

The Propulsion of Reconfigurable Modular Robots in Fluidic Environments



The
University
Of
Sheffield.

Matthew J. Doyle

Supervisors: Dr. Roderich Groß
Dr. Sean Anderson

Department of Automatic Control and Systems Engineering

This dissertation is submitted for the degree of
Doctor of Philosophy

July 2020

Abstract

Reconfigurable modular robots promise to transform the way robotic systems are designed and operated. Fluidic or microgravity environments, which can be difficult or dangerous for humans to work in, are ideal domains for the use of modular systems. This thesis proposes that combining effective propulsion, large reconfiguration space and high scalability will increase the utility of modular robots. A novel concept for the propulsion of reconfigurable modular robots is developed. Termed Modular Fluidic Propulsion (MFP), this concept describes a system that propels by routing fluid through itself. This allows MFP robots to self-propel quickly and effectively in any configuration, while featuring a cubic lattice structure. A decentralized occlusion-based motion controller for the system is developed. The simplicity of the controller, which requires neither run-time memory nor computation via logic units, combined with the simple binary sensors and actuators of the robot, gives the system a high level of scalability. It is proven formally that 2-D MFP robots are able to complete a directed locomotion task under certain assumptions. Simulations in 3-D show that robots composed of 125 modules in a variety of configurations can complete the task. A hardware prototype that floats on the surface of water is developed. Experiments show that robots composed of four modules can complete the task in any configuration. This thesis also investigates the evo-bots, a self-reconfigurable modular system that floats in 2-D on an air table. The evo-bot system uses a stop-start propulsion mechanism to choose between moving randomly or not moving at all. This is demonstrated experimentally for the first time. In addition, the ability of the modules to detect, harvest and share energy, as well as self-assemble into simple structures, is demonstrated.

Acknowledgements

I would firstly like to thank my supervisors Dr. Roderich Groß and Dr. Sean Anderson. Roderich has given me support, guidance and understanding. His academic expertise and knowledge has been paramount to my own development as a researcher, particularly when it comes to producing scientific output.

I would like to thank all the current and past members of the Natural Robotics Lab, who made me feel welcome and included from the moment I arrived. In particular I would like to thank Stefan, Yuri, Gabriel, Fernando, Chris, Wei, Shen, João, Anil, Matt, Isaac and Yue. Fernando, Shen and Haoyu, thanks for the chats and the fun times. João, I'd like to think we made a good team. I would also like to thank other colleagues and office-mates. James and Matt, thanks for all the useful advice.

To all my collaborators I owe a huge debt of gratitude. A problem shared is a problem halved. Thank you in particular to Juan, João, Fernando, Yue and Xinyu. Juan, thank you for developing the fantastic evo-bot platform. Yue and Xinyu, thank you for your great master's project work on the MHP hardware prototype.

Thanks go to my friends outside of the department. Faris, Rob and Chris, thank you for helping to keep me sane.

I would like to thank the IBD team at the Sheffield teaching hospitals. Without their committed expertise and care I would not have been able to undertake this PhD. Long may the NHS continue to serve the public.

To my mother, Christine, goes the ultimate gratitude. Thank you for your ceaseless care and support, and for giving me all of the opportunities I've had. Thank you Norman for helping when I needed it.

Table of contents

List of figures	xi
List of tables	xix
1 Introduction	1
1.1 Motivation: Why fluidic environments?	2
1.2 Problem statement and objectives	4
1.3 Contributions	6
1.4 Publications	8
1.5 Outline	9
2 Related work	11
2.1 Land based systems	11
2.2 Self-propelled fluid based systems	16
2.2.1 AMOUR	18
2.2.2 The Neubots	18
2.2.3 Hydron	19
2.2.4 The Distributed Flight Array	19
2.2.5 ANGELS	20
2.2.6 REMORA	20
2.2.7 The work of Naldi et al.	20
2.2.8 TEMP	21
2.2.9 Roboat	21

2.2.10	ModQuad	21
2.2.11	GridDrones	22
2.2.12	Analysis	22
2.3	Externally-propelled fluid based systems	23
2.3.1	The work of Breivik et al.	23
2.3.2	The work of Griffith et al.	23
2.3.3	The works of White et al. and Tolley et al.	25
2.3.4	Programmable parts	25
2.3.5	Tribolon	26
2.3.6	The work of Virgo et al.	26
2.3.7	Lily	27
2.3.8	The work of Ganesan et al.	27
2.4	Self-replicating and evolving modular ecosystems	27
2.4.1	The work of Chirikjian et al.	28
2.4.2	The REPLICATOR and SYMBRION projects	29
2.4.3	The work of Eiben et al.	29
2.4.4	The work of Brodbeck et al.	29
2.4.5	Virtual creature evolution	30
2.5	Summary	31
3	Modular Fluidic Propulsion	33
3.1	Concept	34
3.1.1	Modular Hydraulic Propulsion	36
3.1.2	Modular Pneumatic Propulsion	36
3.1.3	Sensing and communication	36
3.1.4	Mathematical modelling	37
3.2	Static 2-D force and torque analysis	40
3.2.1	Number of possible forces	41
3.2.2	Number of possible torques	42
3.3	Directed locomotion task and controllers	46

3.3.1	Task definition	46
3.3.2	Occlusion-based motion controller (dec)	47
3.3.3	Communication-enabled motion controller (dec-com)	48
3.3.4	Obstacle avoidance motion controller (dec-obs)	48
3.4	Controller analysis	49
3.4.1	Preliminaries	49
3.4.2	Robot dynamics	50
3.4.3	Occlusion model	51
3.4.4	Distance condition	53
3.5	Assumptions and limitations	57
3.6	Summary	59
4	Validating the MFP concept	61
4.1	Simulations	62
4.1.1	General setup	62
4.1.2	Time to completion	67
4.1.3	Energy expenditure	71
4.1.4	Negotiating obstacles	71
4.1.5	On normality	73
4.1.6	Actuator errors and varying resolution	74
4.1.7	Results	75
4.2	Hardware	78
4.2.1	Chassis	78
4.2.2	Electronics and control	80
4.3	Experiments	82
4.3.1	Setup	82
4.3.2	Results	82
4.3.3	Comparison with theory	83
4.4	Orientation analysis	86
4.5	Summary	87

5	Validating the evo-bot concept	89
5.1	Concept	90
5.2	System modifications	94
5.2.1	Modules	95
5.2.2	Environment	97
5.3	Experiments	99
5.3.1	Light detection and stop-start motion control	99
5.3.2	Energy harvesting and sharing	101
5.3.3	Polymer formation	102
5.4	Summary	104
6	Conclusion	107
6.1	Discussion	109
6.1.1	Achievements	109
6.1.2	Limitations	111
6.2	Future work	112
	Appendix A Static torque analysis of a 2-D MFP robot	115
	Appendix B Outline of a more detailed model of fluid flow within an MFP robot	123
B.1	Flow rate and pressure from conservation laws	123
B.2	Worked example	125
	References	129

List of figures

2.1	(a) PolyBot in rolling track configuration (©2002 IEEE). (b) M-TRAN III in a quadrupedal travel configuration (©2008 SAGE Publishing). (c) Three connected s-bots lifting one another to climb a step (©2006 IEEE). (d) Two M-Blocks modules (©2015 IEEE). Reprinted from [106], [55], [36], [77] respectively.	12
2.2	(a) AMOUR robot with four thrusters visible (©2010 SAGE Publications). (b) DFA with 10 modules (©2014 SAGE Publications). (c) Render of a three module ANGELS robot in chain configuration (©2013 IEEE). (d) TEMP robot with six modules acting as a landing platform for a quadrotor (©2014 IEEE). Reprinted from [93], [69], [90], [67] respectively.	16
2.3	(a) Six programmable parts modules arranged into a hexagon (©2007 IEEE). (b) Six Tribolon modules (with freezing connectors) (©2008 IEEE). (c) 10 passive modules of Tolley et al. [92] self-assembled via fluid pumping (©2010 IEEE). (d) Five Lily modules self-assembled into a cross in their experimental environment (©2016 IEEE). Reprinted from [48], [62], [92], [39] respectively.	24

-
- 3.1 The Modular Fluidic Propulsion (MFP) concept and two implementation options: Modular Hydraulic Propulsion (MHP) and Modular Pneumatic Propulsion (MPP). (a) A generic MFP robot consisting of 12 modules. Each module has one actuator per face that it can use to route fluid. By routing the fluid, the robot propels. Yellow lines denote internal faces, all other faces are external. (b) A 2-D MHP robot. The environmental and routing liquid is indicated in blue. Green and dark grey boxes indicate active and inactive actuators, respectively. (c) A 2-D MPP robot. The gas propellant is indicated in yellow. In both (b) and (c) blue arrows indicate the direction of fluid flow through the robot and the dashed black arrow indicates the resultant direction of motion of the robot. 35
- 3.2 An example 2-D MFP robot consisting of two modules. The goal $\mathbf{g}(t)$ is indicated. An obstacle is shown in dark gray. Sensor/actuator nodes on each module are indicated by small circles. Selected sensors states are indicated. The c_{ij} refer to the connection sensor states, the d_{ij} to the goal sensor states and the r_{ij} to the obstacle sensor states. 37
- 3.3 One of the proposed controllers allows modules to communicate. Illustration of the communication paths for a 2-D MFP robot. Communication between faces A_1 and A_2 takes place over the full width of the robot, as information can be passed from one module to the next. The row below this has a gap in the robot structure. Faces B_1 and B_2 can communicate as information is passed between the two modules. Faces C_1 and C_2 can communicate as they are part of the same module. 40
- 3.4 An l_1 by l_2 robot firing its actuators (shown in green) to achieve maximum force along the x_1 axis. 42
- 3.5 An l_1 by l_2 robot firing its actuators (shown in green) to achieve maximum torque. The blue modules highlight the $l_1 = 1$ case. 43

-
- 3.6 The occlusion-based motion controller `dec` demonstrated with a convex MHP robot. Both the robot center of mass \mathbf{o} and the target point $\mathbf{g}(t)$ are indicated. The small black arrows indicate the individual forces applied by occluded faces. The dashed arrow indicates the net force \mathbf{f} 46
- 3.7 A 2-D MHP robot consisting of modules of side length s . The origin of the coordinate system, \mathbf{o} , is the robot center of mass. The Axis-Aligned Bounding Box of the robot, indicated with a dotted box, has size \mathbf{l} and geometrical center \mathbf{c} . The goal position $\mathbf{g}(t)$ is indicated. Analysis regions A and B are delineated with dashed lines. 50
- 3.8 Casting rays from $\mathbf{g}(t)$ over the height of the bounding box to determine minimum force in the x_1 direction. 51
- 3.9 A 2-D MFP robot composed of 20 modules. The robot center of mass \mathbf{o} and the goal point $\mathbf{g}(t)$ are indicated. The small arrows indicate activated actuators. Green arrows indicate the actuators which make a net contribution to the total force acting on the robot. The angle ϕ between the total force vector \mathbf{F} and the vector $\mathbf{g} - \mathbf{o}$ is greater than $\pi/2$, hence Equation 3.28 does not hold. Note that while this means the robot does not move towards the goal point at this instant, it does not imply that the robot will never reach the goal. In this example, once the force is applied to the robot and the goal position moves, a different set of actuators will be activated which do provide $\phi \leq \pi/2$ 55
- 4.1 An image taken from the simulator showing a randomly generated robot moving towards the target position on the right (green sphere). Green and white module faces represent active and inactive actuators respectively. . . . 63
- 4.2 An image taken from the simulator showing a 5^3 cubic robot in the slalom environment. The robot is using the `dec-obs` controller and moving from its starting position on the left (red point) to the target position on the right (green point). The grey cubes are the slalom obstacles. 63

