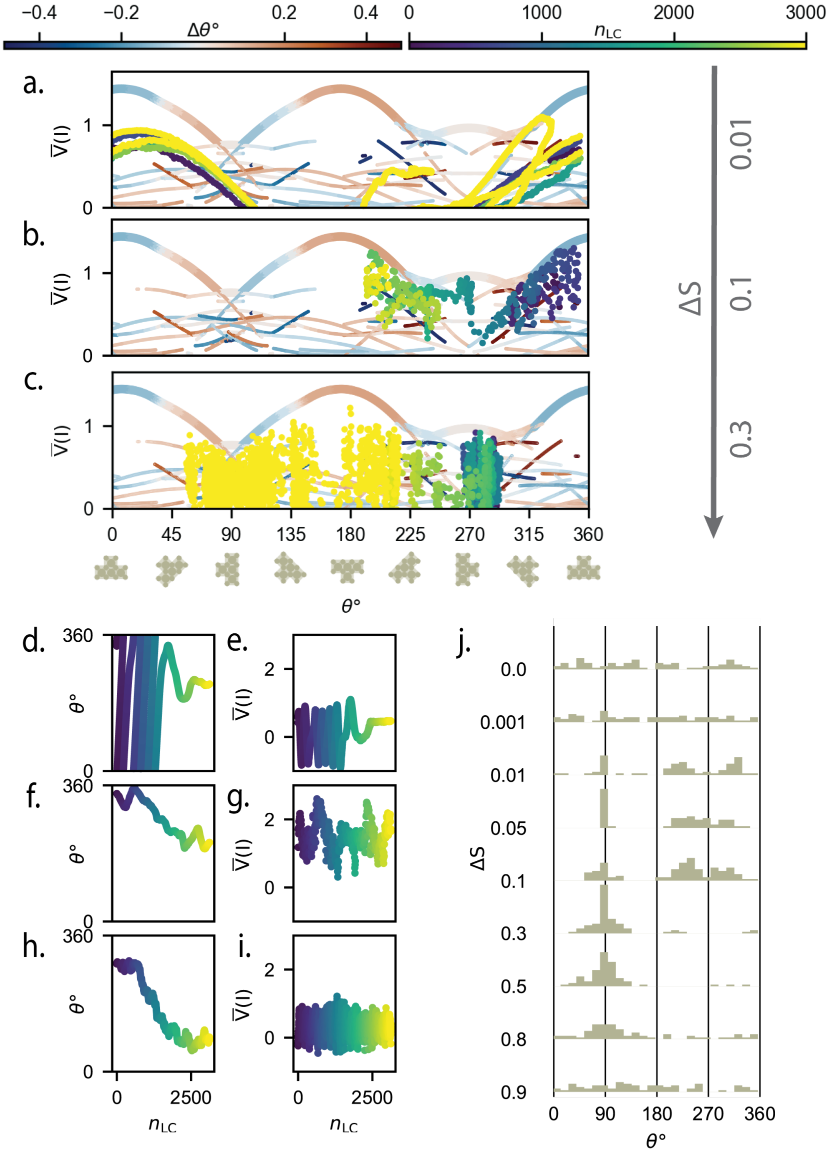
In summary, from the results in Fig. 3.4, we find three characteristic transitions between global optima that govern the long-term behavior of the system. We define the global optimum from the perspective of the system as the set of phases that result in the highest $\overline{V}(I)$ at a fixed orientation (as the system has no notion of orientation) (Fig. 3.5a). First is the convergent behavior, where the geometry drives the system towards a global optimum without rotation, thereby stabilizing the behavior (i.e., no phase changes are required to maintain phototaxis). Secondly, a divergent transition is presented in Fig. 3.4a. Although the system can maintain this orientation for extended periods, the stochastic nature of the control will eventually drive the system out of this orientation. Thirdly, a dynamic behavior, as presented in Fig. 3.4c, leads to persistent orientation fluctuations (i.e., constant phase changes are required to maintain phototaxis). Different geometries can yield combinations of the optima transitions mentioned above.

Fig. 3.5b provides an overview of the difference between the maximum $\overline{V}(I)$ in the fastest and slowest orientation for all three and four unit configurations, along with the $\overline{V}(I)$ distribution at $n_{LC} = 1000$. From this overview, we can conclude that the phototaxic potential of the system heavily depends on its geometry. This difference can also be observed in their trajectories presented in Fig. S3.5. Furthermore, we find that the system's $\overline{V}(I)$ distributions, in general, do not stabilize around their maximum $\overline{V}(I)$ because these orientations are accompanied with rotations of the system.

3.6 EFFECT OF LEARNING PARAMETERS ON THE LONG-TERM DYNAMICS OF THE SYSTEM

Previous simulations on multiple shapes teach us that shape has an important role in the behavior of the assemblies and that the shape directly influences the short- and long-term behavior the system can exhibit. However, the learning parameters can also affect the stability of the system's long-term behavior. In this section, we will further explore the effect that the learning step ΔS has on the potential to perform phototaxis.

The current minimal approach to learning is designed with generality in mind and is dominated by only two parameters, ΔS and the number of actuation cycles (N_{AC}). Both relate to the rate at which the system can adapt its behavior to the changing environment. On the one hand, the number of actuation cycles dominates the signal-to-noise ratio of the light measure by controlling the number of actuation cycles between the first and second light measurements. However, it also determines the amount of rotation between consecutive learning cycles, as the rotation scales linearly with the number of actuation cycles. On the other hand, the amplitude of the phase change between consecutive learning cycles, ΔS, directly controls the algorithm's adaptability (i.e., stochasticity). For this parameter, we predict that there is a direct trade-off between exploration and exploitation, given by high and low ΔS, respectively.



Caption on next page.

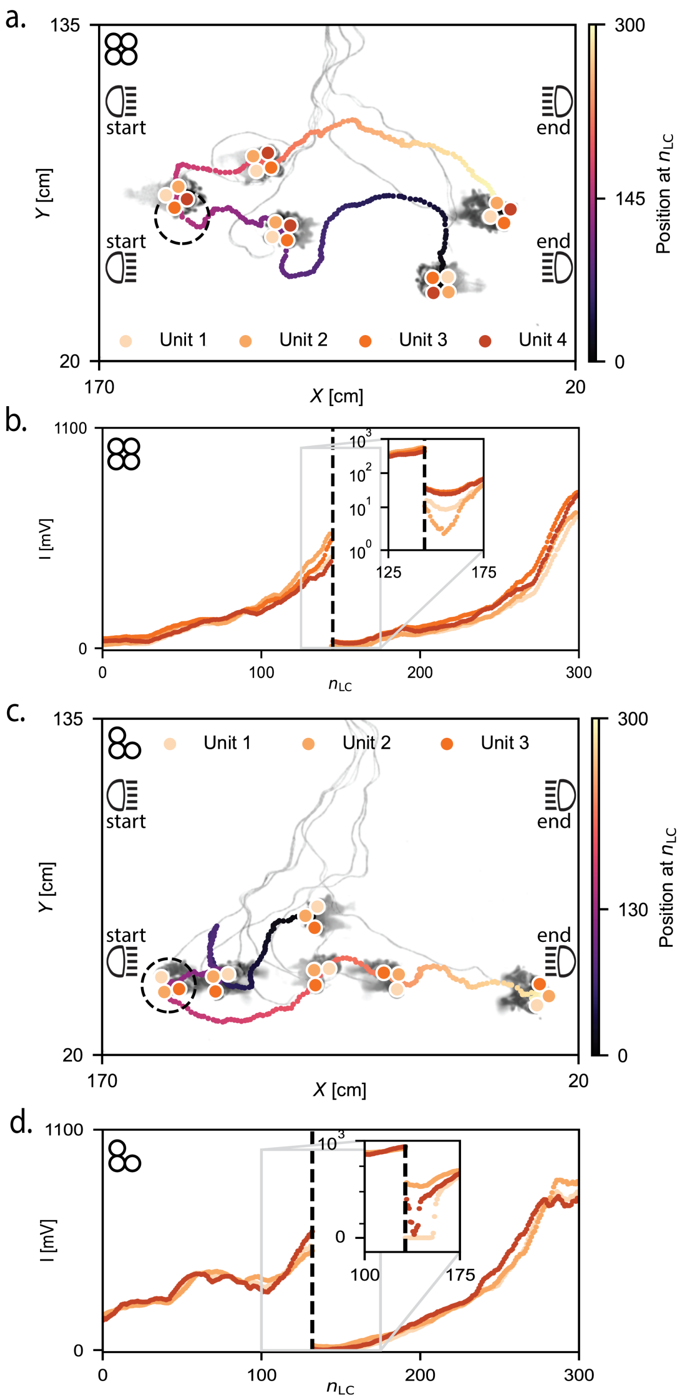
Fig. 3.6: **Effect of the learning step size on the behavior of the system** Evolution of the velocity $\overline{V}(I)$ [mm/LC] of a T-shape robot in the orientation perspective, for (**a**) $\Delta S = 0.01$, (**b**) $\Delta S = 0.1$ and (**c**) $\Delta S = 0.3$. The evolution of the robot's angle for each simulation is given in **d**, **f** and **h**. The evolution of the robot's average velocity $\overline{V}(I)$ [mm/LC] for each simulation is given in **e**, **g** and **i**. **j**, Distribution of the angle at the 1000th learning cycle, for hundred simulations and for a range of learning step sizes ΔS.

To demonstrate the influence that ΔS has on the system's behavior, Fig. 3.6a-i shows three simulations of a T-shape configuration with $\Delta S = 0.01$, 0.1, and 0.3. Having mapped the optima in Fig. 3.6a-c, we find one convergent stable orientation at 90 ° and two dynamically stable orientations at 220 ° and 320 °.

First, for the relatively small learning step $\Delta S = 0.01$ (Fig. 3.6a), we find that the system remains in the local optima and does not jump between local optima as the pay-off declines due to rotation. Fig. 3.6d shows that the assembly keeps rotating until settling in a convergent local optimum around 225 °. Apparently, the small values for the learning step suppress the stochastic exploratory nature of the algorithm. Second, in Fig. 3.6b we find the $\overline{V}$ for $\Delta S = 0.1$ to be more scattered, resulting in global behavior that jumps over the local optima to reach the global optima. As a result, we find less change in the orientation (Fig. 3.6f) and a global behavior that settles around one of the dynamic global optima with a similar orientation as Fig. 3.6a, but considerably higher $\overline{V}$ (Fig. 3.6g). Third, for the simulation with $\Delta S = 0.3$ in Fig. 3.6c we observe even more scattered behavior. With this value of ΔS, the system seems unable to follow any specific optima and rather moves around the entire phase space. This results in an inability to track the dynamic global optima. Yet, the systems still slowly drift towards the convergent behavior at 90 ° (Fig. 3.6h and i).

Looking at the long-term behavior of the T-shape configuration for different values of ΔS (Fig. 3.6j), we find that for $\Delta S = 0$, the end distributions are randomly spread out over all possible orientations. From $\Delta S > 0.01$, we start to see clear peaks in the end distribution, with centers around the convergent point at 90 ° and the two dynamic points around 220 ° and 320 °. Between $0.1 \leqslant \Delta S \leqslant 0.3$, we find the distribution peaks of the dynamic points disappear, resulting in only a single stable orientation at 90 °. Lastly, starting from a $\Delta S > 0.8$, we find the onset of the disappearance of any stable orientations as the system starts to approach a state where it always chooses new random phases. Although we observe phototaxis for all three ΔS, the rate at which phototaxis is achieved and the $\overline{V}(I)$ after the behavior converges are considerably different. Furthermore, we investigate the scalability of the system in lattice configurations with increasing size (Fig. S3.5a). Fig. S3.5b demonstrates that the ΔS significantly impacts the system effectiveness of the phototaxis for larger systems. We see that for both a $\Delta S = 0.1$ and a $\Delta S = 0.05$ the lattice configurations perform phototaxis in a system with up to at least 49 units. However, we do find that for the $\Delta S = 0.1$, the $\overline{V}(I)$ declines more quickly with increasing system sizes compared to the $\Delta S = 0.05$. When evaluating the evolution of the $\overline{V}(I)$ distributions for a lattice of 49 units (Fig. S3.6c), we find that the $\Delta S = 0.1$ learns more quickly. However, Fig. S3.6c also shows that around n_{LC} = 80 the smaller $\Delta S = 0.05$ starts to outperform $\Delta S = 0.1$. From n_{LC} = 100 onwards ΔS = 0.1 simulations stop improving while the smaller ΔS simulations keep on improving their $\overline{V}(I)$.

These results indicate that the phase space of the physical system becomes more complex as the number of units increases. Therefore, the larger ΔS simulations are less effective in exploiting the local optima than the smaller ΔS simulations. While we did not test configurations larger than 49 units due to computational and practical constraints, we have no evidence to suggest that phototaxis would not occur in larger systems, albeit becoming less effective. However, we acknowledge that the efficiency, as indicated by the speed of movement towards the light, may vary with system size and the learning ΔS parameter, which is an important consideration for scalability. Further research is needed to test how the current algorithm would work with an even larger system. Thus, we find the long-term behavior of our system not only emerges from its geometry but is also heavily dependent on the chosen learning parameter. Furthermore, we find that phototaxis remains feasible for a wide range of the ΔS parameter (Fig. S3.7), but that its value does influence the equilibrium behavior. These results further strengthen the claim of robustness in the current learning approach.

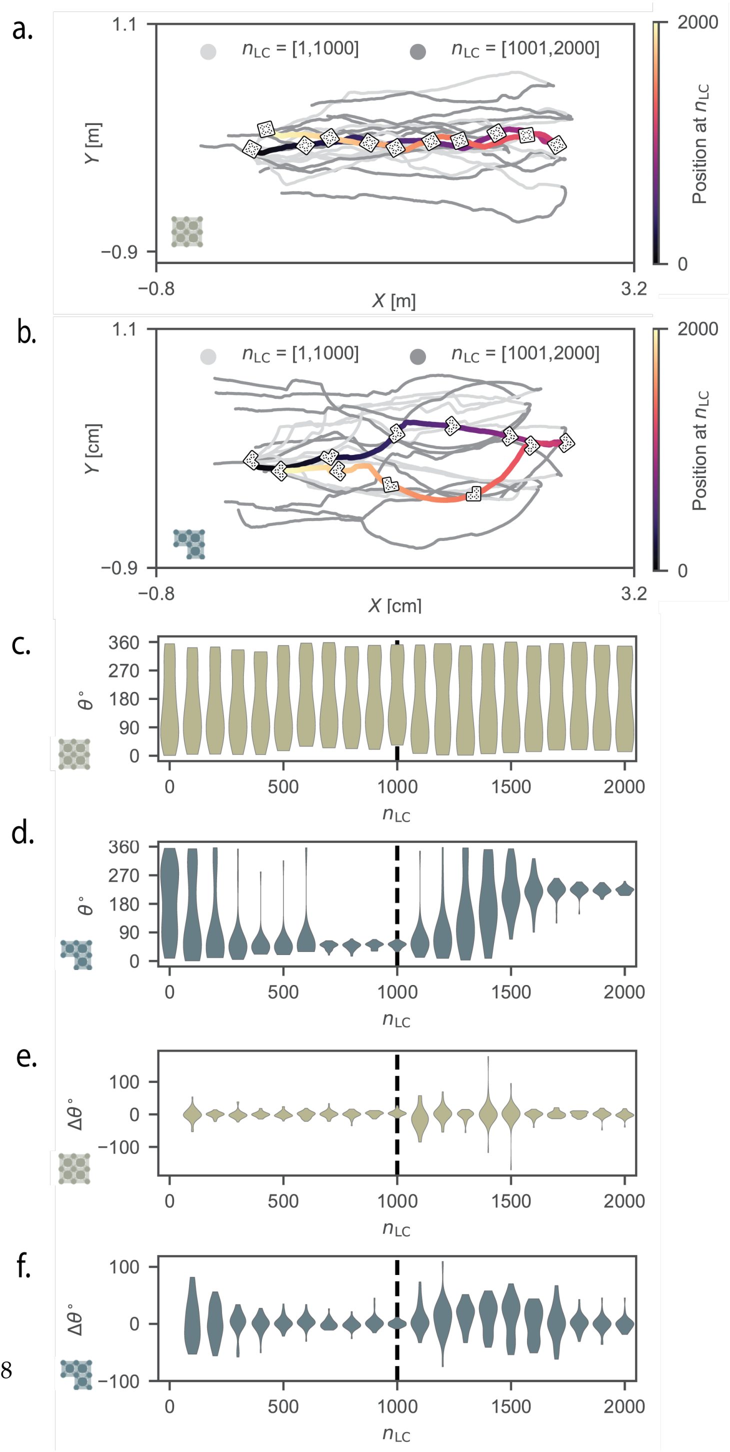


Caption on next page.

Fig. 3.7: **Response of an assembled robot to a sudden change in light direction in experiments.** Experiment of (**a**) a four-unit square robot with 350 N_{LC}. The light source is initially placed on the left side of the canvas and is switched to the right side of the canvas at the 145th learning cycle. The image consists of 6 overlapping photos of the experiment, each 50 N_{LC} apart. The dashed circle indicates the position of the robot when the light source is switched from left to right. The light intensity measured by the individual units is shown in **b**. **c**, present the same experiment as **a** for a three-unit assembly. The light source is switched at $n_{LC} = 130$, with the light measurements presented in **d**. The vertical dashed line indicates the switching of the light source.

3.7 EXPERIMENTAL VALIDATION

Having gained insights into the dynamic learning behavior of our soft modular system using simulations, we next turn to multiple experiments backed up by simulations that demonstrate the main characteristics and, specifically, the robustness of our approach that we also observed in simulations. We do this by *(i)* changing the direction of light during an experiment, *(ii)* increasing the complexity of the environment by adding obstructions, and *(iii)* changing the geometry during the experiment by cutting the assembled robot in two.



Caption on next page.

Fig. 3.8: **Response of an assembled robot to a sudden change in light direction in simulations.** Trajectory of simulations a four-unit square (**a**) system and a three-unit (**b**) system for 2000 N_{LC}. Out of the 30 simulations performed for each configuration, the trajectories of the first 10 are represented in the figure. For each configuration, the first trajectory is highlighted along with snapshots of the system for every 200 n_{LC}. **c-f**, The distributions of the orientations θ for all 30 simulations of both configurations. Change of orientation $\Delta\Theta$ for all 30 simulations in **d** and **f** respectively.

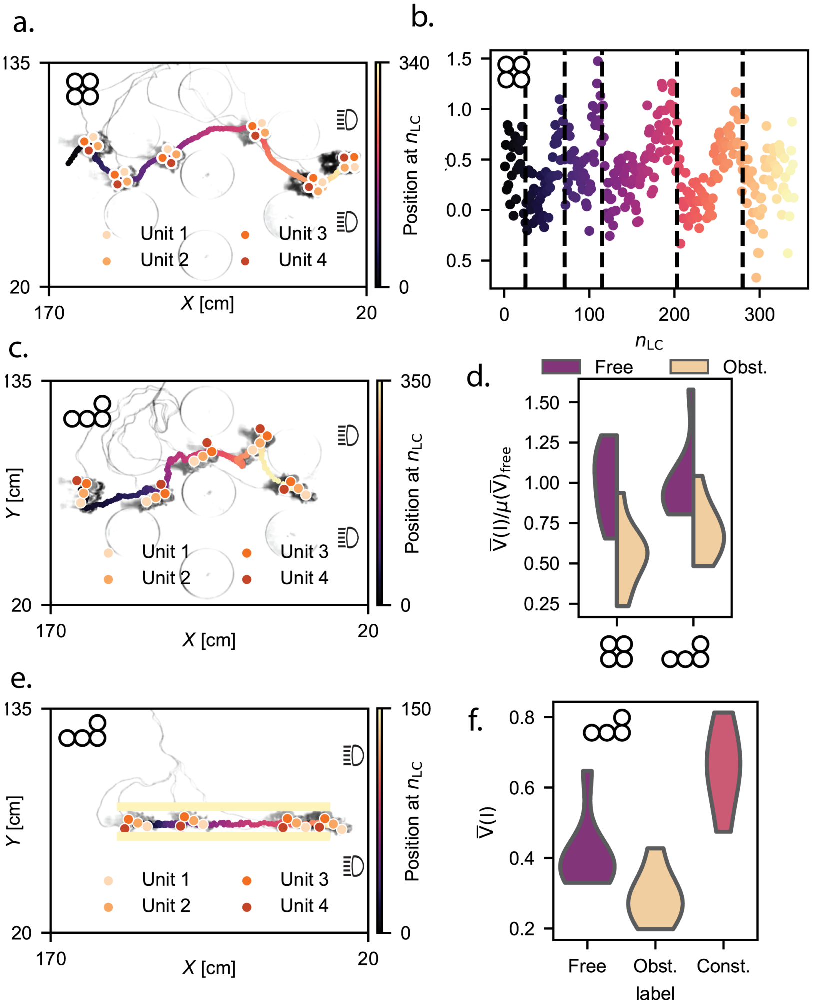
Suddenly changing the direction of light

As the first experimental demonstration, we change the light direction during an experiment, which will require each unit to update its phases. The change in light direction is accomplished by mounting LED panels on both sides of the experimental setup and turning them on and off during the experiment. In Fig. 3.7a, we show a single experiment of the square configuration for $N_{LC} = 400$ learning cycles, where we manually change the direction of the light from left to right approximately halfway through the experiment at $n_{LC} = 23$. We can clearly observe how the robot first moves to the left, after which the direction changes to accommodate displacement in the opposite direction (Movie S3.4). The sudden change can be clearly seen in the discrete jump in the measured light intensity of the four units, which reduces to a value close to zero as shown in Fig. 3.7b. As expected, the system requires some learning steps to adjust its behavior and reach a steady motion in the direction of the light.

Similarly, for a three-unit geometry (Fig. 3.7c-d) we also observe that the system adapts its motion to the change in light direction. It is interesting to note that in this specific case, at least one of the units measures no light intensity ($I = 0$) after the manual change in light direction (see insert in Fig. 3.7d), which means the sensor of this unit is outside of the range of the light source. However, it appears that the other units can compensate for this unit and move the system back to a measurable distance of the light.

Furthermore, Fig. 3.7c demonstrates how the system orients itself towards its preferred orientation (Similar to Fig. 3.3e and Fig. 3.4a) (with the middle unit pointing in the direction of the light) and turning 180 ° when the light is switched. For the square-shape of Fig. 3.7a we do not observe a clear switch of orientation as the light is changed. Next, we turn to simulations to find a more quantitative evaluation of the reorientation due to the switching of the light direction. From Fig. 3.8a, we see that, in general, the trajectories of the square-shape seem to find a straight line back and forwards, whereas the three-unit system (Fig. 3.8b) makes wide loops while rotating. Fig. 3.8c indicates that throughout the simulation, all orientations are used by the square-shape in contrast to the thee-unit configuration where we find that the orientations converge to 45 ° on the first half of the simulation and to 125 ° of the second. Fig. 3.8e provides additional insights in the lack of reorientation compared to the converging behavior found in Fig. 3.8f. Hence, it demonstrates the influence of the more symmetric square-shape geometry as opposed to the three-unit geometry, as also demonstrated by the orientation perspective in Fig. 3.4e

Because the current system only has short-term memory and the units have no notion of orientation, we find similar learning behavior between the start of the experiment and the behavior after the light has switched. Therefore, switching the light can be viewed as nothing more than kicking the dynamical system out of its equilibrium, after which we find robust recovery to phototaxic behavior.



Caption on next page.

Fig. 3.9: **Adaptive behavior of assembled robots in a more complex environment in experiments.** **a**, Single experiment of a square-shape robot moving through circular obstacles. The circular obstacles restrict the motion of the robot, but do not block the light source that is placed on the right. The first five images indicate the moment of contact with an obstacle, and the last image represents the end of the experiment, as indicated by the dashed lines in **b**. **b**, The average velocity $\overline{V}(I)$ [cm/LC] of the system during the experiment performed in **a**. **c**, an experiment with an L-shape robot under the same conditions as in **a**. **d**, Distribution of the $\overline{V}(I)$[cm/LC] for both the square-shape and L-shape robot, with and without obstacles, and normalized by the average $\overline{V}(I)$[cm/LC] without obstacles, each distribution contains 8 experiments of 230 N_{LC} (at $n_{LC} = 230$ the first free experiments reaches the end of the canvas). **e**, A single experiment with the L-shape robot in a confined environment restricted by two bars. **f**, Average $\overline{V}(I)$[cm/LC] for the L-shape robot in experiments without obstacles, with circular obstacles, and in the environment constrained by the two bars. Each distribution consists of 8 experiments with the same N_{LC} as **d**.

Operating in a more complex environment

So far, we have demonstrated the system's ability to generate robust phototaxic behavior, irrespective of the system's configuration. However, we have not considered more complex environments that can obstruct the motion and rotation of the system. As the second experimental demonstration of our approach's robustness, we will explore the system's behavior in more challenging environments. We achieve such environments by fixing impassable circular perspex disks with a height of 1 cm and a diameter of 25 cm to the surface, as shown in Fig. 3.9a. These barriers restrict the system's movement while leaving the light measurements undisturbed.

We next perform a learning experiment using a square assembly (Fig. 3.9a-b). Even though the system is capable of moving past the obstacles, the observed behavior is considerably affected by the obstruction. The effect on the phototaxic behavior is clearly visible by considering the $\overline{V}(I)$ in Fig. 3.9b, which decreases every time the system hits an obstacle. For example, we find five interactions with the barriers in this single experiment. As a result, we observe sharp changes in

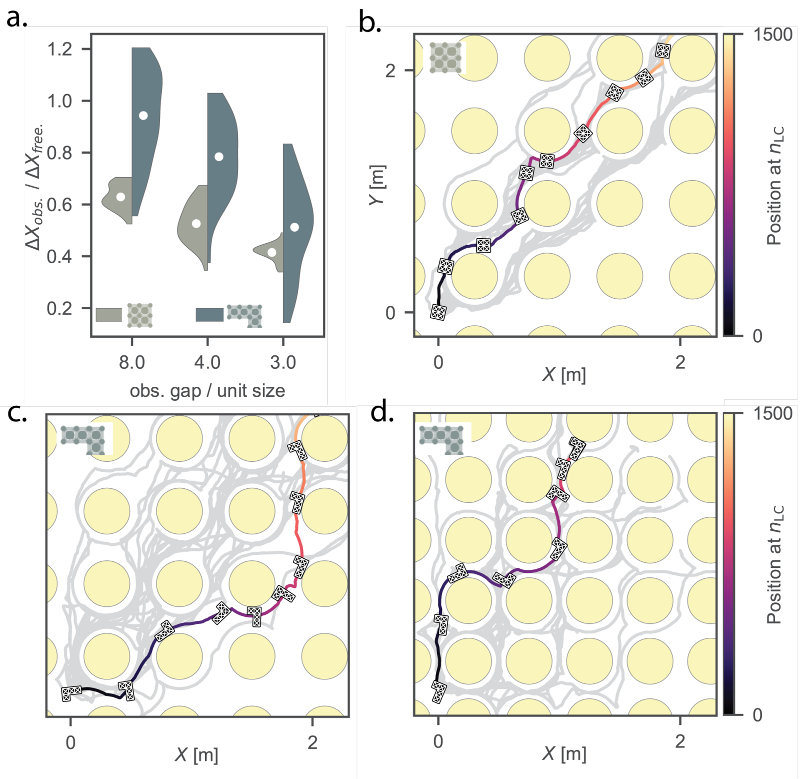


Fig. 3.10: **Adaptive behavior of assembled robots in a more complex environment in simulations. g**, Distributions of the covered distance (in the direction of the light) [m] divided by the covered distance without obstacles for both the square-shape and the L-shape in simulations with obstacles (circles with a diameter of 20 cm). The simulations last 1500 N_{LC}, and the x-axis represents the minimal distance between the outer edges of the obstacles in cm divided by the minimal length of a single unit (5 cm). **h-k**, Trajectories of the simulations of **g** for 2 different configurations and three different obstacle distances. The square shape with an obstacle gap distance of 20 cm is shown in **h** with the obstacles in yellow, the 30 trajectories in gray, and a single simulation highlighted with the colormap for the n_{LC}. **i-j**, follow the same representation as **h** but for the L-shape with an obstacle gap distance of 20 and 15, respectively.

the trajectory of the configuration. Still, for this specific environment, the system is capable of adjusting the behavior after each interaction (Movie S3.5).

The ability of the system to adjust to more complex environments will likely depend on the geometry of the system. To test this, we also run experiments with an L-shape assembly operating in the same environment (Fig. 3.9c). For the L-shape, we observe that the system requires more learning cycles N_{LC} to reach the position with the highest light intensity. To evaluate the influence of the obstacles on both the square and the L-shape geometries, in Fig. 3.9d we compare the distribution of the velocity over a learning cycle normalized by the average speed of experiments without obstacles for that specific shape. We find that this more complex environment influences the L-shape less. One explanation could be the difference in the symmetry between the two geometries, as we know from Fig. 3.4e that the maximum $\overline{V}(I)$ of the square unit is less affected by rotation. The results from Fig. 3.9d could indicate that the obstacles constrain the system into orientations with higher maximum $\overline{V}(I)$, as Fig. 3.4 already indicated that geometries often stabilize in orientations with sub-optimal $\overline{V}(I)$.

Although in the previous case, the environment has a negative effect on the average $\overline{V}(I)$ of both assemblies, to test if the L-shape assembly can also benefit from its surroundings, we next perform an experiment in an environment that contains two parallel placed bars (Fig. 3.9e). In Fig. 3.9f, we compare the average $\overline{V}(I)[cm/LC]$ of the system in the direction of the light between no obstacles, circular obstacles, and the two bars, and interestingly observe that the velocity of the system is, on average higher when constrained by the two bars. These results show that, in some cases, the system could even utilize its environment to increase its displacement toward the light. This can be explained by the fact that the system's rotation is constrained, such that the modular robot can sustain otherwise unstable orientations with higher velocities. In addition, the friction from the bars could aid in an additional contact point.

To further extend our studies of the impact of geometry on the ability to maneuver the obstacles presented in the experiments above, we emulate the experiments with the implemented course-grained simulations. We specifically focus on how the distance between the obstacles influences the behavior. Fig. 3.10a demonstrates how sparsely placed obstacles

result in a better relative performance compared to no obstacles in the L-shape as opposed to the square-shape, similar to the results of the experiments in Fig. 3.9d. Interestingly, we find that, on average, the L-shape experiences more speed degradation due to smaller obstacle distances compared to the square-shape. This is likely because the L-shape has to reorient to fit through the narrow opening between the obstacles, whereas the square-shape still fits through in most of its orientations and only has to change phases without the need to rotate. When evaluating the trajectories for the obstacle distance of 20 cm (equal to the size of 4 units), we find more deviating trajectories for the L-shape (Fig. 3.10c) compared to the square-shape (Fig. 3.10b). Furthermore, the L-shape seems to have a preferred direction to surpass the obstacles as we find more trajectories following the top-left part compared to the bottom-right in Fig. 3.10c, which could potentially be explained by the asymmetry in the orientation perspective (Fig. S3.8a). In Fig. 3.10d, we can clearly see this reorientation of the L-shape geometry as it maneuvers in between the obstacles.
From these experiments and simulations, we conclude that the system can adapt to its environment to move in the light source direction, emphasizing the robust nature of the control strategy (Fig. S3.38). Here, the obstacles can be viewed as changes to the equilibrium behavior of the dynamical system, which is different to the effect of changing the light direction. Where the changing direction of the light slowly changes the dimension of the orientation in the search space, the obstructions actually change the search space as a whole. Note that we have only considered a very limited number of environments, where in both cases, there were possibilities to increase the light intensity even after hitting an obstruction. It could be possible for systems to get stuck in environments, similar to how flies sometimes get stuck behind the glass when trying to fly outside. We expect this to occur when the light intensity decreases in all possible movement directions (even though there is still some stochasticity in the movement that could help escape these situations).

Changing the geometry by damaging the robot

In previous sections, we observed that our system could adapt its behavior to changing objectives and more complex environments. As a

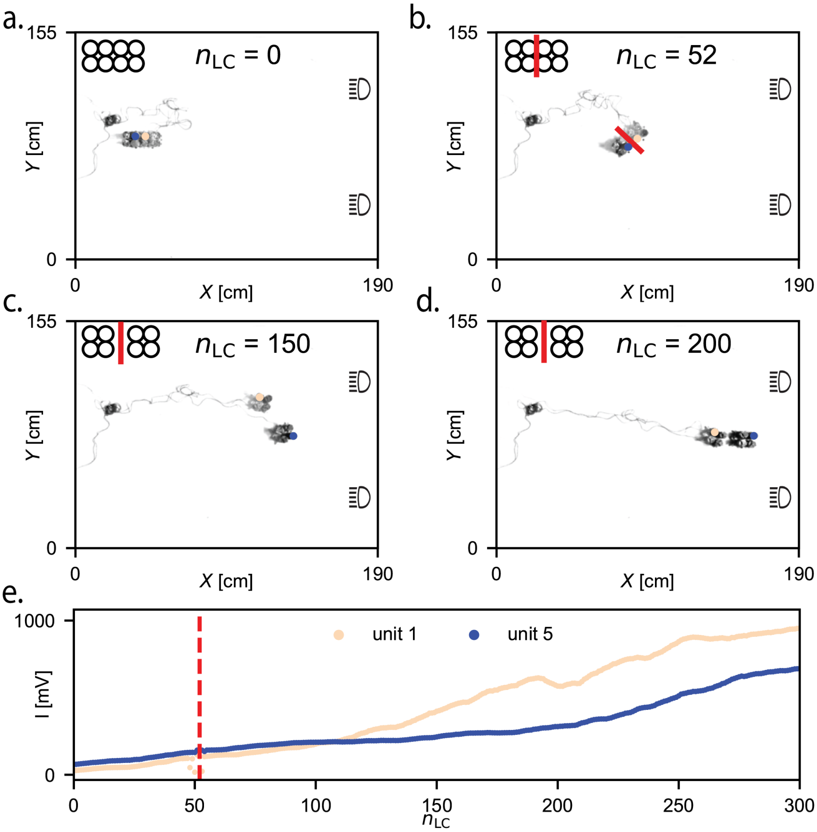


Fig. 3.11: **Separating a robot in two during an experiment. a-d**, depicts the system's time evolution for a single experiment for an 8-unit configuration that is cut in half during the experiment at the 50th learning cycle. **e**, Light measurements at each learning cycle for two units that are each in a separate part of the robot. The red dashed line indicates the moment the robot is separated in two.

final demonstration of robustness, we run an experiment with a larger rectangular assembly comprised of 8 units that we damaged during the experiment (Fig. 3.11a-d). At $n_{LC} = 50$ we split the system in two by manually breaking the physical connection between the units. As a result, both geometries have to adapt their behavior independently to the change in geometry and relearn to move toward the light source (Movie S3.6). We observe in Fig. 3.11e that the system is able to increase its light intensity before and after the damage. When comparing the light readings between two units that are in a different section after the cut, we observe how the units can perform phototaxis separately, without any input, and overcome sudden damage, clearly highlighting the robustness of our decentralized approach.

Apart from cutting the system in two, one could also inflict damage to the extent that the system remains connected, but any number of units become inactive. To study the resilience to failure in the system, we also perform simulations with a 5x5 (25 units) square lattice. Interestingly, Fig. S3.5d shows that the performance first slightly increases for a system with four inactive units. These results indicate a redundancy in the system. Although it has four non-actuating units (dead weight), the search space becomes less complex; therefore, the system can exploit a better solution in the phase space. Furthermore, we find that even though the $\overline{V}(I)$ gradually decreases with the increase in the number of inactive units, the system remains functional until there is only a single unit left.

3.8 CONCLUSION

In this work, we introduced a modular robotic platform to further study decentralized learning algorithms that allow them to adapt to more complex dynamic environments and configurations without needing a centralized controller or electrical connections between units. The current system relies on a form of “collective memory" where each unit explores its own behaviors (the one-dimensional phase parameter). The units evaluate these behaviors using their respective light sensor (LDR) to form a notion of their “environment" (the mapping of the behavior to the phototaxis of the unit). By adjusting its phase, each unit in our modular robotic system influences the collective state of the assembly.

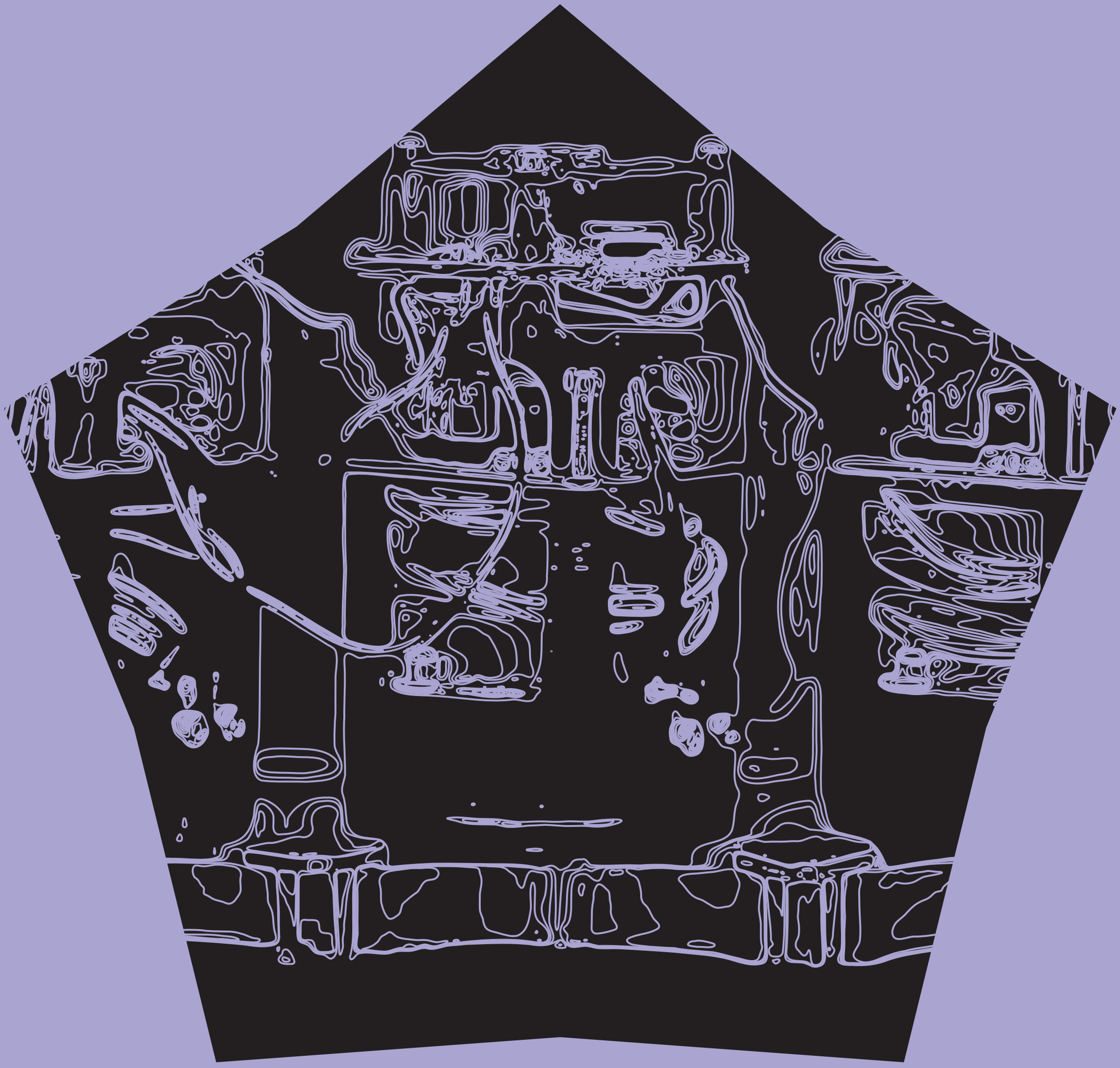
The resulting phototaxis depends on a distributed form of memory and coordination. This indirect communication mechanism is based on the partially shared environment, which is facilitated by the physical connection between units and affects the individual decision-making process in subsequent cycles. While this process bears a resemblance to stigmergy that also depends on decentralized behavior and indirect communication, it does not involve altering the terrain. Instead, it relies on the dynamic interplay of local memory and partially shared sensory inputs across the collective. This is exemplified by the partially shared environment that is constantly changing, as observed for our individual units in Fig. S3.4.

In the current study, we increased the complexity of the task by moving from one-dimensional experiments [88] to a two-dimensional modular system. Even though different phase behavior is needed for different geometries to achieve phototaxis, no information or model was *a priori* needed. We demonstrated that the current short-term memory approach allows the system to adapt to changes in its objective, environment, and geometries. Furthermore, we discovered that relying on short-term memory in this two-dimensional task inherently gives rise to preferential orientations and rotations, thereby strongly influencing the long-term phototaxis of the system. Moreover, we found that these preferential states depend solely on the geometry but can be traversed differently by adapting the learning parameters. Because each unit is identical (both in mechanical behavior as well as in algorithm) and the system does not have any centralized brain, the ability of the system to adjust its behavior to the environment and assembled shape could be viewed as a material property of the system as a whole, which is often referred to as "robotic matter" [62, 73, 88, 106, 110].

Here, the simplicity and universality of the algorithm make it suitable for a wide range of future applications on the interface of soft robots, swarm intelligence, and nanorobotics [10, 80, 89], where computational power is limited due to size and weight restrictions or for which the behavior and interactions are difficult to model *a priori*. Our algorithm could be transferred to the fluidic domain to create more autonomous electronics-free soft robotic systems [25, 44]. Furthermore, the algorithm could be beneficial for micro- and nano-systems in medical applications, where the following chemical gradients aid in site-specific drug delivery [1]. Groups of aggregates can be deployed to explore environments where fault tolerance is of the utmost importance (e.g., space exploration and exploration of underground oil reservoirs) [7]. However, to reach these applications, more knowledge needs to be gained on how decentralized behavior can be embedded in such systems and how specific objectives can be incorporated and emerge. It should be noted that current work focused mostly on gaining a better understanding of how implicit communication and collective memory can lead to robust behavior. Potential applications will likely bring about compromises, and additional work and redesign of the algorithm are likely needed to make our approach applicable to these different physical implementations. Our study provides a platform to study these principles, both numerically and experimentally.

SUPPLEMENTRY MATERIAL FOR CHAPTER 3.

S3.



This Chapter provides additional material to support *Chapter 3*. It details the fabrication of the robotic units, the experimental setup and data-acquisition/processing pipeline, and the unit actuation and learning protocols. We further present the numerical mass–spring model, including the force balance with friction and the time-dependent actuation, together with the parameters used. Supplementary figures and captions for Movies S3.1 to S3.6 are included below.

Other Supplementary Material for this Chapter includes the following:

S3.1 SUPPLEMENTARY MOVIES S3

Movie S3.1. Experimental units. The video on the left side of the frame depicts a top view of a single unit while actuating its four connector arms. The video on the right side of the screen displays a single unit from the side view at two different angles.

Movie S3.2. Phototaxis without any explicit communication. Time-lapse of a single experiment of a three-unit geometry for 350 learning cycles. A planar light source is positioned on the right side of the screen. During the experiment, each immobile light-seeking unit continuously adapts its behavior to increase its sensed light intensity I [mV] without explicit communication between the units. As a result, the three physically connected units coordinate their behavior to achieve phototaxis collectively.

Movie S3.3. Simulation of the long-term behavior of the three-unit system. Simulation of a three-unit geometry for 1000 learning cycles. The top figure presents the local optima of displacement towards the light for each orientation of the system with the displacement and the orientation of the single simulation projected into it. The bottom figure presents the position and orientation of the system during the simulation.

Movie S3.4. Experiment with switching objectives. An experiment of 350 learning cycles where the objective is switched by moving the light in the opposite direction. At the start of the experiment, the light is placed on the left side of the screen. At learning cycle 144, the light is moved to the right side of the screen. The figures on the bottom display the measured light intensity I at the first measurement and the difference between this value and the second measurement (used to evaluate the behavior).

Movie S3.5. Experiment with physical obstacles. In this experiment, physical obstacles are placed on the surface of the experimental area. These obstacles only restrict the system's movement and do not impact the light intensity measurements. In the bottom figure, we display the displacement in the direction of the light (the longitudinal component of the displacement obtained from the processed images) for every learning cycle. The moment of the first impact with an obstacle is indicated both in the trajectory and in the figure below.

Movie S3.6. Eight-unit system split in two during the experiment. In this video, we experiment with an eight-unit geometry. At learning cycle 50, we cut the geometry in two, creating two square geometries. The system adapts its behavior to the new geometry to continue moving in the direction of the light.

Link to the Movies.

S3.2 METHODS

Fabrication of units

The individual robotic units consist of structural components custom designed and 3D printed in polylactide (PLA). An overview of the assembly can be viewed in Fig. S3.10. Two 3D-printed geared disks are positioned on a base disk. A top segment is designed to accommodate a stepper motor (28BYJ-48) and support the electronic components (Custom designed PCB). The robotic units have a 2.1 cm radius, stand 8.5 cm tall, and weighs 75 g when fully assembled. The "brain" of each unit consists of a mico-controller (ESP32) mounted on a custom circular PCB. A second triangular PCB containing three symmetrically positioned Light Detector Resistors (NSL-19M51 Luna Optoelectronics) placed in series connects to the first PCB with 2 mm PCB spacers in between. The PCB is connected to the stepper motor through a stepper driver (ULN2003), all encased in a 3D printed body and attached with inbus bolts (M2X16 and M3X12). The 3D-printed body consists of four printed segments. The first is the top frame which houses the PCB, stepper driver, and stepper motor and sensors. A centrally placed brass axle (6 mm) connects the top frame to the bottom frame. A 3D printed gear connects the stepper motor to two 3D printed disks (top and bottom) aligned by the central brass axle and held in place by the top body and a 3D bottom body piece. The bottom frame also houses a metal bearing (NMB, Radial Ball Bearing) 10x4x4 to keep the 3D stepper gear in place. The rotation of the stepper translates into the bottom and top disk rotations in opposite directions to each other. A Micro-switch (Omron Ultra Subminiature Basis Switch) verifies the starting orientation of the disks mounted to the top frame and a notch in the top disk.

Physical connections between the units are needed to let the units interact with each other, allowing the system to displace its center of mass. We use soft connectors to bind units with each other. We cast the soft connectors using custom-designed molds 3D printed out of PLA (Ultimaker 3). The connections are fabricated with two-component silicone (Smooth-on DS30,in opaque white). The soft connectors have two arms, each fixed to one of the disks of the units. The total length of the soft actuator is $L = 25\,\mathrm{mm}$. The radial position of the soft connectors is dictated by the poses of the two counter-rotating geared disks actuated by the stepper motor. Using this design, we can connect each unit to a maximum of four neighboring units using a 3D-printed pin connector shown inFig. S3.3d.

Experimental setup

The phototaxis experiments are performed under controlled light conditions on a white multiplex surface (190x155cm) encased by a black multiplex frame, as presented in Fig. S3.2c. At the beginning of each experiment, all units are updated and turned on using a master-slave protocol over WiFi, coordinated from a master ESP32 connected in a one-to-many configuration using the esp-now library. After the initiation of an experiment, every unit gathers sensor and learning data individually. At the end of every learning cycle, the unit sends its collected data back to the master ESP32 connected to the lab computer. The master unit, therefore, collects the gathered data for all the units and writes this data to a .tsv file in real-time. Note that the data only needs to be gathered to visualize the results but is not needed to update the phases of the units. The visual and positional data of the experiments are collected using a GoPro (hero 4) positioned at the center of the white multiplex surface at a height of 1.11 m. The GoPro is controlled remotely from the computer that collects the unit's sensor data. The GoPro is prompted to take a photo every time the units have completed a learning cycle.
We synchronize the position data obtained with the GoPro with outputs from the agents to fully reconstruct the dynamics of the system. For example, in the experiments where we map the behavior of a three-unit assembly in phase space (Fig. S3.3b), pictures are collected every ten actuation cycles per phase combinations. This allows for averaging unwanted biases in the actuation and having a large enough displacement to validate the magnitude and direction of locomotion. For all other experiments, instead, we collect one image every learning cycle of the system. In order to take accurate data while inferring the position of the units over time, we correct for optical distortions and camera perspective. On top of that, the visual data is processed using python 3.8 and the OpenCV package in three stages. First, we correct the camera's perspective using four arucos in the corners of the canvas. By measuring the distance between the outer corners of the arucos we can

map the pixel distance to the actual distance on the canvas. Secondly, a color mask and circle detection are used to detect the positions of two 3D-printed discs with $d = 5\,\mathrm{cm}$ and $d = 2.5\,\mathrm{cm}$ that are placed on top of two of the units in the configuration. Lastly, the positions of the two discs are used to infer the center of the configuration and its rotation. We follow the following protocol to ensure the validity of our data. First, we define the working range of the canvas to be $20\,\mathrm{cm}$ from every edge of the canvas to avoid the influence of the borders during the experiment. Experiments are, therefore, cut off if the units surpass this boundary. As a consequence, in Fig. S3.11 and Fig. S3.12, we find that not all experiments are of equal duration, as the experiment is stopped when the units reach the end of the canvas. Furthermore, we do not accept more than one consecutive missing data point. In the case of a single missing data point, we perform linear interpolation to fill the gap. Measured over all the performed experiments, linear interpolation is used for 0.059% of the position data points.

To automate the repetition of the experiments, we implement a mechanism to consistently restore the initial position of the units at the end of each experiment. The solution consists of a physical connection between the assembly and a pulling system positioned above the working area. We position a dedicated servo motor (DFROBOT DF15RSMG) above the experiment and connect it to the units using a nylon thread with $d = 1\,\mathrm{mm}$. When an experiment is over, a flag signal is detected, and the servo motor is activated using the Python interface. As a result, units are pulled to a position approximately beneath the servo.

Unit actuation protocol

As displayed in Fig. 3.1a, the experimental units extend and contract their silicone connectors by rotating the two PLA discs in opposite directions. In the contracted state, the distance between the center of the unit and the outer point of the connector measures $\approx 3.5\,\mathrm{cm}$, while in the fully extended state, this distance becomes $\approx 3.85\,\mathrm{cm}$. The actuation cycle (AC) comprises two stages.

To compare experiments and simulations, we use a parameter called α, which is set to 0.3. Alpha represents the fraction of the total actuation cycle time (T_{AC}), which is $2\,\mathrm{s}$. During this time, the unit extends its connection arms for approximately 0.58 seconds (we denote this time as D_1).

The remaining part of the actuation cycle $(1 - \alpha)$ has two components. First, the unit contracts its four connection arms, which takes roughly the same time as the extension. Second, the unit stays stationary in the contracted state for a duration denoted as D_2.

The last actuation cycle is called the adjustment cycle (AC_a) and differs from the other cycles. In this cycle, we modify the time the unit stays in the contracted state (D_2) to be $D_a dj = D_2 - \Delta\phi_i$. This adjustment changes the total duration of the adjustment cycle (AC_a), enabling the unit to start the next learning cycle (LC) with a new phase ϕ_i.

In between the two light intensity measures I_1 and I_2 there are $N_{AC} = 6$ regular actuation cycles to reduce the noise measured in ΔI and to make the camera tracking feasible. Therefore, there are seven extensions and contractions during a single learning cycle (LC). At the end of every learning cycle, the unit sends its data related to the current learning cycle back to the central computer. Note that every unit operates completely independently from the other units and without synchronization.

S3.3 MODEL

Mass-spring system

To qualitatively capture, and to gain more insight into, the behavior observed in experiments we modeled the soft modular robot as a minimal mass-spring system. Our aim is to develop a model that is simple enough for quick computation, yet comparable enough to qualitatively capture the observed behavior. We do not aim for a perfect quantitative comparison. To achieve this, the system is modeled as a two-dimensional structure consisting of one-dimensional linear elastic elements connected by joints of a certain mass, that interact with the environment through friction. This system decomposes all forces in x and y directions, resulting in a system with n_n nodes connected by e_n elements in a two-dimensional space with $2n_n$ degrees of freedom, where n is the number of units in the assembled robot.

Figure Fig. 3.3a illustrates the nodes and elements represented as masses and springs. Each unit in the assembled robot consists of four nodes and six elements, where an element connects every pair of nodes. The diagonal elements of the module are periodically actuated to achieve symmetric expansion and shrinking. These elements are referred to as the active elements.

Force balance

A multi-module system couples the sides of individual blocks together. Each block has four sides that can be connected to another module. Two connected modules share the two nodes and the connecting element of the linked side, thereby reducing the number of nodes and elements. The mass of the units is equally distributed over their four adjacent nodes.

Let the system consist of N nodes in two dimensions. The position of node i at time t is

$$\mathbf{r}_i(t) = \begin{bmatrix} x_i(t) \\ y_i(t) \end{bmatrix} \in \mathbb{R}^2.$$

We define the global displacement vector as

$$\mathbf{u}(t) = \begin{bmatrix} \mathbf{r}_1(t) \\ \mathbf{r}_2(t) \\ \vdots \\ \mathbf{r}_N(t) \end{bmatrix} \in \mathbb{R}^{2N}.$$

Taking the force balance, we find for each node

$$\mathbf{M}\ddot{\mathbf{u}}(t) = \mathbf{F}_e(t) - \mathbf{F}_f(\dot{\mathbf{u}}(t)) - \mathbf{F}_k(\mathbf{u}(t)),$$

Assuming lumped masses per node:

$$\mathbf{M} = \mathrm{diag}(m_1, m_1, m_2, m_2, \ldots, m_N, m_N).$$

Each elastic bar connects a node pair $(i, j) \in \mathcal{E}$. Let $\mathbf{d}_{ij} = \mathbf{r}_j - \mathbf{r}_i$ be the relative distance vector. Define the unit direction vector and stretch as:

$$\hat{\mathbf{d}}_{ij} = \frac{\mathbf{d}_{ij}}{\|\mathbf{d}_{ij}\|}, \quad \delta l_{ij} = \|\mathbf{d}_{ij}\| - l_{0,ij}.$$

Then, the force on node i is given by:

$$\mathbf{f}_{ij} = K_s \delta l_{ij} \hat{\mathbf{d}}_{ij} = K_s \left(\|\mathbf{d}_{ij}\| - l_{0,ij} \right) \cdot \frac{\mathbf{d}_{ij}}{\|\mathbf{d}_{ij}\|}.$$

Alternatively, using the projection matrix $\mathbf{P}_{ij}$:

$$\mathbf{P}_{ij} = \frac{\mathbf{d}_{ij}\mathbf{d}_{ij}^\top}{\|\mathbf{d}_{ij}\|^2} \in \mathbb{R}^{2\times 2},$$

we can write:

$$\mathbf{f}_{ij} = K_s \left(\|\mathbf{d}_{ij}\| - l_{0,ij} \right) \mathbf{P}_{ij}\mathbf{d}_{ij}.$$

Let $\mathbf{B}_{ij} \in \mathbb{R}^{2\times 2N}$ be the difference operator extracting $\mathbf{d}_{ij}$ from $\mathbf{u}$:

$$\mathbf{d}_{ij} = \mathbf{B}_{ij}\mathbf{u}.$$

Then, the global internal elastic force is:

$$\mathbf{F}_k(\mathbf{u}) = \sum_{(i,j)\in\mathcal{E}} \mathbf{B}_{ij}^\top \cdot K_s \left(\|\mathbf{B}_{ij}\mathbf{u}\| - l_{0,ij} \right) \mathbf{P}_{ij}\mathbf{B}_{ij}\mathbf{u}.$$

The friction force $\mathbf{F}_f$ acting on each node is based on a general nonlinear friction model [88].

Because friction always acts in the direction opposite to velocity, a direction-preserving scaling is used via the unit velocity vector. A smooth approximation of the sign function (via tanh) ensures numerical stability around zero velocity.

Given the empirical model from [88], the friction force acting on node i becomes:

$$\mathbf{F}_{f_i} = -\left(F_C \cdot \tanh\left(\frac{\|\dot{\mathbf{u}}_i\|}{\nu_{\text{Coul}}} \right) + f\|\dot{\mathbf{u}}_i\| \right) \frac{\dot{\mathbf{u}}_i}{\|\dot{\mathbf{u}}_i\|}. \tag{S3.1}$$

Here, $\dot{\mathbf{u}}_i \in \mathbb{R}^2$ is the velocity of node i, and the direction of friction is aligned opposite to this velocity vector. For nodes with zero velocity, this equation can be regularized or evaluated in a limiting sense.

The total global friction force vector is then:

$$\mathbf{F}_f(\dot{\mathbf{u}}) = \begin{bmatrix} \mathbf{F}_{f_1} \\ \mathbf{F}_{f_2} \\ \vdots \\ \mathbf{F}_{f_N} \end{bmatrix} \in \mathbb{R}^{2N}.$$

Symbol	Description	Value
F_C	Coulomb friction	0.172
v_{Coul}	Coulomb velocity threshold	$v_{\text{brk}}/10$
f	Viscous friction coefficient	4

Table S3.1: Friction model parameters based on [88].

To solve the system numerically, we convert it to first-order form. Define the state vector:

$$\mathbf{y}(t) = \begin{bmatrix} \dot{\mathbf{u}}(t) \\ \mathbf{u}(t) \end{bmatrix} \in \mathbb{R}^{4N}.$$

Then the time derivative is:

$$\boxed{\dot{\mathbf{y}}(t) = \begin{bmatrix} \mathbf{M}^{-1}\left(\mathbf{F}_e(t) - \mathbf{C}\dot{\mathbf{u}}(t) - \mathbf{F}_k(\mathbf{u}(t))\right) \\ \dot{\mathbf{u}}(t) \end{bmatrix}}$$

The current system has individual modules that can expand and shrink back to their original shape. These deformations are achieved by simultaneously activating the diagonal elements in each unit. The current model implements actuation by redefining l_0 as the time-dependent variable preferred spring length $L_p(t)$ in every diagonal element. The preferred spring length $L_p(t)$ of the diagonal elements is divined as the sum of the initial spring length l_0 plus the time-dependent actuation length $l_a(t+\phi)$. In order to make the differential equation of the force balance less stiff and speed up the simulation, we implemented a smooth version of the actuation function for $l_a(t+\phi)$ based on the actuation used in the experimental units. By making use of a piece-wise sine function, we qualitatively approximate the extension-contraction cycle of the experimental units. The active spring length during an actuation cycle is given by Eq. S3.2 with the parameters of Table S3.

$$
\begin{aligned}
a &= \omega \cdot \sin(2\pi f(t + \phi_i)) \\
b &= \omega \cdot \sin\left(2\pi f \frac{T_m}{4}\right) \\
c &= \omega \cdot \sin\left(2\pi f\left((t + \phi_i) - D_1\right)\right) \\
d &= \omega \cdot \sin\left(2\pi f \frac{3T_m}{4}\right) \\
e &= \omega \cdot \sin\left(2\pi f\left((t + \phi_i) - D_1 - T_s\right)\right) \\
l_a(t, \phi_i) &= \begin{cases} a & \text{if } t < \frac{T_{mot}}{4}, \\ b & \text{if } t \in \left[\frac{T_m t}{4}, \frac{T_{mot}}{4} + D_1\right), \\ c & \text{if } t \in \left[\frac{T_m}{4} + D_1, \frac{3T_m}{4} + D_1\right), \\ d & \text{if } t \in \left[\frac{3T_m}{4} + D_1 t, \frac{3T_m}{4} + T_s\right), \\ e & \text{if } t > \frac{3T_m}{4} + T_s. \end{cases}
\end{aligned}
\tag{S3.2}
$$

Furthermore, we implement the same learning cycle (LC) as used for the experimental units apart from the number of regular actuation cycles between the two light measurements I_1 and I_2, as there is no noise in the light measurements during the simulations and because we only aim to capture the qualitative behavior of our system we reduce the number of regular actuations N_{AC} to three. Thereby the learning cycle LC of the system in the simulations consists of four expansion-contraction cycles.

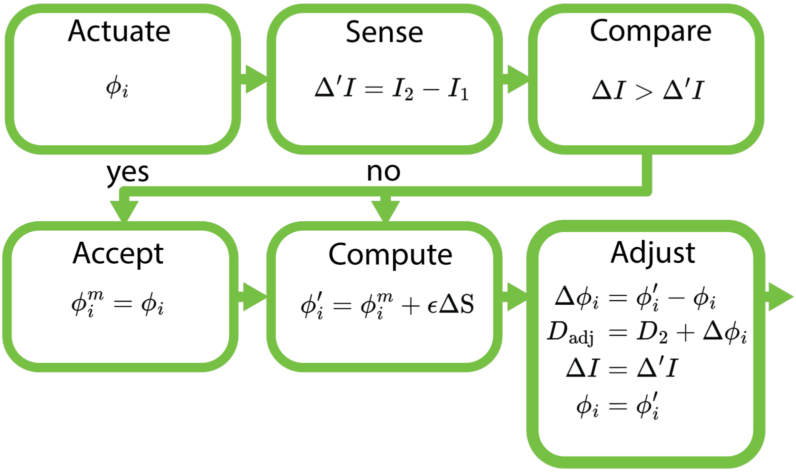
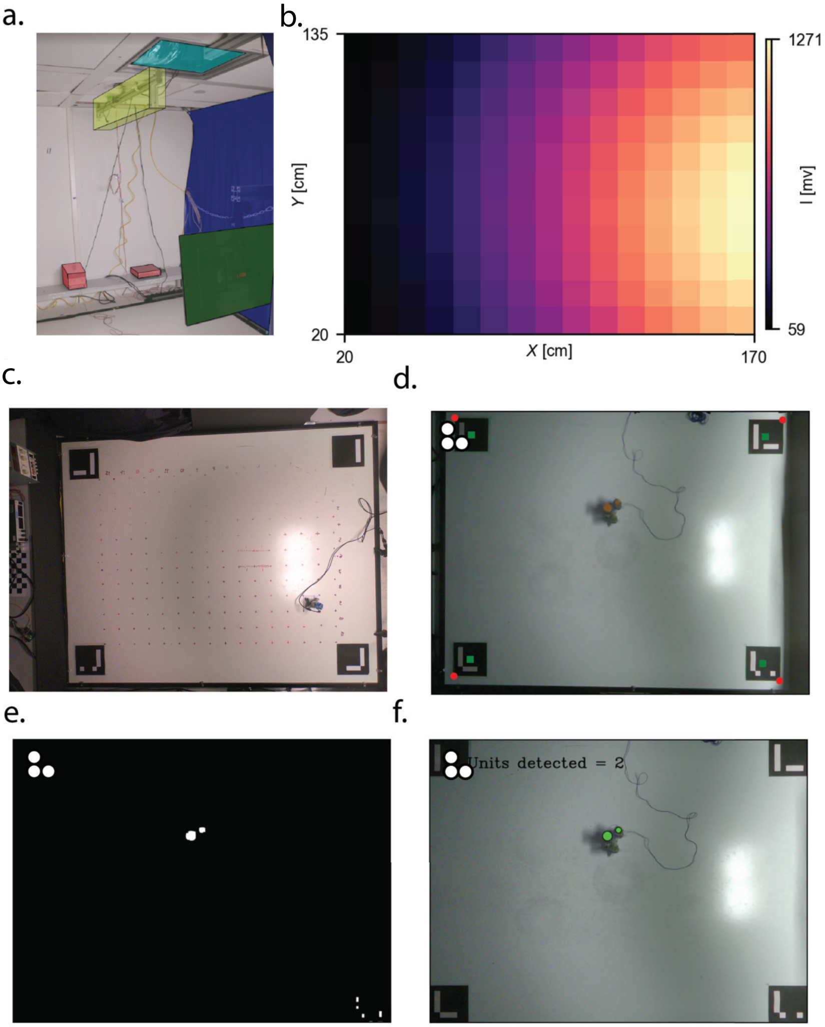
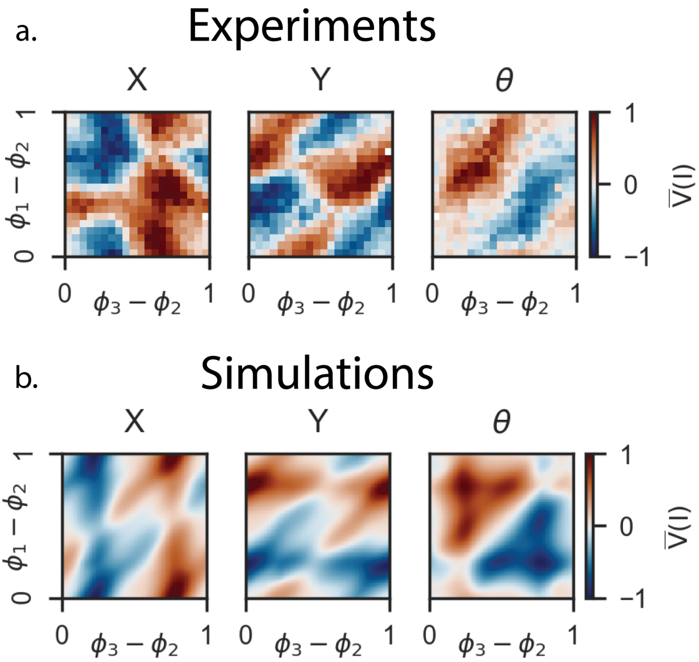


Fig. S3.1: **Learning algorithm schematic.** Schematic of the numerical model describing the steps in each learning cycle following a discrete event model.



Caption on next page.

Fig. S3.2: **Measurement setup for phototaxis experiments. a**, Components of the experimental setup, with the power supplies for the Gridbot units and the restart protocol servo (highlighted in red), the restart protocol apparatus (yellow), the GoPro camera (blue), the LED source (green), and the curtains for shielding the experiment from environmental light (blue). **b**, Light intensity (I) in mV measured by the LDR of a single unit placed on a grid over the canvas. **c**, A picture from the GoPro taken during the measurements of the light intensity of the full canvas. **d-f**, Picture from a single experiment with the three-unit system during the first step in the data processing in which we detect the outer corners of the canvas using the four Arucos ((**d**), the second step of the data processing where the two markers on the units are detected using a color filter and circle detection (**e**), and the last data processing step where we correct the distortion and position of the camera and locate the detected circles to find the position of the small and large unit markers (**f**).



Caption on next page.

Fig. S3.3: **Experiments vs. simulation for the three-unit geometry.** Comparison of the observed velocity and rotation between simulations and experiments for the three-unit system, depicted in the phase space. **a**, Simulation results for the longitudinal displacement, transversal displacement, and the change in rotation Δθ per actuation cycle, respectively. **b**, Experimental results of the same quantities as for the simulation. These experimental results are the averages over three repetitions of the same experiment. Although we find similar qualitative behavior between the thee-unit system in the experiments and the simulations, we can also see apparent differences. Firstly, we find the same general areas in the phase space of positive and negative displacement as well as for the rotation. However, we find a clear difference between the rotation and displacement amplitudes. One explanation for these differences in amplitude could be the differences in the friction between the experiments and the simulations. In the experiments, the units also have four contact points with the ground, but their distances remain stationary, while the simulations treat each friction point as an independent entity. Also, in the experiments, the friction points are positioned closer to the central axis (center of mass) of the unit. Which could explain the differences in the amplitude of the rotation. Lastly, the expansion and contraction of the connection arms likely differ between the experiments and the simulations, as they represent the exact geometry of the experiment and simplify them to one-dimensional springs. Although we implement the same actuation function for both experiments and simulation, the simulations have additional passive springs. These springs result in additional stiffness that limits the expansion and, therefore, limits the velocity of the system.

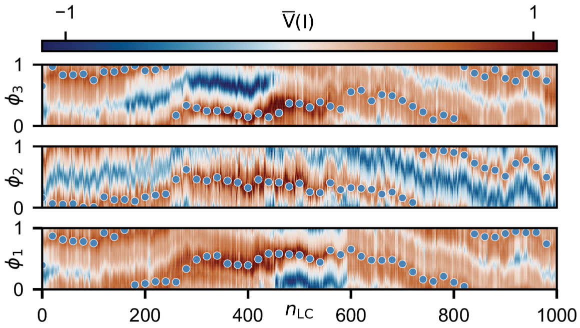


Fig. S3.4: **Visualization of the unit-perspective of a single simulation for the three unit configuration.** Perspective of the individual unit in the single simulation presented in Fig. 3.3b-c. Each of the three vertically aligned heatmaps depicts the perspective of a single unit. The x-axis presents the learning cycle, and the y-axis presents the plausible phases for that single unit normalized between 0 and 1. Within the heatmap, each vertical slice indicates the phototaxic behavior for each possible unit phase at a specific time instance during the simulations, obtained by remapping the system behavior found in Fig. S3.3a to the individual units. The blue dot indicates the chosen phase of the unit at that time instance (every 20 n_{LC}). The vertical slice of the heatmap can be viewed as the "environment" of a single unit, which changes continuously over time as the system rotates and the other units change their phases. Through the implicit communication of their physical connection, the units manage to find the general region of phototaxis. From this Figure, we observe that all three units probe a radically different "environment" but manage to keep up with the continuous change in order to move to the light continuously (Fig. 3.3c).

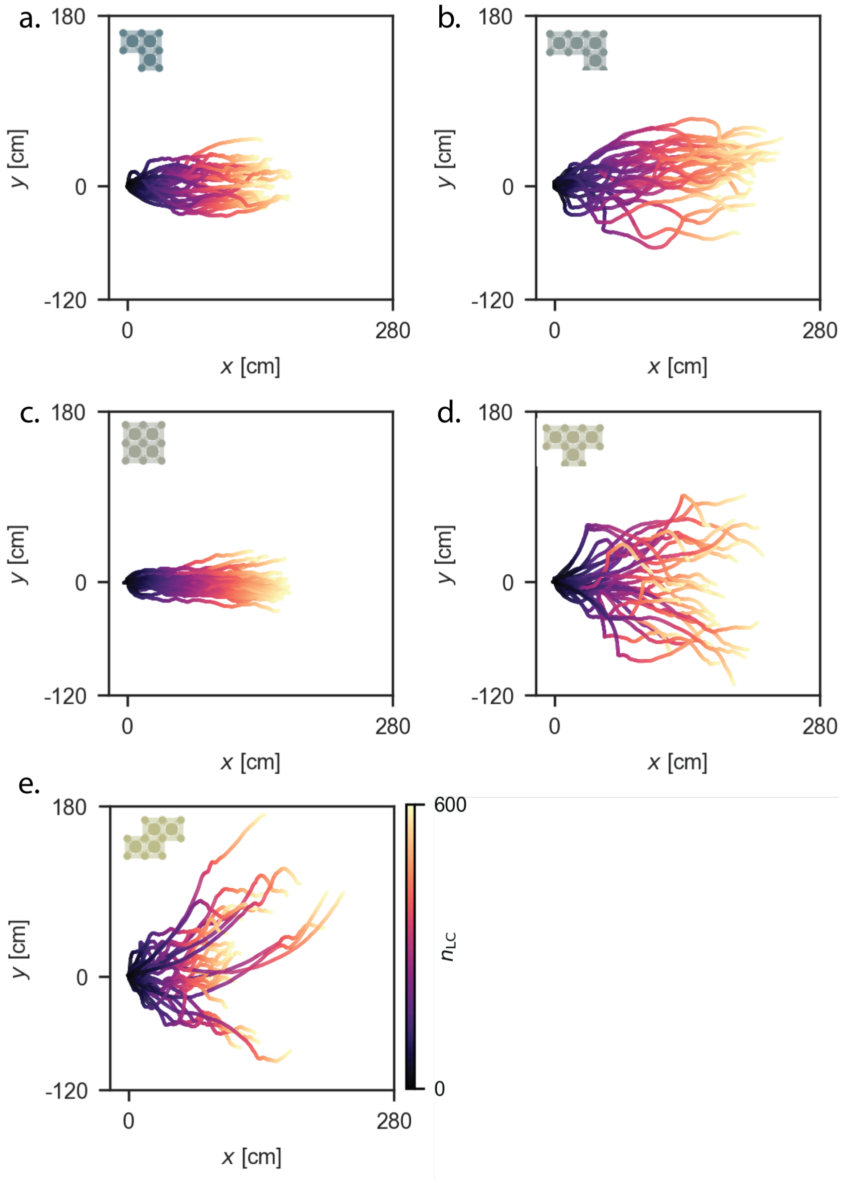
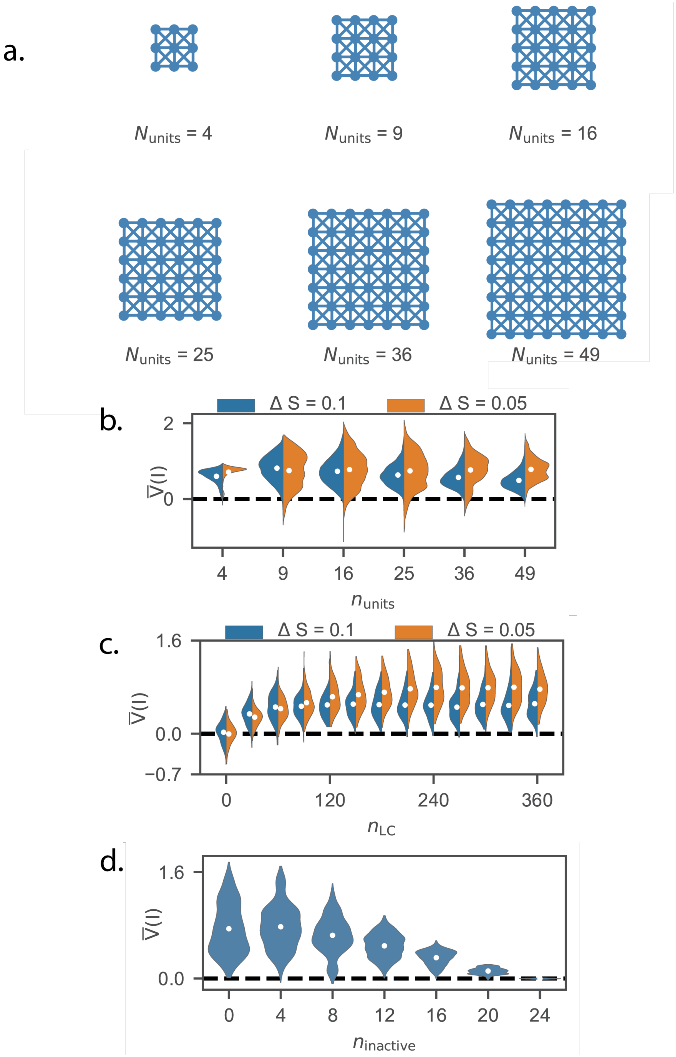


Fig. S3.5: **Overview of the trajectories for all geometries in simulation.a-e** 30 Trajectories of the mass-spring system simulations for 600 learning cycles. The color of the dots corresponds to the $n_{(LC)}$.



Caption on next page.

Fig. S3.6: **Simulations on the scalability and robustness of the system a** a graphical representation of the square lattice configurations with an increasing number of units. We performed simulations of up to 49 units due to computational and practical constraints. **b** presents simulations of square lattice configurations with an increasing number of units for 360 N_{LC}. The distributions present the $\overline{V}(I)$ of the last n_{LC} each for 100 simulations. The color indicates the ΔS used during the simulations. First, **b** demonstrates that the stepsize ΔS has a significant impact on the scalability of the system. We see that although both for a ΔS of 0.1 and for a ΔS of 0.05 the lattice configurations perform phototaxis in a system with up to 49 units. However, we find that for the ΔS of 0.1, the $\overline{V}(I)$ declines more quickly with increasing system sizes compared to the ΔS of 0.05. When evaluating the time evolution of $\overline{V}(I)$ for the 7x7 (49) units lattice in **c**. We find that the ΔS of 0.1 learns more quickly. However, **c** also shows that around $n_{LC} = 80$ the smaller ΔS of 0.05 starts outperforming the ΔS of 0.1. From $n_{LC} = 100$ on-wards the ΔS = 0.1 simulations stop improving while the smaller ΔS simulations keep on improving there $\overline{V}(I)$. These results could indicate that the search space of the physical system becomes more complex as the number of units increases. Therefore, the larger ΔS simulations are less able to narrow down on the local optima compared to the smaller ΔS simulations. **d** presents different simulations of a lattice of 5x5 (25) units for 360 N_{LC}. During these simulations $N_{inactive}$ (presented on the x-axis) units are picked randomly at the start of the simulation. These units are "broken" by turning their active diagonal springs into passive springs. Therefore, these units remain passive and do not actuate. Each distribution depicts the $\overline{V}(I)$ of the final n_{LC} over 10 simulations.

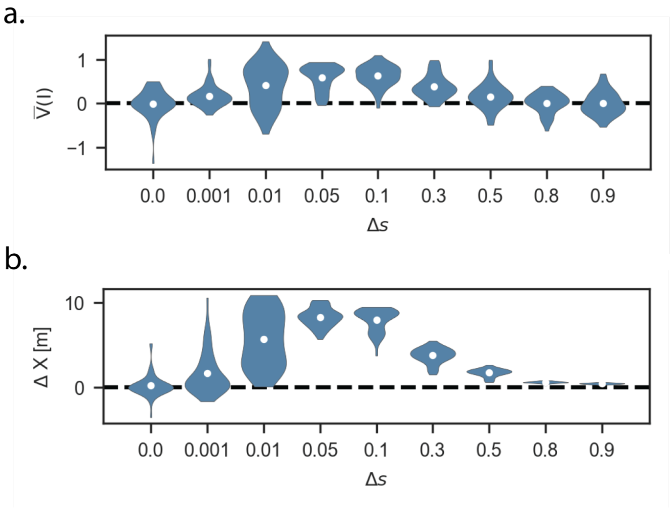


Fig. S3.7: **Additional information on the performance of the system for different** ΔS **values. a**, Score distributions of the triangle shape simulations corresponding to the results of Fig. 3.6j, measured in $\overline{V}(I)$ [mm/LC]. **b**, Total displacement of the system for the simulations in Fig. 3.6j at $n_{LC} = 1000$.

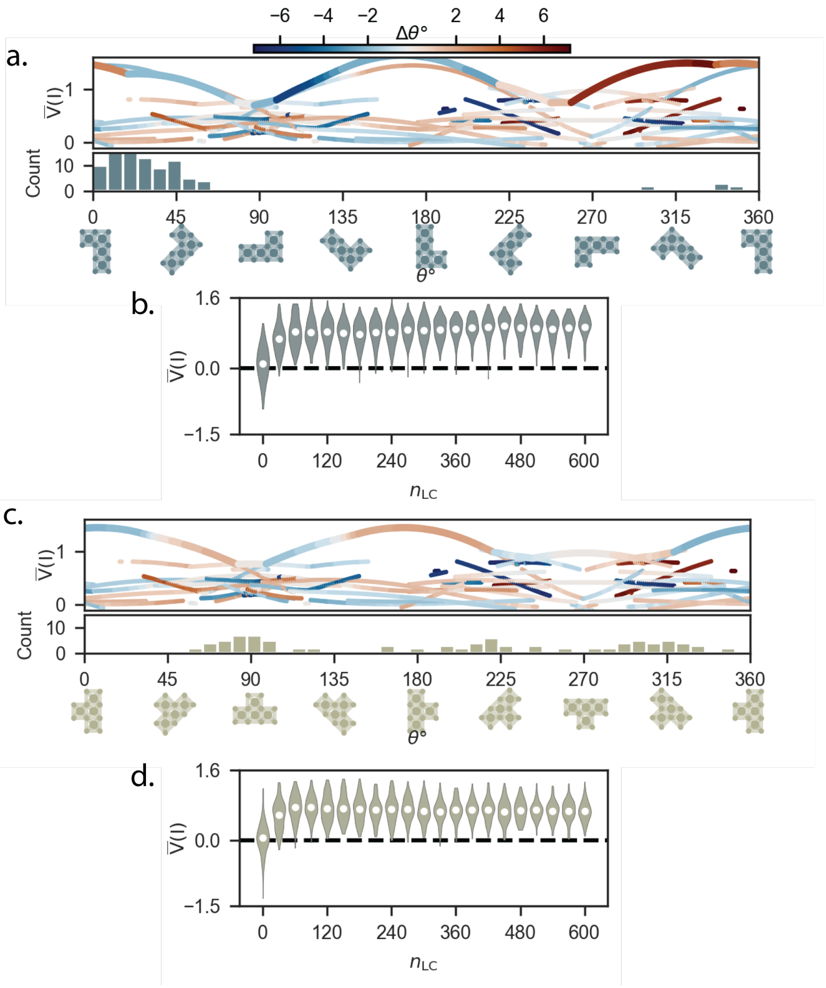


Fig. S3.8: **Simulation data of the projections for two additional geometries.**Projection of the global and local optima for each orientation of (**a**) the L-shape and (**c**) the T-shape assembly. The color of the line corresponds to the change in rotation Δθ for that respective phase combination. The orientation distribution of 100 simulations at $n_{LC} = 3000$ is depicted underneath. The score distributions for $\overline{V}(I)$ [mm/LC] of (**b**) the L-shape and (**d**) the T-shape assembly for every 30th learning cycle.