4.3	The distribution of times taken in the slalom environment using the dec-obs controller with different sensor lengths (σ). Each boxplot is composed of 40 values.	65
4.4	The distribution of number of collisions in the slalom environment using the dec-obs controller with different sensor lengths (σ). Each boxplot is composed of 40 values.	66
4.5	Boxplots showing calibration of the centralized controller. The i th PID set corresponds to values of $P_T = 1$, $D_T = 5^{(i/16)\%4}/10$, $I_T = 0$, $P_R = 5^{(i/4)\%4}/10$, $D_R = 5^{i\%4}/10$ and $I_R = 0$ (where $\%$ is the modulo operator, P_T is the proportional translation component, D_T is the derivative translation component, I_T is the integral translation component, P_R is the proportional rotation component, D_R is the derivative rotation component and I_R is the integral rotation component).	68
4.6	Distribution of times taken for cubic robots and random robots to reach the target position. Each box plot is composed of 100 trials. The dec, dec-com and cen controllers are shown. The horizontal blue line represents the theoretical minimum time for a 5^3 robot to reach the target position. Note that the y axis is truncated for readability.	69
4.7	Distribution of energy required for cubic robots and random robots to reach the target position. Each box plot is composed of 100 trials. The dec, dec-com and cen controllers are shown.	70
4.8	Distribution of time taken for cubic robots using the dec, dec-obs and cen controllers. The boxplots are composed of 100, 100 and 95 trials respectively.	72
4.9	Distribution of number of collisions for cubic robots using the dec, dec-obs and cen controllers. The boxplots are composed of 100, 100 and 95 trials respectively.	73

-
- 4.10 Color maps showing the success rates and median velocities of robots using the dec controller when subjected to permanently failing actuators. The velocities are averaged over successful trials only. Where there is a success rate of zero, the velocity is also recorded as zero. 76
- 4.11 The results of MHP robots working with misaligned actuators. The bars show the success rate of the robots (out of 100) for the dec (orange) and cen (blue) controllers. The overlaid boxplots show the distributions of velocities for successful trials. 77
- 4.12 The hardware prototype of the MHP system floating on water in a tank. Each module is square shaped and contains four micro-pumps and an internal reservoir, four light sensors, a battery, and a microcontroller. 78
- 4.13 Prototype module: (a) Assembled module. (b) Internal circuitry in the opened upper section (top right), micropumps in situ in the bottom section (bottom right) and annotated module (left). A: External programming port. B: Magnetic switch (mounted internally). C: The corresponding magnet. D: LDR mounted behind a clear barrier. E: Micropump output port. F: Connection magnets (mounted internally). 79
- 4.14 The experimental environment and an example path taken by a four module configuration. Each module of the robot executes the decentralized motion controller. The left hand and right hand dashed orange lines show the start and finish lines respectively. The white line shows the path the robot took from left to right. The torch is located outside the water tank on the right hand side. 81
- 4.15 All 9 distinct robot configurations using four modules. These are configurations which cannot be transformed into another configuration by rotation or mirroring. The red dots indicate 2D centers of mass. 83

-
- 4.16 Streamlined or rotated 2-D MFP robots. Each robot is moving to the the right, towards the indicated goal point. Due to the symmetry of the situation none of the robots will rotate. (a) An MFP robot with a streamlined shape. (b) An MFP robot with a less streamlined shape, that will nevertheless experience the drag force as the one in (a). (c) A 5^2 robot moving face-on towards the goal. (d) The same 5^2 robot moving vertex-on towards the goal. 88
- 5.1 The evo-bot environment containing a “primordial soup” of modules. Along with individual modules, growing polymers (a), completed organisms (b) and replicating organisms (c) are present. The arrows indicate the general direction of motion induced by the external propulsion 91
- 5.2 An example of the evo-bot growth process. Black arrows indicate module faces which are accepting new connections. (a) An individual energy module allows an additional module to connect on any face. (b) An interaction module connects to form a 2 module polymer. New connections are only accepted on either end of the polymer. (c) A boundary module connects to form a 3 module organism, which stops growing. 93
- 5.3 An example of the evo-bot replication process. Black arrows indicate module faces which are accepting new connections. Red arrows indicate the disconnection of connected modules. (a) An organism boundary module accepts new connections on one of its perpendicular faces. (b) A new boundary module connects. The next module in the original organism now accepts new modules on the same side. (c) A new interaction module connects. The final module in the original organism now accepts new connections on the same side. (d) A new interaction module connects. This is the wrong type of module, so it is rejected. (e) A new energy module connects and the replicated organism is complete. (f) The original organism and the replicated organism disconnect from one another. 94

5.4	The evo-bot hardware. (a) A single evo-bot module, showing chassis (A), Connectors (B), solar cells (C), servo motors (D) and PCB (E). (b) Six evo-bot modules, five of which are connected.	95
5.5	Modifications made to the evo-bot chassis design to improve successful connection rate. (a) Additional permanent magnets are mounted on top of the module. In addition the main connection magnet has been replaced with a larger one. (b) A new servo horn designed to hold a larger magnet. (c) An alternate connector design with the electrical connectors mounted directly into the face of the chassis. (d) Overlapping jaws highlighted with red triangles and blue lines.	96
5.6	Overview of the experimental environment, an air table, including the overhead lamp (A), the side fans (B_1 to B_8), the overhead camera (C), and the fan control box (D).	98
5.7	Results from the light detection and motion control experiment. The side fans are indicated by large blue circles. The yellow circle in the bottom right indicates the position of the light source, and the green circle in the top left the module starting position. The colored dots represent the final positions of the modules (10 trials per color), from top left to bottom right: orange—0.3 V, teal—0.4 V, blue—0.5 V, red—0.8 V, brown—2.0 V.	100
5.8	Graphs showing the charging and subsequent discharging of an evo-bot module. Charging via (left) charging station, (center) solar panel, or (right) another evo-bot module.	101
5.9	A sequence of stills taken from a video of five evo-bot modules forming a linear polymer.	104
A.1	All unique torques of a 2 by 2 robot.	116
A.2	All unique torques of a 1 by 2 robot.	119
B.1	An MHP robot consisting of two modules. Three pumps are active, five are inactive. Pressure nodes and assumed flow directions are indicated.	126

List of tables

2.1	The capabilities of existing self-propelled fluid based modular robots. A hypothetical system based on the MFP concept is included for comparison.	17
3.1	Sensor to actuator mapping of the occlusion-based motion controller (dec)	47
3.2	Sensor to actuator mapping of the occlusion-based motion controller with communication (dec-com)	48
3.3	Sensor to actuator mapping of occlusion-based motion controller with obstacle avoidance (dec-obs)	49
4.1	Mean time taken and energy required in the open environment	67
4.2	Mean time taken and number of collisions in the slalom environment	72
5.1	Evo-bot module functionality	92
B.1	System constants	125

Chapter 1

Introduction

Robots are increasingly becoming part of both our everyday and not-so-everyday lives. As a means of automation they promise to reduce the workload on humans and allow us to live more carefree lives. As robust artificial constructs with large and fast computational capabilities they can complete tasks which would be dangerous for humans, or carry out complex functions in a small amount of time. And they arguably carry some inherent value by their very existence, both as examples of human engineering ingenuity and as burgeoning artificial creatures in their own right.

When it comes to designing artificial machines, inspiration often comes from nature. Billions of years of evolution have given rise to biological systems that are highly proficient at meeting the demands that their environment requires of them to survive. One repeated motif of the natural world is that of modularity. From the largest scale to the smallest, in biology, chemistry and physics, complex systems are composed of simpler constituent building blocks. Many living creatures consist of organs, each fulfilling a specific role within the body. Organs are modular in cells, cells in molecules, molecules in atoms and so on. Some creatures exhibit modularity even more explicitly. *Physalia physalis*, commonly known as the Portuguese man o' war, superficially resembles a jellyfish. However it is not one single animal but a colony of tiny animals of several different types. Each type fulfils a certain role within the colony. The marine invertebrate *Salpidae* moves by pumping water through its own body, feeding as it does so. Part of the *Salpidae* lifecycle involves reproducing asexually

to form long, physically connected chains of the organisms, which eventually separate from one another.

Promising to provide robust, flexible problem solving capabilities *reconfigurable modular robots* [105, 88] embody the concept of modularity. Such robots are composed of a single set, or a few sets, of homogeneous modules which alone have limited functionality. These modules can be physically, and usually electrically, connected to one another in order to produce a robot capable of solving a certain set of tasks. By changing the configuration of modules a robot with a different morphology and set of capabilities can be produced, with the ability to solve a different set of tasks. The potential of these systems is high. In the workplace a tool could be created on the fly for a specific use. Domestically, a piece of furniture could be created to suit a room. In industry the training demand of human operators could be reduced due to needing to learn to use only a single system. Furthermore, although the upfront cost of developing a modular system may be large, a system that achieves sufficient ubiquity could become cheaper to produce via mass manufacturing of the modules. Modular systems exhibit further advantages during operation. The large number of similar or identical modules mean that such systems typically have a high level of redundancy. If one part of the robot fails then the remaining parts can still maintain some degree of functionality. This makes modular systems well suited to working independently, and ideal for use in environments which are inaccessible or inhospitable to humans.

1.1 Motivation: Why fluidic environments?

Fluidic environments are generally difficult to work in for humans. Fluid pipelines can be too narrow for human technicians or mechanics to access. Deep water requires the use of cumbersome breathing apparatus. Even deeper water cannot be accessed directly by humans, necessitating submersibles built to withstand high pressure. Microgravity environments present similar issues. Inside a spacecraft or space station the lack of gravity can make it difficult for humans to maneuver. Outside, such problems are compounded by the lack of breathable atmosphere. The hostility of these environments to humans makes them an ideal

target for the deployment of robotic systems. Modular robotic solutions have been proposed for exploration, construction and maintenance in space [104, 11], in addition to construction and environmental monitoring [94, 12] in deep water.

Fluidic environments also present challenges for the deployment and operation of robotic systems. A liquid environment requires either some form of waterproofing, or control circuitry which is resistant to damage by the fluid. Some systems work in oil-based fluids, which do not react poorly with the circuitry. However deployment in real-world environments such as rivers or oceans would require waterproofing. Sensing and communication can also be limited in liquid environments, as signals can be attenuated and distorted to a greater extent than in air. These issues can be amplified in the case of modular systems, as modules need to be able to physically and electronically connect while maintaining watertightness and possibly the ability for both connected and unconnected modules to communicate with one another (for example to facilitate docking). Buoyancy control is an additional concern. Systems that float on the surface of a fluid must be sufficiently well balanced in any configuration to avoid tipping over. Systems which need to move around in 3-D within a fluid require either some form of buoyancy control, or must continuously expend energy to maintain a depth at which they are not neutrally buoyant.