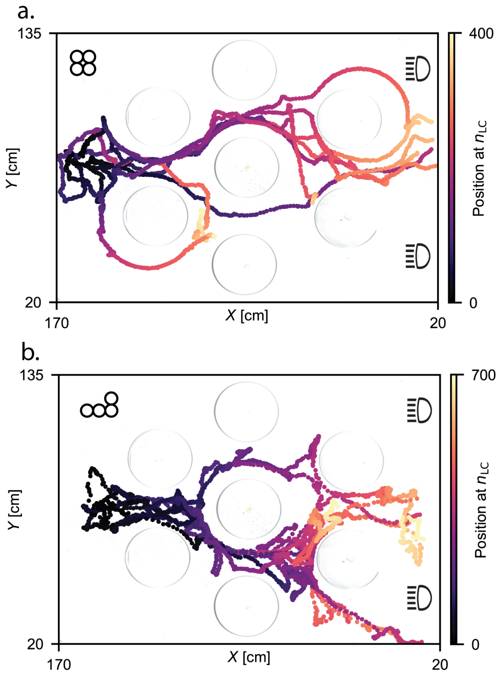
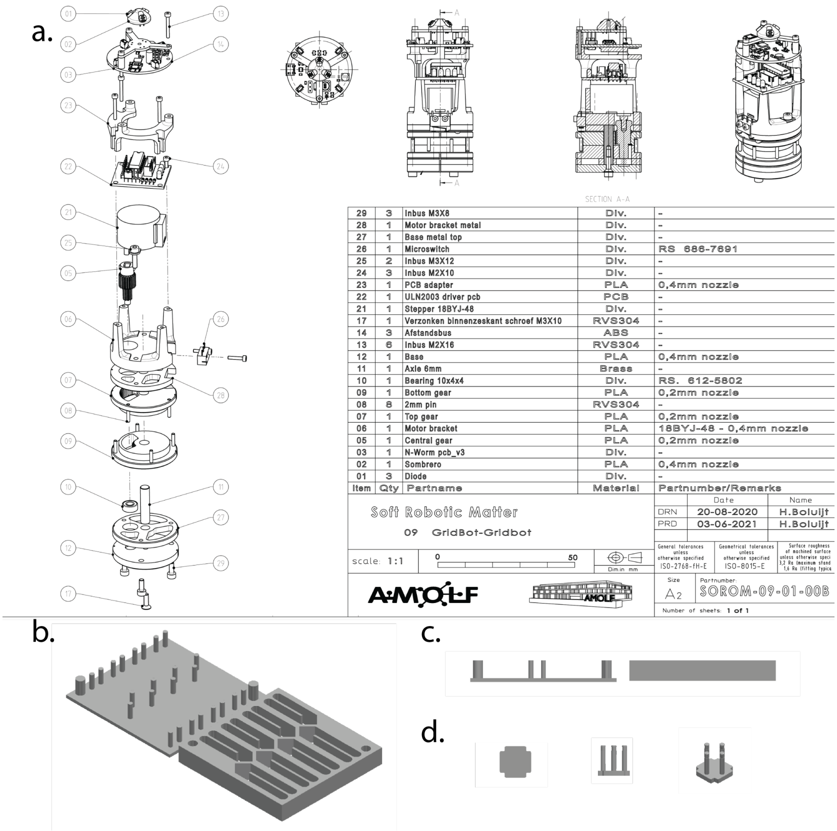


Fig. S3.9: **Obstacle experiments trajectories a**, Trajectories during ten experiments of the square geometry with circular obstacles as presented in Fig. 3.9a. **b**, Trajectories during ten experiments of the L-shape geometry with obstacles as presented in Fig. 3.9c.



Item	Qty	Partname	Material	Partnumber/Remarks
29	3	Inbus M3X8	Div.	-
28	1	Motor bracket metal	Div.	-
27	1	Base metal top	Div.	-
26	1	Microswitch	Div.	RS 686-7691
25	2	Inbus M3X12	Div.	-
24	3	Inbus M2X10	Div.	-
23	1	PCB adapter	PLA	0,4mm nozzle
22	1	ULN2003 driver pcb	PCB	-
21	1	Stepper 18BYJ-48	Div.	-
17	1	Verzonken binnenzeskant schroef M3X10	RVS304	-
14	3	Afstandsbus	ABS	-
13	6	Inbus M2X16	RVS304	-
12	1	Base	PLA	0,4mm nozzle
11	1	Axle 6mm	Brass	-
10	1	Bearing 10x4x4	Div.	RS. 612-5802
09	1	Bottom gear	PLA	0,2mm nozzle
08	8	2mm pin	RVS304	-
07	1	Top gear	PLA	0,2mm nozzle
06	1	Motor bracket	PLA	18BYJ-48 - 0,4mm nozzle
05	1	Central gear	PLA	0,2mm nozzle
03	1	N-Worm pcb_v3	Div.	-
02	1	Sombrero	PLA	0,4mm nozzle
01	3	Diode	Div.	-

Soft Robotic Matter

09 GridBot-Gridbot

	Date	Name
DRN	20-08-2020	H.Boluijt
PRD	03-06-2021	H.Boluijt

scale: 1:1

0 50

Dim.in mm

General tolerances unless otherwise specified ISO-2768-fH-E

Geometrical tolerances unless otherwise specified ISO-8015-E

Surface roughness of machined surface unless otherwise speci 3,2 Ra (maximum stand 1,6 Ra (fitting typica

A·MOLF

AMOLF

Size A2

Partnumber: SOROM-09-01-00B

Number of sheets: 1 of 1

Fig. S3.10: **Unit manufacturing. a,**Exploded view of the unit components and assembly, listing the parts for the units and showing a top view, side view, and cross-section of the units. **b,** three-dimensional view of the mold used to case the soft connection arms, printed with PLA. **c,** side view of the connection arms mold. **d,** top view, side view, and three-dimensional view of the PLA connection pins that are used the connect the soft arms of multiple units.

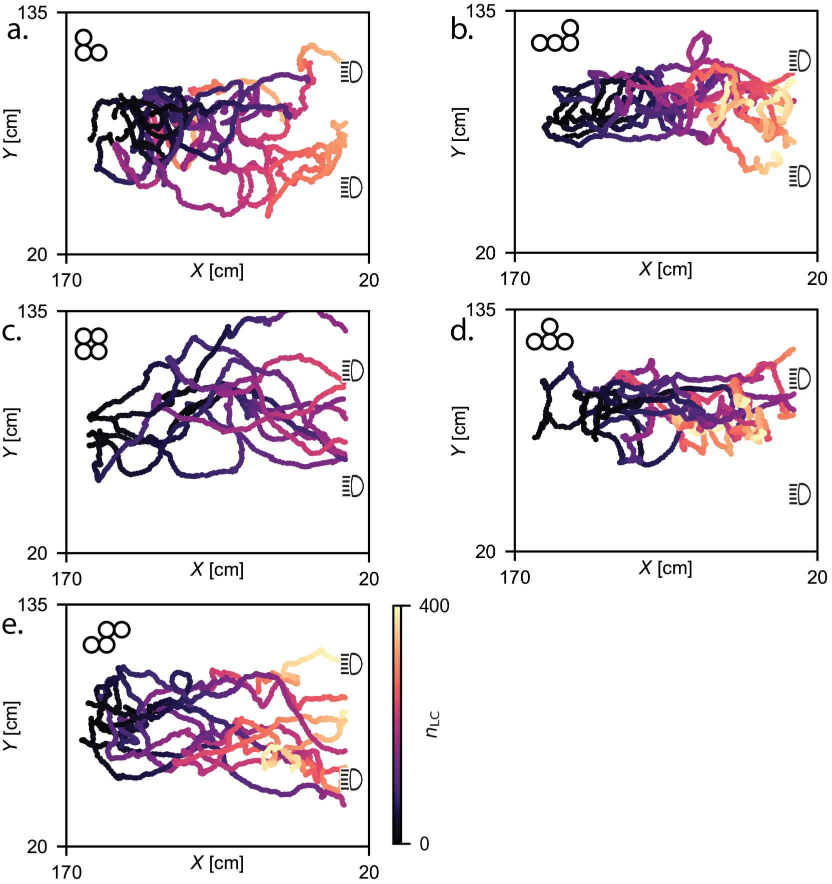


Fig. S3.11: **Overview of experimental results for all the tested geometries. a-e,** Trajectories of the experimental results for all five of the tested configurations. For each of the tested configurations, we show ten trajectories. The geometry of the configuration is depicted in the upper left corner of the canvas. The results are shown for either 400 learning cycles, or the number of learning cycles until the end of the canvas is reached.

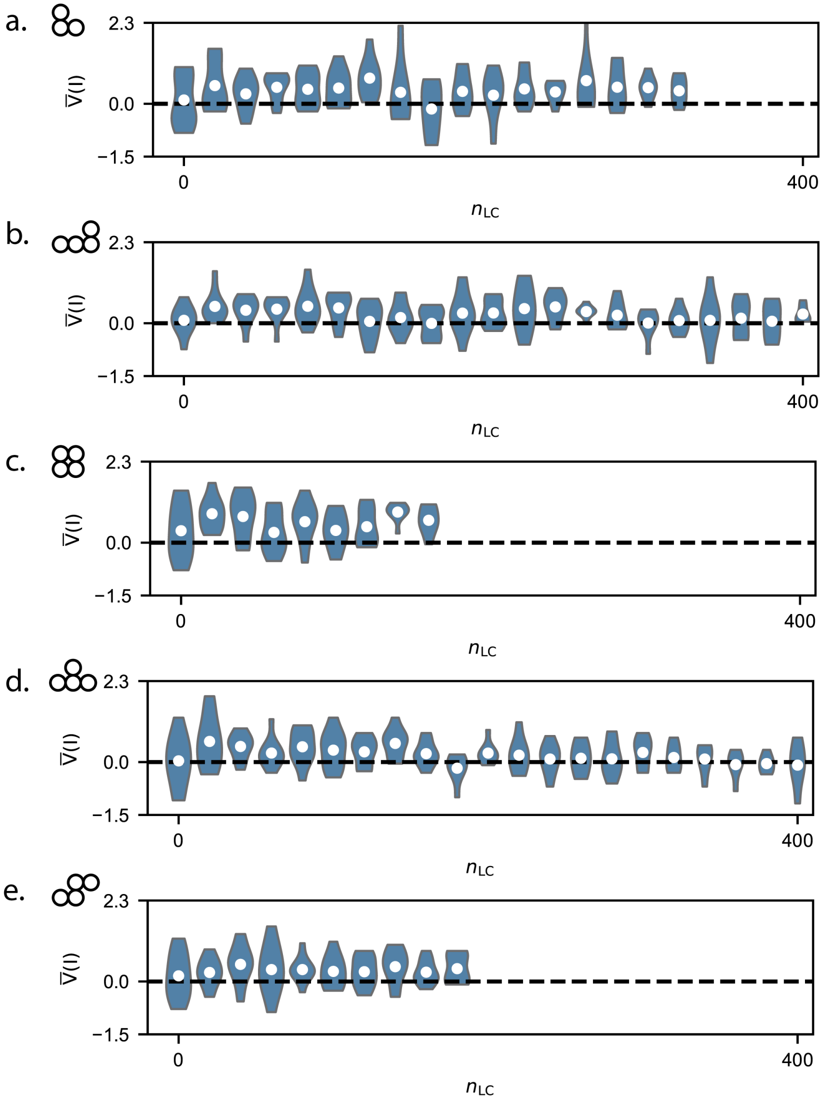


Fig. S3.12: **Overview of experimental results for all the tested geometries. a-e** Present the ΔI for the performed experiments, where the distributions consist of the $\overline{V}(I)$ [cm/LC] of the ten experiments for every learning cycle.

K_s	Spring constant	100
A	Adjacency	binary
c	Damping coefficient	-
$l_{0_{\text{border}}}$	Equilibrium spring length border elements	5 cm
$l_{0_{\text{diagonal}}}$	Equilibrium spring length diagonal elements	$\sqrt{2}\times$ 5 cm

Table S3.2: Model parameters used to model the individual springs.

T_c		Time of a single actuation cycle	2s
f		Frequency of the sin	0.85
T_m	$\frac{1}{f}$	Total movement time	
T_s	$T_c - T_m$	Total static time	
α		Time in extension	0.3
D_1	$\max\left((T_a \cdot \alpha) - \frac{T_m}{2}, 0\right)$	Waiting time at fully extended pose	
D_2	$T_s - D_1$	Waiting time at fully contracted pose	
ω		Stroke of the units	3.5 mm
l_a		Additional length of the diagonal springs	

Table S3.3: Model parameters used to model the actuation of the units.

FLUIDIC OSCILLATORS AS NEURONS FOR SOFT ROBOTICS

4.

Abstract. Soft robotic systems benefit from electronics-free control strategies that integrate actuation and coordination directly into their physical structure. While fluidic oscillators have enabled decentralized and repetitive motion patterns, their sequences have thus far been limited to fixed traveling waves, restricting adaptability. In this work, we introduce a tunable fluidic relaxation oscillator and a bio-inspired coupling framework that draws from central pattern generators (CPGs) in nature. We demonstrate that the oscillator can synchronize with external control signals and, when coupled, can achieve robust phase locking, including both in-phase and anti-phase coordination. This electronics-free approach enables controllable, reprogrammable patterns that are resilient to fabrication imperfections and scalable through localized interactions powered by a single, constant pressure source. By bridging the gap between fluidic circuits and adaptive CPG-like architectures, our findings pave the way for soft robots capable of richer, more flexible, and autonomous locomotion, while remaining simple, robust, and inexpensive to manufacture.

H.A.H. Schomaker*, L. Huntjens*, J.T.B. Overvelde. "Fluidic oscillators as neurons for soft robots." *in preparation*.

4.1 INTRODUCTION

Soft robots exhibit unique intrinsic adaptability and compliance, making them suited for interactions within uncertain and dynamic environments [111]. Practical implementations of soft robots have already found applications as end effectors in robotic arms for tasks such as pick-and-place operations [113], medical rehabilitation [100], and even as implants [93]. Soft robotic systems have also demonstrated effective locomotion, navigating unpredictable terrain with less control complexity compared to traditional rigid robots [111, 126]. Nonetheless, fully autonomous and mobile soft robots remain challenging to realize [65].

One critical barrier hindering autonomous soft robots is the complexity associated with controlling multiple actuators simultaneously [63]. Typically, each actuator demands an individual electronic pump or control channel, complicating the design and reducing scalability [26, 130]. To simplify these systems, recent developments have introduced soft circuits, notably fluidic systems, which minimize or eliminate electronic components by embedding the control directly into the robotic structure [79, 121, 131].

These electronics-free approaches have advantages over electronic circuits for soft robot integration. Firstly, they reduce the number of components, as there are no transitions from electronics to the actuation principle (e.g., fluidics) [72]. Making them cheaper and potentially more reliable due to a decrease in the possible failure points. Secondly, unlike electronics, they are unaffected by radiation, making them suitable for harsh environments where regular electronics fail [114]. Thirdly, they often combine power sensing and actuation in the same circuitry (e.g., the powerline is also the control line); because of this reason, the electronic-free circuits are commonly distributed and integrated throughout the body, making them highly decentralized [72], heaping the benefits of decentralized systems like modularity, robustness to external perturbations and damage [72].

Such fluidic oscillators frequently rely on oscillators to achieve the orchestrated, repetitive -yet adaptive- actuation patterns essential for completing tasks. Existing electronics-free soft robotics have demonstrated control for a limited number of actuators (e.g., using ring oscillators). The sequence of the control circuit has, thus far, always generated a traveling wave (actuating one after the other)[20, 63, 67, 101]. These can be very useful for peristaltic motion, and with innovative circuit design around the oscillators, they can also be transformed into gaits for limbed locomotion [20, 63, 101]. By building additional circuitry around the oscillators that rewire the oscillators to different outputs, soft robots can even make a one-time switch in their behavior[26]. The one drawback of the systems using ring oscillators is that the sequence and phase shifts of the oscillators are intrinsically fixed to a traveling wave in one direction. Research has also been done on soft relaxation oscillators, which, in contrast to ring oscillators, can oscillate without the need for a circuit [63]. They showed that in their circuit, the sequence of the oscillators can be switched by relaying on accurately timed inputs to switch the order [63]. While fluidic relaxation oscillators have successfully embedded repetitive and useful sequences without electronics, they still exhibit limited flexibility in the timing of actuation and switching between predetermined actuation patterns [63, 102]. Coupling the relaxation oscillators through the environment can also lead to more adaptive sequences [19]. However, in this case, the possible sequences are highly limited by body morphology and hard to scale to higher numbers of oscillators.

Nature provides valuable insights into achieving adaptive and controllable sequences through central pattern generators (CPGs), biological neural circuits that orchestrate rhythmic and adaptable sequences of muscle activation during locomotion [52]. These CPGs are key in the control of animal locomotion, providing a feedforward signal for robust and rhythmic muscle activation, which allows for a fast dynamics synchronization of limbs [27]. By providing a physical coupling between the different muscles and the body, animals can partially offload efforts of the central brain to the nervous system [18, 52]. Control strategies inspired by neuronal oscillators, as seen in animal locomotion, have the potential to be more robust, efficient, and adaptable [52]. These neuronal oscillators not only produce robust and predictable sequences but can also dynamically adapt the behavior to environmental feedback [27, 60]. The integration of CPG-inspired principles into soft robotics could, therefore, significantly enhance adaptability and reduce the complexity of controlling.

Research in biomechanics and robotics has demonstrated that coupled neuronal oscillators can switch between different locomotion modes, adapting robustly to changes in environmental conditions [54, 90, 125].

Translating such bio-inspired coupling strategies into fluidic soft robots represents an important step toward enhancing their autonomous capabilities.

This work aims to bridge the gap between soft robotic actuation principles and bio-inspired neuronal oscillator systems, focusing on improving physical control strategies for adaptive and reprogrammable sequences. We introduce a fluidic relaxation oscillator that is, by design, straightforward to adapt to fit the demands of the soft robotic actuators. Next, we propose an implementation to couple these oscillators in predetermined sequences. To this end, we first demonstrate its ability to follow and synchronize with a control signal. Finally, we present the coupling between two oscillators that allows for switching between multiple patterns, including in-phase and anti-phase synchronization. This work sets the stage for implementing this approach in bio-inspired soft robots with CPG-like electronics-free controllers capable of adaptive and controllable locomotion.

4.2 FLUIDIC RELAXATION OSCILLATOR DESIGN

The design of the fluidic relaxation oscillator that mimics the neuron's behavior consists of a soft silicone membrane, two permanent magnets, and a 3D-printed housing to enclose the membrane (Fig. 4.1a-c). The oscillator can be in two states that alternate to self-sustain an oscillatory output for a constant input flow. In the first state, referred to as the 'closed state,' the constant input flow (Q_{in}) at the inlet slowly builds up pressure, which acts on the area (A_1) of the inlet depicted (Fig. 4.1d-1 in orange) to generate an upward force on the membrane (F_{p1}). This upward force on the membrane is counteracted by the force of the magnets on the membrane (F_m) that creates an airtight seal around the inlet of the oscillator (Fig. 4.1d-1). As long as $F_m > F_{p1}$, the oscillator remains closed, but when $F_m < F_{p1}$ at the opening pressure (P_{open}), the membrane snaps to the 'open state' (Fig. 4.1d-2) where the membrane is lifted.

Two factors contribute to the system's bistability within the oscillator's operating range. Firstly, the attractive force of the magnets F_m has an inverse-squared relation to distance, resulting in a rapid decrease of F_m as the magnet is lifted. Secondly, as soon as the membrane is lifted

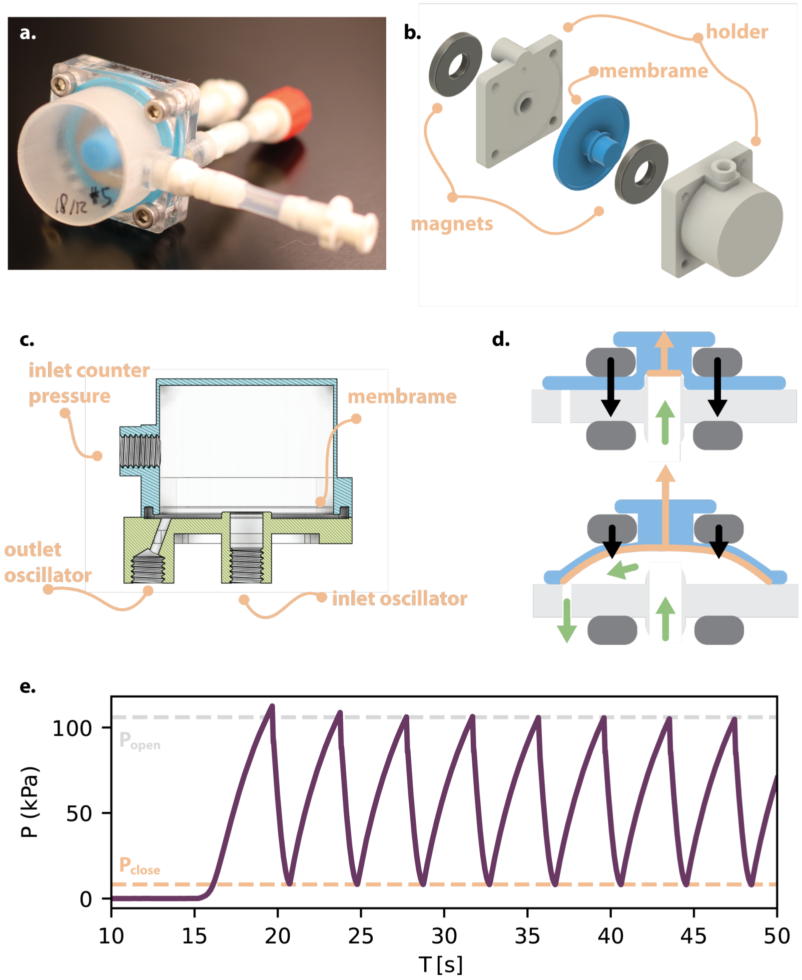
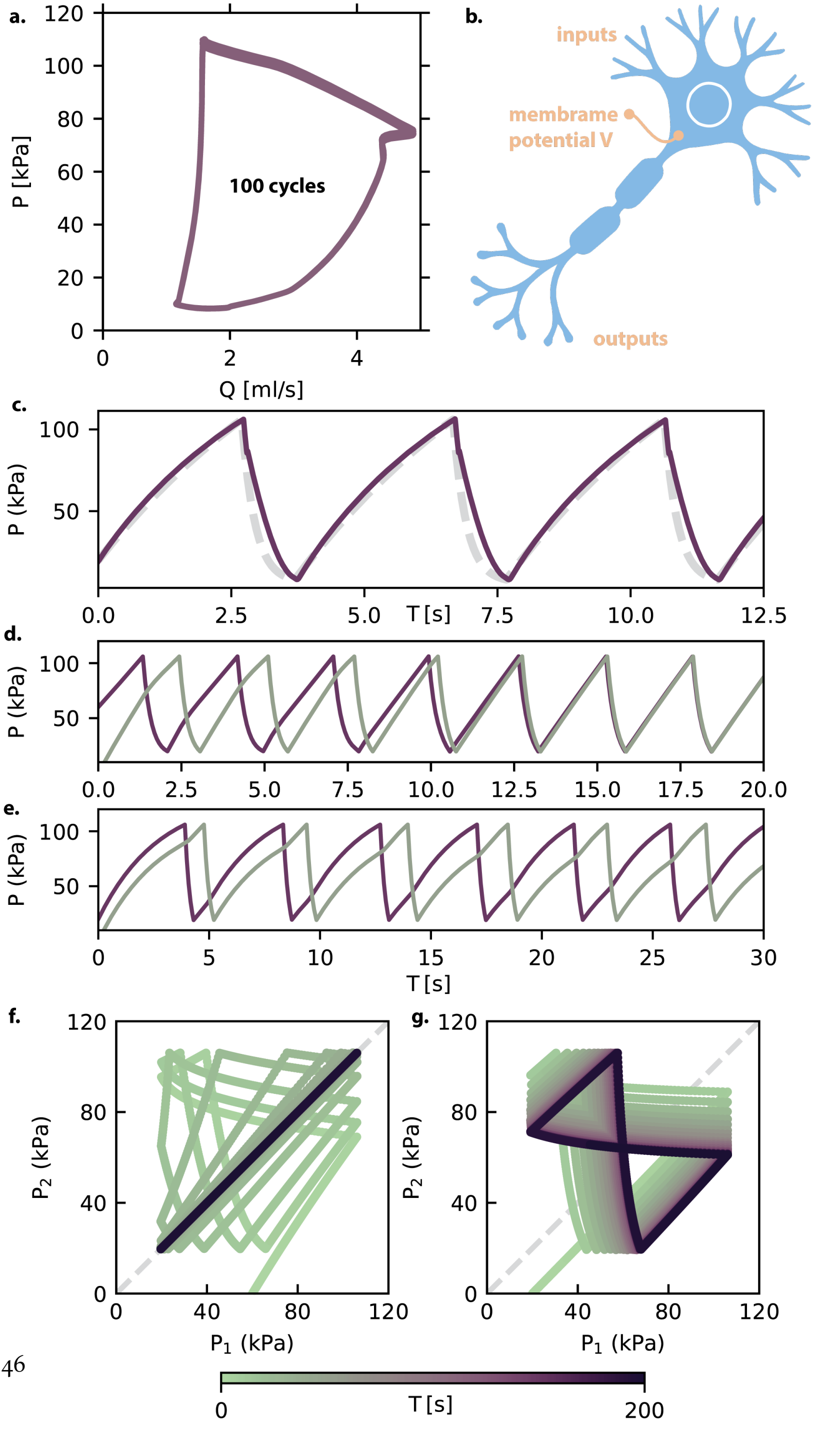


Fig. 4.1: **Fluidic relaxation oscillator design.** **a**, Photo of a single fluidic oscillator. **b**, Exploded view of the oscillator's components. **c**, Schematic demonstrating the inlets and outlets of the holder. **d**, Schematic representation of the two stages of the oscillator with the magnetic (dark gray) forces (black), the area on which the air is pressing (orange), along with a force vector (orange), and the trajectory of the incoming air (green). **e**, Oscillating pressure at the inlet for a constant input flow $Q = 1.8$ and a capacitance parallel to the oscillator's input of 45 ml.

slightly, the pressure P_{open} pressurizes the chamber between the in- and outlet and acts on a larger area (A_2) depicted in Fig. 4.1d-2 in orange, which increases the upward force F_{p2} on the membrane. The interplay between these forces results in a rapid transition from the closed to the open state of the system when the critical limit is reached. The result is a hysteresis loop in the pressure-flow behavior of the oscillator.

In the open state, the air can flow freely to the outlet of the oscillator, thereby continuously reducing the pressure at the inlet. Importantly, the increase of the area between A_1 and A_2 results in an increase of F_p, which means that the oscillator remains open for much lower pressures. When $F_m > F_{p2}$ at (P_{close}), the force of the magnet is able to pull down on the membrane, increasing its force as the distance decreases and closing the seal at the inlet. This resets the system to its initial configuration, where the process starts over. The hysteresis between the opening and the closing of the membrane is what drives the oscillations of the system and prevents an intermediate state where the membrane is partially open and slowly leaking air. The time evolution of the inlet pressure (P_{in}) for a flow rate of 1.8 SLPM (standard liters per minute, a measure of the volumetric flow rate of gas under standard conditions of temperature and pressure) is depicted in Fig. 4.1e and Movie 4.1 for $Q_{in} = 1.8$ SLPM. At $T = 15s$, the flow regulator starts with the constant flow input. From that moment, we see the pressure building up at the oscillator inlet. At 18 seconds, the membrane opens, and the pressure starts to decrease as the air escapes through the outlet, up to P_{close} where the membrane snaps back, and the cycle repeats. Interestingly, we see that in the initial three cycles, P_{open} decreases until it settles at a stable value of 108 kPa. It is likely that in the initial transient cycles, the membrane is still shifting around until it stabilizes its closed position, while there could also be some limited softening of the material often observed for large deformations.



Caption on next page.

Fig. 4.2: **Fluidic oscillator model. a**, Pressure at the inlet vs. the flow at the inlet during 100 oscillation cycles. **b**, Schematic representation of a neuronal cell. **c**, Fit of the adapted LIF model (gray dashed line) and the experimental data (purple). **d**, In-phase synchronization of two oscillators in the model with mutual additive coupling $k = 0.08$. **e**, Out-of-phase synchronization of the same oscillators for mutual additive coupling $k = -0.08$. **f-g**, Limit cycle colored by time for the positive and negative coupling of **d** and **e**, respectively. The $x = y$ diagonal is presented with a dashed gray line.

4.3 MODELING THE FLUIDIC OSCILLATOR AS A NEURON

Single neuron

In Fig. 4.2a, we observe that for over 100 cycles the behavior of the fluidic oscillator remains remarkably consistent. Driven by the build-up and abrupt release of pressure across a membrane valve, these systems can be effectively described using threshold-reset models reminiscent of neuronal dynamics. These natural cells (depicted in Fig. 4.2b) are also relaxation oscillators that function by electrical and chemical signaling. The dendrites close to the cell's nucleus sense the surrounding electrical potential, and when the potential surpasses a specific threshold, it "fires," during which the voltage drops and the process starts anew. Here, we adapt the leaky integrate-and-fire (LIF) neuron model to capture the essential features of a fluidic relaxation oscillator, incorporating a smooth recovery phase that plays a pivotal role in orchestrating phases between coupled oscillators.
The classic LIF model ([116]) describes the sub-threshold dynamics of a membrane potential V by:

$$\tau\frac{dV}{dt} = -(V - V_{rest}) + R_m I,$$

where τ denotes the membrane time constant, V_{rest} is the resting potential, R_m is the membrane resistance, and I is the input current. Once the potential reaches a threshold V_{thres}, it is reset to a lower value V_{reset} and the cycle starts again. While the LIF model captures integrate-and-fire behavior, it lacks a realistic description of the repolarization (recovery) process following threshold crossing.

In our adaptation, we treat the membrane pressure in a fluidic oscillator as analogous to the membrane potential in excitable systems. Air continuously flows into the oscillator, building up pressure until the membrane opens once the threshold pressure is reached. This opening slowly releases the pressure until it drops below a reset target, closing the membrane and restarting the cycle.

We, therefore, reformulate the LIF model as follows:

$$\tau\frac{dP}{dt} = -(P - P_{\mathtt{rest}}) + RQ_{\mathtt{in}},$$

where P is the pressure at the inlet of the oscillator (analogous to membrane voltage), Q_{in} is the input flow (analogous to input current), P_{rest} sets the amount of leakage (analogous to V_{rest}), R (analogous to membrane resistance), P_{open} is the pressure at which the membrane opens (analogous to V_{thres}), and P_{close} is the pressure to which it resets when the membrane opens (analogous to V_{reset}). The term $-(P - P_{\mathtt{rest}})$ most closely resembles leakage of the air over time through a resistor. In our current coarse-grained model, however, this term, together with R and τ, also represents the compression of the air in the combined volume up to the oscillator's inlet (connection tubes, potential actuator, or capacitor).

To capture the membrane's abrupt opening and smooth repolarization, we introduce a "recovery phase" immediately after threshold crossing. During recovery, the system experiences a leak. During the recovery phase (when $P \geqslant P_{\text{open}}$ and *recovering* flag is `True`), the dynamics for each oscillator are given by:

$$\frac{dP}{dt} = \frac{-(P - P_{\text{rest}}) + R\,Q_{\text{in}}}{\tau} - \big(\gamma(P - P_{\text{rest}})\big),$$

Where γ is a recovery-phase leak coefficient. Recovery continues until the pressure P drops below P_{close}. This dynamic replicates the release of pressure followed by the rapid closure of the membrane.

We simulate the system using Euler forward integration with a time step $\Delta t = 0.001$ s. The fit of the model with the experimental data is presented in Fig. 4.2c for the parameters in Table 4.1. The fit is made by setting $P_{\text{close}}, P_{\text{open}}$ and Q_{in} to the values found in the experimental data.

The other parameters are optimized using the least squares method. These fitted parameters are used throughout this chapter unless stated otherwise.

Table 4.1: Model parameters

Parameter	**Value**
τ	100
P_{rest}	$-30\,\text{kPa}$
P_{start}	$0\,\text{kPa}$
P_{open}	$116\,\text{kPa}$
P_{close}	$8\,\text{kPa}$
R	200
Q_{in}	$2\,\text{SLPM}$
γ	1.9

The recovery phase is crucial for coordinating the phases of multiple coupled oscillators. The standard LIF model has a Type 1 PRC (Phase Response Curve), which means that all inputs result in a forward phase shift [61]. With the addition of the recovery phase, where inputs have a negative influence on the phase, the oscillator becomes more similar to continuous oscillators like FHN, known for their Type 2 PRC. This biphasic PRC in the current oscillator promotes not only in-phase synchrony but also stable phase lags, characteristic of Type II oscillators [92].

Consequently, our fluidic oscillator model transcends the purely spiking behavior of LIF neurons and approaches the continuous phase dynamics observed in nonlinear oscillators while maintaining ease of use with fixed opening and closing pressures.

Coupling between neurons

To investigate the coupling, we opt for additive coupling, as it is widely used in CPGs. For the current oscillator, this implementation means that pressure at the inlet of one oscillator couples to the inflow of another oscillator. We extend the model to two mutually coupled fluidic oscillators:

$$\tau\frac{dP_1}{dt} = -(P_1 - P_{\text{rest}}) + R(Q_{\text{in},\,1} + k_1 P_2),$$
$$\tau\frac{dP_2}{dt} = -(P_2 - P_{\text{rest}}) + R(Q_{\text{in},\,2} + k_2 P_1),$$

where k_1 and k_2 are coupling strengths. Each oscillator independently undergoes recovery dynamics upon threshold crossing:

$$\text{if } P_i \geqslant P_{\text{open}} : \text{ recovery activated until } P_i \leqslant P_{\text{close}}.$$

The recovery phase in each oscillator serves as a phase "reset," facilitating robust synchronization or stable phase differences in the presence of coupling. Fig. 4.2d demonstrates the effect of mutual positive coupling, where two identical oscillators start out of phase with mutual positive coupling of a somewhat arbitrarily chosen value $k = 0.08$. We find in-phase synchronization within the first five oscillations, after which the phases are locked. In contrast, for mutual negative coupling presented in Fig. 4.2e, we find that it takes longer for the phases to stabilize, roughly 23 cycles. Fig. 4.2f shows the limit cycle for the positive coupling presented in Fig. 4.2d, where we see that indeed the phases are locked in clear in-phase synchronization, as seen from the linear line following the xy-axis. In Fig. 4.2g, we observe the pressures over time for the negative coupling. Also, in this case, we find a transient followed by a stable limit cycle. The limit cycle demonstrates a clear negative coupling, which is asymmetric on $x = y$.

This adapted LIF framework, with its recovery phase and associated negative PRC dynamics, provides a robust and computationally efficient model for the fluidic relaxation oscillators. Including recovery dynamics bridges the gap between purely spiking neuron models and continuous nonlinear oscillators, enabling the exploration of rich collective behaviors such as phase synchronization, phase lags, and noise resilience in networks of fluidic oscillators and bringing the oscillator closer to the framework of CPGs. With this framework, we can start to use the experimental oscillators in networks to drive the rhythmic actuations of pneumatic soft actuators. These results demonstrate that if we can implement a the directed positive and negative coupling, we are able to program the actuation rhythms for the experimental oscillators.

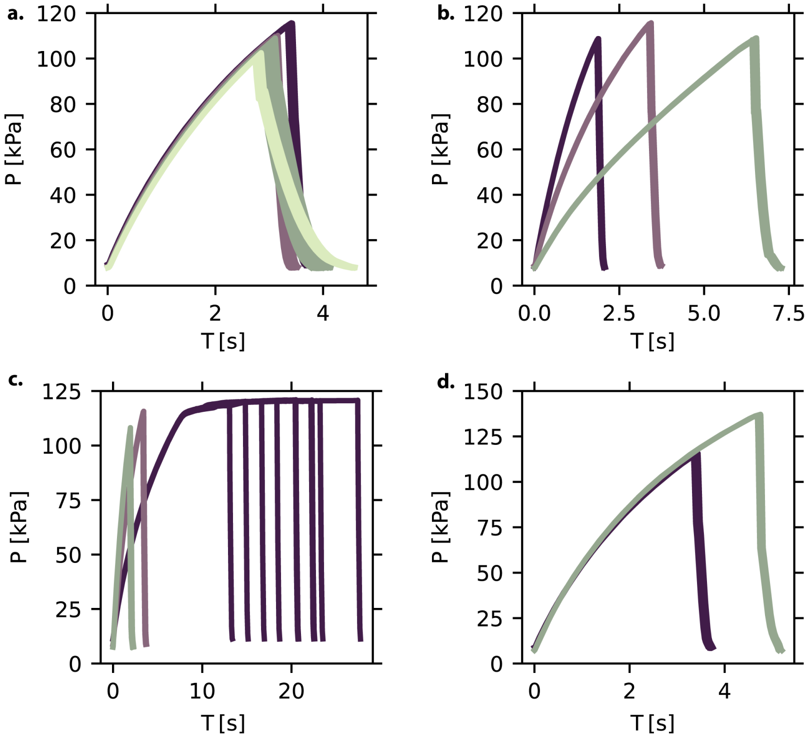
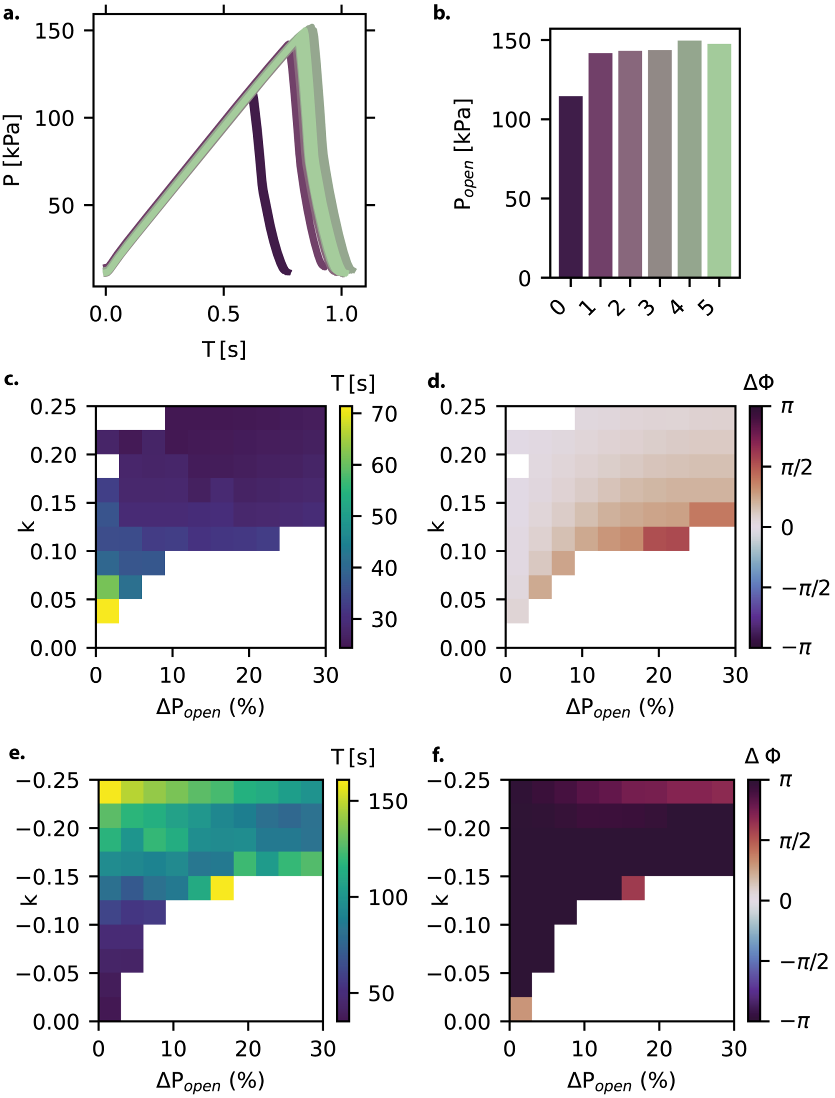


Fig. 4.3: **Characterization of a single oscillator a**, Influence of the resistance at the outlet for 13 mm needles with diameters of 0.61, 0.84, 1.60 mm, colored light green to purple, respectively. For all the results in the current figure, each color represents roughly 50 cycles overlaid by the start and end of the cycle. **b**, Influence of the capacitor parallel to the oscillator inlet for 15 ml (dark purple), 45 ml (light purple), and 170 ml (green). **c**, Influence of the inflow set by the flow regulator for Q_{in} equal to 0.9 SLPM (dark purple), 1.8 SLPM (light purple), and 2.7 SLPM (green). **d**, Effect of the thickness of the membrane for a thickness of 0.8 mm (purple) and 1.1 mm (green).