Despite the difficulties involved with deploying robotic systems in fluid or microgravity environments, such environments could hold certain advantages for modular robots. For a neutrally buoyant system the force of gravity becomes less of a concern. A land based modular robot might be limited in the number or size of its modules due to the strength of the chassis and actuators required to overcome gravity. Systems in fluids which do not have these concerns could potentially be highly scalable. In addition, the reconfiguration space of land based systems can be limited by concerns about stability. A tall and thin configuration might topple over. Systems in fluids are less effected by these concerns (although might require some form of buoyancy control). Furthermore, fluidic environments provide the opportunity for propulsion methods which cannot be applied to land based robots.

1.2 Problem statement and objectives

Despite showing substantial promise modular reconfigurable robotic systems are yet to realize their full potential in the real world. To improve the general utility, a system should combine all of the following properties:

- *effective propulsion*: fast and precise movement as a connected robot, independent of modular configuration;
- *large reconfiguration space*: the number of distinct configurations increases rapidly with the number of modules;
- *scalable design*: performance scales favorably as the number of modules increases and/or their size decreases.

In particular, the ideal is that systems should have effective propulsion in any configuration. As well as making the system more flexible this would cut down the time and energy required to reconfigure (or be reconfigured) into a specific travel form. For a self-reconfigurable robot, this would decouple the net movement of the system from the potentially slow reconfiguration speed.

This thesis hypothesizes that it is possible to combine the above three properties in a fluidic environment, in the form of a highly scalable modular robot that can self-propel effectively in any possible configuration. The main technical challenge is to develop a novel hardware concept that allows actuators, mounted on each module, to provide effective propulsion to the robot independent of the 3-D lattice chassis topology. This problem is currently not well solved. In addition to developing a hardware concept and implementation, motion controllers should be designed, analyzed and proved to be sound. To engender scalability these controllers should be simple and decentralized.

Many current modular systems excel at one or two of the aforementioned properties, though at the expense of the remaining ones. Some systems prioritize effective propulsion, but either have a limited reconfiguration space [93, 69, 79, 56], or have limited propulsion capabilities as a connected unit [36, 3, 13, 72]. Other systems have large reconfigura-

tion spaces, but lack efficient movement capabilities as a connected robot in some [65] or most [110, 78, 45] configurations. Furthermore some systems, particularly those with a programmable matter focus, prioritize scalability [33, 78]. These systems can typically only ambulate, if at all [31], by progressive repositioning of the individual modules [33, 76].

This thesis proposes that the best environment to facilitate a system which displays the above three properties is a fluidic or microgravity environment. Therefore, the first two objectives of this thesis are:

Objective 1. To develop a modular robotic system that has a large reconfiguration space, good scalability and effective propulsion. In order to achieve all of this the system should operate in a fluidic environment. The system should have a cubic lattice structure, simple hardware that can potentially be miniaturized, simple decentralized control, and the ability to move well in any configuration.

Objective 2. To validate the above concept via theory, simulation and experimentation. The robot controllers should be able to provably move the robot towards a goal point. Simulations can analyze 3-D systems with large numbers of modules moving in different environments. A 2-D hardware prototype that floats on the surface of water should be developed. Experiments can be conducted to show that the prototype in a variety of configurations can successfully reach a goal location using the decentralized controller.

The above three properties can be supplemented by the ability of a system to self-reconfigure. Self-reconfigurable systems can adapt to unexpected scenarios on the fly, and potentially operate even more independently from humans. Furthermore, self-reconfiguration allows the possibility of self-repair [89] or self-replication [109]. For situations in which systems are required to spend a very long time out of direct contact with humans a single robot may wear out. In addition, as the environment changes over time the robot may need to change to adapt to it. These problems can be solved if a modular system has the ability to self-replicate and to evolve over successive generations to meet the demands of the environment. The self-reconfigurable *evo-bot*¹ [37] system was designed to investigate self-replication and evolution of morphology of modular robots in an artificial ecosystem. The system floats on

¹Not to be confused with the EvoBot [41]

an air table to mimic microgravity and has a novel stop-start propulsion mechanism, making the evo-bots an ideal target for investigation. Therefore the third objective of this thesis is:

Objective 3. To prepare the evo-bot system for experimentation and validate the functionality of the modules. This shall involve upgrading the experimental environment and refining the module design to improve the ability of the robots to connect and float on the air table. Additionally, experiments should be performed which demonstrate the stop-start motion mechanism, along with the ability of the modules to harvest and share energy, and self-assemble into linear polymers. These represent the first steps to prepare the system for experiments into self-replication and evolution of morphology.

1.3 Contributions

The contributions of this thesis are as follows:

- A novel concept for a modular robot, which moves by routing fluid through itself. The concept, termed *Modular Fluidic Propulsion* (MFP), describes a robot which acts as a mobile fluid network. Each module of the robot contains pipes, actuators and a reservoir. Robots of arbitrary configuration can route fluid between connected modules, and between modules and the environment, thereby providing effective propulsion in any configuration. Two implementations of the concept are discussed. One, Modular Hydraulic Propulsion (MHP), is intended for use in a liquid environment. The actuators take the form of pumps, and route the environmental fluid which permeates the pipes and reservoirs of the robot. The second implementation, Modular Pneumatic Propulsion (MPP), works in a microgravity or low gravity environment. Propulsive, gaseous fuel is stored at pressure within the robot. The actuators, valves, can be opened to selectively release the fluid into the environment, propelling the robot.
- A novel decentralized occlusion-based motion controller for modular robots. The controller is adapted from a swarm robotics transportation controller in the literature [10]. Running on each module face independently, the controller checks for occlusion of a

goal point by the robot structure and activates actuators based on this. The controller uses binary sensors and actuators, and requires no communication, no arithmetic computation and no run-time memory.

- An analysis of the range of motion available to an MFP robot. 2-D MFP robots of size $l_1 \times l_2$ (in units of modules) can generate a total of $(2l_1 + 1)(2l_2 + 1)$ unique forces and $l_1^2 + l_2^2 + 1$ unique torques (for even l_1, l_2).
- The mathematical analysis of the decentralized MFP controller when applied to MHP robots moving slowly in a viscous fluid. Convex robots are guaranteed to reach the goal point successfully. Robots of generic morphology move towards the goal point up to a morphology-dependent minimum distance.
- The extension of the decentralized controller to produce two variants. The first allows communication between modules in order to improve energy efficiency. The second uses obstacle sensors to reduce collisions in environments with obstacles.
- The development of a simulator to study 3-D MFP robots. The simulator supports environments with or without obstacles, robots of cubic or randomly generated morphology, errors in actuator output direction and sensor accuracy, and the use of all variants of the decentralized controller as well as a centralized controller taken from the literature [20]. Simulations in an environment without obstacles show that:
 - The centralized controller is significantly faster than the decentralized controller.
 - Using the decentralized controller, random robots are not significantly slower than cubic robots.
 - Using any controller, random robots are significantly less energy efficient than cubic robots.
 - Enabling communication for the decentralized controller makes it significantly more energy efficient.
 - When actuator or sensor errors are introduced, robots with a higher number of modules suffer a smaller and more gradual loss of functionality.

Simulations in an environment with obstacles show that:

- Enabling object avoidance for the decentralized controller does not make it significantly faster or slower.
- Enabling object avoidance for the decentralized controller does significantly reduce the rate of collisions of the robot with the obstacles.
- The development of a 2-D hardware prototype of the MFP system as a proof-of-concept. The system was designed and built by two master’s students, Xinyu Xu and Yue Gu, under the co-supervision of the author. The prototype is equipped with fluid pumps, light sensors and connection sensors. Experiments are performed using the decentralized controller. The light sensors are used to detect a torch used as a goal. The prototype is consistently successful in reaching the goal.
- The investigation of a second modular robotic system, the evo-bots. The modules float on a air table, simulating a microgravity environment. The system uses a novel stop-start motion mechanism, with which robots can choose to move around semi-randomly due to external agitation, or stop in place, depending on the presence of external stimuli. The evo-bot experimental environment is prepared and the evo-bot module design is refined.
- The validation of the evo-bot system through experimentation. Experiments demonstrate the first use of the novel stop-start motion mechanism, the ability of the modules to harvest energy from external light sources via solar cells, the trophallactic transfer of energy between modules, and the ability of the modules to communicate with one another and self-assemble into linear polymers.

1.4 Publications

The research described within this thesis represents the original efforts of the author. Some of it has previously been published in the form of peer-reviewed papers. These papers are as follows.

1. **M. J. Doyle**, X. Xu, Y. Gu, F. Perez-Diaz, C. Parrott, and R. Groß. “Modular Hydraulic Propulsion: A robot that moves by routing fluid through itself”, in *Proceedings, 2016 IEEE International Conference on Robotics and Automation*, IEEE, 2016, pp. 5189-5196
2. J. A. Escalera, **M. J. Doyle**, F. Mondada, and R. Groß. “Evo-bots: A simple, stochastic approach to self-assembling artificial organisms”, *DARS 2016, Springer Proceedings in Advanced Robotics 6*, Springer (2018) 373-385
3. **M. J. Doyle**², J. V. A. Marques², I. Vandermeulen, C. Parrott, Y. Gu, X. Xu, A. Kolling and R. Groß. “Modular Fluidic Propulsion Robots”, *IEEE Transactions on Robotics* (*accepted pending revisions*)

Papers 1 and 3 provide content for both Chapter 3 and Chapter 4. Paper 2 forms the foundation of Chapter 5. Papers 1 and 2 were presented by the author at conferences in Stockholm, Sweden and London, UK respectively.

During his PhD studies the author also contributed to the following work, which does not feature in this thesis.

1. J. V. A. Marques, A. Özdemir, **M. J. Doyle**, D. Rus and R. Groß. “Decentralized Pose Control of Modular Reconfigurable Robots Operating in Liquid Environments”, in *Proceedings, 2019 IEEE International Conference on Intelligent Robots and Systems*, IEEE, *in press*

1.5 Outline

The thesis is organized as follows.

- Chapter 2 reviews the modular robotics literature. Section 2.1 inspects selected land based systems. Sections 2.2 and 2.3 look in detail at fluidic modular systems which are self-propelled and externally propelled respectively. Section 2.4 overviews the use of

²Both authors contributed equally to this work.

modular systems in self-replicating and evolving ecosystems. Section 2.5 summarizes the chapter.

- Chapter 3 introduces the novel MFP system. Section 3.1 introduces the concept, describing possible hydraulic and pneumatic implementations, detailing the sensing and actuation capabilities and developing a mathematical model. Section 3.2 analyzes the number of unique forces and torques that can be developed by 2-D MFP robots. Section 3.3 introduces a directed locomotion task for the MFP system and describes three variants of a decentralized controller to solve the task. Section 3.4 provides an analysis of the ability of the robot and controller to complete the task. Section 3.5 evaluates the assumptions and limitations of the model and analysis. Section 3.6 summarizes the chapter.
- Chapter 4 describes validation of the MFP system via simulation and experimentation. Section 4.1 introduces the simulator and describes the setup and results of simulations to investigate speed and energy consumption, obstacle avoidance, and resistance to errors. Section 4.2 describes the development of a set of hardware prototype modules, including the chassis and electronics. Section 4.3 describes the setup and results of experiments undertaken with the hardware prototype. Section 4.4 analyzes the effect of starting orientation on the time taken for a robot to reach the goal. Section 4.5 summarizes the chapter.
- Chapter 5 describes the refinement and validation of the evo-bot system. Section 5.1 describes the evo-bot concept. Section 5.2 details modifications made to the module design and experimental environment. Section 5.3 reports experiments in light detection and motion control, energy harvesting and sharing, and polymer formation. Section 5.4 summarizes the chapter.
- Chapter 6 concludes the thesis. Section 6.1 discusses its achievements and limitations. Section 6.2 suggests avenues for further research.