4.4 OSCILLATOR CHARACTERIZATION

Having demonstrated the potential of the fluidic oscillators to generate phase-locking and synchronization, we next characterize the oscillator and show the adaptability of the fluidic oscillator's oscillations. Firstly, Fig. 4.3a demonstrates how we can tune the recovery phase of the oscillator by changing the resistance at the outlet. By increasing the outlet resistance, we increase the recovery phase, as the air escapes the oscillator less quickly. Secondly, Fig. 4.3b demonstrates that we can tune the duration of the loading phase of the oscillator by changing the capacitance at the inlet of the oscillator. Note that driving a soft actuator with the oscillator will have a similar effect, as this will act as a volume of air placed at the inlet of the oscillator. Hence, using pneumatic actuators with more internal volume at inflation will result in a more extended loading phase of the oscillator. These two variables can be tuned together to obtain the desired PRC for the actuator-oscillator combination.

Fig. 4.3c demonstrates the influence of the constant flow input. We find that for $Q_{in} = 0.9$, the oscillator stalls close to the P_{open}, where it seems to find an intermediate state. Although the oscillator still oscillates, the frequency F becomes erratic. This is an undesired behavior of the oscillator, and as such, when designing the networks of the oscillators, it should be taken into account that we provide enough input flow to the oscillator to obtain stable oscillation cycles. Lastly, in Fig. 4.3d, we demonstrate how tuning the stiffness of the membrane can directly control the P_{open} of the oscillator. This is a useful property because it enables us to adapt the oscillator to the required pressures of any pneumatic actuator one would like to connect to the oscillator. Furthermore, the model can be used to guide the experimental realization of the coupled networks.



Caption on next page.

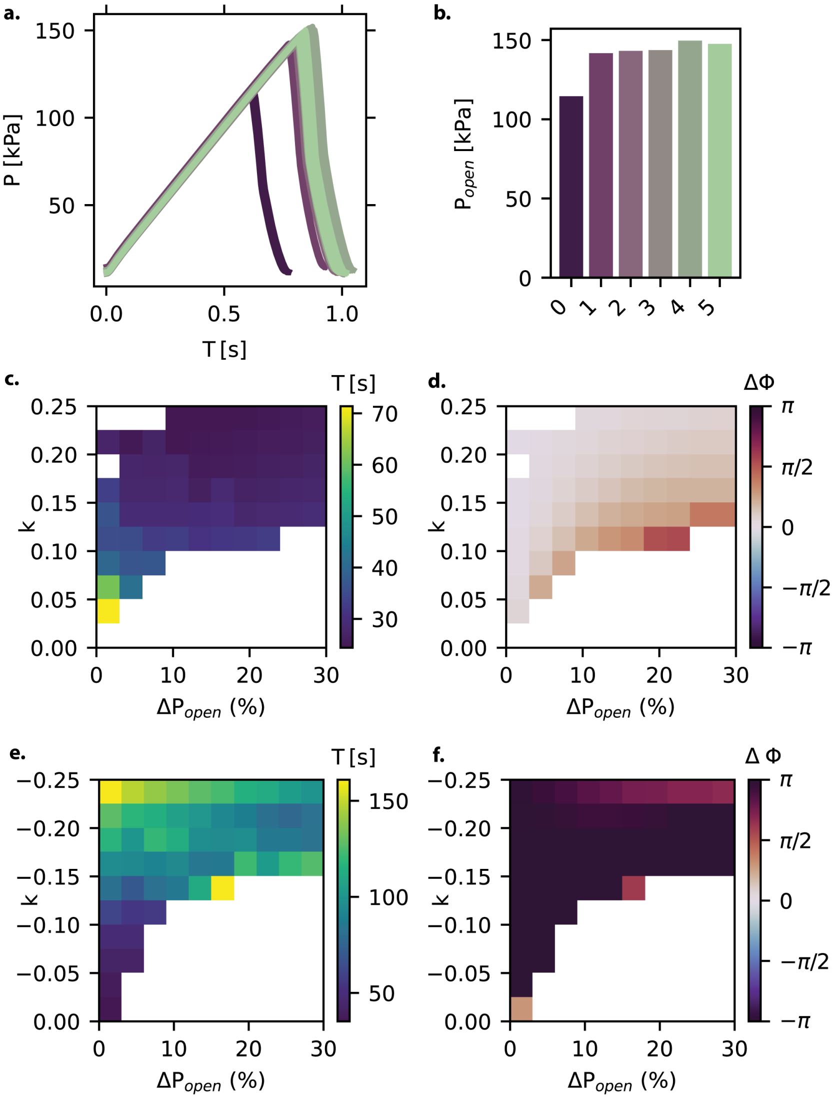
Fig. 4.4: **Characterization of a single oscillator a**, 100 oscillation cycles for five different oscillators with the same manufacturing parameters, attached directly to the flow regulator at $Q_{in} = 1.8$ SLPM. **b**, Average opening pressures for the 5 oscillators presented in **a**. **c**, Time it takes for the phases to lock for simulations of 500 s for mutual positive coupling values. Measured as the first phase difference that is repeated for 10 cycles within a five percent error margin from the first measure. For the black entries in the heatmap, phase locking did not occur within the simulation time. **d**, Phase shifts for the same simulations as **c**, measured as the average phase shift over the last 10 cycles of the simulation if phase locking occurred. **e-f**, The same simulation results as **c-d** for negative mutual coupling values.

4.5 OSCILLATOR VARIABILITY

One of the main challenges of working with experimental oscillators in a network configuration is the variability and repeatability of the oscillator fabrication process. To test this, we fabricated five of the fluidic oscillators with identical parameters. Fig. 4.4a shows the oscillation cycle for each of these oscillators. The main variability that we find is the difference in P_{open} between the oscillators depicted in Fig. 4.4b.

To test the effect of this variability on the coupling of the oscillators, we turn to the model where we can more easily investigate the general effect of these differences in the oscillators. Firstly, we run the model with the same parameters as described earlier, varying both the coupling strength for mutual positive coupling and the difference in P_{open}. The difference in P_{open} is created by increasing the P_{open} for only one of the oscillators by a specific percentage. Fig. 4.4c presents the time it takes to lock the phases, measured by the first phase difference where the 10 following phase differences are the same within a five percent error margin. We find two interesting trends in Fig. 4.4c. Firstly, we find that, in general, both the higher ΔP_{open} and higher coupling result in faster phase locking. Secondly, we find that for higher ΔP_{open}, more coupling is needed to lock the phases, as seen by the blank areas at the bottom of the heatmap. In Fig. 4.4d, we find the $\Delta\phi$ after the phases have locked. Interestingly, we observe a similar trend where for higher ΔP_{open}, more mutual positive coupling is needed to drive the oscillators to in-phase synchronization. At the border between phase locking and no phase locking, we find that although the phases are locked, they lock into a phase shift somewhere between in-phase and anti-phase.

Next, we repeat the same simulations for mutual negative coupling in Fig. 4.4e-f. In Fig. 4.4e, we find an opposite effect of the coupling strength k on the lock time. Also, we do not find an apparent impact of ΔP_{open} on the time required to lock the phases. We do find a clearly similar trend where higher differences in the opening pressure require stronger mutual coupling to lock the phases. For the phase difference while locking, depicted in Fig. 4.4f, we find that for most of the combinations where phase locking occurs, the phase difference is anti-phase. In the top row, we see some deviation from the anti-phase synchronization, where the negative coupling is likely so strong that it is overpowering one of the oscillators. Furthermore, we find some phase shifts at the boundary between locking and not locking, but these are less apparent compared to the positive coupling. From these results, we conclude that the deviations that we are finding in the experimental fluidic oscillators are within the margin that can still be coupled successfully, assuming we can implement strong enough coupling.



Caption on next page.

Fig. 4.5: **Characterization of a single oscillator a**, Effect of the γ parameter in the oscillation curve in simulation. **b**, Phase shift for the different values of γ and k using the same criteria as presented in Fig. 4.4, starting with one starting pressure of zero and one of 30 kPa. **c**, Phase differences over time for three individual simulations presented in **b** by dots in their respective colors. **d-f**, Pressures over time when the phase difference has stabilized for the three simulations represented with the same colors in **b** and **c**.

4.6 IMPORTANCE OF THE RECOVERY VARIABLE

As described earlier, the recovery of the oscillator is hypothesized to play a crucial role in finding predictable phase locking for the coupled oscillators. To test this theory, we investigate the influence of the variable γ in the model that controls the duration of the recovery phase. Fig. 4.5a shows how increasing γ decreases the duration of the recovery phase of the oscillator. In Fig. 4.5b, we perform the same phase locking simulation as in Fig. 4.4c-f, but instead of changing the ΔP_{open}, we vary γ for both oscillators while keeping P_{open} the same. Fig. 4.5b shows that we find phase locking for all values of γ. However, we find a trend at the lower diagonal (higher γ and lower coupling) where the phase shift between the oscillators becomes more erratic. The colored dots represent three individual cases that are presented in Fig. 4.5c. Interestingly, from Fig. 4.5c, we see that all three lines first shift their relative phases to lock at a specific phase offset. Furthermore, although the line in dark gray for a γ of 2.5 is still close to anti-phase, the other two lock the phases at phase shifts somewhere between in- and anti-phase. The time evolution of the oscillators for the three simulations presented in Fig. 4.5c, after the phase difference has stabilized, can be found in Fig. 4.5d-f. We find that for Fig. 4.5f, the curve seems to be more affected due to the recovery phase effectively making the oscillator maintain higher pressures for longer.

Fig. 4.6a presents the same simulations for positive coupling, where we find a similar effect. For higher values of γ, there is a stronger coupling needed to obtain a predictable phase offset. Fig. 4.6b shows that for the purple line, the positive coupling actually increases $\Delta\phi$, moving it away from in-phase synchronization.

4.7 COUPLING THE EXPERIMENTAL FLUIDIC OSCILLATORS

For the coupling of the experimental oscillators, we aim for a framework that closely resembles the formulation of Kuramoto oscillators and CPGs with additive coupling. As with those oscillators, it is known that they can have rich phase locking with predictable patterns. There are three critical factors to consider when coupling the experimental fluidic oscillators in a similar fashion to additive coupling. Firstly, since we focus on additive coupling between the oscillators, there should be no flow interaction between the oscillators. Secondly, the coupling has to be nonreciprocal, meaning that pressure from the first oscillator has to have the same effect on the second oscillator, irrespective of the second oscillator's pressure. Thirdly, the coupling mechanism has to be simple enough to fabricate many of them, as the maximum number of couplings that can be used in the network scales with N^2. Taking these three design restrictions into account, we opt for the fluidic circuit presented in Fig. 4.7a, where the air coming from the wall pressure can flow in two directions: either past a resistance to the atmosphere or through another resistance toward the oscillator. By placing two normally open and continuously closing fluidic transistors (analogous to depletion-mode MOSFETs), we can apply a pressure (voltage) to the transistors to steer the flow towards (pressure on the blue transistor) or away from (pressure on the orange transistor) the oscillator. This creates a form of non-reciprocal additive coupling, which can be both positive and negative in nature.

The design of the coupling mechanism (fluidic transistor) consists of a larger flexible silicone tube (outer diameter 3.6 mm and inner diameter 3 mm), a smaller stiff silicone tube (outer diameter 1.8 mm and inner diameter 1 mm), and a thin PVA plate (1 mm thick) sandwiched between the tubes (Fig. 4.7b-c). Two thick 3D-printed plates encapsulate the structure to keep it in place and confine the space for the tubes. The pressure in the larger tube drives coupling. The smaller tube acts as the driveline. The PVA interface plate serves as the weight (k) that determines how much force the thick control tube can exert on the

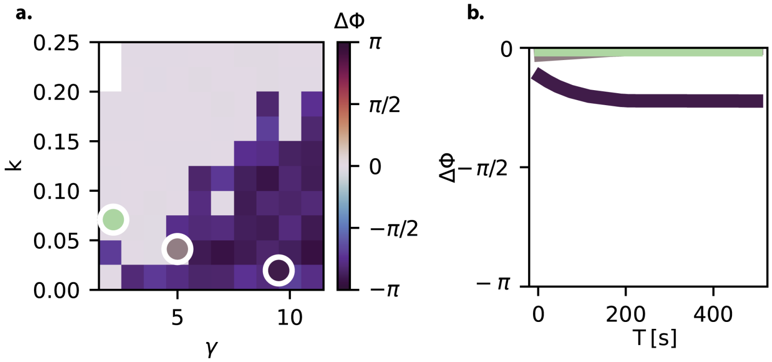


Fig. 4.6: **Characterization of a single oscillator a**, The same simulation results as Fig. 4.5b for positive values of k, starting with one starting pressure of zero and one of 60 kPa. **b**, Phase over time for three individual simulations presented with dots in **a** in their respective colors.

smaller driveline tube, in other words, how strong the coupling is (Fig. 4.7c).

To achieve the desired $Q_{in} = 1.8$ SLPM, we use an analogy between fluidic systems and electronic circuits. We do this by using the physical similarities between Hagen-Poiseuille's law (see Equation 4.1a) for fluidic systems and Ohm's law for electronic circuits (see Equation 4.1b) [85]. The equation (4.1c) for hydraulic resistance (R_H) shows that we can tune the resistance in a fluidic system by modifying the length (L) and radius (r) of the tubes. Additional fluidic resistances in the form of silicone tubes with a small lumen (i.e., an inner area that is at least an order of magnitude smaller compared to the other connection tubes) are used as a replacement for the flow regulator that controls the input flow, making the entire system softer and more compact. Instead of setting the Q_{in} using the flow regulator, we tune the silicone tubing from the wall pressure source to get the desired Q_{in} value for actuating the oscillator (Fig. 4.7a). As the wall pressure is much higher than the rest of the system, we expect this silicone tubing to act as a resistance that transforms the constant wall pressure into a constant flow. The effect of the length of the silicone tubing can be found in Fig. 4.8a.

$$\Delta P = Q \cdot R_H \tag{4.1a}$$

$$\Delta V = I \cdot R_E \tag{4.1b}$$

$$R_H = 8\mu \cdot \frac{\pi L}{A^2} = 8\mu \cdot \frac{L}{\pi r^4} \tag{4.1c}$$

The coupling mechanism uses this same analogy to increase the resistance by manipulating the inner diameter (r) of the small tube. Initially, the flexible large tube is not inflated and wraps around the interface plate (i.e., weight). This leaves the inner diameter of the stiffer small tube intact, allowing air to flow through the small tube. When the large tube inflates, it presses the weight down into the small tube, reducing the inner diameter. By increasing the resistance, the coupling manipulates the amount of flow that is directed toward the oscillator, which influences the overall dynamics of the system. Although the effect of the weight on the inner diameter of the small tube is comparable to a variable resistor, the entire coupling mechanism shows more similarities to a normally open transistor. The coupling mechanism has a gate (the flexible tube) and an inlet and outlet (the small tube going in and out of the mechanism). For a transistor, this is equivalent to the base (B), emitter (E), and collector (C) pins.

The final setup consists of the following four elements (Fig. 4.7a). Firstly, the source connection is a 150 cm tube going from the pressure source to the oscillator system used to generate a constant input flow.

Secondly, the negative coupling (Fig. 4.7a in orange) is a 30 cm tube connected to the 150 cm tube and the tubing that goes towards the valve inlet. A coupling mechanism is connected to this tubing, thereby establishing the negative coupling. This coupling increases the resistance towards the inlet of the valve, thus restricting the flow. The flow gets restricted when the coupling is activated, as can be seen from

Fig. 4.8b. We find a close-to-linear slope for the negative coupling up to 100 kPa, where the control pressure completely closes the driveline tube. This is not ideal, as the oscillator used for the final experiments has a $P_{open} = 148$ kPa, which means that it will have a weakened effect during these higher pressures.

Thirdly, the positive coupling is a 200 cm tube with an additional needle resistor connected to the 150 cm tube (parallel to the 30 cm tube) that serves as a "leaking" resistor that allows the flow to escape towards the environment. We find that the leaking increases the loading phase of the oscillator as depicted in Fig. 4.8c. A coupling mechanism is connected to this tubing, thereby establishing the positive coupling. This coupling increases the resistance towards the environment, thereby increasing the flow that goes towards the inlet of the valve. The flow towards the inlet is amplified when the coupling is activated. The effect of the positive coupling on the flow towards the oscillator can be found in Fig. 4.8d. We see that the relative change in flow is much less compared to the negative coupling, meaning that the constructed positive coupling has a lower coupling strength. Also, we see that the transition between the maximum and minimum flow is much quicker, which means that there is a larger area where the coupling has no effect. From these results, we expect that it will be harder to find in-phase synchronization using the positive coupling compared to finding anti-phase synchronization with the negative coupling. Lastly, we have the oscillator and the capacitance, where the capacitance could be interchanged with an actuator to achieve the actuation of the soft robots.

4.8 PHASE-LOCKING TO A PRESSURE CONTROL SIGNAL

Using the circuit described in the previous section, we first focus on a controlled case where we aim to test if the oscillator can be phase-locked with a control signal generated by a pressure regulator. Instead of providing a constant pressure that increases stepwise, the pressure regulator mimics the oscillatory behavior to see if phase locking can occur in a controlled setting. We control the coupling signal in both frequency and pressure to isolate the dynamics of the single oscillator and analyze the response of the coupling on the oscillator.

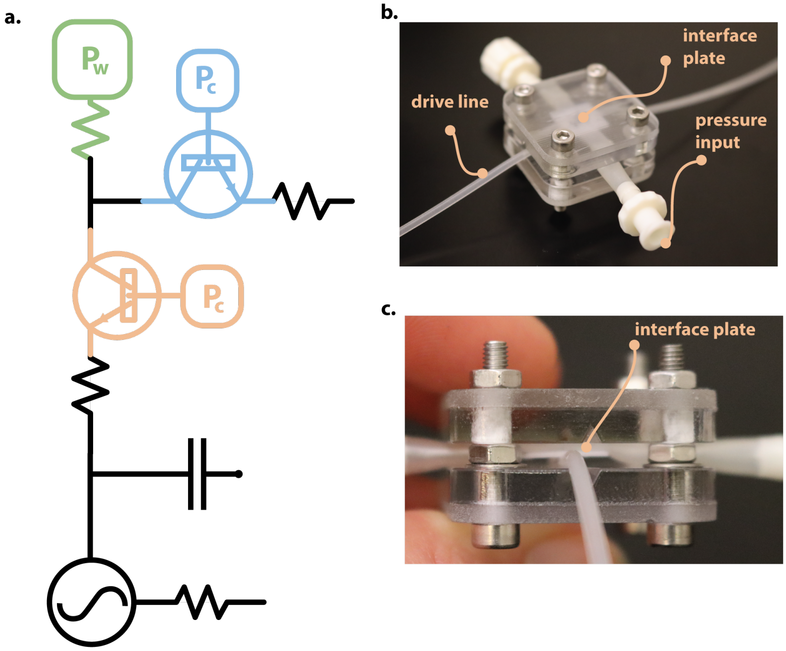


Fig. 4.7: **Coupling a single oscillator to environmental input a**, Schematic representation of the circuit for a single oscillator with the positions of the positive (blue) and negative (orange) coupling devices. The wall pressure with its dedicated resistance is presented in green. The capacitance used for the experiment with this circuit has a volume of 15 ml. **b**, Photo of the coupling device. **c**, Zoomed side view of the coupling device. Indicated the PVA interface plate between the pressure input and the drive line.

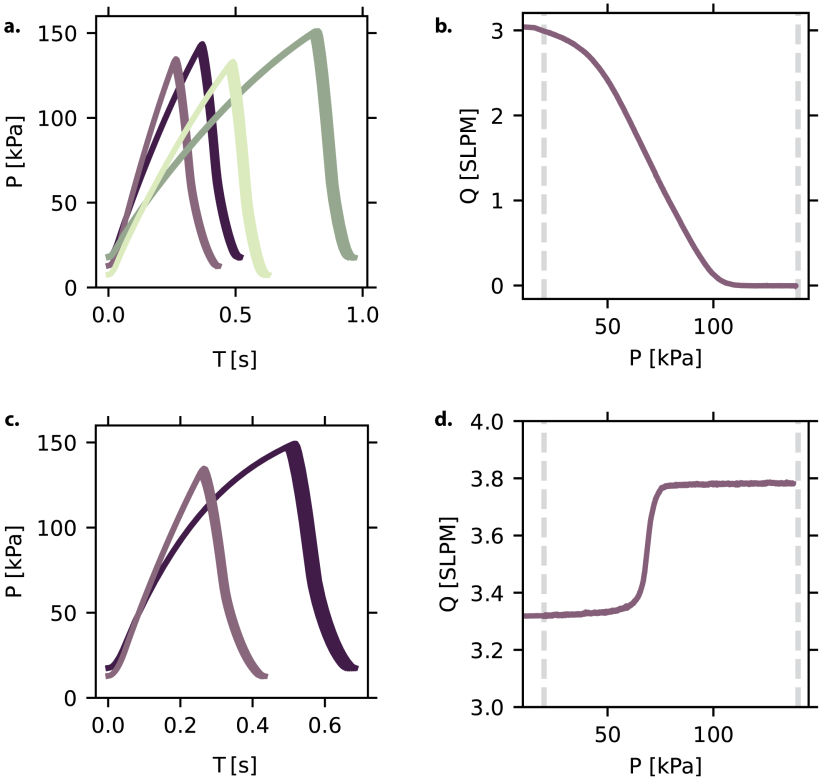


Fig. 4.8: **Effect of the coupling in the oscillator circuit a**, Effect of different lengths of tubing for the P_{wall} resistance, measure in purple for tubing with an inner diameter of 1 mm with lenght 100 cm (light purple) and 200 cm (dark purple). Measured for tubing with an inner diameter of 0.3 in green with length 94 cm (light green) and 34 cm (dark green). **b**, Effect of pressure applied at the negative coupling element on Q_{in} measured at the oscillators inlet, for the circuit described in Fig. 4.7a. **c**, Effect of the parallel resistance that vents the air to the atmosphere (added in Fig. 4.7) on the curve of the oscillator. Measure with (dark purple) and without (light purple) the parallel resistance. **e**, Effect of pressure applied at the positive coupling element on Q_{in} measured at the oscillators inlet, for the circuit described in Fig. 4.7a.

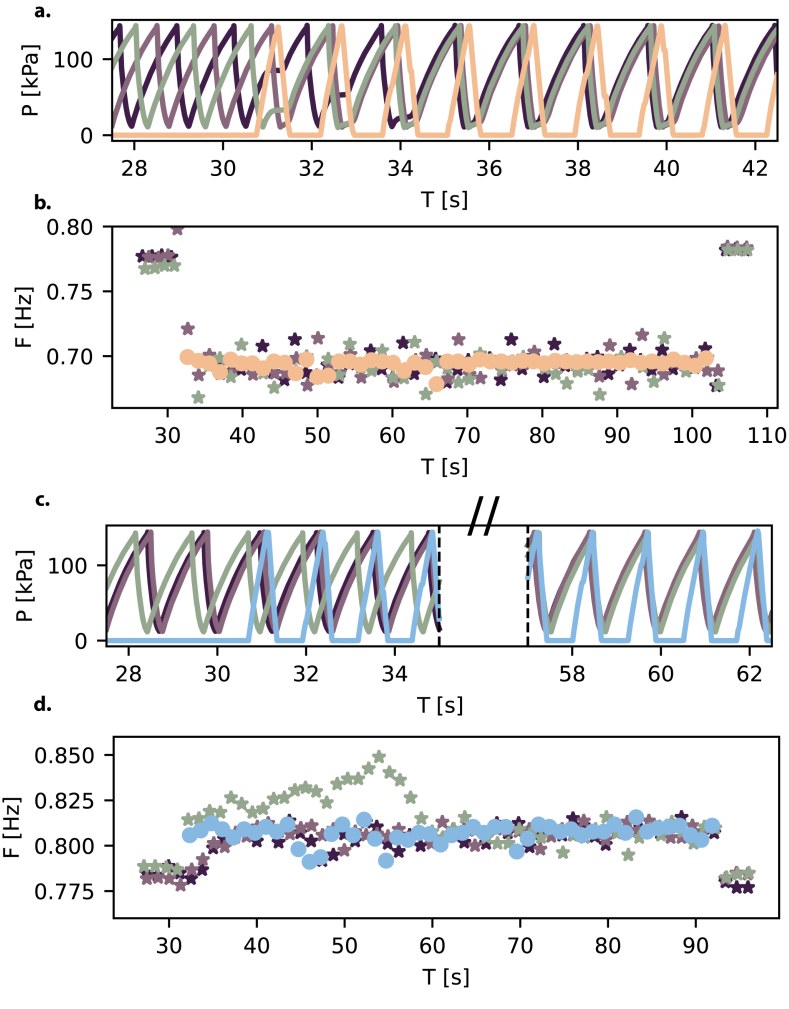
In Fig. 4.9, we connect the pressure regulator (presented in orange in Fig. 4.9a) to the negative coupling element. For the three consecutive experiments with the same oscillator and different starting conditions, we find phase locking in all three experiments. Fig. 4.9a shows that the phases lock in a phase shift where the oscillator closes when the pressure regulator is at its highest pressure and opens when the pressure regulator is at its lowest pressure (Movie 4.2). Furthermore, we find that for all three experiments, the phases are locked within the first four cycles of the pressure regulator, or 5 s. From Fig. 4.9b, we find that also the frequency of the oscillator adapts to match the frequency of the control signal. By maintaining the same frequency, the phases are able to lock and prevent phase drift with respect to the pressure regulator. As soon as the pressure regulator stops, the frequency returns back to its original value.

Fig. 4.9c presents the same experiments, but now the pressure regulator is connected to the positive coupling. We find that after 60 s, the phases lock at an in-phase synchronization where the peak of the pressure regulator aligns with the peak of the oscillator pressure at the point where the oscillator opens. Furthermore, the closing event of the oscillators aligns with the moment the pressure regulator stops depressurizing. From Fig. 4.9a-b, it is clear that the negative coupling is much stronger compared to the positive and is, therefore, able to lock the phases more quickly (Movie 4.3). This is likely due to the difference in how much they can influence Q_{in}, where the negative coupling has a larger range of flows compared to the positive coupling (Fig. 4.8b and d, respectively). For the frequencies depicted in Fig. 4.9d, we find similar results where the frequency of the oscillators adapts to the frequency of the pressure regulator.

The behavior observed in Fig. 4.9 is in line with the hypothesis of the coupling mechanisms given by the model (Fig. 4.2). When the negative coupling is activated, meaning the pressure regulator is in the loading phase, it inhibits the inflow going to the valve. This increases the loading phase of the valve. When the pressure regulator is in the recovery phase, the negative coupling is not activated. Inflow can now go to the valve, which allows for faster pressure build-up and brings the membrane into the fully open state. For the positive coupling, the loading phase of the pressure regulator increases the inflow to the valve, decreasing the loading phase of the valve, which leads to in-phase locking (Fig. 4.2).

Interestingly, the negative coupling currently does not lock the oscillators fully anti-phase (i.e., with a phase shift of the peaks equal to π), but rather somewhere between in-phase and anti-phase (Fig. 4.10a). This is attributed to limitations in the pressure regulator inflow toward the coupling, which is unable to fully mimic the pressure outputs of the oscillators and remains longer at a pressure value of zero during the recovery phase. During this zero-pressure phase of the pressure regulator, there is effectively no influence of the coupling, and we suspect that this enables the phases of the oscillators to shift slightly out of phase, as observed in Fig. 4.10a. Furthermore, for the positive coupling, we find that the experiment presented in green switches its trajectory towards in-phase synchronization at around 37 s, causing it to phase lock much later compared to the other two experiments. Lastly, we found the phase locking to be sensitive to the frequency of the pressure regulator. Only pressure regulator frequencies within roughly five percent of the frequency of the oscillator showed phase locking.

These experiments demonstrate that the current framework can follow a control signal and phase lock, given a non-reciprocal coupling with the environment. This means that the oscillator is coupled to the pressure regulator's input and not vice versa, thereby confirming the effect of the negative and positive one-direction coupling in the experimental system.



Caption on next page.

Fig. 4.9: **Experiment of a single oscillator following a control signal a**, Three separate experiments with the circuit described in Fig. 4.7, where the positive coupling element remains at zero pressure and a pressure controller controls the negative coupling element. The control signal is depicted in orange, and the pressure at the inlet of the oscillator is presented in three colors: purple to green. **b**, Frequencies of the oscillator during the three experiments in **a** along with the control signal (orange). **c**, Three experiments with the same setup as **a**, but with the pressure controller connected to the positive coupling element (blue) and the negative coupling element fixed at zero pressure. **d**, Frequencies of the oscillator during the three experiments in **c** along with the control signal (blue).

4.9 MUTUAL COUPLING OF TWO OSCILLATORS

Having confirmed the effect of the experimental negative and positive coupling framework, the next step is to perform experiments on a network consisting of two oscillators without the need for pressure and flow regulators to control the phases of the oscillators. To perform these experiments, we constructed the circuit schematically depicted in Fig. 4.11a. Importantly, all inputs come from the same pressure source P_{wall}, meaning that even if the number of oscillators increases, it would still only require a single constant pressure to operate. This is possible because the large resistances connected to the wall pressure decouple the system's inputs. In this circuit, we integrate the positive and the negative coupling for both oscillators with a manual valve in front of each so that they can be turned on and off individually during the experiments (Fig. 4.11b).

For the first experiment, we turned off all the couplings in the system to verify that the system does not phase lock without active coupling. This could occur due to parasitic coupling that could potentially be generated by the vibration of the table or backpropagation of pressures to P_{wall}. The results from this experiment are presented in Fig. 4.12a, where the pressure in the tank is set to 200 kPa at 14 s into the experiment. We find a slight difference in the loading phase during the first oscillation of both oscillators, which might be due to slight differences in leakage

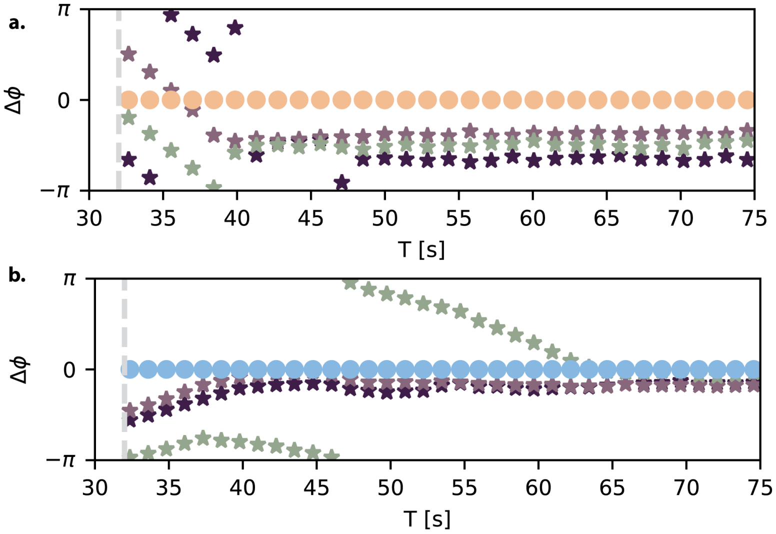


Fig. 4.10: **Characterization of a single oscillator a**, Phase differences over time for the experiment of Fig. 4.9a. **b**, Phase differences over time for the experiment of Fig. 4.9c. Both **a** and **b** use the same color indication as Fig. 4.9, and the gray dashed line indicates the start of the cycle activations of the pressure controller.

or a difference in the resistances. More importantly, we also find a slight difference in the opening pressures of the two oscillators, which results in a difference in frequency. Over time, this difference in frequency results in phase drift between the oscillators that is not reconciled by any potential parasitic coupling. These results confirm the absence of phase locking within the first 30 s of the experiment.

Next, we only turn on the negative coupling for both oscillators and repeat the experiment (Fig. 4.12b and Movie 4.4). We observe that although the loading of both oscillators starts equally, imperfections combined with the strong negative coupling drive the oscillators out of phase within the first cycle. For the remainder of the experiment, we observe clear anti-phase locking, confirming the hypothesis from the simulations. Furthermore, we find that the negative coupling is so strong that it seems to temporarily completely repress the loading of the other oscillator. This results in almost constant pressure in one oscillator while the other is close to opening.

Lastly, in Fig. 4.12c, the same experiment is presented where only the positive couplings are active. Although the start of the experiment loads both oscillators equally during the first oscillation, it seems like the difference in the initial opening pressure drives the two oscillators slightly out of phase during the first cycle. Over time, we see that this phase shift disappears due to the positive coupling, confirming results found in the simulation (Movie 4.4).

To obtain more conclusive evidence of the phase locking over longer time scales, we show the inlet pressures of the two oscillators with respect to each other for all three experiments in Fig. 4.13a-c. In Fig. 4.13a, we see that without any of the couplings activated, the pressure relation between the two oscillators keeps changing without a clear sign of a recurring pattern. Hence, a clear limit cycle is absent, and there is no sign of phase locking. For the positive coupling in Fig. 4.13b, we do find a clear limit cycle along the diagonal, similar to the simulation result (Fig. 4.2f), confirming the in-phase synchronization of the oscillators. Interestingly, the start of the trajectory (light green) also shows that indeed the oscillators were initially pushed out of the in-phase synchronization during the initial cycles. Furthermore, for the final limit cycle, at lower pressures up to 70 kPa, there is relatively little deviation from the diagonal. At the same time, we find that at roughly 70 kPa, the trajectory of the limit cycle deviates from the diagonal (Fig. 4.13b). This

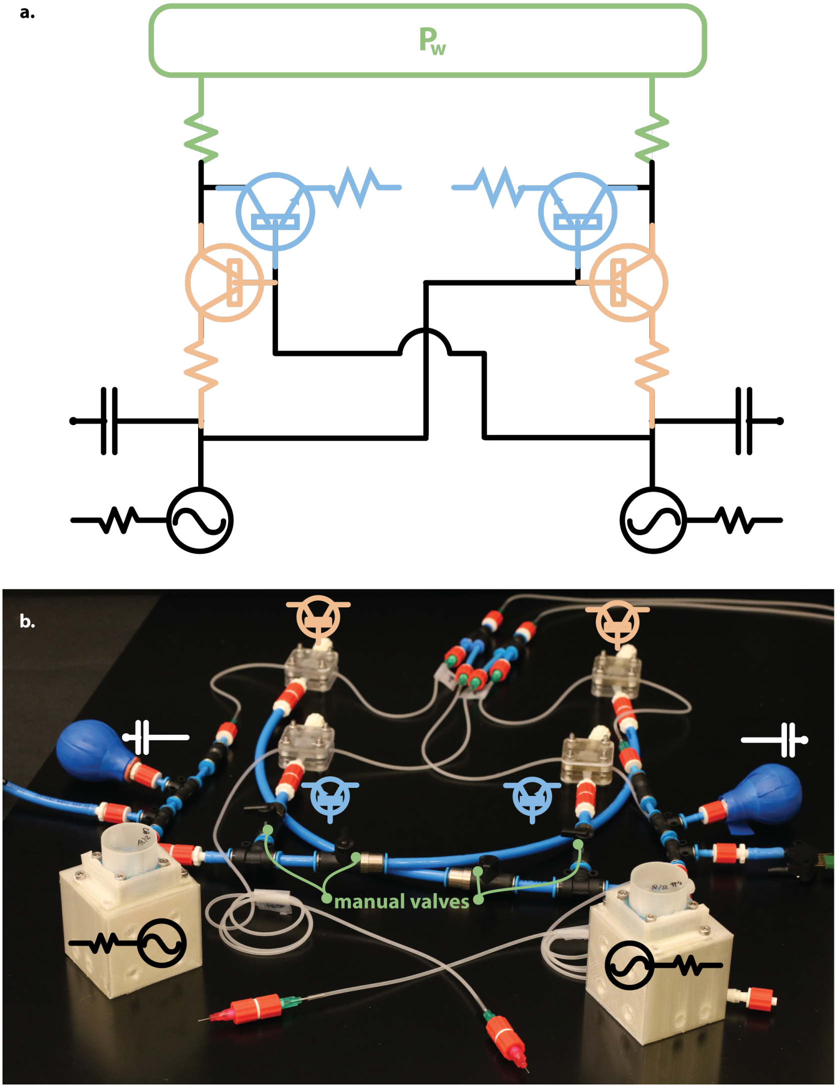


Fig. 4.11: **Ciruit for a network of two oscillators. a**, Schematic circuit diagram for a network of two oscillators operating from a single pressure source (green). Both are integrated with both positive coupling (blue) and negative coupling (orange) that can be turned on and off with a valve individually. **b**, Photo of the experimental setup for the network of the two oscillators. Schematic icons highlight the position of individual components in **a**.

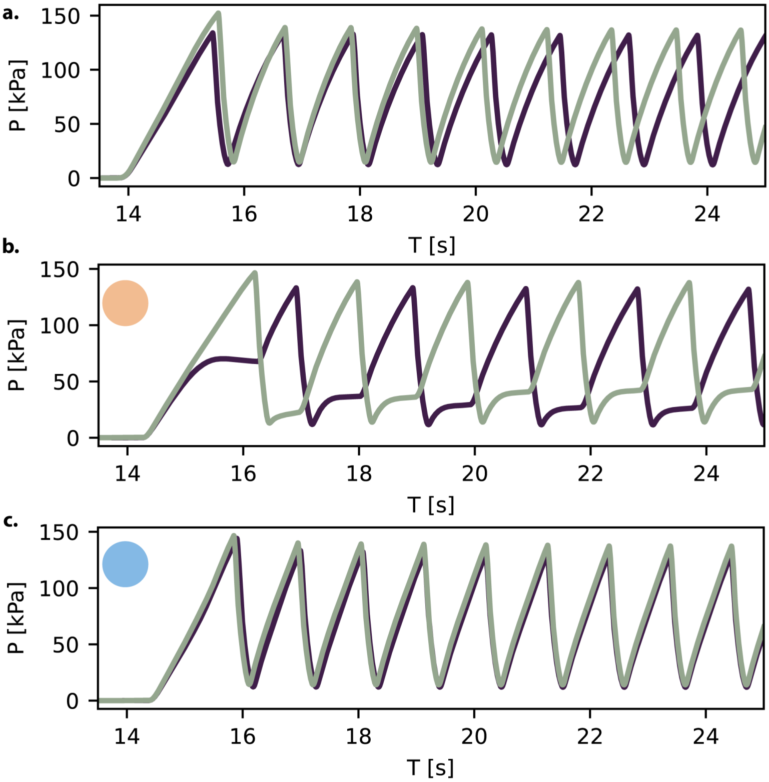


Fig. 4.12: **Characterization of a single oscillator a**, Experimental data of the pressures for the two oscillators presented in Fig. 4.11, measured at the inlet of the oscillators. At 14 s the pressure tank P_{wall} is set to 200 kPa which starts the oscillators. During the experiment, all the valves that control the couplings are turned off. **b** Repeats the experiment of **a**, but before the experiment has started, both valves linked to the negative coupling elements are turned on, activating the negative coupling. The positive coupling elements remain inactive during the experiment. **c** Repeats the experiment of **a**, but before the experiment has started, both valves linked to the positive coupling elements are turned on, activating the positive coupling. The negative coupling elements remain inactive during the experiment.

could be a result of the sharp transition in the coupling device for the positive coupling, as presented in Fig. 4.8d. If one of the oscillators is slightly in front of the other in the phase space, it will boost the other oscillator earlier, resulting in a slight deviation from the diagonal.

Fig. 4.13c depicts the limit cycle from the mutual negative coupling between the oscillators. These results confirm that within the first three cycles, the phases are locked into the limit cycle, and once the limit cycle is reached, we find almost no deviations in its trajectory. These results are coherent with previous findings on the strength of this negative coupling. Importantly, the overall shape of the limit cycle differs from the results obtained in the simulations (Fig. 4.2g). In the simulation results, we observed a shape for the limit cycle best described as an hourglass, which was axisymmetric along two axes. In contrast, the results from the experiments present a shape that is more reminiscent of the outline of a butterfly. Apart from the slight asymmetry on the xy-axis, which is likely due to differences between the oscillators, we find slender, stretched corners when one of the oscillators is at high pressure. We found that these results could be replicated with the model when we make the negative coupling stronger and set it to $k = 0.3$ in the model. The resulting limit cycle is presented in Fig. 4.13d, where the increased coupling strength stretches the corners for high pressures as a result of inhibiting the loading of the other oscillator. The time evolution of the simulation is presented in Fig. 4.13e. Indeed, the loading of the second oscillator is repressed at high pressures.