Chapter 2

Related work

This chapter reviews the current literature on modular systems. Systems which propel in fluids or microgravity are the focus. Land based systems are also reviewed to provide contrast. Both self-propelled and externally propelled systems are discussed. Additionally, systems which use (or could use) modular robots as a basis for self-replication or evolution of morphology are reviewed.

The chapter is laid out as follows. Section 2.1 overviews notable land based systems. Section 2.2 reviews self-propelled modular systems that operate in fluidic or microgravity environments. Section 2.3 reviews externally-propelled systems that operate in fluidic or microgravity environments. Section 2.4 investigates modular systems which self-replicate or evolve in morphology. Section 2.5 summarizes the chapter.

2.1 Land based systems

Although this thesis focuses on systems that operate in fluidic or microgravity environments, land based systems offer an interesting and varied point of comparison. Some systems prioritize movement capabilities, either as a connected structure or as individual units. Others prioritize a larger reconfiguration space or high scalability. In part due to the constraints of the environment most systems do not fully combine all three properties.

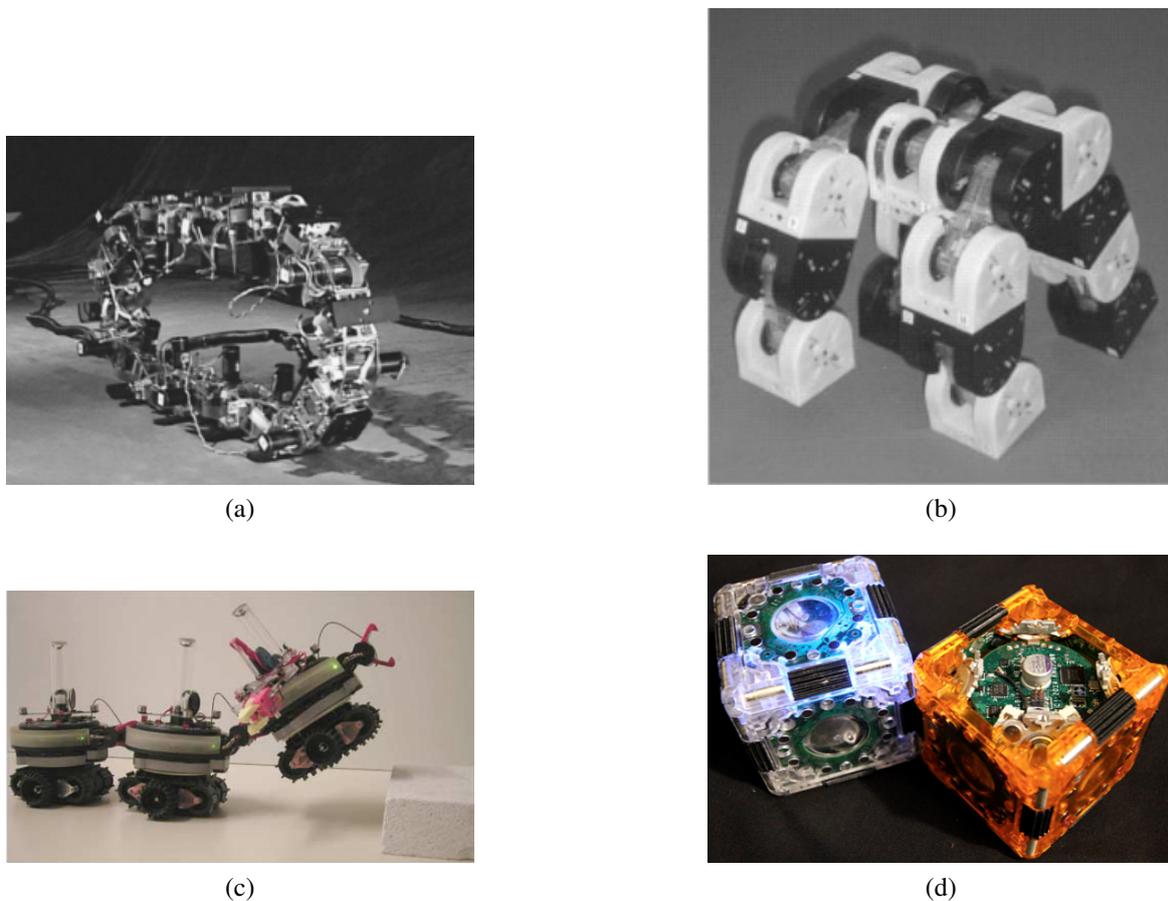


Figure 2.1 (a) PolyBot in rolling track configuration (©2002 IEEE). (b) M-TRAN III in a quadrupedal travel configuration (©2008 SAGE Publishing). (c) Three connected s-bots lifting one another to climb a step (©2006 IEEE). (d) Two M-Blocks modules (©2015 IEEE). Reprinted from [106], [55], [36], [77] respectively.

The modern field of modular robotics arguably began with the development of CEBOT [28] in 1988 (followed by CEBOT II [27]). CEBOT modules can both move independently and self-assemble in 2-D; initial work conceptualized individual modules moving into a container before self-assembling inside. Experimentally, alignment and docking of two modules was demonstrated.

The Polypod [103, 102] of Yim et al. is a chain type system that consists of two types of modules: segments and nodes. Nodes, which house the robot power supply, are passive, cubic modules with six connectors. Segments are actuated with two motors, and can both rotate and undergo compression or expansion. Robots are formed by connecting chains of

segments to nodes. One configuration, the rolling track [80], demonstrates a novel form of propulsion. This work was followed up by Polybot [106] (Figure 2.1a), which exchanges the ability to perform compression and expansion for a much larger range of rotational motion.

The lattice type M-TRAN [65, 53–55] is a highly influential system in the field. Each module has two parallel rotational degrees of freedom, and can form six connections. The latest version, M-TRAN III, uses a mechanical connection mechanism to self-reconfigure in a 3-D lattice. Various forms of travel have been demonstrated such as crawling and quadrupedal walking (Figure 2.1b).

ATRON[45, 68] is a lattice based system which can reconfigure in 3-D. Each module is a pair of hemispheres connected by a rotational joint. Despite the minimalist design the system is highly self-reconfigurable. Propulsion methods include both progressive motion via self-reconfiguration, as well as the use of the modules as wheels.

The swarm-bots [36] system (Figure 2.1c) comprises fully mobile, autonomous robots that can move around in 2-D. Self-propulsion is achieved through the use of both tracks and wheels, using differential drive. The wheels bestow the robots with accurate steering ability, while the tracks allow rough terrain to be traversed. Each module is equipped with a gripper arm which can be used to attach to nearby modules, forming self-reconfigurable structures. The gripper arm and traction system can be rotated with respect to one another. This allows connected structure to align their tracks and wheels and move around as a single unit.

Each Molecube [109] module is a cube split in half along its longest diagonal. The two halves of the cube are actuated by a single rotational degree of freedom, resulting in a deliberately simple design. The original system features only two active connectors per module, and therefore cannot form a lattice. An updated version of the system [107] replaces the two active connectors with six passive connectors, removing the ability to self-reconfigure. Instead, the updated system investigates locomotion, introducing specialized wheel modules among others [110]. A similar design, the Roombots [87], uses modules which consist of two diagonally split cubes connected together. This allows each module to have 10 connectors.

The MarXbot [3] is similar in function to the swarm-bots, being fully mobile and autonomous, and able to self-assemble using an attachment mechanism. Unlike the swarm-

bots, the attachment mechanism of the MarXbot has no dead zones. Additionally, the MarXbot employs an energy management and battery hotswap mechanism, which allows continuous experiments.

The SMORES system [13] is capable of acting like a lattice, chain or mobile modular robot. Each module has three rotatable connectors which double as wheels, allowing the modules to drive around independently on a flat surface. Additionally, one of the connectors can be tilted up to 90 degrees to aid with reconfiguration. The modules can form 3-D lattice or chain structures. Connected structures can move around in 2.5-D if provided with a base plate to which they can connect. Alternatively, in certain configurations the structure can drive around like a single module.

The M-Blocks [78] (Figure 2.1d) are cubic modules that can form lattice structures in 3-D. Modules can both travel individually and reconfigure by using a novel angular momentum based motion system. Flywheels inside the modules are spun up to store angular momentum and then braked in order to generate high amounts of torque and flip the module. Individually this allows a module to move around the environment. When connected to a structure, this allows the module to pivot out of one connection and into an adjacent one.

The HyMod [72] system can form 3-D cubic lattice structures. Each module uses four HiGen [71] connectors, which allow single-sided disconnect, and has three rotational degrees of freedom. This allow modules to rotate while in place in a connected structure, and move about individually in 2-D by using two of the rotatable connectors as differential drive wheels. Additionally, a connected structure as a whole can self-propel in certain configurations. This is aided by the use of heterogeneous extension modules, including mecanum wheels [16], which allow omni-directional motion.

Claytronics [33] (a form of programmable matter [32]) describes a system of catoms (claytronic atoms) which can self-organize into the shape of an object and take on its appearance. A prototype system of circular modules that can reconfigure in 2-D is produced. Each module can move around the edge of the connected structure using electromagnetic connectors. Karagozler et al. [46] produce a system of catoms with 1 mm modules. The modules can roll over the surface of one another to reconfigure, using electrostatic forces and

an external power supply. Piranda et al. [76] design a new type of catom which combines large contact surface, high volume filling and efficient movement between grid positions. The module, a faceted sphere with rounded surfaces, can additionally be fabricated from a flat shape. Collective actuation [7] shares similarities with the concept of claytronics. Circular robots with discrete actuators are arranged into octagons or hexagons to form cell units. Individual cells can expand and contract to produce forces. Multiple cells can be stacked in series or parallel to increase actuation distance or force respectively, and also allow flexible structures.

The Robot Pebbles [31] are cubic modules which boast an impressive side length of 1 cm while managing to incorporate a connection mechanism, power transfer and communication. The modules have no explicit propulsion system and instead self-reconfigure via self-disassembly.

Inout et al. [42] propose a set of pneumatic cellular modules. The system is not capable of net movement but can self-reconfigure. The modules are cubic and reconfigure using pneumatic bellows. Air is externally pumped into each module individually, although the authors propose a system for internally routing air between modules. The system can form a 2-D lattice, although only experiments with two modules are performed.

Thus far the review has focused mostly on hardware capabilities. Decentralized propulsion control is also relevant. Yim's gait control tables [102] specify at every time step what action each module must take in a given configuration. Although the control is centralized, it could potentially be made decentralized. Shen et al. [82] present a biologically inspired control method which relies only on local interactions between modules, in a similar manner to hormones. Experiments with CONRO [8] demonstrate "caterpillar" motion of chains of modules, legged gaits and rolling track motion. For each configuration the robot can adapt to changes in the modular configuration. The swarm-bots system uses evolved decentralized controllers [21] for both aggregation and coordinated movement of a connected structure. A distributed controller [55] for M-TRAN III is used to demonstrate various legged gaits, along with rolling and crawling.