In conclusion, the results from the experiments are coherent with the simulation and show that solely by changing the coupling between the oscillators, we can start to program the phase differences between the oscillators. Furthermore, we find that we can match the qualitative behavior of the oscillators with our relatively simple model.

4.10 CONCLUSION

In line with the bio-inspired philosophy of soft robotics, this work draws inspiration from the bio- and neuromechanics found in nature to enrich the behavioral patterns we can encode in soft robots. We showcase the design of a fluidic relaxation oscillator with a highly tunable flow response. Along with this oscillator, we introduce a framework

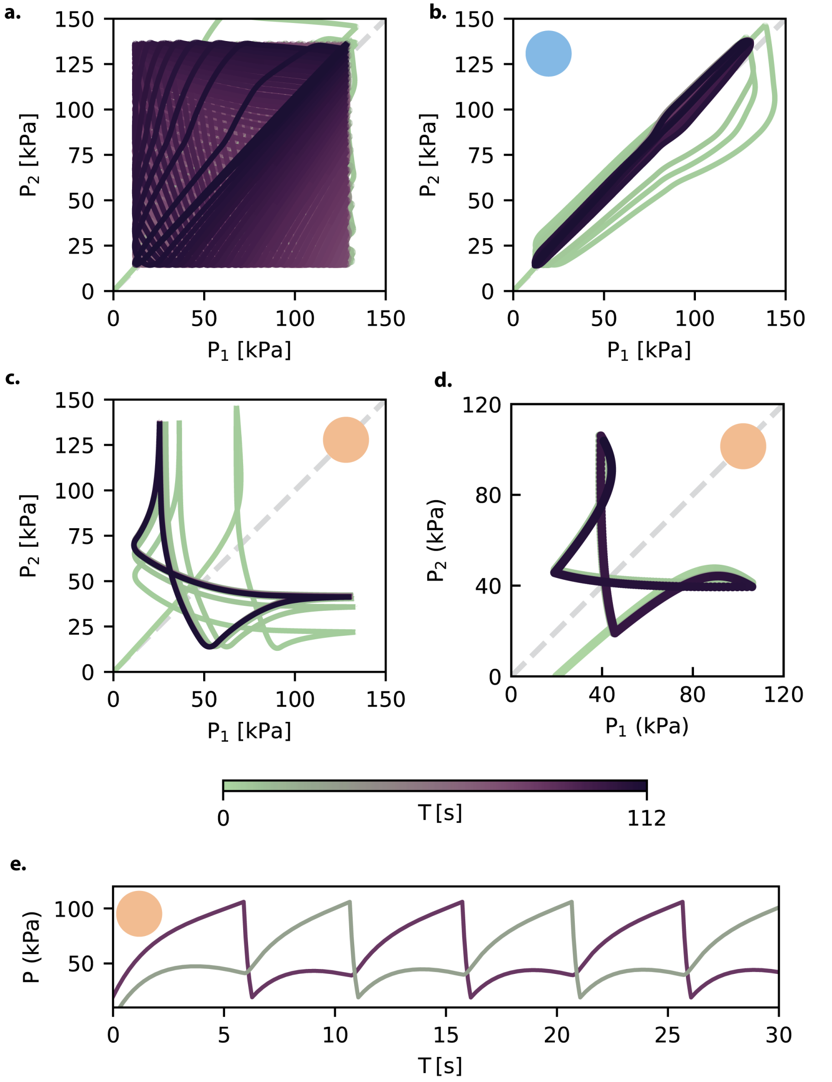


Fig. 4.13: **Characterization of a single oscillator a**, Pressures presented in Fig. 4.12a colored by time for $t = 0$ to $t = 112$ seconds. **b**, Pressures presented in Fig. 4.12c color by the same time range. **c**, Pressures presented in Fig. 4.12b color by the same time range. **d**, Simulation as presented in Fig. 4.2g with the k changed from -0.08 to -0.3. **d**, Time evolution simulated pressure in **d**.

inspired by the behavior of neurons in nature to couple these fluidic oscillators for richer and more programmable behavioral encoding. This coupling strategy allows for robust phase locking and both in-phase and anti-phase synchronization of the oscillators without the need for any electronics (Movie 4.5). The system shows promising results for better understanding and controllability of the emerging behavioral patterns, irrespective of small fabrication defects in the design. The current framework also enables the system to scale by keeping interactions local and driving the system from a single constant pressure source. Overall, with this work, we hope to move electronic-free circuits towards more adaptive and controllable outputs. This work thereby addresses the earlier defined gap between the limiting propagating waves observed in soft robots with fluidic oscillators and the rich, adaptive behavior exhibited in CPG models and nature.

Although the current experimental setup is not small compared to the average size of pneumatic soft robots, most of the current elements can be scaled down and integrated (e.g. using microfluidics [86]). In future work, it would be interesting to focus on integrating the individual elements and coupling into the oscillators and start expanding the networks to more oscillators. Once integrated into soft robots with pneumatic actuators, the actuators could function as a coupling with the environment through forces applied to the actuator. Studying these environmental couplings could result in programmed behavioral switching in response to environmental changes, as known in nature and CPG models [109].

Overall, future research on control based on soft logic and biomechanics would support the development of these alternative control strategies for soft robots. These alternative strategies reduce the need for complex hardware and electronics and show potential in their increased robustness, inexpensive manufacturability, and responsiveness in soft robotic systems.

4.11 SUPPLEMENTARY MOVIES 4

Movie 4.1. Single oscillator. Oscillatory behavior of a single oscillator for a constant input flow.

Movie 4.2. Negative coupling. Single oscillator coupling connected to a constant flow input. The oscillator has a negative coupling to an external pressure controller facilitated by the fluidic coupling device.

Movie 4.3. Positive coupling. Single oscillator coupling connected to a constant flow input. The oscillator has a positive coupling to an external pressure controller facilitated by the fluidic coupling device.

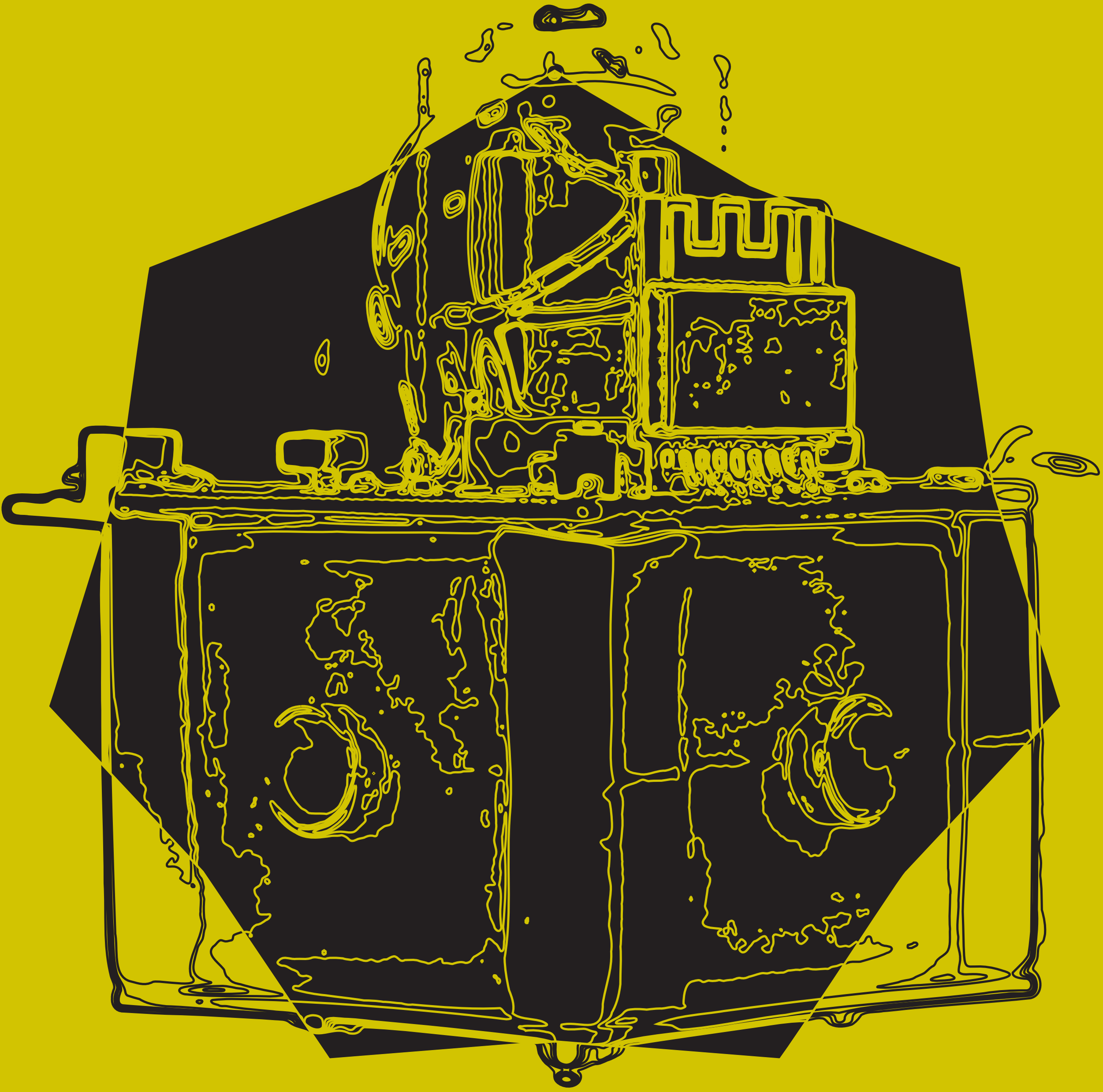
Movie 4.4. Coupling of two oscillators. In-phase and out-of-phase synchronization of two fluidic oscillators. The movie shows that the fluidic coupling between the oscillators can achieve phase locking between the oscillators in a system powered by a single constant pressure source.

Movie 4.5. Switching between positive and negative coupling. Switching the coupling between positive and negative shows the transition between in-phase and out-of-phase synchronization and phase locking.

Link to the Movies.

ADAPTIVE BEHAVIOR IN MINIMAL OSCILLATOR NETWORK THROUGH COEVOLUTION OF BODY AND BRAIN

5.



Abstract. In biological systems, central pattern generators (CPGs) produce robust yet adaptive rhythmic patterns by tightly coupling morphology, sensors, and oscillatory neural circuits. Translating these principles into soft robotics has proven difficult, as most fluidic oscillator networks generate only fixed sequences and rely on external supervision for task execution. Here, we numerically investigate how adaptive behavior can emerge from the co-evolution of soft morphologies and minimal networks of relaxation oscillators. Using voxel-based simulations, we demonstrate that just three coupled oscillators are sufficient to generate diverse gaits, enable task switching, and achieve behavioral switching from a single input. Importantly, the same oscillator network can settle into distinct behaviors depending on environmental interactions, highlighting the role of body–environment coupling as an additional mode of computation. These findings suggest that morphology and controller should be treated as a single dynamical unit, whose interaction yields scalable and multifunctional behavior. By demonstrating that complex locomotion and environmental adaptation can emerge from minimal oscillator networks, this work lays the groundwork for fluidic soft robots that leverage co-designed body–brain architectures for electronics-free adaptive control.

H.A.H. Schomaker, J.T.B. Overvelde. "Adaptive behavior in minimal oscillator networks through coevolution of body and brain." *in preparation*.

5.1 INTRODUCTION

In nature, central pattern generators (CPGs) are specialized circuits comprised of networks of oscillating neurons that produce rhythmic motor outputs with minimal input from higher-level control centers [41]. Such networks are crucial for locomotion in a wide array of species, ranging from simple invertebrates to complex vertebrates (Fig. 5.1a). They can be modulated by sensory inputs (Fig. 5.1a-b in green) to accommodate environmental or physiological constraints [35, 53, 138]. In practice, CPGs exemplify how rhythmic motions (Fig. 5.1c), such as those needed to walk or swim, can be reliably generated while maintaining enough flexibility to cope with changing conditions [53, 94, 96]. Though capable of generating basic rhythmic patterns on their own, local sensory input is crucial for CPGs [94]. This interplay between central networks and peripheral feedback mechanisms enables animals to react almost instantaneously to perturbations in their environment, whether they face uneven terrain or unexpected obstacles, without the intervention of the brain [2, 29, 53]. Sensors in muscles, joints, or skin further modulate the ongoing motor program to maintain stability and efficiency [50, 76, 95]. Especially in animals with a more minimal neural architecture, it is speculated that the CPGs take the primary role in executing actions [53]. In this way, sensory-driven regulation exemplifies the "closed-loop" control that biologists and engineers alike strive to replicate in artificial systems.

Fluidic actuators have gained traction for their ability to produce compliant, programmable motion using fluidic circuits. These systems increasingly rely on fluidic oscillators to produce repetitive motions from continuous inputs, in an attempt to make the entire soft robot electronics-free [25]. These electronics-free soft robots have the advantage that they reduce cost because they do not require electronics [25]. Furthermore, the reduction of components (e.g., number of pumps) has the potential to reduce failure and make the system scalable to smaller sizes [64]. Coupling these oscillators together has demonstrated potential for encoding locomotion primitives into soft robots [25, 82, 101]. Nonetheless, existing fluidic oscillator systems, such as ring oscillators, are constrained by fixed sequential motion patterns, offering little capacity for reconfiguration or tuning based on sensory information or higher-level inputs [25, 82, 101]. Consequently, these devices often require additional circuits or external supervision to perform tasks of any complexity [25]. This limitation restricts the development of soft robotic systems that can autonomously adapt, learn, or refine their behavior in real-world settings [122, 136]. The development of soft relaxation oscillators has shown that coupling self-oscillating fluidic oscillators can lead to sequence changes within the same circuit based on external inputs [64]. These findings bridge the gap between reflex neural circuits in living systems and current soft robotics designs.

In biological organisms, form and function evolve in tandem. The design of limbs or body segments, tissue elasticity, and the placement of sensors all shape and are shaped by their neural control architectures [40]. A prominent example in humans is the heart, whose characteristic pumping arises from closely coupled electrical and mechanical processes. Research in electromechanical cardiology demonstrates that understanding one aspect, the electrical or the mechanical, cannot fully explain the emergent behavior without accounting for their continuous interaction [127]. Translating these findings to soft robotic designs suggests that computational and physical aspects must co-develop to produce robust and adaptable behavior.

On the one hand, the evolution of soft robots has led to the development of voxel-based soft robotic simulations. These numerical simulations place square pixels in a grid to form and evolve the body of a soft robot. Using a minimal variety of different voxels, including stiff voxels (bone), periodic inflating voxels (muscle), and passive voxels (soft tissue), robots evolve complex walking gaits solely due to their soft morphology [14]. Previous work demonstrates that for fixed periodic inputs, both 2D and 3D morphologies can evolve to swim and run [14, 21, 22, 128]. The behavior, however, remains fixed to a single gait as the actuation timing is prescribed beforehand.

On the other hand, to achieve adaptive behavior for completing more complex tasks, reinforcement learning and neural networks have been applied to fixed morphologies to control the behavior of these soft voxel-based robots [75, 115]. This demonstrates how complex tasks can be performed with systems comprising only a limited number of voxels. Even neural controllers in a decentralized architecture show great potential when optimized for fixed geometries [123]. However, the morphologies in these cases are predefined.

Research on the co-evolution of a voxel body in tandem with a reinforcement-learning policy controller shows the importance of sensory feedback and the synergy between a body and its controller. While this work highlights the complexity of tasks that can be executed when the body and the brain are optimized together (e.g., picking up objects and transporting them to a new location), it requires a highly sensitive and complex centralized controller.

In the current work, we aim to numerically explore the gap between experimental electronics-free soft robots based on fluidic oscillators and the adaptive behavior found in both CPG controllers and neural networks using numerical voxel-based simulations. Drawing inspiration from biology's example of co-evolution, where body morphologies evolve in tandem with their underlying neural oscillator networks and sensors, this work aims to investigate how a similar process can be replicated in a simulated environment to design adaptive behavior in soft robots using minimal oscillator networks (using a similar number of oscillators as in current experimental soft robots). By iteratively refining both physical designs and oscillator architectures, we aim to investigate how a close interaction between body, brain, and sensors can lead to more robust locomotion as well as multifunctional task execution within the same minimal oscillator architecture. With this work, we hope to pave the way for more adaptive behavior in soft robots that can be experimentally implemented in fluidic networks due to their minimal architecture.

5.2 METHODS

We pose the problem of co-designing a soft robot's body and its oscillator network that controls actuation, to generate task-level motion on a static terrain. The body is discretized as a 5×5 grid of square pixels (size limit), where each pixel is absent, passive, or an actuator coupled to one of the relaxation oscillators (Fig. 5.2a shows an example for two oscillators). By coupling the oscillators directionally in networks, we can enforce phase-locking with predictable phase offsets between the oscillator outputs (Fig. 5.2b). Actuation of the non-passive pixels is produced by directly coupling the oscillators to the homogeneous expansion and contraction of the pixels (Fig. 5.2c). Given the terrain

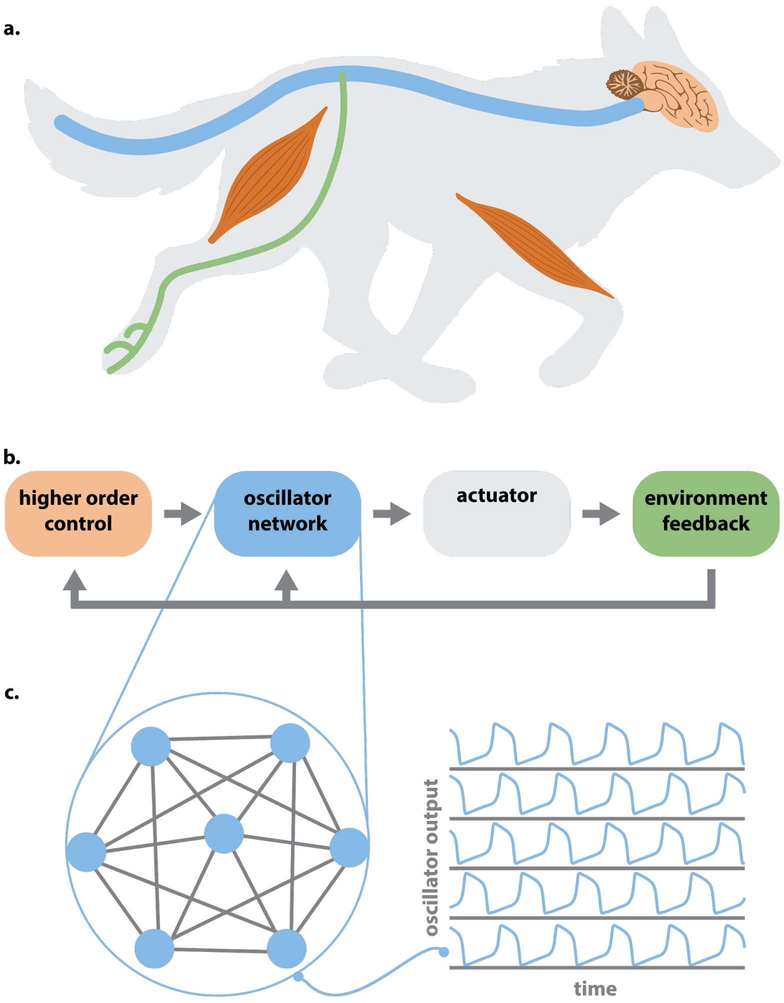


Fig. 5.1: **Central pattern generator (CPG)-based locomotion control. a**, Schematic representation of the signal flows involved in locomotion control. **b**, framework of the signal feedbacks involved in the control of **a**. **c**, Schematic representation of the oscillator network and oscillator outputs.

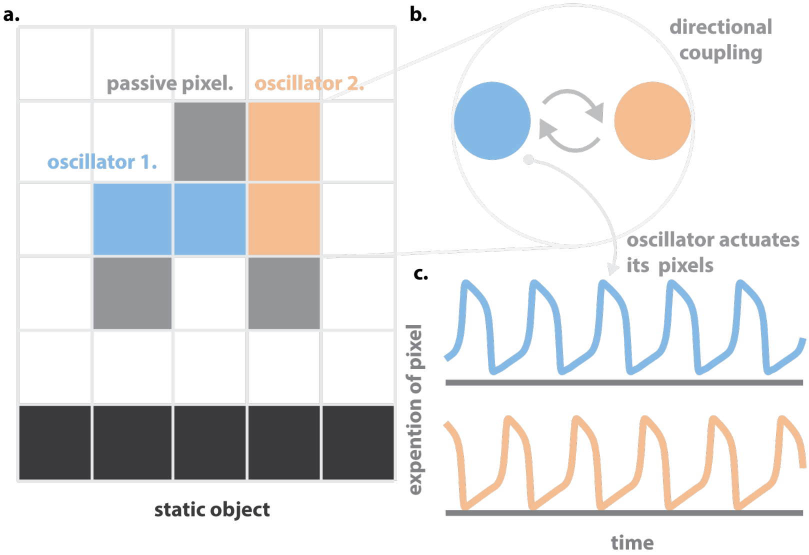


Fig. 5.2: **Problem setup and example for a two oscillator system. a**, The soft robot morphology is encoded on a 5 × 5 pixel grid (size limit). Each pixel is either absent, passive, or active (blue/orange). The dark region indicates the static object that forms the terrain on which the robot moves. **b**, Schematic of the directional coupling between the two oscillators. **c**, Directional coupling can yield phase locking with a specific phase offset, which then drives coordinated expansion/contraction of the actuated pixels.

(static object) and the size constraint, our objective is to find the morphology and coupling parameters that maximize task performance (e.g., forward displacement or climbing height) while keeping the controller minimal and the simulation computationally efficient. Keeping the simulation computationally low-cost enables us to use a genetic algorithm to co-evolve the body morphology and oscillator network in tendam, generating solutions with synergy between body and brain.

Mechanical Model

We model the mechanical system as a network of point masses (nodes) interconnected by linear springs, forming deformable square pixels.

Our model and parameter choices are designed to make the simulation computationally efficient, enabling its use in evolutionary optimization. Each square pixel comprises four nodes linked by six springs: four along the edges, which remain passive with fixed rest lengths, and two diagonal springs that are active (Fig. 5.3a-b). These diagonal springs undergo periodic changes in their rest lengths, driven by FitzHugh–Nagumo oscillators (Fig. 5.3c), causing each square to expand and contract rhythmically. This coupling leads to synchronized expansion and contraction of the square's diagonals, Fig. 5.3a-b, while the edge springs provide passive structural integrity. The rhythmic actuation drives deformation and, when assembled into larger lattices, can induce locomotion or coordinated shape change. All baseline parameters for the model are listed in Table 5.1 and were chosen to optimize simulation speed while maintaining rich dynamics for the structure.

More specifically, we model a planar structure composed of N point masses (nodes) connected by E linear springs. A *square pixel* consists of four nodes connected by six springs: four passive *edge* springs with fixed rest lengths and two *diagonal* springs whose rest lengths are actively modulated in time. Let

$$\mathbf{q}(t) = \begin{bmatrix} x_1 & y_1 & \cdots & x_N & y_N \end{bmatrix}^\top \in \mathbb{R}^{2N}, \qquad \dot{\mathbf{q}}(t),\ \ddot{\mathbf{q}}(t) \in \mathbb{R}^{2N},$$

denote positions, velocities, and accelerations, respectively. The diagonal mass matrix is

$$\mathbf{M} = \operatorname{diag}(m_1, m_1, \dots, m_N, m_N) \in \mathbb{R}^{2N\times 2N} \quad [\mathrm{kg}].$$

Bulk viscous drag is modeled as an isotropic, piecewise-constant dashpot,

$$\mathbf{C}(t) = c(t)\,\mathbf{I}_{2N} \quad [\mathrm{N\,s/m}], \qquad c(t) = \begin{cases} 0.6, & t < 1.5\ \mathrm{s}, \\ 0.2, & t \geqslant 1.5\ \mathrm{s}, \end{cases}$$

chosen to suppress initial transients while preserving steady actuation dynamics.

Spring Kinematics and Matrix Assembly

Springs are indexed by $e = 1, \dots, E$, with endpoints (i_e, j_e), stiffness k_e [N/m], and rest length $L_e(t)$ [m]. For each spring we define a *selection matrix* $\mathbf{S}_e \in \mathbb{R}^{2\times 2N}$ such that

$$\mathbf{r}_e(t) = \mathbf{S}_e\,\mathbf{q}(t) = \begin{bmatrix} x_{j_e} - x_{i_e} \\ y_{j_e} - y_{i_e} \end{bmatrix} \in \mathbb{R}^2, \quad \ell_e(t) = \|\mathbf{r}_e(t)\|_2, \quad \mathbf{n}_e(t) = \frac{\mathbf{r}_e(t)}{\max\big(\ell_e(t), \varepsilon\big)},$$

with a zero-length regularization $\varepsilon = 10^{-9}$ m to prevent division by zero. Concretely, $\mathbf{S}_e$ has nonzero values only in the four columns corresponding to $(x_{i_e}, y_{i_e}, x_{j_e}, y_{j_e})$:

$$\mathbf{S}_e = \begin{bmatrix} \cdots & -1 & 0 & \cdots & +1 & 0 & \cdots \\ \cdots & 0 & -1 & \cdots & 0 & +1 & \cdots \end{bmatrix}.$$

The Hookean force contributed by spring e and assembled to the global DOFs is

$$\mathbf{f}_e\big(\mathbf{q}(t), t\big) = \mathbf{S}_e^\top\, k_e\,\big(\ell_e(t) - L_e(t)\big)\,\mathbf{n}_e(t) \in \mathbb{R}^{2N},$$

which automatically applies equal and opposite forces to nodes i_e and j_e. Summing over springs yields

$$\mathbf{f}_{\text{int}}\big(\mathbf{q}(t), t\big) = \sum_{e=1}^{E} \mathbf{f}_e\big(\mathbf{q}(t), t\big) \in \mathbb{R}^{2N}.$$

Equivalently, the stored elastic energy is

$$U(\mathbf{q}, t) = \tfrac{1}{2}\sum_{e=1}^{E} k_e\big(\ell_e(\mathbf{q}) - L_e(t)\big)^2, \qquad \mathbf{f}_{\text{int}}(\mathbf{q}, t) = -\frac{\partial U}{\partial \mathbf{q}}.$$

Actuation via FitzHugh–Nagumo Oscillators

Each pixel $p = 1, \dots, P$ is actuated by a FitzHugh–Nagumo (FHN) oscillator with states (V_p, W_p) (dimensionless),

$$\dot{V}_p = V_p - \frac{V_p^3}{3} - W_p + I_{\text{base},p}, \tag{5.1}$$

$$\dot{W}_p = \frac{V_p + a_p - b_p W_p}{\tau_p}, \qquad \tau_p > 0 \ \ \text{[s]}, \tag{5.2}$$

where $I_{\text{base},p}$ is a constant input. The oscillator modulates the two diagonal springs in pixel p through

$$L_{\text{rest},e}(t) = L_{0,\text{base}} + \alpha\, V_p(t), \qquad e \in \mathcal{D}_p, \tag{5.3}$$

with α [m] and $L_{0,\text{base}}$ [m]. For $\mathcal{O}(1)$ excursions of V_p, α sets the diagonal stroke amplitude (m). Edge springs retain fixed rest lengths.

For compactness, we define disjoint index sets $\mathcal{E}_{\text{edge}}$ and $\mathcal{E}_{\text{diag}} = \bigcup_p \mathcal{D}_p$, and let $\mathbf{V} = [V_1, \dots, V_P]^\top$. With a diagonal-incidence matrix $\mathbf{A}_{\text{diag}} \in \{0,1\}^{E \times P}$ that has exactly two nonzero values per column (the two diagonals of pixel p), the assembled rest-length vector reads

$$\mathbf{L}(t) = \mathbf{L}_{\text{edge}} + \mathbf{A}_{\text{diag}}\big(L_{0,\text{base}}\mathbf{1}_P + \alpha\,\mathbf{V}(t)\big) \in \mathbb{R}^E.$$

Environment, Contact, and Friction

Static obstacles are given as polygons via axis-aligned bounding boxes (broad-phase culling), vertex lists, and edge connectivity. For node i, we compute the closest point $\mathbf{p}_i^\star$ on the polygon boundary by orthogonal projection onto edges; outward normal orientation is enforced consistently by reference to the polygon centroid. Let $\mathbf{n}_i$ be the outward unit normal at $\mathbf{p}_i^\star$. The signed penetration depth (positive when the node is *inside*) is

$$d_i = \max\Big\{0,\ -\mathbf{n}_i^\top\big(\mathbf{x}_i - \mathbf{p}_i^\star\big)\Big\}\ \ \text{[m]}, \qquad \mathbf{x}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}.$$

We apply a linear penalty in the normal direction and a linear tangential dashpot,

$$\mathbf{f}_{\text{pen},i} = k_c\, d_i\, \mathbf{n}_i - c_\mu\big(\mathbf{I}_2 - \mathbf{n}_i\mathbf{n}_i^\top\big)\,\mathbf{v}_i \in \mathbb{R}^2 \quad \big[\text{N}\big], \tag{5.4}$$

with k_c [N/m], c_μ [N s/m], and $\mathbf{v}_i = \dot{\mathbf{x}}_i$. Contact handling is applied to a designated subset of nodes (mass threshold) to determine which parts of the structure interact with the substrate. A uniform gravitational body force acts in $-y$,

$$\mathbf{f}_g = \mathbf{M}\,\mathbf{g}, \qquad \mathbf{g} = \begin{bmatrix} 0 & -g & \cdots & 0 & -g \end{bmatrix}^\top \ \big[\text{m/s}^2\big].$$

Equations of Motion in Assembled Matrix Form

To minimize the computations, contact is only checked for the nodes on the boundary of the structure. Let $\mathcal{I}$ denote the set of nodes that participate in contact handling, and let $\mathbf{P}_i \in \mathbb{R}^{2\times 2N}$ select the (x_i, y_i) block:

$$\mathbf{P}_i = \begin{bmatrix} \cdots & 1 & 0 & \cdots \\ \cdots & 0 & 1 & \cdots \end{bmatrix}, \qquad \mathbf{f}_{\text{pen}}(\mathbf{q}, \dot{\mathbf{q}}) = \sum_{i\in\mathcal{I}} \mathbf{P}_i^\top \, \mathbf{f}_{\text{pen},i}.$$

The coupled mechanical–actuation system is

$$\boxed{\mathbf{M}\,\ddot{\mathbf{q}}(t) \;+\; \mathbf{C}(t)\,\dot{\mathbf{q}}(t) \;+\; \mathbf{f}_{\text{int}}\big(\mathbf{q}(t), t\big) \;+\; \mathbf{f}_{\text{pen}}\big(\mathbf{q}(t), \dot{\mathbf{q}}(t)\big) \;=\; \mathbf{f}_g}\,, \tag{5.5}$$

coupled to (5.1)–(5.2) for each pixel. Unless otherwise stated, all nodes are free; any kinematic constraints (if used) are specified where results are presented.

Numerical Integration and Stability Considerations

Equations (5.2)-(5.5) are integrated using an adaptive Dormand–Prince explicit Runge–Kutta scheme of order $5(4)$ (Boost.Odeint). Unless stated otherwise, the second-order system is expressed as a first-order ODE in $[\mathbf{q}, \dot{\mathbf{q}}]$ for numerical integration. Absolute and relative tolerances are 10^{-4}, the initial trial step is 0.1 s, and simulations run to $T = 300$ s. State snapshots are recorded every 0.1 s for analysis. Spring-force evaluations are vectorized with AVX (batch size 8) to accelerate large lattices without altering the continuous model.

As k_c increases, the contact stiffness shortens the fastest time scale; we verified that the tolerance and step settings in Table 5.1 keep accepted steps within stability bounds for the explored parameter ranges.

Initialization and Boundary Data

Initial node positions, velocities, spring data (connectivity, stiffnesses, edge rest lengths), oscillator states and parameters $(a_p, b_p, \tau_p, I_{\text{base},p})$, and environment geometry (polygons) are loaded from external files. Transient environmental inputs $I_{\text{env},p}(t)$ are activated when a pixel's

Table 5.1: Baseline simulation parameters.

Mechanical network

Quantity (symbol / config key)	Value	Units / Notes
Edge spring stiffness (k_s / `spring_constant`)	300	N/m
Diagonal spring stiffness (k_{diag} / `diag_spring_constant`)	300	N/m
Node mass (m_i / `mass`)	0.073	kg
Edge rest length (L_{edge} / `rest_length`)	0.025	m
Height offset (h_{offset} / `hight_offset`)	0.01	m (initial vertical offset)
Gravity (g)	9.8	m/s^2 (acts in $-y$)
Contact normal stiffness (k_c)	1000	N/m
Tangential dashpot (c_μ)	100	N s/m

Actuation (FitzHugh–Nagumo)

Quantity (symbol)	Value	Units / Notes
FHN parameters (a, b, τ)	$(0.7, 0.8, 12)$	τ in s; V, W dimensionless
Bias; environmental input (I_{base}; I_{env})	const.; impulse on contact	Applied for one accepted step then reset
Diagonal actuation map L_{rest}	$L_{0,base} + \alpha V(t)$	Both diagonals share this rest length
Diagonal base rest length ($L_{0,base}$)	$L_{edge} * \sqrt{2}$	m
Actuation gain (α)	0.04	m per unit V

Damping / drag

Quantity (symbol)	Value	Units / Notes
Bulk viscous drag $c(t)$	0.6 for $t < 1.5$; 0.2 for $t \geqslant 1.5$	N s/m; isotropic dashpot on all DOFs

Time integration and output

Quantity	Value	Units / Notes
Integrator	Dormand–Prince RK45 (adaptive, Boost.Odeint)	Explicit, error-controlled
Absolute / relative tolerances	10^{-4} / 10^{-4}	—
Initial step / final time / log interval	0.1 / 300 / 0.1	s / s / s
Environment specification	polygons (bbox, vertices, edges)	Loaded from external files
Output metrics	COM displacement; pixel strains; oscillator traces	Recorded every 0.1 s

node(s) enter contact and are applied for one accepted integrator step, after which they are reset to zero.

Computational Setup

Simulations are coded in C++ for the core solver, using Eigen for matrix operations and SIMD for efficient batched spring force computation. Data analysis and visualization are handled in Python using Jupyter notebooks, matplotlib, and OpenCV. Computations run on a Windows workstation (Intel i7, 8 cores).

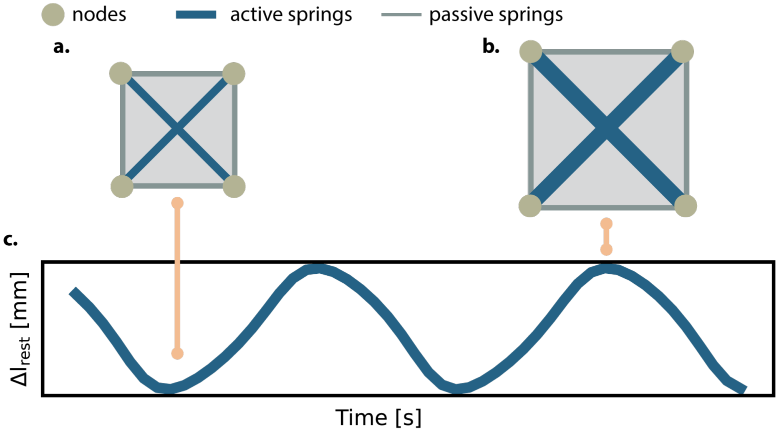


Fig. 5.3: **Minimal mass-spring model of a body with actuation coupled to a FHN oscillator.** **a**, Mass-spring representation of a single pixel consisting of four nodes and six springs, with two active springs represented in blue. **b**, Changing the rest length of the active springs causes symmetric inflation of the cube. **c**, The equilibrium length of both active springs coupled to the output of the FHN oscillator.

Geometry and Oscillator Coupling Optimization Approach

We conducted geometry optimization of our pixel-based mass-spring system using a Genetic Algorithm (GA). Genetic algorithms are stochastic optimization methods inspired by the process of natural selection and evolution. They iteratively improve a population of candidate solutions through operations such as selection, crossover, and mutation, guided by an objective (fitness) function. Each candidate in the GA represents a unique pixel structure, defined by a float-valued matrix where each entry specifies pixel presence and oscillator coupling status. The GA begins by initializing a diverse population of candidate geometries. Each candidate undergoes mechanical simulation, and its fitness is quantitatively evaluated based on mechanical performance metrics, such as stability, efficiency of locomotion, structural integrity, and compliance. Candidates with superior fitness scores have higher probabilities of being selected for subsequent breeding (crossover) and mutation processes, enabling progressive improvement of geometries over successive generations.

Geometry Parameters and Constraints

The geometry parameters are encoded as a 5x5 float-valued matrix. Each matrix element defines the properties of a pixel in the following way:

- Positive values represent a pixel.
- Negative values indicate the absence of a pixel.
- For the positive floats, the range [0,1] is equally divided over the active and passive roles the pixel can play. For a system with a single oscillator, values smaller than or equal to 0.5 are considered passive, while those larger than 0.5 couple to the oscillator.

To ensure mechanical feasibility, constraints are explicitly enforced before each simulation of a new specimen is executed. The critical constraint is structural connectivity: all present pixels in an individual specimen structure must form a connected unit without isolated elements. Connectivity is determined using an eigenvector analysis of the mass-spring stiffness matrix, computed as follows:

1. The stiffness matrix of the structure is assembled based on pixel presence and spring connections.
2. Eigenvalue decomposition is performed to obtain eigenvalues and eigenvectors.
3. Structures displaying eigenvectors corresponding to zero or near-zero eigenvalues indicate mechanical disconnection or trivial deformation modes.
4. Candidates failing the connectivity test are subject to targeted mutations (mutations on the mechanism and its direct neighbors) designed to reconnect isolated or loose pixels with minimal geometric changes.

This rigorous eigenvector-based connectivity analysis ensures that the structures created from cossovers and mutations are mechanically meaningful and feasible for physical realization. Moreover, this constrains and speeds up the optimization process as a large portion of the low-quality specimens are filtered out and replaced in each generation.

Numerical Implementation

The genetic algorithm was implemented in Python, leveraging libraries such as NumPy for numerical computations, SciPy for mathematical optimization routines, and multiprocessing for parallel execution. Custom scripts were developed to handle specific genetic operations and constraint evaluations.

Genetic Algorithm Parameters:

- **Population size**: 500 individuals per generation, balancing genetic diversity and computational manageability.
- **Number of generations**: 100 generations, chosen empirically to achieve convergence within computational constraints.
- **Mutation rate**: 0.6, a high mutation rate selected to maintain genetic diversity and avoid premature convergence.
- **Elitism fraction**: 0.2, ensuring the top 20% highest-fitness individuals are directly preserved to subsequent generations.

Crossover and Mutation Methods:

- **Single-point crossover**: Selects a random crossover point within the candidate matrices, combining rows or columns from two parent matrices to generate offspring.
- **Two-point crossover**: Uses two randomly selected crossover points to combine segments of parent matrices.
- **Uniform crossover**: Randomly combines elements from parent matrices based on probabilistic masks.
- **Random float mutation**: Adds uniformly distributed random values to elements of candidate matrices within specified mutation bounds.
- **Row-column mutation**: Randomly selects entire rows or columns and applies uniform random perturbations to those segments.
- **Block mutation**: Randomly selects sub-blocks within matrices, perturbing multiple elements simultaneously to encourage structural diversity.

Candidates were initialized with random matrices drawn uniformly from the range $[-1, 1]$, ensuring broad initial genetic variability. The bounds for all elements were strictly enforced within the same range during mutation and crossover operations to maintain numerical stability. Once the candidates are selected for crossover, one of the crossover methods is selected with an equal probability for each; the same holds for the mutations.

The genetic algorithm utilized tournament selection to pick candidates for reproduction, with tournament sizes of 3, ensuring balanced selective pressure towards higher fitness individuals without overly diminishing genetic diversity.

Stopping criteria were based on completing a predefined number of generations (100), with interim convergence monitored through fitness score distributions.