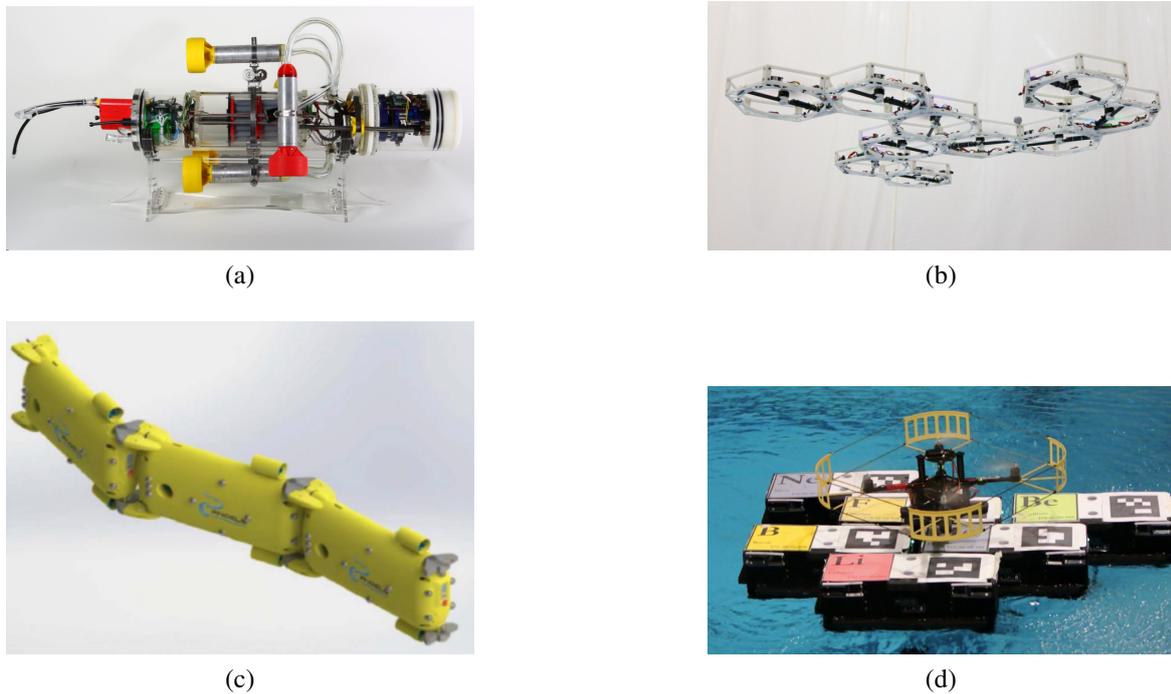


Figure 2.2 (a) AMOUR robot with four thrusters visible (©2010 SAGE Publications). (b) DFA with 10 modules (©2014 SAGE Publications). (c) Render of a three module ANGELS robot in chain configuration (©2013 IEEE). (d) TEMP robot with six modules acting as a landing platform for a quadrotor (©2014 IEEE). Reprinted from [93], [69], [90], [67] respectively.

2.2 Self-propelled fluid based systems

More recently modular systems have been developed to function in the air or water. Similarly, some systems have been designed for use in microgravity environments such as space. As discussed in Section 1.1 these systems can come closer to achieving effective propulsion, large reconfiguration space and high scalability. Drawbacks relate to the difficulties encountered in designing robots for fluidic environments. For example a robot operating in a liquid will require waterproofing. In addition fluid could effect sensing or communication. This section focuses on systems that are self-propelled.

Table 2.1 The capabilities of existing self-propelled fluid based modular robots. A hypothetical system based on the MFP concept is included for comparison.

System	Propulsion	Structure	Control	Limitations
AMOUR [94]	3-D	1-D lattice	centralized	
Neubots [96]	3-D	branching chain	centralized	
Hydron [51]	3-D	3-D lattice	decentralized	Lacks full orientation control and propulsion of connected structure
DFA [70]	3-D	2-D lattice	decentralized	
ANGELS [61]	3-D	linear chain	decentralized	
REMORA [29]	3-D	3-D lattice	centralized	Lacks explicit connection mechanism
Naldi et al. [66]	3-D	3-D lattice	centralized	Only horizontal connections demonstrated
TEMP [67]	2-D	2-D lattice	centralized	
Roboat [98]	2-D	2-D lattice	centralized	
ModQuad [79]	3-D	2-D lattice	decentralized	
GridDrones [4]	3-D	—	centralized	
MFP robot	3-D	3-D lattice	decentralized	

2.2.1 AMOUR

Vasilescu et al. [94] introduced the first version of AMOUR, an Autonomous Modular Optical Underwater Robot. The system is composed of cylindrical, stackable modules which can form both mobile robots and static undersea sensor networks. The modules are heterogeneous, with different types of modules performing different functions such as power supply, propulsion, buoyancy, computation and sensing.

The current incarnation of the system (Figure 2.2a) was presented by Vasilescu et al. [93]. The robot is designed as a single cylindrical hull with a set of reconfigurable thrusters. It is capable of hovering underwater, and can function in either a horizontal or vertical configuration. A buoyancy control mechanism is used to allow the robot to maintain balance even while transporting payloads of varying mass [15]. The payloads include sensor nodes for a static sensor network with which the robot can communicate. Doniec et al. [19] further present a communication system that allows the robot, while underwater, to communicate with a base station on land.

The centralized motion controller for AMOUR is described and tested by Doniec et al. [20]. The controller is designed for use with a variable number of thrusters with reconfigurable positions and orientations, and can handle non-holonomic configurations. Both rotation and translation can be controlled. The controller is expanded by Doniec et al. [18] to allow the automated estimation of a robot's thruster configuration. The robot activates its thrusters and measures the response using an IMU (Internal Measurement Unit) and pressure sensor. This allows the controller to build a model of the thruster configuration. The process takes around 40 seconds (for a robot with five thrusters) and the authors find that the resulting controllers match the performance of those that have been manually tuned.

2.2.2 The Neubots

Individual Neubot [96] modules are unable to self-propel, instead, connected groups of modules move in 3-D by generating bio-inspired swimming motions. The modules are faceted spheres with six connectors that can be configured into 3-D structures. Each connector

forms a rotational joint between connected modules. Control is provided by central pattern generators (CPGs) which, along with the robot morphology, are evolved in simulation to produce systems which can move efficiently.

2.2.3 Hydron

The Hydron system [51] is used for distributed self-assembly. Each Hydron module is a sphere which is suspended in water. Fluid is routed through a single module to produce movement in the horizontal plane—four nozzles on the sides of the module expel water which is drawn in by an impeller on the bottom. Vertical motion is controlled by the use of a syringe which draws or expels water into or out of the module to control buoyancy. A single module is not capable of rotating about the vertical axis, which can result in modules not being able to align with one another. Additionally, no explicit connection mechanism between the modules is proposed.

2.2.4 The Distributed Flight Array

The Distributed Flight Array (DFA) of Oung et al. [70] (Figure 2.2b) is a reconfigurable modular system consisting of hexagonal modules which can connect to form 2-D structures. The modules each have three powered omni-directional wheels which allow them to move around and connect with one another on the ground. Module to module connection is facilitated by a set of four permanent magnets on each side of the module. The magnets are strong enough to hold a connected structure together, while being weak enough that modules can disconnect by pulling away from one another. Each module also contains a rotor. The spin direction of the rotor depends on the module type. The rotors of CW modules spin in the clockwise direction. The rotors of CCW modules spin in the counterclockwise direction. Connected structures composed of at least two CW modules and two CCW modules can take flight. Oung et al. [69] demonstrate structures with up to 12 modules taking flight. The controller is scalable and decentralized; each module has full knowledge of the structure and no communication between modules is required.

2.2.5 ANGELS

Developed by Mintchev et al. [61], ANGELS (ANGuilliform robot with ELectric Sense) is a bio-inspired fish-like robot. A set of electrodes mounted on the chassis generate a dipolar electric field around the chassis. Disturbances in this field can be measured and used for navigation.

Individually, each robot uses a set of three propellers to control forward, backward, pitch and yaw motion. An adjustable buoyancy system controls vertical motion. Roll is not controlled explicitly. Robots can dock with one another, connecting nose to tail, to form chains (Figure 2.2c). An automated docking process is developed by Sutantyó et al. [90] and Mintchev et al. [60]. Connected chains have an additional mode of propulsion enabled to them. Each robot has a rotational joint that allows it to pivot with respect to its connected neighbours. This allows an undulatory swimming motion to be generated.

2.2.6 REMORA

The REMORA project [12] examines the use of aquatic modular UAVs for the inspection and maintenance of off-shore structures such as oil platforms. The UAVs can move and in principle reconfigure in 3-D. Furno et al. [29] produce a hardware prototype of the system and develop a reconfiguration strategy that minimizes energy consumption. Experiments using three UAVs demonstrate reconfiguration from an I-shaped configuration to an L-shaped configuration, although the UAVs do not physically dock with one another.

2.2.7 The work of Naldi et al.

Naldi et al. [66] study the dynamics and control of a proposed modular aerial robot. Each module consists of a ducted fan (a fan surrounded by a tube) with a set of actuated vanes which control the lift generated by the fan. Unlike other aerial systems, the design allows modules to connect vertically, in addition to horizontally, and still fly. This means that the system could be classed as one which can reconfigure and move in 3-D. Only manual

reconfiguration in the horizontal plane is demonstrated. In addition, only two prototype modules in a single configuration (horizontally connected) are tested.

2.2.8 TEMP

The Tactically Expendable Marine Platform, presented by O'Hara et al. [67] (and further developed by Paulos et al. [73]), is a self-reconfigurable modular system which floats on the surface of water. The cuboid modules can self-propel, either individually or as a connected structure, using a set of four thrusters mounted on the bottom corners of the hull. The modules can connect and disconnect to one another using a mechanical hook and loop connector. Originally envisioned to provide on-demand support in tactical situations, structures such as landing platforms (Figure 2.2d) and bridges can be formed. An assembly sequence planner was developed by Seo et al. [81]. This builds on the previous controller by allowing for structures which have internal holes.

2.2.9 Roboat

The Roboat [98] is a large ($0.9 \text{ m} \times 0.45 \text{ m}$) modular robot that floats on the surface of water. The self-reconfigurable system is proposed to be used to dynamically form 2-D structures and provide transportation across rivers or along the coast. The Roboat propels using four actuators that are arranged in a cross-shaped configuration that produces efficient motion. The robot uses a large suite of sensors including a GPS, IMU and 3-D laser scanner. Using this input the robot generates trajectories via nonlinear model predictive control (NMPC). The system was validated in both indoor and outdoor experiments. Docking experiments, in the presence of turbulent currents, are reported in [58].

2.2.10 ModQuad

ModQuad, developed by Saldaña et al. [79], is an aerial system with cubic modules that can form 2-D structures. Each module consists of a quadrotor platform with a cuboid frame mounted on it. On each vertical face of the cuboid, four permanent magnets are attached to

allow modules to connect to one another. Connections can be formed while the modules are airborne, although as of yet the modules do not have the ability to self-disconnect. Given that each module is a functional quadrotor by itself, modules can take flight individually, and there is no minimum number of modules necessary to form a structure capable of hovering. Experiments have been conducted with up to seven modules.

2.2.11 GridDrones

The GridDrones of Braley et al. [4] are an aerial implementation of the programmable matter concept. A set of cube-shaped nanocopters are used to form 2.5-D relief maps. Although the nanocopters are not physically connected to one another, they can be given topological relationships to their neighbours on the fly. This allows human operators to modify the map by physically interacting with it.