Computational Efficiency and Performance

Optimization efficiency was enhanced through parallelization using Python's multiprocessing library, enabling simultaneous evaluation of multiple candidate geometries. Additional computational improvements included caching intermediate simulation results to avoid redundant calculations, strategic serialization of data for quick access, and real-time monitoring and visualization of optimization progress to promptly identify convergence or stagnation.

5.3 EVOLVING A BODY FOR LOCOMOTION

As a first optimization and demonstration, we start with a single oscillator (Fig. 5.4a) and optimize the morphology of a soft voxel-based design with three options for each voxel in a 5x5 lattice. Each voxel can be a passive unit where the diagonal springs are passive and not coupled to an oscillator. It can be an active unit where the diagonal springs are coupled to the oscillator and the rest length of the diagonal springs is modulated by the output of the oscillator. Lastly, a unit can be absent in the voxel. The system is tasked with moving towards the right, and the fitness function is the distance it covers over 60 seconds of simulation time. Fig. 5.4b shows how the morphology of the best specimen develops over the generations. We observe clear signs of convergence in the solution as the change in morphology decreases for later generations. In Fig. 5.4c, we observe snapshots of the walking gait over time for the best specimen of the first generation. As we start from a random population, we find that even for a randomly drawn set of 500 samples, we already see movement in the target direction.
The same snapshots for the optimal morphology after 100 generations are displayed in Fig. 5.5a. Here, we observe an apparent increase in displacement over the same time interval. Furthermore, we find the absence of a voxel in the middle of the structure. Interestingly, it appears that the system uses this absent voxel as a hinge to increase deformation. We find that the maximum fitness of the solutions doubles throughout the 100 generations, as depicted in Fig. 5.5b. For the average fitness, we find that although it improves over generations, it converges to a

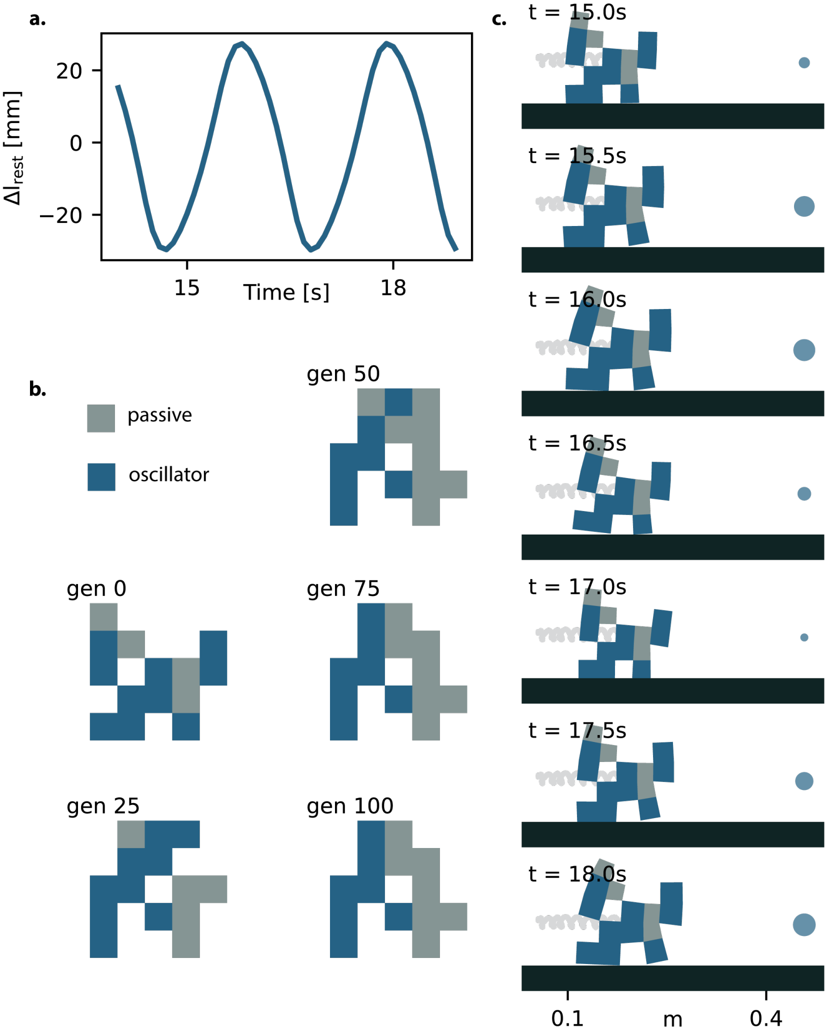


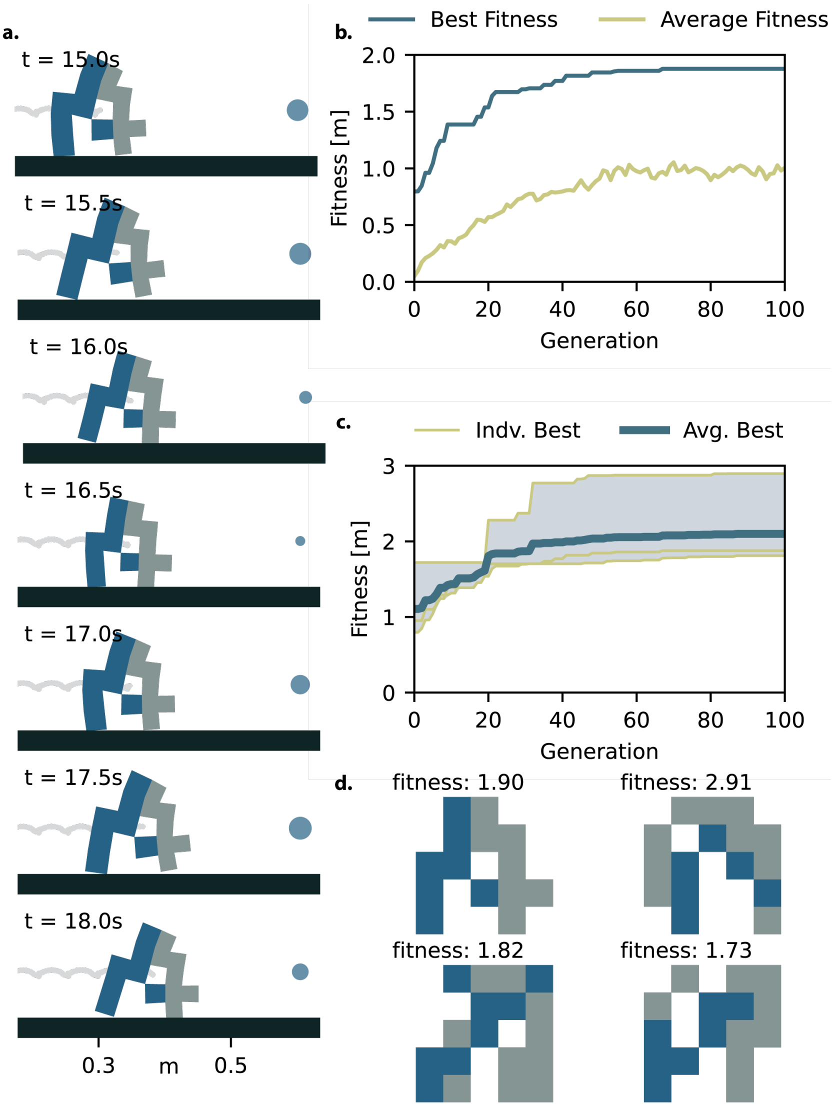
Fig. 5.4: **Optimizing geometry for locomotion.** **a**, Output of a single oscillator. **b**, Best performing morphology over the generations. **b**, Snapshots of the walking gait of the best-performing specimen of the first generation. The size of the blue circle on the right indicates the amplitude of the oscillator at the moment of the snapshot.

value lower than the maximum fitness, indicating that even after 100 generations, there is still considerable diversity in the population.

Next, we repeat the same evolutionary optimization process four times. The trajectories of the maximum fitness show that three of the four simulations converge to fitness values around 2. Only a single optimization significantly outperforms the others, achieving a final fitness value of nearly 3. This could be an indication of the presence of many local minima, which make it harder for individual runs to find a global maximum. This is expected, as even this seemingly simple problem has 3^{25} solutions. Fig. 5.5d compares the morphologies of the best solutions. We find that all solutions use this hinge mechanism to increase the amplitude of the actuation. Interestingly, for the best performer overall (top right), we observe a half-circle with an inner layer of active voxels and an outer layer of inactive voxels, which vaguely resembles the actuation of a bi-layer material to increase the stroke.

5.4 MULTIPLE OSCILLATORS

To increase the complexity of the possible solutions and behaviors, we next perform simulations with more than one oscillator. Similar to CPG controllers, we couple the oscillators using directional additive coupling. This means that the coupling of the oscillators can exhibit both excitatory (positive coupling) and inhibitory (negative coupling) interactions. By coupling multiple oscillators together in a coupled network, the coupling dictates the phase offset or the sequence between the oscillators.



Caption on next page.

Fig. 5.5: **Outcome of the geometry optimization with a single oscillator. a**, Walking gait of the best-performing specimen of the last generation (100). The size of the blue circle on the right indicates the amplitude of the oscillator at the moment of the snapshot. **b**, Fitness over the generations. **c**, Results of four repetitions of the same optimization. In green, the best fitness over time for the individual optimizations. In blue, the average of the best fitness among the optimizations. In light blue, the range of the best fitness for all the optimizations. **d**, The optimized structures of the initial optimization and the best over the four optimizations.

We consider a system of n FitzHugh–Nagumo oscillators with additive coupling. The state of each oscillator $i \in \{1, 2, \ldots, n\}$ is defined by a voltage-like variable $V_i(t)$ and a recovery variable $W_i(t)$, governed by:

$$\dot{V}_i = V_i - \frac{V_i^3}{3} - W_i + I_{\text{base},i} + \gamma \sum_{j=1}^{n} A_{ij} V_j, \tag{5.6}$$

$$\dot{W}_i = \frac{V_i + a - bW_i}{\tau}, \tag{5.7}$$

with parameters:

$$a = 0.7, \quad b = 0.8, \quad \tau = 12, \quad \gamma \in \mathbb{R}.$$

The matrix $A \in \mathbb{R}^{n \times n}$ defines the coupling topology, where $A_{ij} = 1$ if oscillator j influences oscillator i, and $A_{ii} = 0$. Each oscillator's voltage $V_i(t)$ remains linearly mapped to the rest length of both diagonal springs:

$$L_{rest,i}(t) = L_{0,\text{base}} + \alpha V_i(t), \tag{5.8}$$

where $L_{0,\text{base}}$ is the base rest length and α is a scaling factor.
Assuming a fully coupled network of three oscillators, equal positive coupling between the oscillators will result in in-phase synchronization of the oscillators, as depicted in Fig. 5.6a, regardless of the starting conditions of the oscillators. The effective coupling matrix for this simulation is presented in Fig. 5.6b, where we use the same positive value for all the couplings (0.05), and because we do not consider

coupling of an oscillator to itself, the diagonal remains zero. Similarly, if we start with the same initial conditions and equal negative coupling (-0.05) between the oscillators, Fig. 5.6c-d show how the oscillators drive each other out of phase, resulting in phase-locked oscillators with equal phase differences between the oscillators. The behavior becomes more interesting when we assign random couplings between the oscillators in Fig. 5.6e-f. Now we find a combination of negative and positive couplings that can steer the phase differences and slightly adjust the amplitudes of the oscillators.

Whereas, for a single oscillator, the evolutionary algorithm only optimized the morphology of the structure, the addition of multiple oscillators poses a vast increase in the complexity of the search space. For more than one oscillator, we now optimize both the structure and network architecture of the oscillators. As an example, for two oscillators, each voxel now has four possible states (absent, passive, oscillator 1, oscillator 2). In tandem with the morphology, the initial state and the two couplings between the oscillators are optimized to find a control sequence that aids in the displacement of the system.

To investigate how the evolutionary algorithm handles the additional complexity of the optimization problem, we repeat the same optimization task of moving towards the right for increasing numbers of oscillators in Fig. 5.7a. We observe that the first increase in fitness occurs when the system transitions from a single oscillator to two oscillators, roughly doubling the average maximum fitness achieved during optimization. Overall, we observe an asymptotic trend where the increase from three to four oscillators yields only a minimal increase in fitness. Fig. 5.7b-c depict the maximum fitness trajectories of the optimizations for three and four oscillators, respectively. Interestingly, we find very similar variance between the three- and four-oscillator systems. This could suggest that the asymptotic convergence for the increasing number of oscillators in Fig. 5.7a is not caused by a search space that is too complex for the optimization, but rather a saturation of complexity needed for the task. This is a key insight, as it suggests that there is a limit to the extent to which system complexity is beneficial to the fitness outcome, given a specific task and geometry size.

Evaluating the outputs produced by the best-performing two- and four-oscillator networks in Fig. 5.7d-e, we find that the evolutionary process is both modulating the amplitude and the phases. Even for

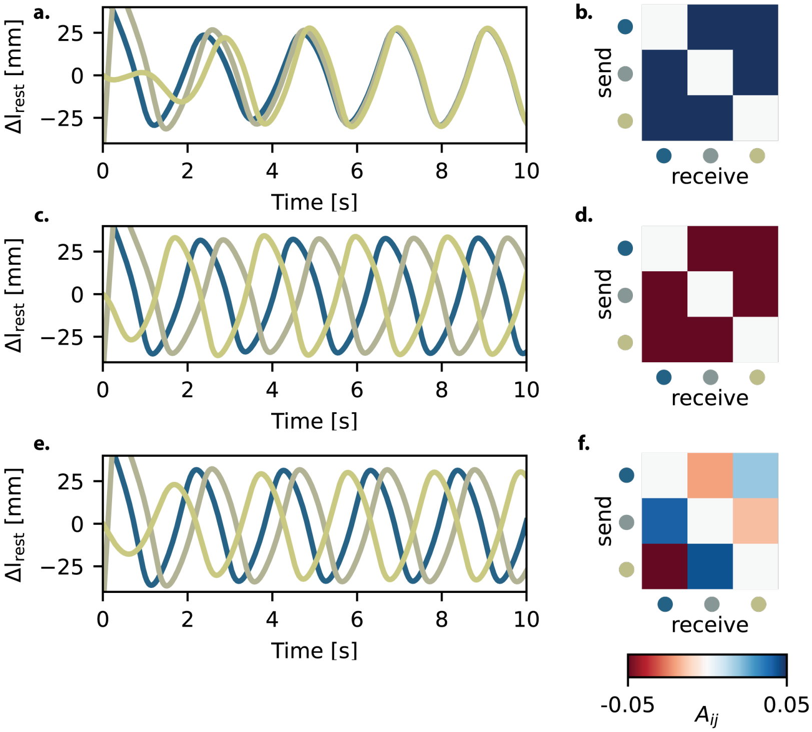
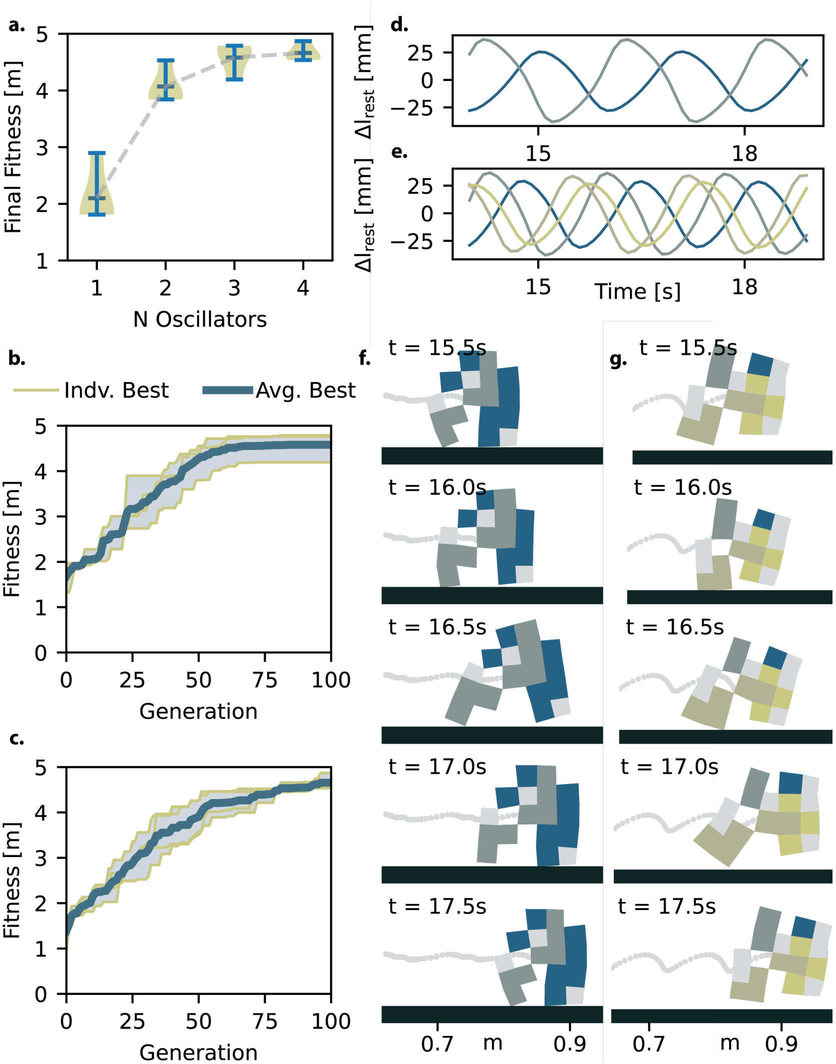


Fig. 5.6: **Coupling multiple oscillators. a-b**, Three oscillators with equal positive coupling between them. **a** depicts the output of the oscillators over time, and **b** represents the coupling matrix for the three oscillators. **c-d** represent the three oscillators with the same initial conditions but with equal negative coupling between the oscillators. **e-f**, The three oscillators with the same initial conditions and random coupling between them.

the four-oscillator system, we see a coupling matrix that diversifies the outputs by phase-locking the oscillators at different phase differences and different amplitudes. In the snapshots of the system over time, we observe similar behavior to that of the single-oscillator system, where a rear limb with a hinge mechanism is used to increase deformation in Fig. 5.7f-g. For the four-oscillator systems, we find that although they often utilize all the different voxels, an additional diversification of the fourth oscillator often appears to be redundant. As exemplified in Fig. 5.7g, where the diversification of the voxel coupled to the oscillator in blue appears to be redundant, as it is only used once in a position that seems to have little effect on the global deformation of the system. Taking these results and the aim of exploring architectures for future experimental systems into account, we focus further simulations and results on three-oscillator systems. An example of the evolutionary process of three oscillators is illustrated in Movie 5.1.

5.5 TWO DIFFERENT GAITS APPEAR

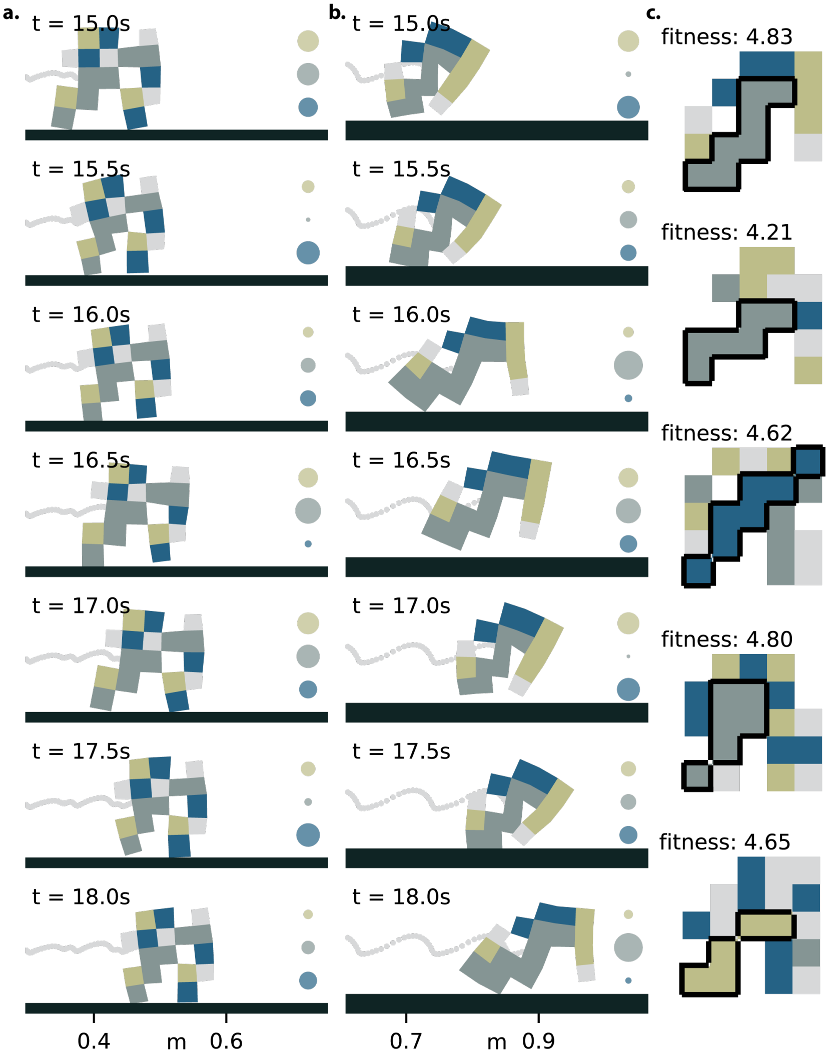
In total, we performed eight simulations for the three-oscillator system. Two qualitative observations are worth noting: in six of the eight cases, we find a gait transition during the optimization. In these cases, the system finds a two-limb solution within the first ten generations. These limbs are used to develop a gait in which the weight constantly shifts back and forth between the front and rear limbs. There is minimal or no flight, with the body in contact with the ground most of the time, similar to a canter or walking gait.



Caption on next page.

Fig. 5.7: **Locomotion with varying numbers of oscillators. a**, Optimization results for increasing numbers of oscillators. Each distribution consists of four repetitions of the same optimization. **b-c**, Optimization trajectories of the four repetitions for three oscillators and four oscillators, respectively. **d-e**, Optimized oscillator outputs for the optimizations with two and four oscillators, respectively. **f-g**, Optimized structures and gaits for the two (**f**) and four-oscillator (**g**) optimizations.

Fig. 5.8a shows an example of this gait with snapshots of the best-performing optimization at generation 50. The hindlimb and forelimb both contribute to propulsion. In contrast, the gait illustrated in Fig. 5.8b more closely resembles a gallop, bound, or hop. This is a faster gait: the hindlimb strikes the ground, followed shortly by little or no contact for the forelimb. A clear aerial phase follows, during which all limbs are off the ground. This gait emphasizes speed and dynamic movement, characterized by pronounced extension and flexion of the spine, as well as a more forceful push-off by the hindlimb (Movie 5.1). Compared to the first gait in Fig. 5.8a, the gallop-like gait in Fig. 5.8b shows greater stride length, reduced ground contact time, and increased reliance on the hindlimb for propulsion. The convergence towards this gallop-like locomotion can also clearly be observed in the best-scoring morphologies for each evolutionary optimization. Fig. 5.8c shows five examples, where, for all five examples, we find an interconnected section of the same type of voxel that reaches from the hindlimb up to the spine. By expanding and contracting these sections in synchrony, the body generates the spinal extension and push-off by the hindlimb, similar to the gallop of the cheetah.



Caption on next page.

Fig. 5.8: **Gaits for three-oscillator geometries. a**, Canter gait found as the best specimen at generation 50 for the optimization with three oscillators. **b**, Gallop gait found as best specimen at generation 100 of the same optimization as **a**. **c**, Best five specimens over eight optimizations with three oscillators. The black outline indicates the observed similarity between the specimens: a diagonal arrangement of the same oscillator that controls the rear limb and mid-body.

5.6 CLIMBING IN A CONFINED SPACE

To build towards adaptive behavior, we next investigate the three-oscillator system for a different task, climbing in a confined space. For this task, we apply the same parameters as for the previous task, but change the environment by adding two walls spaced 0.15 m apart. The fitness function is changed to the average position of the system on the y-axis after 60 s of simulation time. Fig. 5.9a depicts the best-performing first-generation specimen out of four repetitions of the optimization. In contrast to the running task, where we already found walking specimens in the random initial generation, we find that the initial generations of the climbing optimization are barely able or unable to climb. An example of the evolutionary process is illustrated in Movie 5.2.
Nonetheless, the optimization is able to adapt the structure and oscillator architecture to achieve climbing after 100 generations (Fig. 5.9b). From the optimization trajectories in Fig. 5.9c, we observe that in the first generation of all four repetitions, the average position of the system after 60 s of simulation time does not exceed the height of the structure. Despite this steep increase in task complexity, we find that for all four optimizations, the system evolves to climb, increasing its fitness by 15 to 20 times over 100 generations. The outputs of the oscillators over time for the first and last generations in Fig. 5.9d and Fig. 5.9c, respectively, show how the evolution has both shifted the phase lag between the oscillators and used the coupling to increase the amplitude of the oscillator in blue. These results indicate that the system can handle the increase in task complexity, enabling us to test its ability to switch between tasks of climbing and running.

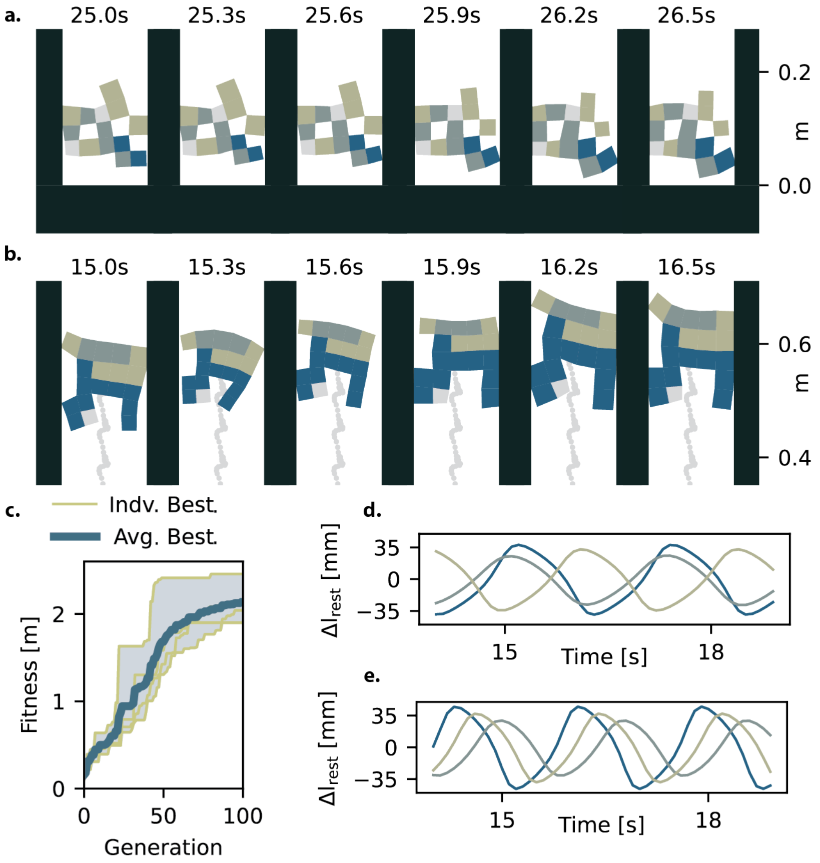


Fig. 5.9: **Climbing as increased task complexity. a-b**, Snapshots of the gaits and trajectory during the first and last generation, respectively. **c**, Climbing fitness over the generations for four repetitions of the same optimization. **d** depicts the oscillators of the best specimen during the first generation, and **e** depicts the oscillators of the last generation.

5.7 ENVIRONMENTAL FEEDBACK

For the system to adapt its behavior to the task, it needs a way to sense changes in its task. Inspired by how CPGs in nature process sensory information, we aim to couple the network of oscillators directly to the environment. We directly couple the input current of the oscillators to the environment, similar to how the coupling between oscillators modulates their input current. Although there are numerous ways to implement this coupling, our primary consideration is that the implementation must be as computationally inexpensive as possible to maintain the feasibility of running these large, computationally intensive optimizations.

During simulations, we model the environmental coupling to each oscillator using a simple heuristic. Specifically, when a voxel associated with a given oscillator comes into contact with the environment (interpreted as penetration or sustained contact), the oscillator's input current is increased from a baseline value of 0.4 to 0.8 for the duration of the contact. This modulation serves as a computationally inexpensive substitute for mechanical interaction with the environment. In an experimental setting, this would be analogous to mechanically kinking or unkinking a tube that leaks air parallel to the oscillator's inlet Chapter 3. Fig. 4.7a. Fig. 5.10a shows the results of four repetitions of the climbing task for the adaptive oscillator network. We find significantly more variance in the fitnesses of the four repetitions. Although the average best fitness is relatively similar between the adaptive and the non-adaptive networks, Fig. 5.10b also demonstrates that the overall best specimen after 100 generations of the adaptive oscillator network significantly outperforms the non-adaptive counterpart. Snapshots of this specimen are presented in Fig. 5.10c. When we compare the limit cycles of the best non-adaptive and adaptive solutions in Fig. 5.10d and e, we find that both solutions find a stable limit cycle. However, we observe that the adaptive oscillator network follows a trajectory with small deviations, whereas the non-adaptive network exhibits no deviations once the limit cycle is reached (an example of a stable limit cycle for the non-adaptive oscillators is presented in Movie 5.3). Lastly, we find that the adaptive network reaches its limit cycle after a single oscillation, while the non-adaptive network needs three cycles to reach its limit cycle. These results demonstrate that the coupling with the environment makes the optimization problem more complex, as seen by the variation in the quality of the solutions. However, they also demonstrate that it is capable of finding solutions that outperform the non-adaptive oscillators, which could be the result of increased robustness.

To further investigate the quality of the solutions found in Fig. 5.10, we investigate the robustness of the solutions to perturbations to the initial conditions of the oscillators and the coupling between the oscillators by repeating the simulation of the best result for each optimization repetition (four in total). For each repetition, we draw 100 perturbations from a normal distribution with a mean of zero and a fixed standard deviation. This process is repeated for increasing standard deviations, and the average fitness within each standard deviation is then normalized by the original fitness of the solution. We first compare the adaptive networks with the non-adaptive networks in response to perturbations in the initial conditions, for both the climbing task in Fig. 5.11a and the running task in Fig. 5.11b. We find that the adaptive networks are more robust to initial conditions. The same simulation setup is also used to test robustness against perturbations in the coupling network for climbing and running, as shown in Fig. 5.11c-d. We observe a similar effect, where adaptive networks exhibit greater overall robustness and become even more robust during the task with increased interaction with the environment. One explanation for this difference is that, in the case of the adaptive network, the environment acts as additional coupling, making the system more strongly coupled overall. This also supports the results found in Fig. 5.10d and e, where an adaptive network resulted in a shorter transient before reaching the limit cycle. Overall, these results demonstrate how sensory feedback from the environment leads to more robust behavior, even in this minimal network with a fixed objective.

5.8 ADAPTATION TO THE TASK

While the adaptive networks in running and climbing tasks show that the adaptive networks are capable of finding higher-performing solutions, the system still settles in stable limit cycles (Fig. 5.10e). This is likely because the interactions with the environment are relatively stable. To explore the system's capabilities, we next perform multiobjec-

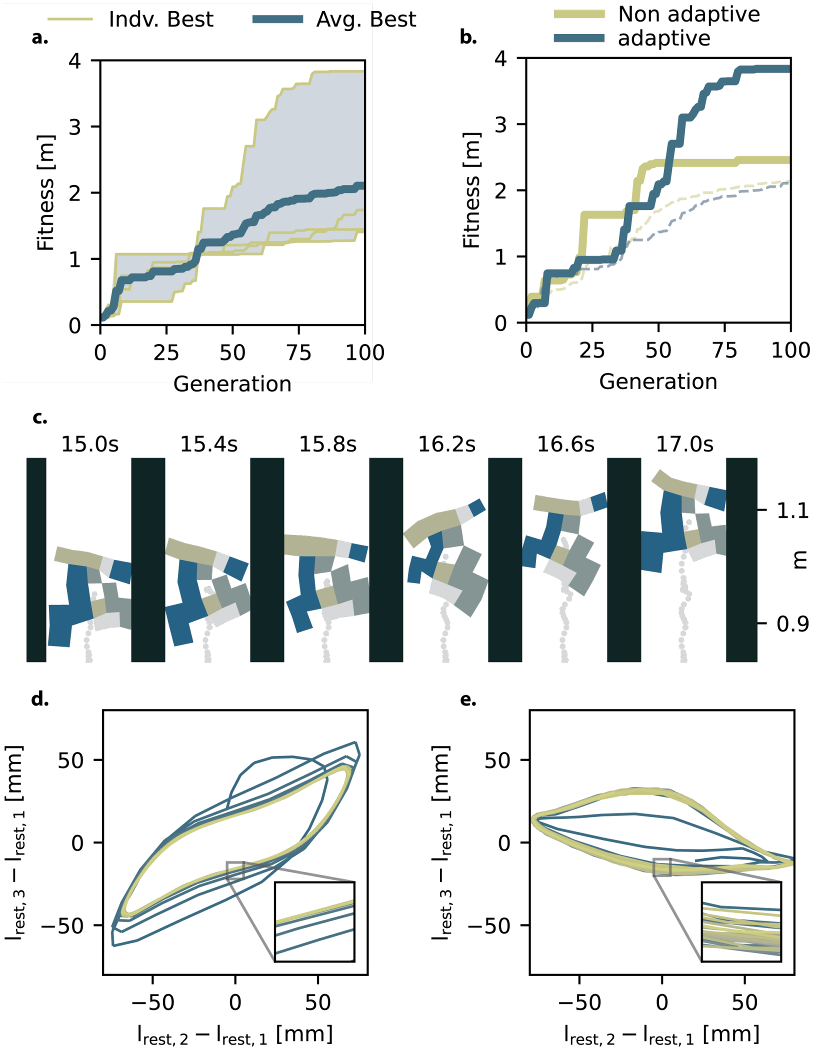


Fig. 5.10: **Making the oscillator network adapt to the environment.** **a**, Fitness over the generations for adaptive oscillators in the climbing optimizations over four repetitions of the optimization. **b**, Comparison of the average best fitness over four repetitions for 100 generations for the adaptive and non-adaptive oscillators. **c**, Snapshots of the best specimen for the adaptive oscillator optimizations. **d-e**, Comparison of the oscillator limit cycles for the best non-adaptive (**d**) and adaptive (**e**) climbing optimizations.

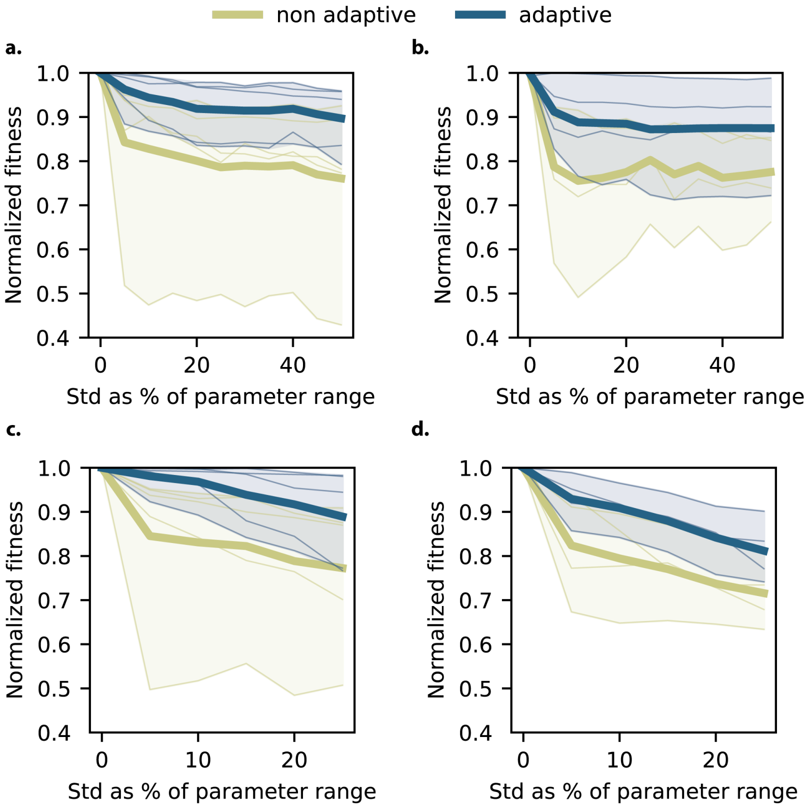


Fig. 5.11: **Robustness of adaptive and non-adaptive networks. a–b**, Robustness against perturbations in the initial conditions for the climbing task (**a**) and the running task (**b**). **c–d**, Robustness against perturbations in the coupling network for the climbing task (**c**) and the running task (**d**). For each optimization repetition (four in total) as presented in earlier figures, the best solution was resimulated under 100 perturbations drawn from a normal distribution with zero mean and increasing standard deviation. The original fitness of the solution normalized the average fitness within each standard deviation.

tive optimizations while switching tasks and environments. During the first stage of the optimization, the climbing task is active (as described in Fig. 5.9). When the system has climbed 0.8 m, the environment is switched to stage two, as depicted in Fig. 5.12a. The side wall disappears, and a horizontal floor appears at 0.6 m. The fitness function during optimization is set to the sum of the y-axis and x-axis displacements of the system. Effectively, the system optimizes for climbing during the first stage and running during the second stage of optimization. We perform four optimizations, both with the non-adaptive and adaptive oscillator networks, with the results presented in Fig. 5.12b and c, respectively. We find that the average best fitness score for both non-adaptive and adaptive oscillators is similar. Furthermore, similar to the climbing experiments, we see a larger variation in the results for the adaptive oscillator networks. The best-performing simulation of the adaptive networks has a fitness that is 1.5 times higher than that of the best-performing simulation of the non-adaptive networks. More interestingly, we observe vast differences in the limit cycles of the non-adaptive (Fig. 5.12d) and the adaptive (Fig. 5.12e) networks (Movie 5.4). The non-adaptive phase diagram exhibits a clear, stable limit cycle with a short transient of roughly four cycles, as shown in Fig. 5.12d. The adaptive oscillator simulation results in a phase diagram without a stable limit cycle, as shown in Fig. 5.12e, indicating that the oscillators change their respective behavior over time.

Fig. 5.13a and b show the climbing and running behavior of the single adaptive oscillator simulation corresponding to Fig. 5.12e. Although we can observe a difference in deformations between the two environments, this difference becomes much clearer when examining the oscillator outputs during these time intervals in Fig. 5.13c and d. We find that both the phase differences between the oscillators and the amplitudes have changed in response to the environmental change. Fig. 5.13e shows the gradual transition between the gaits as the environment is switched, around the 40 s mark. Reevaluating the phase diagram of the oscillators for the two isolated time intervals in Fig. 5.13f reveals two distinct limit-cycles for the two environments. Importantly, the system remains the same, and the change in behavior is solely due to the change in environment.

These results demonstrate that coupling oscillators to the environment, in conjunction with a co-evolved morphology and oscillator network,

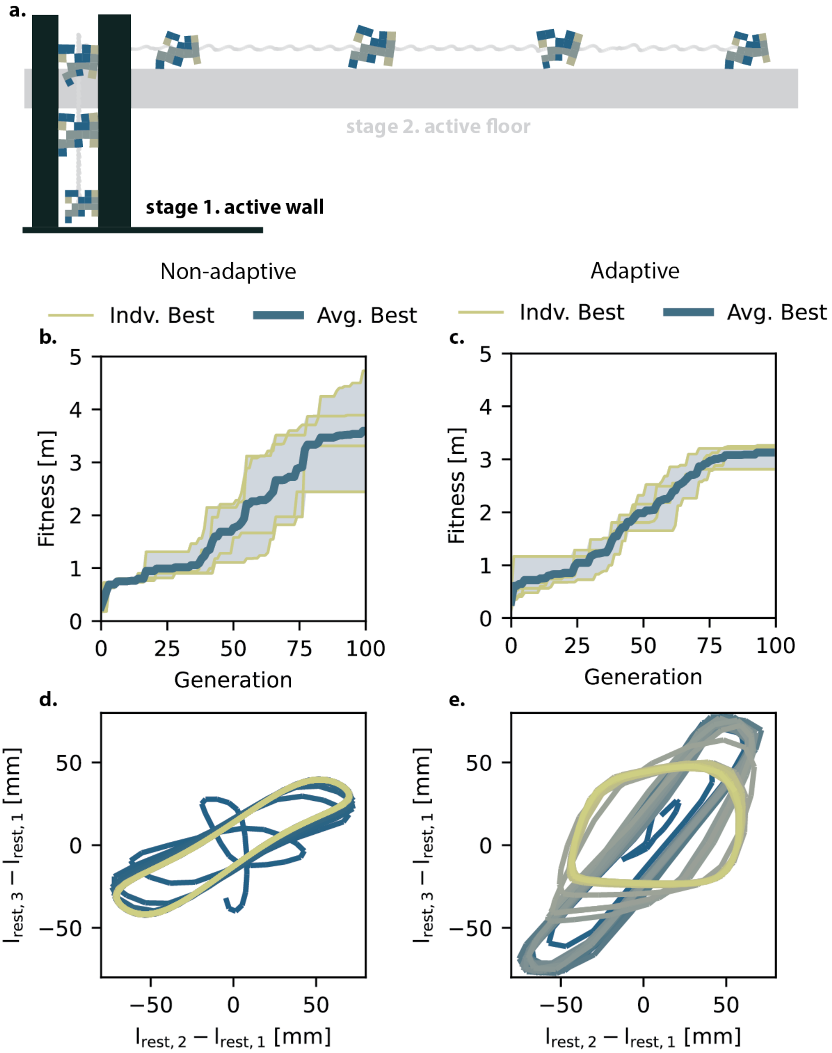


Fig. 5.12: **Optimizations with switching tasks and environments. a**, Schematic of the switching simulations: in stage one, the climbing task is active (as in Fig. 5.9); once the system has climbed 0.8 m, the environment switches to stage two, where the side wall disappears and a horizontal floor appears at 0.6 m. The fitness function is defined as the sum of vertical and horizontal displacements. **b–c**, Results of four optimizations with non-adaptive (**b**) and adaptive (**c**) oscillator networks. **d–e**, Phase diagrams of the best simulations for non-adaptive (**d**) and adaptive (**e**) networks.

can lead to task-dependent behavioral switching, even for networks with a minimal number of oscillators. Moreover, the intrinsic stability of the FHN oscillators facilitates smooth transitions between the limit cycles.

5.9 BINARY INPUTS FOR CONTROL

As a last demonstration, we show how the behavior of these minimal oscillator networks can also be controlled by higher-order control, as depicted in Fig. 5.1a-b. For this demonstration, we add a control oscillator to the network that can be turned on and off by setting its base current I_{base} from 0.8 to zero; the network is depicted in orange in Fig. 5.14a. Our goal is to have the system move to the left when the oscillator is off and to the right when the oscillator is on, thereby emulating how the system could be controlled by a sensor or higher-order brain (e.g., MCU) from a single input (oscillator on or off). To evolve the system for both tasks, we run the simulation twice for every specimen, once with the control oscillator on and once off. To force equal priority for both tasks, we employ a geometric mean (emphasis on balance) of the two simulations.
We define the fitness function as

$$F = \sqrt{(d_- + \varepsilon)(d_+ + \varepsilon)}, \quad \varepsilon > 0$$

where

$$d_- = \max\bigl(0, -\Delta x_A\bigr), \qquad d_+ = \max\bigl(0, \Delta x_B\bigr).$$

Here, d_- is the distance moved in the negative x-direction in simulation A (oscillator off), and d_+ is the distance moved in the positive x-direction in simulation B (oscillator on). The small constant ε prevents the fitness from becoming zero when either distance is zero. This geometric mean formulation ensures that both d_- and d_+ must be large to achieve high fitness, thereby rewarding a balance between the two simulations. Fig. 5.14b shows the best-performing specimen after 100 generations of optimization. We find that the control oscillator is only used in a single pixel, likely because it is turned off for the first simulation and is therefore considered less useful. Fig. 5.14c-d show the oscillators over time for simulations A and B, respectively. These

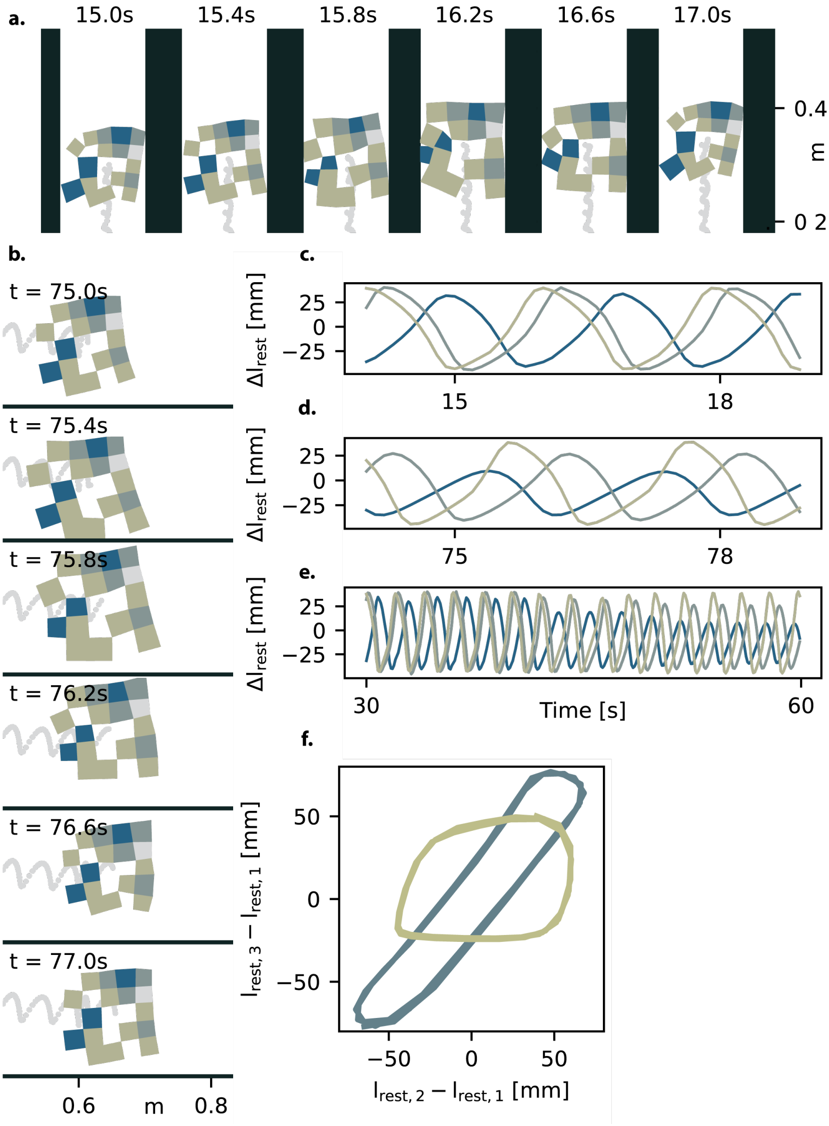


Fig. 5.13: **Task-dependent gait switching in a single adaptive oscillator network. a–b**, Climbing (**a**) and running (**b**) behaviors of the adaptive-network simulation corresponding to Fig. 5.12e. **c–d**, Oscillator outputs for the time windows around **a** (climbing) and **b** (running). **e**, Oscillator outputs across the task switch (at ~40 s). **f**, Phase diagram computed for the two isolated intervals in **c–d**.

show clear phase differences between simulations A and B. Specifically, the relation between the blue and dark gray oscillators changes drastically from anti-phase to nearly in-phase when the control oscillator is turned on. From the snapshots of the simulations in Fig. 5.14e-f, we can observe how the minimal network can move both left and right in response to a single input (Movie 5.5). These results demonstrate that with a well-suited morphology, minimal oscillator networks can switch between different behaviors in response to a single higher-order input, thereby demonstrating how these minimal oscillator networks could offload computation from higher-order controllers.

5.10 CONCLUSIONS

This chapter shows that adaptive behavior does not require large neural controllers or dense sensor networks; it can emerge from the coevolution of a soft body and a minimal network of coupled oscillators. By co-evolving morphology and a relaxation oscillator network with only a handful of nodes, we obtained robust gaits, task switching, and single-input higher-order control, which, in principle, can be mapped to fluidic implementations. The key insight is that morphology and controller are not independent modules, but a single dynamical object whose potential appears when both are allowed to shape one another.

A practical implication emerging from these results is that, for the tested environments and design space, three oscillators may be sufficient to generate the required phase–amplitude diversity, while adding an oscillator offers only marginal benefits. These insights yield significant benefits for future experimental work. Direct coupling between the environment and the oscillators appears to potentially enhance fitness, reduce transients toward stable limit cycles, and increase robustness, allowing the body–environment interaction to act as an additional coupling edge in the network. Notably, behavioral switching occurs without reprogramming: the same hardware can settle into distinct limit cycles under different environmental conditions, and a single gated "control" oscillator could toggle locomotion direction. Taken together, these observations suggest that sequencing and part of the sensing could be offloaded from central processors to body mechanics and

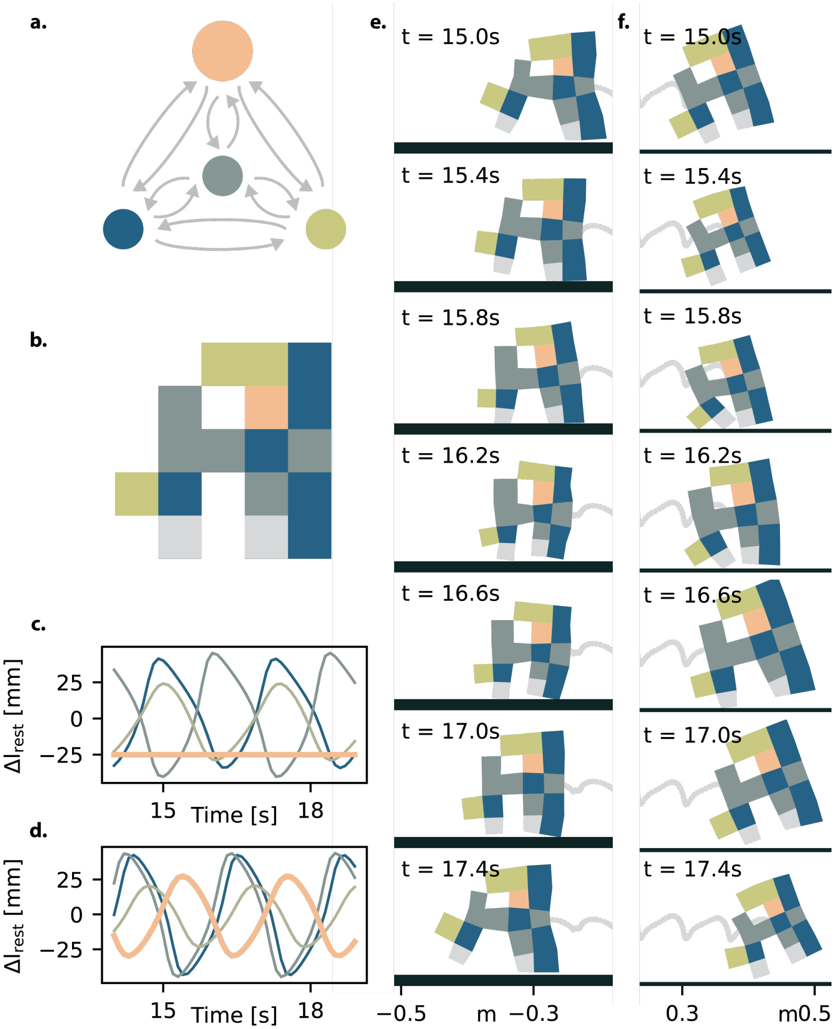


Fig. 5.14: **Higher-order control from a single input. a**, Network schematic with an added control oscillator (orange) whose base current I_{base} toggles between 0.8 (on) and 0 (off), as in Fig. 5.1a–b. **b**, Morphology of best-performing specimen after 100 generations evolved with paired simulations (A: control off, B: control on) and a balanced objective. **c–d**, Oscillator time series for simulations A (**c**, control off) and B (**d**, control on). **e–f**, Snapshots showing leftward motion when the control is off (**e**) and rightward motion when it is on (**f**).

local oscillators, indicating a path toward electronics-free or minimal electronics soft robots that adapt to their environment.

On the experimental side, a three-oscillator fluidic network with tunable coupling could be used to reproduce these behaviors, with minimal sensing implemented as a contact-based inflow or simple “kinked tube” [17]. While the present model employs idealizations (linear springs, penalty contact, simplified damping and sensing, and a planar lattice) to make it computationally feasible to explore many directions, the overall trends, however, appear robust enough to motivate experimental work in this direction. Follow-up work could include hardware-in-the-loop optimization to account for manufacturing tolerances and pneumatic dynamics ([78]), the incorporation of sparse yet richer sensors to shape oscillator inputs ([45]), and manufacturability-aware constraints embedded directly into the optimization.

More broadly, these minimal oscillator networks coupled to morphology and environment have the potential to extend beyond legged locomotion. Similar principles could be applied to peristaltic pumps with reconfigurable wave orders, timing-aware grippers, and medical crawlers that adapt to changing substrates. In this view, these minimal networks with co-evolved bodies that “compute as one” show a practical route to scalable, adaptive soft robotics.

5.11 SUPPLEMENTARY MOVIES 5

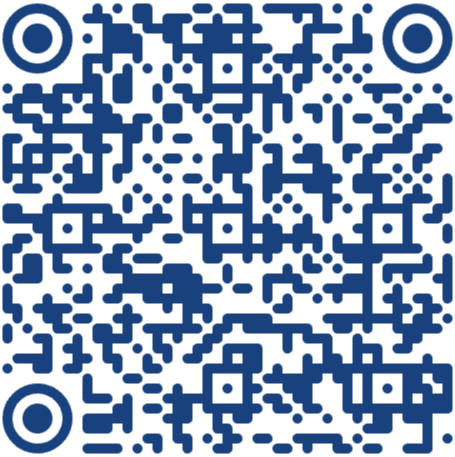
Movie 5.1. Evolution of the morphology over time for the running task. The movie depicts the behavior of the best-performing candidate over time at different generations of the evolution, during one evolutionary optimization of the running task. The colors indicate to which oscillator each voxel is coupled.

Movie 5.2. Evolution of the morphology over time for the climbing task. The movie depicts the behavior of the best-performing candidate over time at different generations of the evolution, during one evolutionary optimization of the climbing task. The colors indicate to which oscillator each voxel is coupled.

Movie 5.3. Oscillators and limit cycle over time for a running structure. Three windows show the morphology, oscillators, and the limit cycle of the oscillators over time, from left to right, respectively.

Movie 5.4. Change in behavior for the adaptive oscillator network during an environmental switch. Three windows show the morphology, oscillators, and the limit cycle of the oscillators over time, from left to right, respectively. During the experiment, the environment is switched from climbing to running, and the system adjusts its behavior accordingly.

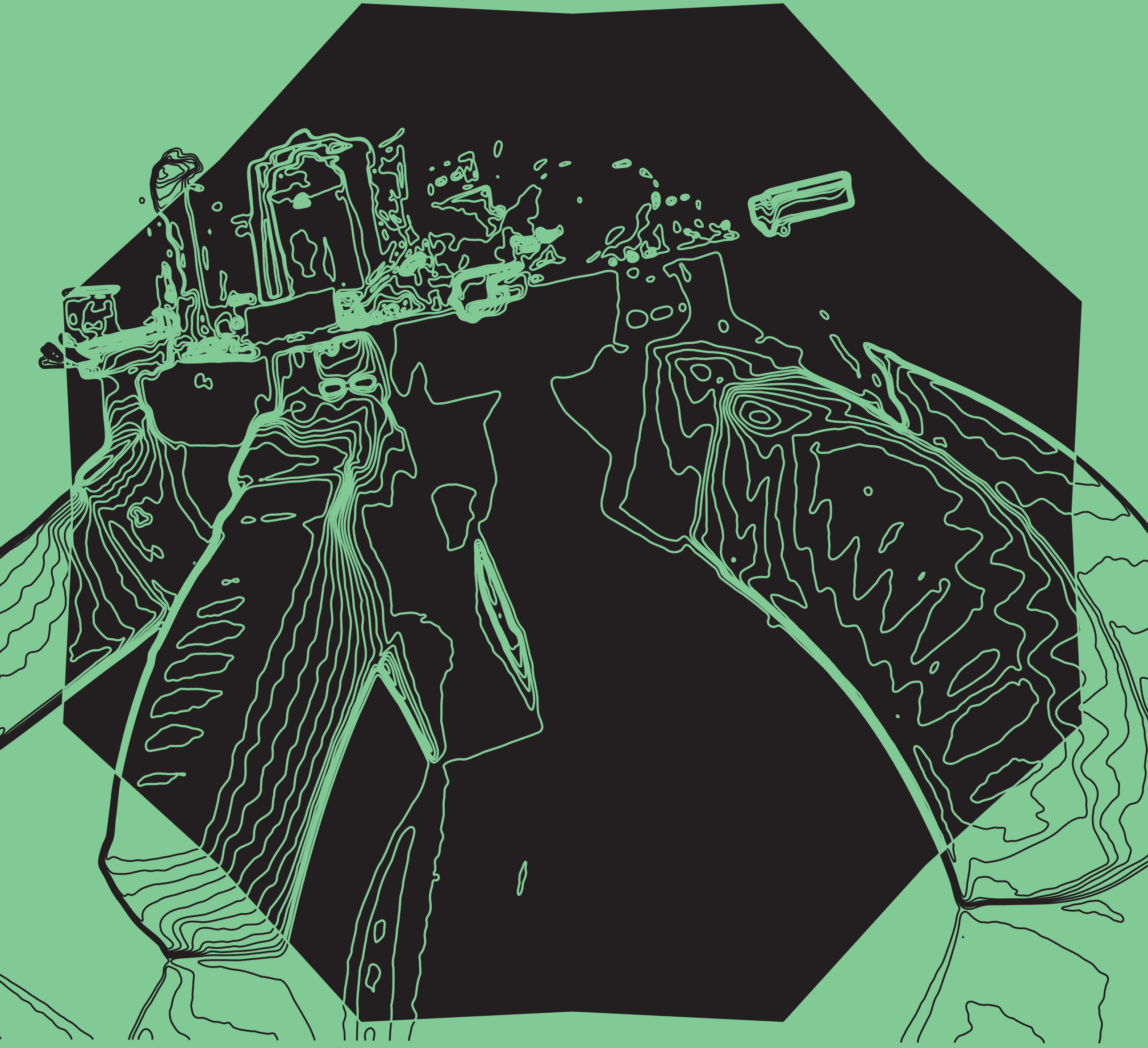
Movie 5.5. Binary inputs for control. The movie demonstrates how a single structure can change the orientation of locomotion by switching a single oscillator on and off.



Link to the Movies.

CONCLUSIONS AND OUTLOOK

6.



6.1 CONCLUSIONS

This thesis asks how computation can be embodied for emergent, goal-oriented behavior in soft robots. Embodied computation in the context of this thesis refers to the capacity of the body, local controllers, and environment to work together, much like biological systems that distribute control across tissues and internal networks. These complex, decentralized interactions within biological systems of all scales grant them the autonomy to survive in the real and unstructured world. This remains a significant challenge in robotics to this day.

> *What constitutes decentralized autonomy in soft robotics, and how does it relate to biological systems?*

Across design, simulation, and experiment, we show that coordination of locomotion can be distributed across compliant bodies, local controllers, and environmental interactions such that purposeful motion in soft robots arises without the need for central control or a full-body representation.

The main difference between natural systems and the design of soft robots is that nature has had billions of years to evolve, resulting in a seemingly limitless level of complexity. This is unattainable for robotics today; however, there are key principles we can learn from. By using local sensing, computation, and processing in each limb, *Chapter 2* shows how goal-directed behavior can emerge even without explicit communication or central control.

> *How can we design soft robots that achieve complex goal-directed behaviors through localized sensing and minimal processing?*

The algorithm proposed in *Chapter 2* and *Chapter 3* teaches us that designing and optimizing for an emergent behavior using a minimal form of decentralized learning results in complex dynamical systems where behavior is not predefined as in inverse kinematics for rigid robots but instead emerges from the combination of environmental input, the soft deforming body, and the algorithm. These results lead us to view soft robotic systems and their surroundings as one closed-loop dynamical system in which materials, morphology, and the environment affect and thus contribute to the composition of system-level behavior, much like their biological counterparts.

> *Can we characterize and understand the emergence of phototaxis in decentralized soft robots, with minimal and local sensing, memory, and actuation?*

The characterization clarifies both strengths and limits. The decentralized algorithm proposed in *Chapter 2* and *Chapter 3* produces robust goal-directed behavior when the learning rate matches the morphology and actuation time scale of the system. From *Chapter 3*, we further learn that the system-level behavior is primarily dictated by body composition and that the optimization, due to its limited form of memory, merely follows locally optimal behavior. This gives rise to structured trajectories through the behavioral search space of the system, which can be utilized and designed for a given specific task or environment. Yet, the discrete-time updates and stochastic perturbations make direct hardware realizations in analog or fluidic circuits costly because continuous signals must be quantized, gated, and randomized. Although possible, this approach results in integrated circuits with considerable additional components, as it requires converting continuous electrical signals into discrete and stochastic processes. This top-down approach, which starts with an algorithm on a computer and works toward a hardware implementation, therefore introduces overhead that is at odds with embedding computation to reduce the complexity of components. By contrast, nature builds coordination from continuous dynamics at the component level, for example, through self-oscillatory elements. Framing locomotion as an emergent property of such continuous dynamics explains why bottom-up designs can achieve similar functions with fewer resources.

> *Can we create controllable self-sustaining rhythmic patterns and coordination in soft oscillators, through local fluidic coupling and without electronic control systems?*

Chapter 4 adopts a bottom-up approach by designing a self-oscillating component that directly regulates the pressures in the soft actuators, thereby minimizing the overhead posed by electronics entirely. Using directional coupling, inspired by biological networks of neurons, shows that we can coordinate the phases of the oscillators solely through fluidics.

The transition to this bottom-up approach, combined with the need to design for the system as a whole, also poses the question of how a coupling between oscillators can be learned to produce a desired goal-directed behavior at the system level. In *Chapter 3*, the system can learn and adapt its phases freely during the task, which would be challenging to implement directly in an oscillator network due to the nonlinear relationships between oscillator couplings and their emerging phases.

Using the limited short-term memory of the proposed algorithm yielded great benefits for its ability to adapt. However, this approach also has limitations, as the absence of long-term memory causes it to quickly forget and prevents learning from past experiences on longer time scales. Natural systems learn from past experiences on many different time scales. The most fundamental and universal is evolution, where specimens that perform well pass on their genes to the next generation.

> *How does co-evolutionary design of morphology and embedded computation enhance autonomous adaptability in soft robotics?*

By co-evolving the morphology of a soft robot and its oscillator network that controls its actuation in different task settings with simulations in *Chapter 5*, we can optimize the synergy between morphology, embedded computation, and environment in a closed loop. Co-evolution of morphology with a small relaxation oscillator network produces robust gaits, task switches, and direction toggling via a single gated input. Importantly, for the tested environments, three oscillators often suffice to span the required phase and amplitude diversity, and direct coupling through the environment shortens transients to stable limit cycles, thereby improving robustness. These findings point the way toward compact physical implementations with minimal sensing that retain adaptability. The main drawback of this approach is that it is not trivial to realize the simulated system as a physical counterpart. First, it is notoriously hard to model all the physics and nonlinearities of soft robots in detail. Second, non-gradient-based stochastic optimizations, such as evolutionary algorithms, are computationally costly. However, as evidenced in both *Chapter 3* and *Chapter 4*, coarse-grained qualitative simulations can provide meaningful insights and give engineers design directions for physical implementations.

Returning to the central question of how computation can be embodied for emergent, goal-oriented behavior in soft robots, this work demonstrates that modules with minimal computation and limited memory can, through repeated oscillatory actuation and environmental coupling, yield goal-oriented locomotion that adapts to changing conditions. The body's compliant mechanics do not merely execute commands; they compute by filtering, shaping, and synchronizing local actions into system-level trajectories, aligning with biological strategies that distribute control across material and morphology to offload central processing. Because the approach relies on the design of the dynamical system that consists of the entire body together with its environment, it scales to domains where modeling is impractical and computational resources are scarce, including swarm, micro, and nano robotics [140]. By unifying mechanical intelligence with minimal computation, the thesis charts a path for soft robots to approach biological versatility, robustness, and autonomy while maintaining a minimal design approach.

6.2 OUTLOOK

Building on these conclusions, this section translates the central lesson of this thesis into a research agenda: task, environment, body, and control should be designed and studied as a single dynamical system, allowing compliant materials to share the work of sensing, computation, and control. Insights from *Chapter 2* and *Chapter 3* on decentralized learning and environment-mediated feedback, from *Chapter 4* on electronics-free rhythmic control, and from *Chapter 5* on co-evolution of morphology and minimal oscillator networks, motivate concrete questions and future experimental work. The following section proposes a direction for investigating these questions.

> *How can we combine implicit and explicit forms of communication in decentralized systems to gain adaptability?*

Chapter 4 and *Chapter 5* demonstrate that explicit communication between oscillators, facilitated by their fluidic coupling, can produce robust and versatile phase patterns between the oscillators. However, *Chapter 1*, *Chapter 2*, and *Chapter 3* show that explicit communication is not strictly required to produce phase locking and achieve a desired system-level behavior. Although it is more complex to design for, because the phase locking and desired behavior are solely environment dependent, it has the benefit of reducing component complexity, as no coupling is required, as seen in the example in *Chapter 1*. An interesting direction is to explore how we can combine the predictable design and network scalability of explicit coupling with the adaptability and tolerance to fabrication variance of implicitly coupled systems. For limb systems, one could explicitly couple more local sections within each limb while retaining implicit coupling between the limbs, thereby reducing the overhead of extensive wiring between limbs. In this example, the implicit coupling could also be used to switch between limit cycles in response to a changing environment, as seen in *Chapter 1*, while the more local explicit coupling could be of use to increase the number of actuators and generate more complex and reliable gaits.

> *Can we combine these decentralized emergent forms of control with existing central controllers?*

Throughout the thesis, we have seen that these decentralized systems are capable of goal-oriented behavior. However, due to their minimal design approach, these systems lack a form of long-term memory or action planning. Achieving true autonomy in more complex tasks and real-world scenarios appears to require hierarchical control. There is a vast literature on central controllers and computer algorithms. A promising research direction could be to combine these central controllers, which facilitate high-level thinking such as action planning and decision-making, with decentralized networks that can filter sensory information and construct actuation gaits locally. As an example, *Chapter 5* demonstrates how a single binary input can switch the limit cycle of the oscillator network to change the direction of locomotion. By designing binary switches for different system-level behaviors, the central controller will have a set of behaviors that require minimal inputs for action planning. At the same time, the oscillator network handles gait formation and the transition between gaits. This way, we offload control from the central controller to the body, analogous to the relationship between the central nervous system and CPGs in natural systems.

The implicitly coupled algorithm from *Chapter 2* and *Chapter 3* could find similar use in larger, more complex robotic systems, performing local sensing and computation to carry out tasks away from the central controller, such as adapting tactile skins based on surface composition, similar to the camouflage of cuttlefish, which arises from local sensing and actuating in the skin.

> *Do these fluidic oscillators scale in size, and how do we design for large networks?*

The principles of the fluidic relaxation oscillator proposed in *Chapter 4* could be scaled down significantly. Currently, these oscillators are centimeter-scale, which limits their practical implementation in larger networks. However, in theory, one could implement these same principles into printed microfluidic networks [4], opening up the possibility to scale the network sizes to hundreds of fluidic neurons. This would be in the same order of magnitude as simple natural life forms, like *C. elegans* [129]. Scaling the networks to larger numbers of oscillators increases both the number of potential sensory inputs and the number of potentially stable limit cycles within the system. Already, with the present oscillator system, we must start designing for soft robots that incorporate a number of behavioral primitives, where probability or environmental cues can switch the system to a different limit cycle and adapt its behavior. This way, the system-level behavior starts to emerge and adjust based on local sensing, similarly to the minimal algorithm proposed in *Chapter 2* and *Chapter 3*. Therefore, we can begin to develop systems that can adapt their behavior and accomplish tasks in real-world scenarios.

Even though this would grant soft robots a vast increase in their potential complexity, it also poses a fundamental question: how do we design for such complex behavior with potentially thousands of degrees of freedom? Computer simulations and reinforcement learning could provide a potential solution, as research on physics engines is progressing quickly and has shown the ability to train large neural networks for robotics [71]. Combining these with the ability to print microfluidic chips could give rise to a domain of scalable, bio-inspired systems that do not require electricity and that actuate, sense, and compute using the same medium.

> *Is it possible to translate short-term and long-term memory learning into coupled oscillator networks?*

The oscillator networks introduced in *Chapter 4* and *Chapter 5* show great potential for making robotic systems that have the adaptability and robustness of their biological counterparts. One crucial difference, however, is that in biology, the neuronal connections do not merely change over generations but also within the lifetime of a single specimen. It is this ability that enables natural systems to adapt and learn and gives them memory. Although ambitious, the emerging field of physical learning has started to bridge the gap between learning in computers and their physical and biological counterparts [117]. By creating fluidic oscillator networks with adjustable directional couplings, behaviors can be trained directly on the physical system itself. In this way, principles of reinforcement learning and physical learning could be applied to give the soft robots a sense of memory and learning.

A long-term goal for soft robotics and robotics in general could therefore be to envision networks of neural elements distributed throughout a robot's body, acting as a biological nervous system and brain that learns throughout its life cycle.

In conclusion, designing the task, environment, body, and control as a single closed-loop dynamical system achieves robustness not by modeling everything in detail, but by combining emergence with minimal tunability and soft bodies. This is the main takeaway: decentralized, emergent forms of coordination offer a minimal and scalable way to steer soft-robot behavior toward a goal robustly while retaining the adaptability to bring future soft robots outside the lab.

BIBLIOGRAPHY

[1] Gursel Alici. "Towards soft robotic devices for site-specific drug delivery." In: *Expert Review of Medical Devices* 12 (2015), pp. 703 –715.

[2] O. Andersson, S. Grillner, and E. Lennander. "Peripheral feedback mechanisms acting on the central pattern generators for locomotion in fish and mammals." In: *Canadian Journal of Physiology and Pharmacology* 59.7 (1981), pp. 713–726.

[3] Henry C Astley. "Getting around when you're round: quantitative analysis of the locomotion of the blunt-spined brittle star, Ophiocoma echinata." In: *Journal of Experimental Biology* 215.11 (2012), pp. 1923–1929.

[4] Anthony K Au, Wilson Huynh, Lisa F Horowitz, and Albert Folch. "3D-printed microfluidics." In: *Angewandte Chemie International Edition* 55.12 (2016), pp. 3862–3881.

[5] Frederico Augugliaro, Sergei Lupashin, Michael Hamer, Cason Male, Markus Hehn, Mark W Mueller, Jan Sebastian Willmann, Fabio Gramazio, Matthias Kohler, and Raffaello D'Andrea. "The flight assembled architecture installation: Cooperative construction with flying machines." In: *IEEE Control Systems Magazine* 34.4 (2014), pp. 46–64.

[6] Danielle K Barnes and Jonathan D Allen. "Predators Induce Phenotypic Plasticity in Echinoderms across Life History Stages." In: *The Biological Bulletin* 244.2 (2023), pp. 103–114.

[7] Levent Bayindir. "A review of swarm robotics tasks." In: *Neurocomputing* 172 (2016), pp. 292–321.

[8] Michael A Bell, Luca Cattani, Benjamin Gorissen, Katia Bertoldi, James C Weaver, and Robert J Wood. "A soft, modular, and bi-stable dome actuator for programmable multi-modal locomotion." In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 6529–6535.

[9] Florian Berlinger, Melvin Gauci, and Radhika Nagpal. "Implicit coordination for 3D underwater collective behaviors in a fish-inspired robot swarm." In: *Science Robotics* 6.50 (2021).

[10] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. "Swarm robotics: a review from the swarm engineering perspective." In: *Swarm Intelligence* 7 (2013), pp. 1–41.

[11] MD Candia Carnevali. "Regeneration in Echinoderms: repair, regrowth, cloning." In: *Invertebrate Survival Journal* 3.1 (2006), pp. 64–76.

[12] Chi Chen, Pengju Shi, Zixiao Liu, Sidi Duan, Muqing Si, Chuanwei Zhang, Yingjie Du, Yichen Yan, Timothy J White, Rebecca Kramer-Bottiglio, et al. "Advancing physical intelligence for autonomous soft robots." In: *Science Robotics* 10.102 (2025), eads1292.

[13] Jianing Chen, Melvin Gauci, Wei Li, Andreas Kolling, and Roderich Groß. "Occlusion-based cooperative transport with a swarm of miniature mobile robots." In: *IEEE Transactions on Robotics* 31.2 (2015), pp. 307–321.

[14] Nicholas Cheney, Jeff Clune, and Hod Lipson. "Evolved electrophysiological soft robots." In: *Artificial life conference proceedings*. MIT Press One Rogers Street, Cambridge. 2014, pp. 222–229.

[15] Ken Cheng. "Cognition Beyond Representation: Varieties of Situated Cognition in Animals." In: *Comparative Cognition & Behavior Reviews* 13 (2018).

[16] Chuan-Chin Chiao, Charles Chubb, and Roger T Hanlon. "A review of visual perception mechanisms that regulate rapid adaptive camouflage in cuttlefish." In: *Journal of Comparative Physiology A* 201 (2015), pp. 933–945.

[17] Alberto Comoretto, Stijn Koppen, Tanaya Mandke, and Johannes TB Overvelde. "Embodying mechano-fluidic memory in soft machines to program behaviors upon interactions." In: *Device* (2025).

[18] Alberto Comoretto, Harmannus AH Schomaker, and Johannes TB Overvelde. "Physical synchronization of soft self-oscillating limbs for fast and autonomous locomotion." In: *Science* 388.6747 (2025), pp. 610–615.

[19] Alberto Comoretto, Harmannus AH Schomaker, and Johannes TB Overvelde. "Physical synchronization of soft self-oscillating limbs for fast and autonomous locomotion." In: *Science* 388.6747 (2025), pp. 610–615.

[20] Stefan Conrad, Joscha Teichmann, Philipp Auth, N Knorr, K Ulrich, D Bellin, Thomas Speck, and Falk J Tauber. "3D-printed digital pneumatic logic for the control of soft robotic actuators." In: *Science robotics* 9.86 (2024), eadh4060.

[21] Francesco Corucci, Nick Cheney, Francesco Giorgio-Serchi, Josh Bongard, and Cecilia Laschi. "Evolving soft locomotion in aquatic and terrestrial environments: effects of material properties and environmental transitions." In: *Soft robotics* 5.4 (2018), pp. 475–495.

[22] Francesco Corucci, Nick Cheney, Hod Lipson, Cecilia Laschi, and Josh Bongard. "Evolving swimming soft-bodied creatures." In: *ALIFE XV, the fifteenth international conference on the synthesis and simulation of living systems, late breaking proceedings*. Vol. 6. 2016.

[23] Cosimo Della Santina, Manuel G. Catalano, and Antonio Bicchi. "Soft Robots." In: *Encycl. Robot.* Vol. 23. 15. https://doi.org/10.1007/978-3-642-41610-1_146-2. Springer Berlin Heidelberg, 2021, pp. 1–15. ISBN: 9783642416101.

[24] Cosimo Della Santina, Christian Duriez, and Daniela Rus. "Model-based control of soft robots: A survey of the state of the art and open challenges." In: *IEEE Control Systems Magazine* 43.3 (2023), pp. 30–65.

[25] Dylan Drotman, Saurabh Jadhav, David Sharp, Christian Chan, and Michael T Tolley. "Electronics-free pneumatic circuits for controlling soft-legged robots." In: *Science Robotics* 6.51 (2021), eaay2627.

[26] Dylan Drotman, Saurabh Jadhav, David Sharp, Christian Chan, and Michael T. Tolley. "Electronics-free pneumatic circuits for controlling soft-legged robots." In: *Sci. Robot.* 6.51 (2021). https://doi.org/10.1126/SCIROBOTICS.AAY2627, eaay2627.

[27] Sourav Dutta, Abhinav Parihar, Abhishek Khanna, Jorge Gomez, Wriddhi Chakraborty, Matthew Jerry, Benjamin Grisafe, Arijit Raychowdhury, and Suman Datta. "Programmable coupled oscillators for synchronized locomotion." In: *Nature Communications* 10.1 (July 24, 2019), p. 3299. ISSN: 2041-1723. DOI: 10.1038/s41467-019-11198-6.

[28] TA Ebert. "Adaptive aspects of phenotypic plasticity in echinoderms." In: *Oceanologica acta* 19.3-4 (1996), pp. 347–355.

[29] Ö. Ekeberg and S. Grillner. "Simulations of neuromechanical control in lamprey swimming." In: *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences* 354.1385 (1999), pp. 895–902.

[30] Johannes F Elfferich, Dimitra Dodou, and Cosimo Della Santina. "Soft robotic grippers for crop handling or harvesting: A review." In: *Ieee Access* 10 (2022), pp. 75428–75443.

[31] Tamar Flash and Binyamin Hochner. "Motor primitives in vertebrates and invertebrates." In: *Current opinion in neurobiology* 15.6 (2005), pp. 660–666.

[32] L Formery, P Peluso, I Kohnle, J Malnick, JR Thompson, M Pitel, KR Uhlinger, DS Rokhsar, DR Rank, and CJ Lowe. "Molecular evidence of anteroposterior patterning in adult echinoderms." In: *Nature* 623.7987 (2023), pp. 555–561.

[33] Yoël Forterre, Jan M. Skotheim, Jacques Dumais, and L. Mahadevan. "How the Venus flytrap snaps." In: *Nature* 433.7024 (2005). https://doi.org/10.1038/nature03185, pp. 421–425.

[34] Cody A Freas and Ken Cheng. "Neuroecology beyond the brain: learning in Echinodermata." In: *Learning & Behavior* 50.1 (2022), pp. 20–36.

[35] W Otto Friesen and William B Kristan. "Leech locomotion: swimming, crawling, and decisions." In: *Current opinion in neurobiology* 17.6 (2007), pp. 704–711.

[36] Michel Fruchart, Ryo Hanai, Peter B Littlewood, and Vincenzo Vitelli. "Non-reciprocal phase transitions." In: *Nature* 592.7854 (2021), pp. 363–369.

[37] Gaurav Gardi and Metin Sitti. "On-demand breaking of action-reaction reciprocity between magnetic microdisks using global stimuli." In: *Physical review letters* 131.5 (2023), p. 058301.

[38] Bruno L Gianasi, Jean-François Hamel, Emaline M Montgomery, Jiamin Sun, and Annie Mercier. "Current knowledge on the biology, ecology, and commercial exploitation of the sea cucumber Cucumaria frondosa." In: *Reviews in Fisheries Science & Aquaculture* 29.4 (2021), pp. 582–653.

[39] Benjamin Gorissen, David Melancon, Nikolaos Vasios, Mehdi Torbati, and Katia Bertoldi. "Inflatable soft jumper inspired by shell snapping." In: *Sci. Robot.* 5.42 (2020). https://doi.org/10.1126/scirobotics.abb1967, p. 1967.

[40] Martyn Goulding. "Circuits controlling vertebrate locomotion: moving in a new direction." In: *Nature Reviews Neuroscience* 10.7 (2009), pp. 507–518.

[41] Sten Grillner and Alexander Kozlov. "The CPGs for limbed locomotion–facts and fiction." In: *International Journal of Molecular Sciences* 22.11 (2021), p. 5882.

[42] Elliot W Hawkes, Laura H Blumenschein, Joseph D Greer, and Allison M Okamura. "A soft robot that navigates its environment through growth." In: *Science Robotics* 2.8 (2017), eaan3028.

[43] Elliot W Hawkes, Charles Xiao, Richard-Alexandre Peloquin, Christopher Keeley, Matthew R Begley, Morgan T Pope, and Günter Niemeyer. "Engineered jumpers overcome biological limits via work multiplication." In: *Nature* 604.7907 (2022), pp. 657–661.

[44] Qiguang He, Rui Yin, Yucong Hua, Weijian Jiao, Chengyang Mo, Hang Shu, and Jordan R Raney. "A modular strategy for distributed, embodied control of electronics-free soft robots." In: *Science Advances* 9.27 (2023), eade9247.

[45] Chidanand Hegde, Jiangtao Su, Joel Ming Rui Tan, Ke He, Xiaodong Chen, and Shlomo Magdassi. "Sensing in soft robotics." In: *ACS nano* 17.16 (2023), pp. 15277–15307.