2.2.12 Analysis

Table 2.1 summarizes the most relevant properties of the systems discussed in this section, including a hypothetical MFP system. The problem of connected modular structures self-propelling in 2-D has been addressed by a number of solutions. The TEMP, Roboat, DFA and ModQuad systems all form 2-D lattices and can move about in 2-D or 3-D. Fewer systems have a 3-D lattice structure while being able to self-propel in 3-D. Of the systems that are capable of this, detail is often lacking about how the modules connect together, or how they can self-propel as a connected unit. The REMORA system proposes a connection mechanism but experiments are conducted only using individual robots. Additionally, centralized control and the use of an external localization system is required. The robots of Naldi et al. can theoretically form both vertical and horizontal connections, however only horizontal connections are demonstrated. The movement of connected structures of Hydron modules is not investigated, and additionally no hardware experiments are performed.

In light of this, the MFP concept could be well situated as a modular robot system with a reconfigurable 3-D lattice structure that can self-propel in 3-D as a connected unit.

2.3 Externally-propelled fluid based systems

This section reviews fluidic systems that are propelled by external means, such as fluid mixing, or by some internal mechanism which causes random motion. Also included are systems that float on air tables, thereby simulating microgravity. These systems are typically designed to investigate self-assembly or similar processes.

2.3.1 The work of Breivik et al.

Breivik [5] presents a thermomechanical system of two types of inert modules. The modules float on top of water, and use permanent magnets as connectors. Different connection sites use magnets of different Curie temperatures, so that by varying the temperature of the surrounding water, the formation and replication of different polymers can take place. Experiments with 70 modules performing formation and replication of strings are reported.

2.3.2 The work of Griffith et al.

Griffith [35] presents a set of square modules which float on an air table and can perform string replication. Each module has a lightweight plastic chassis with two-fold rotational symmetry. Connections can be formed on two sides, and are facilitated by electromagnetically actuated latches. Coils mounted on the modules sides allow connected modules to communicate. Griffith et al. [34] demonstrate the replication of strings of modules of different types, in a similar manner to that of DNA. Replication proceeds via the random accretion of modules by the parent string. If the module is of the correct type and in the correct position it will remain attached to the parent string and form part of the child string. Else, the module is detached from the parent string. Each module is controlled by a seven-state state machine which provides full control for the replication process—no module requires knowledge of the overall structure of the string.

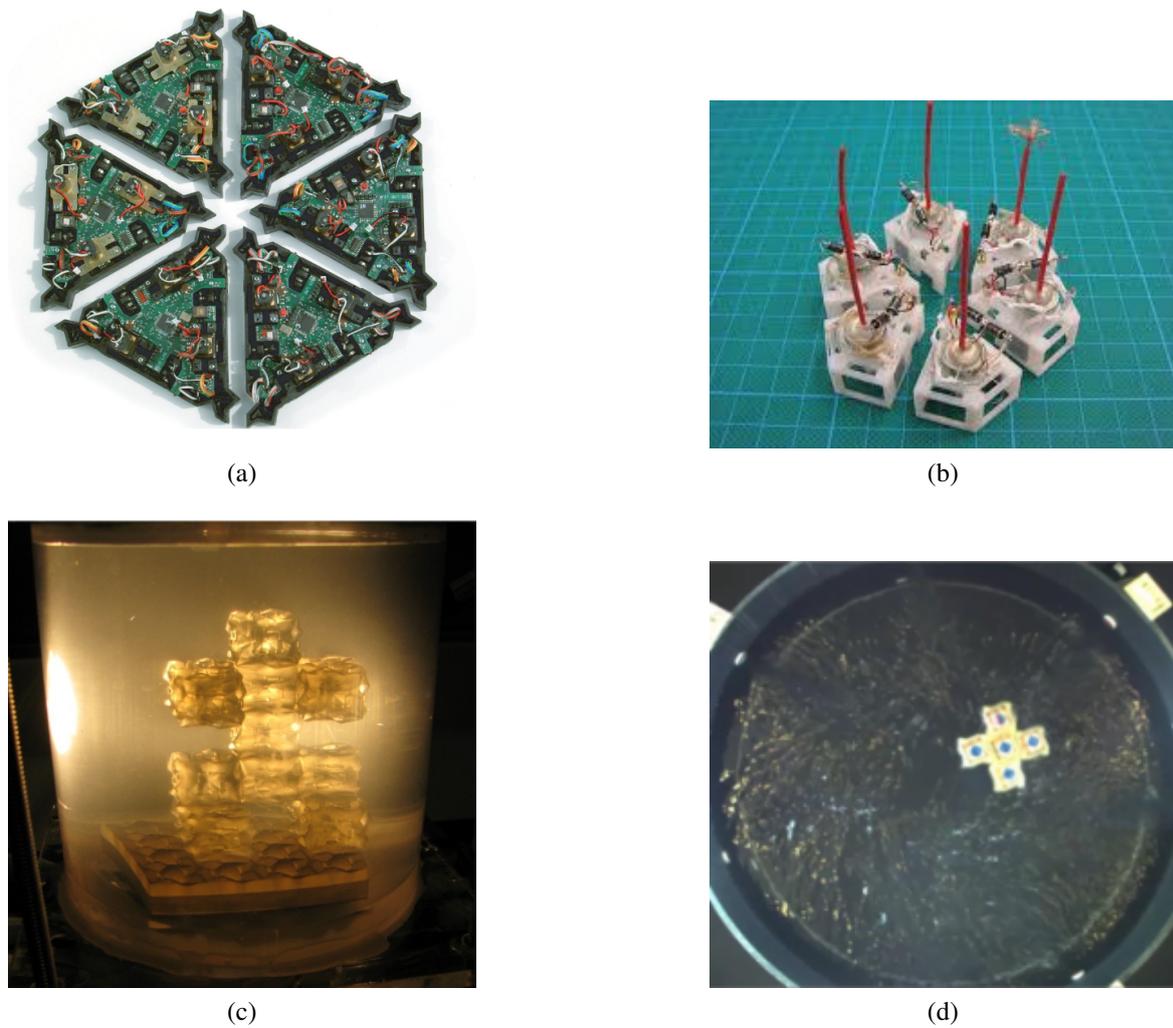


Figure 2.3 (a) Six programmable parts modules arranged into a hexagon (©2007 IEEE). (b) Six Tribolon modules (with freezing connectors) (©2008 IEEE). (c) 10 passive modules of Tolley et al. [92] self-assembled via fluid pumping (©2010 IEEE). (d) Five Lily modules self-assembled into a cross in their experimental environment (©2016 IEEE). Reprinted from [48], [62], [92], [39] respectively.

2.3.3 The works of White et al. and Tolley et al.

White et al. [100] present two similar modular systems which can assemble to form 2-D structures. One system has square modules that use electromagnets as a connection mechanism. The other has triangular modules that use rotatable permanent magnets to connect to one another. Both systems work on an air table mounted on an orbital shaker. The shaker oscillates to provide the modules with random motion. Power is supplied to connected modules from a single seed module, which also serves as a root from which structures grow.

A 3-D system is also presented by White et al. [101]. The modules, which float within an oil filled tank, are neutrally buoyant cubes of side length 10 cm. A substrate is fixed to the bottom of the tank. This substrate acts as a base from which structures can grow, and supplies power to attached modules. Each module face has a set of passive magnets along with an electromagnet which can be used to form connections to other modules or the substrate. Additionally, each face features an opening which leads through a short pipe to a reservoir in the center of the module. Each pipe contains a valve which can either prevent or allow the flow of water.

Tolley et al. [92] present a smaller (15 mm side length) 3-D cubic system (Figure 2.3c) which also floats within oil. The modules are completely passive and are propelled by agitation of the surrounding fluid. Structures are grown from a powered substrate on the base of the environment, which pumps fluid from the environment through the growing structure and itself. The resultant pressure differential caused by the pumping acts as a connection mechanism between modules.

2.3.4 Programmable parts

The programmable parts (Figure 2.3a), introduced by Bishop et al. [2], are a set of triangular modules which float on an air table. Side fans around the table provide the modules with random motion. Each module face has a fixed magnet and movable magnet pairing to allow modules to connect with one another. Additionally, each module face incorporates an infra-red transmitter and receiver which allow connected modules to communicate. Modules

follow a set of local rules, in the form of graph grammars (discussed further by Klavins et al. [49]). These local rules lead to global results—the formation of structures of certain morphologies. As modules move around their environment they randomly come into contact. If the modules successfully connect, they can begin to communicate. Modules can then check their internal states against the graph grammar rules, and decide whether they should stay connected, or disconnect. Klavins et al. [48] demonstrate experiments with up to nine modules.

2.3.5 Tribolon

The Tribolon robots, developed by Miyashita et al. [63] are triangular modules which float on water. Unlike other stochastic systems, propulsion is not provided via agitation of the water. Instead, a pager motor mounted on top of each module vibrates to induce semi-random motion. Power is supplied to the modules externally. Each module has a pair of electrodes. The anode, working as a pantograph, is mounted on top of the module and touches an aluminium ceiling. The cathode is attached to the bottom of the module and immersed in the water. A potential difference is applied between the water and the aluminium ceiling, powering the modules via the electrodes and an electrolyte mixed into the water. The modules use a single permanent magnet to connect with each other. Self-assembly is mediated by the shape of the modules and the interplay between the magnet strength and the size of the potential applied (and therefore frequency of vibrations). Miyashita et al. [62] present a more advanced version of the system (Figure 2.3b) which features a peltier based freezing connector. In later work [64] the investigation of triangle shaped robots is extended to rectangles and circles.

2.3.6 The work of Virgo et al.

Virgo et al. [95] present a set of physical self-replicating modules. The modules float randomly on an air table, and are completely inert. Interactions and connection formation between the modules are mediated only by the shape of the modules and the positioning

of permanent magnets on the module chassis. Two different module shapes are used. The modules can connect to form strings, and replicate in a similar manner to DNA. The modules are designed to prevent unnecessary or incorrect side reactions forming unwanted structures.

2.3.7 Lily

The Lily robots of Haghghat et al. [38] float on the surface of water. The environment is agitated by fluid pumps in order to produce semi-random motion of the modules. Each module is cubic, of approximately 35 mm side length. Modules can connect to one another on their four vertical faces, using electro-permanent magnets (EPMs), to form 2-D structures. The EPMs are also used for inductive communication between connected modules. In addition, each module has a light sensor and is capable of radio communication with a centralized base station. Similar to the programmable parts, Lily modules use graph grammars for structure formation. This is extended by Haghghat et al. [40] to the development of formal methods which can automatically generate the local rules for self-assembly. Experiments involving six modules forming line and cross shapes are demonstrated.

2.3.8 The work of Ganesan et al.

In simulation Ganesan et al. [30] investigate a system of stochastically moving cubic modules which self-assemble into 3-D structures in a fluidic environment. Structure formation is entirely decentralized. A rule based procedure is used to carry out self-assembly of different structures, and an analysis of the statistical dynamics of the system is provided.

2.4 Self-replicating and evolving modular ecosystems

In the 1940s von Neumann designed his “universal constructor” [97], a self-replicating machine in a cellular automata. Each constructor contains a tape, analogous to a genome, which carries instructions with which the constructor can build a copy of itself. The tape itself is replicated by a separate copy machine. While simpler methods of self-replication exist,

von Neumann's approach allowed the possibility of significant evolvability of the system. Each constructor is a highly complex machine. It took until 1995 before Pesavento et al. [75] produced the first implementation, which consisted of more than 150,000 cells in total. Since von Neumann came up with the universal constructor many self-replicating systems, both digital and physical have been developed.

Some systems focus on self-replication as an interest in and of itself. Jacobson [43] studied self-replicating sequences of track-bound modules. Penrose et al. developed a self-replicating mechanical system [74]. Zykov et al. [108] present a process by which a modular robot consisting of four modules can build a copy of itself. At the outset of the process, a robot composed of four Molecube [107] modules is placed on a base plate. The robot can move around on the base plate and pick and place modules from the environment to form a copy of itself. Although the robot acts autonomously, the spare modules have to be placed in the environment during the experiment by a human operator. This work is extended to investigate the replication of different structures in both hardware and simulation [109].

Other systems explore self-replication in the context of a self-propagating and potentially evolving ecosystems of artificial constructs. The remainder of this section highlights some of these systems.

2.4.1 The work of Chirikjian et al.

Chirikjian et al. [11] propose a self-replicating modular system for the development of infrastructure on the moon. The system consists of a set of modular robots, a solar powerplant, a casting plant, a processing plant and a rail gun. The system is seeded from a group of robots and a set of starting materials. The robots can both construct the other parts of the systems, and harvest material from the lunar regolith. Via the processing plant and the casting plant, new robots can be constructed. The rail gun can fire robots and resources across the moon to enable the construction of new sites.

Several hardware prototypes of the modular robot are demonstrated. One prototype consists of a remote controlled robot which can produce a copy of itself from seven base

parts. A second, semi-autonomous prototype uses a set of construction states to allow it to produce a copy of itself.

2.4.2 The REPLICATOR and SYMBRION projects

The REPLICATOR and SYMBRION projects [47] aim to investigate how individual mobile robots can physically cooperate as well as evolve in both function and morphology (although the work on evolution is largely hypothetical). An energy foraging task is used as an example. To access an energy source robots must navigate over a wall much taller than they are. In order to do this they must self assemble into a legged structure capable of climbing over the wall.

2.4.3 The work of Eiben et al.

Eiben et al. propose the EvoSphere [22], an ecosystem of robots which undergo open-ended evolution of morphology and controller. Robots are constructed by a centralized birth clinic. Newly created robots undergo a period of training and, if successful, are released into the main arena. The robots in the main arena must survive, complete tasks and ultimately reproduce. Reproduction requires the interaction of two or more robots which agree to share their genetic code. The resultant offspring genome is sent to the birth clinic to be constructed.

Weel et al. [99] propose a hardware implementation of the EvoSphere concept (although only simulations are conducted), using robots composed of Roombot [87] modules. However energy gathering is not explicitly simulated, and robots are selected based on their ability to propel themselves, rather than more implicitly on their ability to survive within their environment.

2.4.4 The work of Brodbeck et al.

Brodbeck et al. [6] develop a hardware system that automatically evolves and constructs successive generations of robots. A “mother robot” is used to construct the individual offspring robots according to their genomes. The mother robot can pick and place plastic

cubes to form the structure of an offspring, and glue them together with hot glue. Some cubes are inert, while others consist of two rotating parts actuated by a servo motor. Once construction is complete, the offspring robot is placed in a testing environment and activated.

The offspring is evaluated according to the distance it travels from its starting point over a set amount of time. Once all offspring in a given generation have been constructed and tested, a new generation is produced via mutation and crossover. In addition, the best performing offspring of the previous generation are transferred unchanged into the new generation. Most of the process is automated, however human interaction is required for the disassembly of offspring after testing.

2.4.5 Virtual creature evolution

Many systems exist which evolve both morphology and control of virtual creatures in simulation. Typically such systems stem from the seminal work of Sims [84, 83], who evolves virtual creatures to perform explicit tasks such as walking, jumping, swimming and light following. This line of study was extended by Lipson et al. [57] who evolve virtual creatures in simulation, and then construct physical versions of the most successful ones. Framsticks [50] uses organisms which are formed of jointed rods. Various senses such as touch, smell and equilibrium can be stimulated. The authors suggest that human design of hardware should be combined with evolutionary design of control. The Division Blocks [86] system constructs organisms out of blocks of varying size. Organisms gather energy (via photosynthesis or transfer between neighbours), create waste, reproduce and interact. The system is modelled with collisions and gravity, as well as a day-night cycle. Miconi et al. [59] use a spherical environment, designed to mimic a planet. Block based organisms traverse the surface of the planet and fight other organisms by swinging their appendages. Organisms are seen to adopt different strategies, such as minimizing their size to avoid encounters and take less damage, or maximizing their size to find and defeat more opposition.

2.5 Summary

This chapter reviewed the literature on reconfigurable modular robots, with a particular focus on propulsion mechanisms in fluids, and self-replicating ecosystems. The area of land based modular systems has seen significant interest and many different systems have been developed. Despite this, and despite the large theoretical potential for modular robots, there is limited evidence of systems being successfully and consistently used in the field.

As discussed in Section 1.1 fluidic environments may prove ideal for the deployment of modular systems, however comparatively few fluidic systems exist. In particular there is limited study of lattice based systems which self-propel in 3-D. Some systems move in 3-D but do not self-propel [100, 92]. Others self-propel in 3-D but are only capable of 1-D [93] or 2-D [70] reconfiguration [93]. Some lattice based systems self-propel in 2-D, but it is not immediately obvious as to how the design could be extended into 3-D [67, 98]. This thesis aims to begin to fill this gap by proposing a 3-D lattice based system that can potentially self-propel in 3-D by routing fluid through itself.

Physically implemented ecosystems of self-replicating, evolving modular robots are another focus. Existing systems are typically composed of complex elements that would be difficult or expensive to fully implement in hardware [11, 22]. Other work is highly abstract [86, 59]. Those systems that do have a hardware implementation [57, 6] typically have an explicit fitness function with a focus on completing a single type of task, rather than on surviving indefinitely in an environment. In light of this, this thesis also investigates the evo-bot system, the ultimate goal of which is to act as a physically embodied artificial ecosystem.

Chapter 3

Modular Fluidic Propulsion

Section 1.2 discussed three properties which can be combined to improve the utility of modular systems, namely *effective propulsion*, *large reconfiguration space* and *scalable design*. In particular the first two properties can be combined to produce a system that can self-propel in a wide variety of configurations, making the use of the system more flexible as well as cutting down on the time and energy needed to reconfigure to a specific travel form. High scalability not only allows robots to fit into tighter spaces, but combined with effective propulsion can also improve maneuverability.

Fluids or microgravity provide ideal environments for modular systems which embody the above three properties. Due to the characteristics of such environments, concerns such as the ability of the system to support its own weight are lessened.

In order to develop a fluidic system that embodies the above three characteristics this chapter introduces the *Modular Fluidic Propulsion* (MFP) concept for modular robot propulsion in fluids or microgravity. Connected MFP modules form an internal fluid network consisting of binary actuators, pipes and reservoirs. The actuators route fluid through the robot, thereby providing propulsion, making MFP robots the first modular systems which propel by routing fluid through themselves. MFP robots are intended to combine a scalable design and large configuration space with the ability to move effectively in any configuration.

The MFP concept is contextualized with a directed locomotion task. The task requires an MFP robot to move towards a goal point up to a minimum distance. In order to solve the

task, a decentralized controller is adapted from the literature [10]. Formerly the controller was used for a swarm of robots pushing an object towards a goal. Here, the object is the MFP robot itself.

The chapter begins in Section 3.1 by introducing the MFP concept in detail and providing a formal definition for an MFP robot. Furthermore, two specific implementations for use in liquid and microgravity environments respectively are described. Section 3.2 analyzes the propulsion capabilities of the concept and proves how many unique forces and torques a 2-D convex MFP robot can generate. Section 3.3 describes the directed locomotion task and presents the adapted controller solution, along with two variants. Section 3.4 analyzes the controller and proves that a 2-D robot of generic shape will reach a distance from the goal which is a function of the number of modules in the robot. In addition a convex robot will always reach the goal. Section 3.5 evaluates the assumptions and limitations of the theoretical work. Section 3.6 summarizes the content and findings of the chapter.

3.1 Concept

The MFP concept is illustrated in Figure 3.1. An MFP *robot* is composed of a set of homogeneous *modules*, forming a single connected entity. The modules are squares in 2-D or cubes in 3-D. With a slight abuse of terminology we use *face* to refer to both 3-D module faces and 2-D module sides. A module can be physically connected to other modules via each of its faces. By doing so, robots of different shapes can be built.

MFP robots propel by routing fluid through themselves. Their modules form a fluid network. Each module incorporates an internal fluid reservoir at its center. Routing pipes run from the reservoir to the center of each face. Each pipe includes an actuator which can be active or inactive. The routing pipes of corresponding faces of connected modules join to form a direct link between the internal reservoirs. We refer to the modules' faces that are not in face-to-face contact with another module as *external* module faces. All other faces are *internal* module faces. Active actuators connected to external module faces move fluid from the internal reservoir into the environment. Active actuators connected to internal module

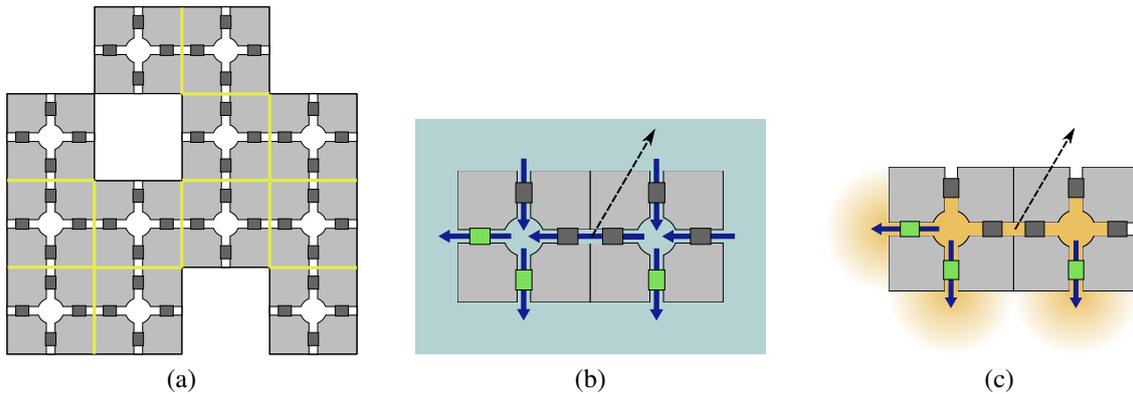


Figure 3.1 The Modular Fluidic Propulsion (MFP) concept and two implementation options: Modular Hydraulic Propulsion (MHP) and Modular Pneumatic Propulsion (MPP). (a) A generic MFP robot consisting of 12 modules. Each module has one actuator per face that it can use to route fluid. By routing the fluid, the robot propels. Yellow lines denote internal faces, all other faces are external. (b) A 2-D MHP robot. The environmental and routing liquid is indicated in blue. Green and dark grey boxes indicate active and inactive actuators, respectively. (c) A 2-D MPP robot. The gas propellant is indicated in yellow. In both (b) and (c) blue arrows indicate the direction of fluid flow through the robot and the dashed black arrow indicates the resultant direction of motion of the robot.

faces enable the movement of fluid from one internal reservoir into the adjacent one. Note that the control algorithms used throughout this thesis do not activate any actuators at internal faces. Inactive actuators allow fluid to pass through freely. By controlled routing in this manner, the robot can propel itself as a reaction to the induced fluid flow. Note that it is possible for the structure of an MFP robot to fully enclose a section of the environment. Figure 3.1a shows a 2-D example of this.