[46] Sina Heydari, Amy Johnson, Olaf Ellers, Matthew J McHenry, and Eva Kanso. "Sea star inspired crawling and bouncing." In: *Journal of The Royal Society Interface* 17.162 (2020), p. 20190700.

[47] Binyamin Hochner. "An embodied view of octopus neurobiology." In: *Current biology* 22.20 (2012), R887–R892.

[48] Guodong Hou, Xu Zhang, Feihong Du, Yadong Wu, Xing Zhang, Zhijie Lei, Wei Lu, Feiyu Zhang, Guang Yang, Huamiao Wang, et al. "Self-regulated underwater phototaxis of a photoresponsive hydrogel-based phototactic vehicle." In: *Nature Nanotechnology* 19.1 (2024), pp. 77–84.

[49] Adam D Hughes, Lars Brunner, Elizabeth J Cook, Maeve S Kelly, and Ben Wilson. "Echinoderms display morphological and behavioural phenotypic plasticity in response to their trophic environment." In: (2012).

[50] Cynthia L Hughes and John B Thomas. "A sensory feedback circuit coordinates muscle activity in Drosophila." In: *Molecular and Cellular Neuroscience* 35.2 (2007), pp. 383–396.

[51] Fumiya Iida and Cecilia Laschi. "Soft robotics: Challenges and perspectives." In: *Procedia Computer Science* 7 (2011), pp. 99–102.

[52] Auke J. Ijspeert and Monica A. Daley. "Integration of feedforward and feedback control in the neuromechanics of vertebrate locomotion: a review of experimental, simulation and robotic studies." In: *Journal of Experimental Biology* 226.15 (Aug. 1, 2023), jeb245784. ISSN: 0022-0949, 1477-9145. DOI: 10.1242/jeb.245784.

[53] Auke J Ijspeert and Monica A Daley. "Integration of feedforward and feedback control in the neuromechanics of vertebrate locomotion: a review of experimental, simulation and robotic studies." In: *Journal of Experimental Biology* 226.15 (2023), jeb245784.

[54] Auke Jan Ijspeert, Alessandro Crespi, Dimitri Ryczko, and Jean-Marie Cabelguen. “From Swimming to Walking with a Salamander Robot Driven by a Spinal Cord Model.” In: *Science* 315.5817 (Mar. 9, 2007). Publisher: American Association for the Advancement of Science, pp. 1416–1420. DOI: 10.1126/science.1138353.

[55] Auke Jan Ijspeert, Alessandro Crespi, Dimitri Ryczko, and Jean-Marie Cabelguen. “From swimming to walking with a salamander robot driven by a spinal cord model.” In: *science* 315.5817 (2007), pp. 1416–1420.

[56] Trevor J Jones, Etienne Jambon-Puillet, Joel Marthelot, and P-T Brun. “Bubble casting soft robotics.” In: *Nature* 599.7884 (2021), pp. 229–233.

[57] Takeshi Kano, Eiki Sato, Tatsuya Ono, Hitoshi Aonuma, Yoshiya Matsuzaka, and Akio Ishiguro. “A brittle star-like robot capable of immediately adapting to unexpected physical damage.” In: *Royal Society open science* 4.12 (2017), p. 171200.

[58] Takeshi Kano, Ryo Yoshizawa, and Akio Ishiguro. “Tegotae-based decentralised control scheme for autonomous gait transition of snake-like robots.” In: *Bioinspiration & biomimetics* 12.4 (2017), p. 046009.

[59] Sangbae Kim, Cecilia Laschi, and Barry Trimmer. “Soft robotics: a bioinspired evolution in robotics.” In: *Trends in biotechnology* 31.5 (2013), pp. 287–294.

[60] Jérémie Knüsel, Alessandro Crespi, Jean-Marie Cabelguen, Auke J. Ijspeert, and Dimitri Ryczko. “Reproducing Five Motor Behaviors in a Salamander Robot With Virtual Muscles and a Distributed CPG Controller Regulated by Drive Signals and Proprioceptive Feedback.” In: *Frontiers in Neurorobotics* 14 (Dec. 23, 2020). Publisher: Frontiers. ISSN: 1662-5218. DOI: 10.3389/fnbot.2020.604426.

[61] Nancy Kopell and G Bard Ermentrout. “Mechanisms of phase-locking and frequency control in pairs of coupled neural oscillators.” In: *Handbook of dynamical systems* 2 (2002), pp. 3–54.

[62] Arda Kotikian, Connor McMahan, Emily C Davidson, Jalilah M Muhammad, Robert D Weeks, Chiara Daraio, and Jennifer A Lewis. “Untethered soft robotic matter with passive control of shape morphing and propulsion.” In: *Science robotics* 4.33 (2019), eaax7044.

[63] Lucas C. van Laake, Jelle de Vries, Sevda Malek Kani, and Johannes T. B. Overvelde. “A fluidic relaxation oscillator for reprogrammable sequential actuation in soft robots.” In: *Matter* 5.9 (Sept. 7, 2022). Publisher: Elsevier, pp. 2898–2917. ISSN: 2590-2393, 2590-2385. DOI: 10.1016/j.matt.2022.06.002.

[64] Lucas C van Laake, Jelle de Vries, Sevda Malek Kani, and Johannes TB Overvelde. “A fluidic relaxation oscillator for reprogrammable sequential actuation in soft robots.” In: *Matter* 5.9 (2022), pp. 2898–2917.

[65] Lucas Carolus van Laake and Johannes Tesse Bastiaan Overvelde. “Bio-inspired autonomy in soft robots.” In: *Communications Materials* 5.1 (2024), p. 198.

[66] Chiwon Lee, Myungjoon Kim, Yoon Jae Kim, Nhayoung Hong, Seungwan Ryu, H Jin Kim, and Sungwan Kim. “Soft robot review.” In: *International Journal of Control, Automation and Systems* 15.1 (2017), pp. 3–15.

[67] Won-Kyu Lee, Daniel J Preston, Markus P Nemitz, Amit Nagarkar, Arthur K MacKeith, Benjamin Gorissen, Nikolaos Vasios, Vanessa Sanchez, Katia Bertoldi, L Mahadevan, et al. “A buckling-sheet ring oscillator for electronics-free, multimodal locomotion.” In: *Science Robotics* 7.63 (2022), eabg5812.

[68] Shuguang Li, Richa Batra, David Brown, Hyun-Dong Chang, Nikhil Ranganathan, Chuck Hoberman, Daniela Rus, and Hod Lipson. “Particle robotics based on statistical mechanics of loosely coupled components.” In: *Nature* 567.7748 (2019), pp. 361–365.

[69] Danli Luo, Aditi Maheshwari, Andreea Danielescu, Jiaji Li, Yue Yang, Ye Tao, Lingyun Sun, Dinesh K Patel, Guanyun Wang, Shu Yang, et al. “Autonomous self-burying seed carriers for aerial seeding.” In: *Nature* 614.7948 (2023), pp. 463–470.

[70] Quanxia Lyu, Shu Gong, Jialiang Yin, Jennifer M Dyson, and Wenlong Cheng. "Soft wearable healthcare materials and devices." In: *Advanced healthcare materials* 10.17 (2021), p. 2100577.

[71] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. "Isaac gym: High performance gpu-based physics simulation for robot learning." In: *arXiv preprint arXiv:2108.10470* (2021).

[72] Yoichi Masuda and Masato Ishikawa. "Review of electronics-free robotics: Toward a highly decentralized control architecture." In: *Journal of Robotics and Mechatronics* 34.2 (2022), pp. 202–211.

[73] Michael Andrew McEvoy and Nikolaus Correll. "Materials that couple sensing, actuation, computation, and communication." In: *Science* 347.6228 (2015), p. 1261689.

[74] Ross M McKenzie, Mohammed E Sayed, Markus P Nemitz, Brian W Flynn, and Adam A Stokes. "Linbots: soft modular robots utilizing voice coils." In: *Soft robotics* 6.2 (2019), pp. 195–205.

[75] Eric Medvet, Alberto Bartoli, Andrea De Lorenzo, and Stefano Seriani. "2D-VSR-Sim: A simulation tool for the optimization of 2-D voxel-based soft robots." In: *SoftwareX* 12 (2020), p. 100573.

[76] C. S. Mendes, D. B. Bartos, G. Akay, and K. Z. Stanley. "Neural control of limb coordination in fruit flies during adaptive locomotion." In: *PLoS Biology* 11.4 (2013), e1001564.

[77] Gianmarco Mengaldo, Federico Renda, Steven L Brunton, Moritz Bächer, Marcello Calisti, Christian Duriez, Gregory S Chirikjian, and Cecilia Laschi. "A concise guide to modelling the physics of embodied intelligence in soft robotics." In: *Nature Reviews Physics* 4.9 (2022), pp. 595–610.

[78] Franc Mihalič, Mitja Truntič, and Alenka Hren. "Hardware-in-the-loop simulations: A historical overview of engineering challenges." In: *Electronics* 11.15 (2022), p. 2462.

[79] Edoardo Milana, Cosimo Della Santina, Benjamin Gorissen, and Philipp Rothemund. "Physical control: A new avenue to achieve intelligence in soft robotics." In: *Science Robotics* 10.102 (2025), eadw7660.

[80] Marc Z Miskin, Alejandro J Cortese, Kyle Dorsey, Edward P Esposito, Michael F Reynolds, Qingkun Liu, Michael Cao, David A Muller, Paul L McEuen, and Itai Cohen. "Electronically integrated, mass-manufactured, microscopic robots." In: *Nature* 584.7822 (2020), pp. 557–561.

[81] Bobak Mosadegh, Panagiotis Polygerinos, Christoph Keplinger, Sophia Wennstedt, Robert F Shepherd, Unmukt Gupta, Jongmin Shim, Katia Bertoldi, Conor J Walsh, and George M Whitesides. "Pneumatic networks for soft robotics that actuate rapidly." In: *Advanced functional materials* 24.15 (2014), pp. 2163–2170.

[82] Mostafa Mousa, Ashkan Rezanejad, Benjamin Gorissen, and Antonio E Forte. "Frequency-Controlled Fluidic Oscillators for Soft Robots." In: *Advanced Science* 11.43 (2024), p. 2408879.

[83] Reed A Myers, Carolyn M Furlong, Murray K Gingras, and John-Pal Zonneveld. "Locomotion traces emplaced by modern stalkless comatulid crinoids (featherstars)." In: (2023).

[84] Nguyen HP Nguyen, Daphne Klotsa, Michael Engel, and Sharon C Glotzer. "Emergent collective phenomena in a mixture of hard shapes through active rotation." In: *Physical review letters* 112.7 (2014), p. 075701.

[85] Kwang W. Oh, Kangsun Lee, Byungwook Ahn, and Edward P. Furlani. "Design of pressure-driven microfluidic networks using electric circuit analogy." In: *Lab Chip* 12.3 (2012), pp. 515–545. ISSN: 1473-0197, 1473-0189. DOI: 10.1039/C2LC20799K.

[86] Kwang W Oh, Kangsun Lee, Byungwook Ahn, and Edward P Furlani. "Design of pressure-driven microfluidic networks using electric circuit analogy." In: *Lab on a Chip* 12.3 (2012), pp. 515–545.

[87] Giorgio Oliveri, Lucas C van Laake, Cesare Carissimo, Clara Miette, and Johannes TB Overvelde. "Continuous learning of emergent behavior in robotic matter." In: *Proceedings of the National Academy of Sciences* 118.21 (2021), e2017015118.

[88] Giorgio Oliveri, Lucas C van Laake, Cesare Carissimo, Clara Miette, and Johannes TB Overvelde. "Continuous learning of emergent behavior in robotic matter." In: *Proceedings of the National Academy of Sciences* 118.21 (2021), e2017015118.

[89] Eneko Osaba, Javier Del Ser, Andres Iglesias, and Xin-She Yang. *Soft computing for swarm robotics: new trends and applications*. 2020.

[90] Dai Owaki and Akio Ishiguro. "A Quadruped Robot Exhibiting Spontaneous Gait Transitions from Walking to Trotting to Galloping." In: *Scientific Reports* 7.1 (Mar. 21, 2017). Publisher: Nature Publishing Group, p. 277. ISSN: 2045-2322. DOI: 10.1038/s41598-017-00348-9.

[91] Yasemin Ozkan-Aydin, Daniel I Goldman, and M Saad Bhamla. "Collective dynamics in entangled worm and robot blobs." In: *Proceedings of the National Academy of Sciences* 118.6 (2021), e2010542118.

[92] Youngmin Park, Stewart Heitmann, and G Bard Ermentrout. "The utility of phase models in studying neural synchronization." In: *Computational models of brain and behavior* (2017), pp. 493–504.

[93] Christopher J. Payne et al. "An Implantable Extracardiac Soft Robotic Device for the Failing Heart: Mechanical Coupling and Synchronization." In: *Soft Robot.* 4.3 (2017). https://doi.org/10.1089/soro.2016.0076, pp. 241–250.

[94] Alessandro Pazzaglia, Andrej Bicanski, Andrea Ferrario, Jonathan Arreguit, Dimitri Ryczko, and Auke Ijspeert. "Balancing central control and sensory feedback produces adaptable and robust locomotor patterns in a spiking, neuromechanical model of the salamander spinal cord." In: *PLOS Computational Biology* 21.1 (2025), e1012101.

[95] K. Pearson. "Generating the walking gait: role of sensory feedback." In: *Progress in Brain Research* 143 (2004), pp. 123–129.

[96] Cengiz Pehlevan, Paolo Paoletti, and L Mahadevan. "Integrative neuromechanics of crawling in D. melanogaster larvae." In: *Elife* 5 (2016), e11031.

[97] Orit Peleg, Jacob M Peters, Mary K Salcedo, and Lakshminarayanan Mahadevan. "Collective mechanical adaptation of honeybee swarms." In: *Nature Physics* 14.12 (2018), pp. 1193–1198.

[98] Rolf Pfeifer, Max Lungarella, and Fumiya Iida. "Self-organization, embodiment, and biologically inspired robotics." In: *science* 318.5853 (2007), pp. 1088–1093.

[99] Sergio Picella, Catharina M van Riet, and Johannes TB Overvelde. "Pneumatic coding blocks enable programmability of electronics-free fluidic soft robots." In: *Science Advances* 10.51 (2024), eadr2433.

[100] Panagiotis Polygerinos, Nikolaus Correll, Stephen A. Morin, Bobak Mosadegh, Cagdas D. Onal, Kirstin Petersen, Matteo Cianchetti, Michael T. Tolley, and Robert F. Shepherd. "Soft Robotics: Review of Fluid-Driven Intrinsically Soft Devices; Manufacturing, Sensing, Control, and Applications in Human-Robot Interaction." In: *Adv. Eng. Mater.* 19.12 (2017). https://doi.org/10.1002/adem.201700016, p. 1700016.

[101] Daniel J Preston, Haihui Joy Jiang, Vanessa Sanchez, Philipp Rothemund, Jeff Rawson, Markus P Nemitz, Won-Kyu Lee, Zhigang Suo, Conor J Walsh, and George M Whitesides. "A soft ring oscillator." In: *Science Robotics* 4.31 (2019), eaaw5496.

[102] Daniel J. Preston, Haihui Joy Jiang, Vanessa Sanchez, Philipp Rothemund, Jeff Rawson, Markus P. Nemitz, Won-Kyu Lee, Zhigang Suo, Conor J. Walsh, and George M. Whitesides. "A soft ring oscillator." In: *Science Robotics* 4.31 (June 26, 2019). Publisher: American Association for the Advancement of Science, eaaw5496. DOI: 10.1126/scirobotics.aaw5496.

[103] John W Romanishin, Kyle Gilpin, Sebastian Claici, and Daniela Rus. "3D M-Blocks: Self-reconfiguring robots capable of locomotion via pivoting in three dimensions." In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 1925–1932.

[104] Daniela Rus and Michael T Tolley. "Design, fabrication and control of soft robots." In: *Nature* 521.7553 (2015), pp. 467–475.

[105] Daniela Rus and Michael T. Tolley. "Design, fabrication and control of soft robots." In: *Nature* 521.7553 (May 28, 2015), pp. 467–475. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature14543.

[106] Baudouin Saintyves, Matthew Spenko, and Heinrich M Jaeger. "A self-organizing robotic aggregate using solid and liquid-like collective states." In: *arXiv preprint arXiv:2304.03125* (2023).

[107] William Savoie, Thomas A Berrueta, Zachary Jackson, Ana Pervan, Ross Warkentin, Shengkai Li, Todd D Murphey, Kurt Wiesenfeld, and Daniel I Goldman. "A robot made of robots: Emergent transport and control of a smarticle ensemble." In: *Science Robotics* 4.34 (2019), eaax4316.

[108] HAH Schomaker, S Picella, A Küng Garcia, LC van Laake, and JTB Overvelde. "Robust Phototaxis by Harnessing Implicit Communication in Modular Soft Robotic Systems." In: *Advanced Functional Materials* (2024), p. 2310932.

[109] Milad Shafiee, Guillaume Bellegarda, and Auke Ijspeert. "Viability leads to the emergence of gait transitions in learning agile quadrupedal locomotion on challenging terrains." In: *Nature Communications* 15.1 (2024), p. 3073.

[110] Suraj Shankar, Anton Souslov, Mark J Bowick, M Cristina Marchetti, and Vincenzo Vitelli. "Topological active matter." In: *Nature Reviews Physics* 4.6 (2022), pp. 380–398.

[111] Robert F Shepherd, Filip Ilievski, Wonjae Choi, Stephen A Morin, Adam A Stokes, Aaron D Mazzeo, Xin Chen, Michael Wang, and George M Whitesides. "Multigait soft robot." In: *Proceedings of the national academy of sciences* 108.51 (2011), pp. 20400–20403.

[112] Masahiro Shimizu, Akio Ishiguro, and Toshihiro Kawakatsu. "Slimebot: A modular robot that exploits emergent phenomena." In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE. 2005, pp. 2982–2987.

[113] Jun Shintake, Vito Cacucciolo, Dario Floreano, and Herbert Shea. "Soft Robotic Grippers." In: *Adv. Mater.* 30.29 (2018). https://doi.org/10.1002/adma.201707035, p. 1707035.

[114] Yuanping Song, Robert M Panas, Samira Chizari, Lucas A Shaw, Julie A Jackson, Jonathan B Hopkins, and Andrew J Pascall. "Additively manufacturable micro-mechanical logic gates." In: *Nature communications* 10.1 (2019), p. 882.

[115] Andrew Spielberg, Allan Zhao, Yuanming Hu, Tao Du, Wojciech Matusik, and Daniela Rus. "Learning-in-the-loop optimization: End-to-end control and co-design of soft robots through learned deep latent representations." In: *Advances in Neural Information Processing Systems* 32 (2019).

[116] Richard B Stein. "Some models of neuronal variability." In: *Biophysical journal* 7.1 (1967), pp. 37–68.

[117] Menachem Stern and Arvind Murugan. "Learning without neurons in physical systems." In: *Annual Review of Condensed Matter Physics* 14.1 (2023), pp. 417–441.

[118] Klaus M Stiefel and Glyn A Barrett. "Sea urchins as an inspiration for robotic designs." In: *Journal of Marine Science and Engineering* 6.4 (2018), p. 112.

[119] Xin Sui, Hegao Cai, Dongyang Bie, Yu Zhang, Jie Zhao, and Yanhe Zhu. "Automatic generation of locomotion patterns for soft modular reconfigurable robots." In: *Applied Sciences* 10.1 (2020), p. 294.

[120] Tzer Han Tan, Alexander Mietke, Junang Li, Yuchao Chen, Hugh Higinbotham, Peter J Foster, Shreyas Gokhale, Jörn Dunkel, and Nikta Fakhri. "Odd dynamics of living chiral crystals." In: *Nature* 607.7918 (2022), pp. 287–293.

[121] Falk Tauber, Marc Desmulliez, Olivier Piccin, and Adam A Stokes. "Perspective for soft robotics: the field's past and future." In: *Bioinspiration & Biomimetics* 18.3 (Mar. 2023). Publisher: IOP Publishing, p. 035001. ISSN: 1748-3190. DOI: 10.1088/1748-3190/acbb48.

[122] Falk Tauber, Marc Desmulliez, Olivier Piccin, and Adam A Stokes. "Perspective for soft robotics: the field's past and future." In: *Bioinspiration & Biomimetics* 18.3 (2023), p. 035001.

[123] Maryam Tebyani, Alex Spaeth, Nicholas Cramer, and Mircea Teodorescu. "A geometric kinematic model for flexible voxel-based robots." In: *Soft robotics* 10.3 (2023), pp. 517–526.

[124] Seppe Terryn, Joost Brancart, Dirk Lefeber, Guy Van Assche, and Bram Vanderborght. "Self-healing soft pneumatic robots." In: *Science Robotics* 2.9 (2017), eaan4268.

[125] Robin Thandiackal, Kamilo Melo, Laura Paez, Johann Herault, Takeshi Kano, Kyoichi Akiyama, Frédéric Boyer, Dimitri Ryczko, Akio Ishiguro, and Auke J. Ijspeert. "Emergence of robust self-organized undulatory swimming based on local hydrodynamic force sensing." In: *Science Robotics* 6.57 (Aug. 11, 2021). Publisher: American Association for the Advancement of Science, eabf6354. DOI: 10.1126/scirobotics.abf6354.

[126] Michael T. Tolley, Robert F. Shepherd, Bobak Mosadegh, Kevin C. Galloway, Michael Wehner, Michael Karpelson, Robert J. Wood, and George M. Whitesides. "A Resilient, Untethered Soft Robot." In: *Soft Robot.* 1.3 (2014). https://doi.org/10.1089/soro.2014.0008, pp. 213–223.

[127] N. Trayanova. "Whole-heart modeling: applications to cardiac electrophysiology and electromechanics." In: *Circulation Research* 108.1 (2011), pp. 113–128.

[128] Merel Van Diepen and Kristina Shea. "A spatial grammar method for the computational design synthesis of virtual soft locomotion robots." In: *Journal of Mechanical Design* 141.10 (2019), p. 101402.

[129] Lav R Varshney, Beth L Chen, Eric Paniagua, David H Hall, and Dmitri B Chklovskii. "Structural properties of the Caenorhabditis elegans neuronal network." In: *PLoS computational biology* 7.2 (2011), e1001066.

[130] Michael Wehner, Ryan L. Truby, Daniel J. Fitzgerald, Bobak Mosadegh, George M. Whitesides, Jennifer A. Lewis, and Robert J. Wood. "An integrated design and fabrication strategy for entirely soft, autonomous robots." In: *Nature* 536.7617 (2016). https://dx.doi.org/10.1038/nature19100, pp. 451–455.

[131] Michael Wehner, Ryan L. Truby, Daniel J. Fitzgerald, Bobak Mosadegh, George M. Whitesides, Jennifer A. Lewis, and Robert J. Wood. "An integrated design and fabrication strategy for entirely soft, autonomous robots." In: *Nature* 536.7617 (Aug. 2016). Publisher: Nature Publishing Group, pp. 451–455. ISSN: 1476-4687. DOI: 10.1038/nature19100.

[132] Justin Werfel, Kirstin Petersen, and Radhika Nagpal. "Designing collective behavior in a termite-inspired robot construction team." In: *Science* 343.6172 (2014), pp. 754–758.

[133] IC Wilkie. "Autotomy as a prelude to regeneration in echinoderms." In: *Microscopy research and technique* 55.6 (2001), pp. 369–396.

[134] Nialah Jenae Wilson, Steven Ceron, Logan Horowitz, and Kirstin Petersen. "Scalable and robust fabrication, operation, and control of compliant modular robots." In: *Frontiers in Robotics and AI* 7 (2020), p. 44.

[135] Jianguo Wu and Orie L Loucks. "From balance of nature to hierarchical patch dynamics: a paradigm shift in ecology." In: *The Quarterly review of biology* 70.4 (1995), pp. 439–466.

[136] Matheus S Xavier, Charbel D Tawk, Ali Zolfagharian, Joshua Pinskier, David Howard, Taylor Young, Jiewen Lai, Simon M Harrison, Yuen K Yong, Mahdi Bodaghi, et al. "Soft pneumatic actuators: A review of design, fabrication, modeling, sensing, control and applications." In: *IEEE Access* 10 (2022), pp. 59442–59485.

[137] Zechen Xiong, Yufeng Su, and Hod Lipson. "Fast untethered soft robotic crawler with elastic instability." In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 2606–2612.

[138] Xintian Yu, Binh Nguyen, and W Otto Friesen. "Sensory feedback can coordinate the swimming activity of the leech." In: *Journal of Neuroscience* 19.11 (1999), pp. 4634–4643.

[139] Eduardo S Zeron. "Positive and Negative Feedback in Engineeringand Biology." In: *Mathematical Modelling of Natural Phenomena* 3.2 (2008), pp. 67–84.

[140] Dandan Zhang, Thomas E Gorochowski, Lucia Marucci, Hyun-Taek Lee, Bruno Gil, Bing Li, Sabine Hauert, and Eric Yeatman. "Advanced medical micro-robotics for early diagnosis and therapeutic interventions." In: *Frontiers in Robotics and AI* 9 (2023), p. 1086043.

[141] Yao Zhao, Yinding Chi, Yaoye Hong, Yanbin Li, Shu Yang, and Jie Yin. "Twisting for soft intelligent autonomous robot in unstructured environments." In: *Proceedings of the National Academy of Sciences* 119.22 (2022), e2200265119.

[142] Yusen Zhao, Qiaofeng Li, Zixiao Liu, Yousif Alsaid, Pengju Shi, Mohammad Khalid Jawed, and Ximin He. "Sunlight-powered self-excited oscillators for sustainable autonomous soft robotics." In: *Science Robotics* 8.77 (2023), eadf4753.

[143] Shibo Zou, Sergio Picella, Jelle de Vries, Vera G Kortman, Aimée Sakes, and Johannes TB Overvelde. "A retrofit sensing strategy for soft fluidic robots." In: *Nature communications* 15.1 (2024), p. 539.

[144] Olga Zueva, Maleana Khoury, Thomas Heinzeller, Daria Mashanova, and Vladimir Mashanov. "The complex simplicity of the brittle star nervous system." In: *Frontiers in zoology* 15 (2018), pp. 1–26.

[145] Letizia Zullo and Binyamin Hochner. "A new perspective on the organization of an invertebrate brain." In: *Communicative & integrative biology* 4.1 (2011), pp. 26–29.

LIST OF PUBLICATIONS

[1] H. A. H. Schomaker, S. Picella, A. Küng Garcia, L. C. van Laake, and J. T. B. Overvelde. Robust phototaxis by harnessing implicit communication in modular soft robotic systems. *Advanced Functional Materials*, 2024.

[2] A. Comoretto, H. A. H. Schomaker, and J. T. B. Overvelde. Physical synchronization of soft self-oscillating limbs for fast and autonomous locomotion. *Science*, 2025.

[3] H. A. H. Schomaker, J. de Vries, and J. T. B. Overvelde. Echinoderm-inspired autonomy for soft-legged robots. *submitted*.

[4] H. A. H. Schomaker and J. T. B. Overvelde. Adaptive behavior in minimal oscillator networks through coevolution of body and brain. *in preparation*.

[5] H. A. H. Schomaker*, L. Huntjens*, and J. T. B. Overvelde. Fluidic oscillators as neurons for soft robots. *in preparation*.

[6] S. Mohanty, P. Baconnier, H. A. H. Schomaker, A. Comoretto, M. van Hecke, and J. T. B. Overvelde. Buckling-induced navigational autonomy and sensing in soft robots. *in preparation*.

ACKNOWLEDGMENTS

I want to begin by thanking my PhD supervisor Bas for all the compassion and the endless curiosity you brought into our collaboration. Your positive outlook and inquisitive nature have been a constant source of inspiration. You always managed to bring an uplifting note to our meetings, especially during moments when motivation or clarity felt far away. Working with someone who views the world with such openness has shaped the way I approach research and has made this journey far more enjoyable.

I also want to thank my co-promotor Simon for his interest in my work and for his guidance in the final stages of this thesis. Your feedback helped me bring everything together and gave me the confidence to take the last steps that once felt impossibly large.

My deepest gratitude goes to my ohana, Dorelies, Veer and Annelies. You have given me a warm and loving foundation and shaped me into the person I am today. Dorelies, thank you for sparking endless creative projects in me. Nothing was ever too big or too absurd to try when you were there beside me, always ready to help and always trusting that I could pull it off. Veer, thank you for being the steady listening ear that I have so often relied on. Your calmness taught me how to gather my thoughts, how to build a compelling argument, and how to listen to others just as carefully. Annelies, you gave me the little rascal outlet that every child needs. Standing on the back seat of your bike or cycling through an orange light just as it turned red gave me a taste of freedom and mischief that I still cherish. Time with the three of you always leaves me laughing, lighter, and with a renewed smile. I am endlessly grateful for the joy and balance you bring into my life.

Eva, thank you for supporting me throughout this entire PhD journey. You reminded me again and again that there is more to life than work, especially during the moments when I tended to forget. Thank you for listening to me go on about deadlines, experiments and planning, and for teaching me that it is okay to share the things that worry me. Being with an academic is not always easy, especially in the final year of a PhD when stress becomes a constant companion, yet you never complained. Your patience, warmth and love carried me through more than you know.

I also want to thank Mirjam Schomaker. More than four years ago you helped me navigate my first long pieces of scientific writing, at a time when I had no idea how to structure a text or make it readable. Those early lessons stayed with me and still form the foundation of this book today. Without that early guidance, writing this thesis would have been a much more daunting task.

Finally, I want to thank Juul for supporting me at the very beginning of my PhD, when everything felt new and overwhelming and I had no idea where I was heading. Your encouragement helped me take the first steps with confidence.

And of course, I want to thank my AMOLF family. Working in such a warm, curious and collaborative environment has made these years truly special. I am grateful for every discussion, every shared frustration, every breakthrough and every moment of laughter that carried us through the challenges of research. Niels, you are the stepping stone for all of us in the Soft Robotic Matter group. You provide a foundation of knowledge that we simply cannot do without. You taught me everything I know about printing and fabrication, and even now, whenever I need to make something and I am not sure how to begin, you are always the first person I go looking for. It is not only your vast technical knowledge that brings us to your door, but also the supportive and caring way you carry yourself. Working with you has been both grounding and inspiring.

I still remember my very first day at AMOLF during my application for a master internship. Giorgio, you opened the door to your office with a warm Italian smile and told me I could sit on your drum music box. You showed me what you were working on and it looked absolutely spectacular. Thank you for guiding me through my initiation at AMOLF and for giving me such an incredibly warm welcome. It made all the difference.

To the members of what felt to me like the first generation of the Soft Robotic Matter group, thank you. Your presence, your discussions, and yes, your beards laid the foundation on which we are all standing today. You created a culture of curiosity, humour and craftsmanship that defined the group long before many of us joined.

Agustín, thank you for your sharp and critical mind, and perhaps even more for your witty character and jokes. You always managed to keep things light while still pushing us all to think one step deeper.

Luuk, I am still mesmerized by your endless stories and ideas. You taught me that everything can be engineered and often overengineered, especially during lunch and coffee. Your creativity and enthusiasm have left a lasting mark on me.

Maziar, even though our projects were far apart, your presence in the lab always brought fun and excitement. I learned a great deal from your presentations and from the way you approached engineering challenges with such clarity and joy.

I would also like to thank what feels like my own generation of the Soft Robotic Matter group.

Alberto, office mate, coffee companion and partner in crime. You are hilarious, and I deeply admire how comfortable you are in being yourself. We had so much fun together and I consider you a true friend, even now that you live in another country. The office was never the same after you left.

Jelle, we started our internships together, and together with Alberto we made a great team. Those early days shaped so much of what was to come, and I look back on them fondly.

Sergio, you are an incredible scientist and an even kinder person. Thank you for all the conversations we had. You were always ready to help and think along, and your patience and clarity made every discussion valuable. We collaborated on my first project in the group and you showed me what perseverance in academia truly looks like. I have deeply enjoyed our time together over these past four years, and I am grateful for everything you taught me.

Paul, your witty jokes and your lively presence always made my day. Your interest in the smallest details and your deep dives into ideas reminded me that science does not always need to be practical. There is joy in the theory as well, and you helped me rediscover that.

Shibo, you are one of the funniest, kindest, and hardest-working colleagues I have ever had. You were always ready to help, always eager to learn, and always enthusiastic. Your dedication lifted the whole group.

Katrien, your unparalleled mastery of the English language and your love for etymology made lunch and coffee breaks a treasure trove of fun facts and practical knowledge. I learned a lot from your industrial design and manufacturing skills, which you were always ready to explain and demonstrate. What amazes me the most is your ability to acquire new knowledge and make it your own. Within a month, it felt like you had been a soft robotic engineer in our group for all your life.

Sumit, you are full of stories and full of energy. I loved collaborating on your ciliary walkers, and I am grateful for the creativity you brought into the lab.

And to the new generation of the Soft Robotic Matter group, thank you for continuing the spirit of the group and shaping it in your own way.

Stijn, Mister Cool. When you joined our office, it immediately felt like the perfect fit. Your view of the world and your sense of how life should be lived are things I genuinely admire.

Nienke, you brought a perspective that science is not only driven by curiosity, but also by the desire to make something that helps others. It has been a pleasure working with you and seeing how your practical vision enriches the research around you.

Bob, what a man. I love listening to your presentations. You put tremendous effort into your slides and posters, turning complex science into something clear, beautiful and exciting. Your clay creations and the Pokémon cards you bring to AMOLF show perfectly that science and art go hand in hand. You make the group brighter in many ways.

Elif, thank you for the wonderful times when you were at AMOLF and for your fearless spirit during the group outings. Your biking skills and the bold way you rode your bike down the path are moments I will never forget. You brought joy and energy wherever you went.

Sara, you were the first student I supervised, and you taught me more than you might realize. Guiding you through your project helped me understand how to explain, structure and support someone in the academic process. You showed me what mentorship can look like, and I am grateful for that experience.

Yorick, your endless curiosity and your fascination with the most niche and unexpected topics are things I still remember with joy. Working with you was a true pleasure. You approached every question with enthusiasm, and I learned so much from the fresh perspective you brought.

Antonio, you were an incredible help with the Gridbot project. Your dedication and careful work played a central role in shaping it into a fantastic paper. I am grateful for the precision, creativity, and motivation you brought into the project.

Erwin, thank you for your deep understanding of reservoir computing and for the insights you shared about harnessing the complexity hidden in fluidic interactions with soft silicone tubes. Even though your contributions do not appear directly in the thesis, the knowledge you provided helped build the foundations that underpin several of the chapters. I am truly grateful for your thoughtful explanations and for the time you invested in helping me understand these systems.

Luca, you are an awesome person, and working with you has been a great pleasure. Your research on variable resistances and your ability to draw meaningful analogies with electronic circuits gave us the knowledge we needed to design and fabricate the oscillator networks. Your contributions were essential, and I appreciate both your hard work and your enthusiastic spirit.

Lio, you accomplished so much in the short time we worked together. I came to you with little more than an idea, still full of gaps and uncertainties. You were thrown into the deep, yet you learned incredibly quickly and approached every challenge with determination. You never doubted the broader vision of the project, and you worked tirelessly to make the experiments succeed. Thanks to your perseverance and dedication, we were able to write an exceptional chapter about the work. I am very grateful to have had the opportunity to supervise you, and I am excited to see the many great things you will achieve in the future.

Moreover, I want to thank all the other scientists and colleagues who have been part of the AMOLF community and our soft robotic matter group throughout my years at AMOLF.

Thank you to all my roommates throughout the years of my PhD. Living with each of you shaped these years in ways that stretched far beyond academia. You created a home where laughter, late-night conversations, and shared routines became the balance I needed next to research.

Lieve, heart-warming, kind, and with a touch of misophonia, there is truly no one I know with a stronger work ethic. Yet it is your norms and values that I admire most. You bring people together and make them feel comfortable, both in life and in our home at the Alexanderkade. Your weekly schedule still haunts me a little, and probably will for years, but living with you has been a gift.

Melle, living with you during the years of my PhD was its own adventure. You arrived at the Alexanderkade during your first experience living on your own, and by the time I left, you had transformed into one of the patriarchs of the house. We leave not just as roommates, but as friends who once conquered the Himalaya together, and that bond will last far beyond the walls of our home.

Lola, best baker of Amsterdam Oost. Living with you was the best culinary experience I could imagine. Fresh bread that lasted the entire week and cakes that could win any bake off made the Alexanderkade feel like the coziest place on earth. Your warmth and your kitchen creations brightened so many days.

Henriëtte, with the energy of ten people and a movie knowledge that beats my own, it has been an absolute pleasure living with you. Your enthusiasm filled the house, and your presence made every shared moment lively and memorable.

Job, sharing a floor with you has been epic. Coming home from work and finding you in the living room meant I was set for a great evening at home or going to a party. I probably learned more about tennis from you than from all my tennis lessons combined. Your good humor and calm presence made the house feel like home.

Tess, we started as roommates, then I third-wheeled your relationship with Bart for a while at the Alexanderkade, and we ended my time at AMOLF as colleagues. What a ride. Your endless creative ideas and side projects still surprise me, and your drive and imagination inspire me. I am proud to call you my friend.

Dirk, you got me into the best housing situation I could have dreamed of, and we have been friends ever since. Doing our master's together was a blast, and your work-hard-play-hard mentality did wonders for me. You helped me get the best grades of my study career, and I am grateful for the years of friendship and shared ambition.

Last but certainly not least, I want to thank all of my dear friends for their support and for helping me become who I am today. A special thanks has to be given to the following:

Luc, you are always ready to listen, and there is nothing I cannot share with you. You are not afraid to talk about your deeper emotions, and that is something I admire deeply and continue to learn from.

Jip, also known as dokter Bibber, you are the embodiment of a great friend. You sense when something is wrong long before someone realizes it themselves. With you, anyone can open their heart, and you will always stand up for your friends no matter what. That is a rare gift.

Bart, PhD brother from another mother and definitely and specifically not to be listed under roommates. Over the past fifteen years, we have done so many things together that I have lost count: high school, South America, our bachelor's, our master's and living together, just to name a few. And specifically in relation to this thesis, it has been fantastic to have a friend who not only understands what it is like to do a PhD but also grasps the content I am working on. Your support has meant more than I can express.

Mila, besides being friends for so many years, I am so grateful that we studied together, especially during the years when I was still a whirlwind of disorganized chaos. You always made sure I met my deadlines and knew where to show up for exams, something I will never forget. No one will be left behind is a principle you live by, and it is something I admire dearly.

Rafael and Boris, who would have thought that when we started high school together, we would still be friends in our thirties. Raffie, besides being incredibly funny, you give me a perspective on society that I would never have found on my own. You care deeply about the world around you and choose to work on it directly, not with research that may or may not help in the long run, but by standing on the streets and carrying your ideals into real, stone-cold politics. Boris, cycling with you is always filled with laughter, and I have always admired your down-to-earth view on life. You know very well what you like and what you do not like, and that clarity is something I can learn a great deal from.

Many more people helped me get where I am today, and I am grateful to all of you. I am terribly afraid that I forgot someone, but please know that I am just as thankful to you. Your presence, whether small or large, shaped this journey.

ABOUT THE AUTHOR

Mannus Schomaker was born on November 30th, 1994, in Amsterdam, The Netherlands. He completed his secondary education at the Montessori Lyceum Amsterdam. Fascinated by the interplay between science, nature, and human behaviour, he enrolled in the interdisciplinary Bachelor programme Beta-Gamma at the University of Amsterdam, majoring in Brain and Cognition and minoring in Programming. Already during these years, he became inspired by optimization algorithms and how nature itself often provides elegant solutions to complex problems.

Driven by this fascination, he continued his studies in the joint Computational Science Master programme of the University of Amsterdam and Vrije Universiteit Amsterdam. Influenced by computational biology, complexity theory and machine learning, he became intrigued by the way simple organisms can display surprisingly complex behaviour in order to survive in an unstructured world. For his Master's thesis, he joined the Soft Robotic Matter Group at AMOLF, under the supervision of Professor J.T.B. (Bas) Overvelde, where he investigated how learning could emerge in soft robots in a decentralized manner.

In December 2020, Mannus began his PhD research in the same group. His work explores embodied computation, drawing inspiration from natural systems to understand how soft robots can develop goal-directed behaviour through local sensing and feedback rather than through centralized control. Throughout his doctoral years, he spent countless hours experimenting with simulations, soft materials, and unconventional robotic systems, accompanied by long discussions and longer coffee breaks at the AMOLF coffee corner, where no idea was considered too wild to engineer into existence.

Defending his thesis in January 2026, Mannus continues to search for the principles that allow complex behaviour to emerge from simple interacting parts.