We assume that each actuator's state is binary—active or inactive. This means that not only can the actuators themselves be mechanically simple, but the need for complex control circuitry is reduced. This simplicity enables future miniaturization of the system, and could also allow a large number of modules to be produced. At the same time the system retains effective movement capabilities, and even a robot with a small number of modules has a wide range of motion available to it. A 3-D robot with 8 modules arranged in a $2 \times 2 \times 2$ cube would have 24 actuators on its external module faces. Without considering the actuation of internal thrusters, a total of $2^{24} \approx 10^7$ unique actuator firing configurations can be achievable.

A 3-D robot with 1000 modules arranged in a $10 \times 10 \times 10$ cube would have $2^{600} \approx 4 \times 10^{180}$ unique actuator firing configurations.

3.1.1 Modular Hydraulic Propulsion

Modular Hydraulic Propulsion (MHP) is a specific implementation of MFP that uses a liquid as the routing fluid. For this implementation, the robot requires a liquid environment, such as a lake or water pipe. The robot can either float on the surface of the liquid and move in 2-D, or move in 3-D beneath the surface.

The interior of the robot is permeated by the environmental liquid and all of the routing pipes and reservoirs are filled (shown in Figure 3.1b). The actuators take the form of pumps, which can displace liquid between modules, or between module and environment. Inactive pumps allow liquid to pass through in any direction. As liquid is pumped from the robot into the environment, liquid from the environment is drawn into the robot via the inactive pumps of other faces.

3.1.2 Modular Pneumatic Propulsion

An alternative implementation, Modular Pneumatic Propulsion (MPP), uses gas as the routing fluid. The gas is stored within the robot, at a higher pressure than the environmental medium. The actuators take the form of valves. When active, a valve can release gas from within the robot into the environment, providing propulsion. One advantage of this implementation over MHP is the ability for the system to work in a microgravity vacuum, such as in space. However, this comes at the cost of a limited supply of fuel. As the gas used for propulsion is stored inside the robot, and cannot be reclaimed, once it is exhausted the robot will not be able to control its own movement.

3.1.3 Sensing and communication

In addition to the above core characteristics of the MFP concept, we consider sensors which may be specific to the tasks we attempt to solve with the MFP system.

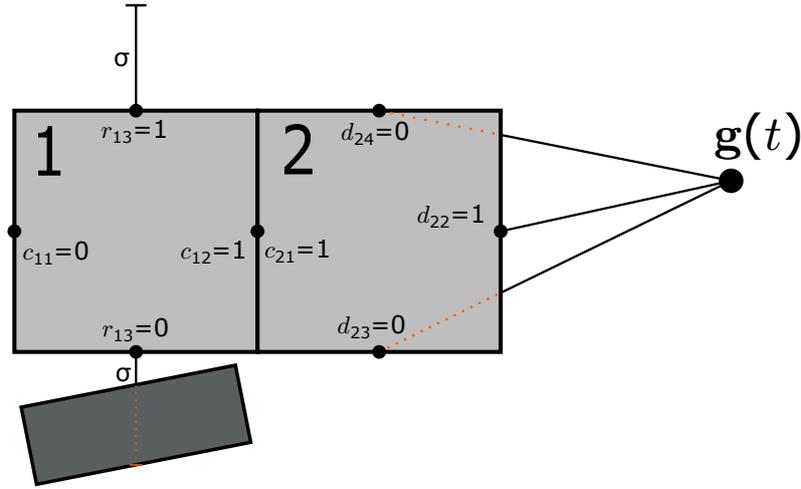


Figure 3.2 An example 2-D MFP robot consisting of two modules. The goal $\mathbf{g}(t)$ is indicated. An obstacle is shown in dark gray. Sensor/actuator nodes on each module are indicated by small circles. Selected sensors states are indicated. The c_{ij} refer to the connection sensor states, the d_{ij} to the goal sensor states and the r_{ij} to the obstacle sensor states.

- Each module has binary *connection* sensors, one per face. The connection sensors detect whether any neighboring modules are present, that is, physically attached.
- Each module has binary *goal* sensors, one per face, and mounted in the face center. The goal sensors detect whether a goal object is visible, that is, in the line of sight of the respective sensor. The range of these sensors is unlimited.
- Each module has binary *obstacle* sensors, one per face, and mounted in the face center. These sensors detect the presence of nearby obstacles. The range of these sensors is limited.
- Each module is capable of communicating with neighboring modules directly connected to each of its faces.

3.1.4 Mathematical modelling

Consider an unbounded fluidic environment $\mathcal{W}' = \mathbb{R}^N$, $N \in \{2, 3\}$. The environment contains an N -D MFP robot, occupying $\mathcal{R}'(t) \subset \mathcal{W}'$ at time t , a static goal point $\mathbf{g}' \in \mathcal{W}'$, and, optionally, a static obstacle region $\mathcal{O}' \subset \mathcal{W}'$. We use $'$ for variables described in the environment

reference frame. By default the analysis is conducted with respect to the reference frame of the robot, in which we have a static robot \mathcal{R} , and time-varying goal point and obstacle region, $\mathbf{g}(t)$ and $\mathcal{O}(t)$.

A point \mathbf{x}' , a velocity $\dot{\mathbf{x}}'$ and an acceleration $\ddot{\mathbf{x}}'$ transform between the two reference frames as:

$$\mathbf{x} = \mathbf{R}'^{-1}(\mathbf{x}' - \mathbf{q}') = \mathbf{R}'^T(\mathbf{x}' - \mathbf{q}') \quad (3.1)$$

$$\dot{\mathbf{x}} = \dot{\mathbf{R}}'^T(\mathbf{x}' - \mathbf{q}') + \mathbf{R}'^T(\dot{\mathbf{x}}' - \dot{\mathbf{q}}') \quad (3.2)$$

$$\ddot{\mathbf{x}} = \ddot{\mathbf{R}}'^T(\mathbf{x}' - \mathbf{q}') + 2\dot{\mathbf{R}}'^T(\dot{\mathbf{x}}' - \dot{\mathbf{q}}') + \mathbf{R}'^T(\ddot{\mathbf{x}}' - \ddot{\mathbf{q}}'), \quad (3.3)$$

where $\mathbf{q}' = \mathbf{q}'(t)$ and $\mathbf{R}' = \mathbf{R}'(t)$ correspond to the position and orientation of the robot in the environment reference frame at time t , and \mathbf{X}^T is the transpose of \mathbf{X} .

The robot consists of m modules, which are N -hypercubes of side length s and uniform density ρ . We use $i \in \{1, 2, \dots, m\}$ to index the modules. The position of module i is denoted by $\mathbf{v}_i \in \mathbb{R}^N$. The robot center of mass is given by $\mathbf{o} = \frac{1}{m} \sum_i \mathbf{v}_i$. We assume that the module positions are defined such that $\mathbf{o} = \mathbf{0}$.

We use $j \in \{1, 2, \dots, 2N\}$ to index the module faces, and $k \in \{1, 2, \dots, N\}$ to index the world dimensions. We assume that the module face normals are given by $\{\mathbf{u}_j\}_{j=1}^{2N} = (-\mathbf{e}_1, \mathbf{e}_1, \dots, -\mathbf{e}_N, \mathbf{e}_N)$, where \mathbf{e}_k is the k _{th} natural basis vector aligned along robot coordinate axis x_k . The position of the center of face j of module i is given by $\mathbf{h}_{ij} = \mathbf{v}_i + 0.5s\mathbf{u}_j$. Although the notation implies that all modules share a common orientation, this is not an assumption of the controllers to be investigated.

The modules form a connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{\mathbf{v}_i\}_{i=1}^m$, where $\mathcal{E} = \{(\mathbf{v}_i, \mathbf{v}_l) \subseteq \mathcal{V}^2 \mid s = \|\mathbf{v}_i - \mathbf{v}_l\|\}$ defines the connections among the modules.

We have $\mathcal{R} = \bigcup_i \mathcal{R}_i$, with $\mathcal{R}_i = \{\mathbf{x} \in \mathbb{R}^N \mid \max_k (\|v_{ik} - x_k\|) \leq 0.5s\}$. For a robot, \mathcal{R} , we define the *axis-aligned bounding box* as the set $\mathcal{B} = \mathcal{B}(\mathcal{R}) = [-l_1/2 + c_1, l_1/2 + c_1] \times \dots \times [-l_N/2 + c_N, l_N/2 + c_N] \subset \mathbb{R}^N$, where $\forall k : l_k = s + \max\{v_{ik} - v_{lk} \mid \mathbf{v}_i, \mathbf{v}_l \in \mathcal{V}\}$, and \mathbf{c} , the geometric center, is chosen to ensure $\mathcal{R} \subseteq \mathcal{B}(\mathcal{R})$.

The state of the robot actuators is represented by family $\{a_{ij}\}$, with $a_{ij} \in \{0, 1\}$. The actuator positioned at \mathbf{h}_{ij} fires along the associated normal vector \mathbf{u}_j .

Figure 3.2 shows examples of an MFP robot's sensor states. The state of the robot's *connection* sensors is represented by $\{c_{ij}\}$, with $c_{ij} \in \{0, 1\}$ given by

$$c_{ij} = \begin{cases} 0, & \mathcal{V} \cap \{\mathbf{v}_i + s\mathbf{u}_j\} = \emptyset; \\ 1, & \text{otherwise.} \end{cases} \quad (3.4)$$

The *goal* sensor detects the goal if not occluded by the body of the robot, that is, within the direct line of sight of the sensor. Figure 3.2 demonstrates the occlusion mechanism. The state of all goal sensors are represented by $\{d_{ij}\}$, with $d_{ij} \in \{0, 1\}$ given by

$$d_{ij} = \begin{cases} 1, & \mathcal{R} \cap \{\alpha\mathbf{h}_{ij} + (1 - \alpha)\mathbf{g}(t) | \alpha \in [0, 1]\} = \emptyset; \\ 0, & \text{otherwise.} \end{cases} \quad (3.5)$$

We assume that the obstacle region is transparent, in other words, the region does not occlude the goal. The robot *obstacle* sensor casts a ray from the module face center along the face normal up to distance σ and checks if this ray intersects with an obstacle. The state of all robot obstacle sensors are represented by $\{r_{ij}\}$, with $r_{ij} \in \{0, 1\}$ given by

$$r_{ij} = \begin{cases} 1, & (\mathcal{O}(t) \cup \mathcal{R}) \cap \{\mathbf{h}_{ij} + \alpha\sigma\mathbf{u}_j | \alpha \in (0, 1]\} = \emptyset; \\ 0, & \text{otherwise,} \end{cases} \quad (3.6)$$

where $\sigma \geq 0$ is the obstacle sensor range.

We assume that each external module face has access, via internal communication, to the goal sensor of a paired external module face (see Figure 3.3). The state of this paired sensor is

$$\hat{d}_{ij} = \begin{cases} d_{i\hat{j}}, & c_{ij} = 0; \\ 0, & \text{otherwise,} \end{cases} \quad (3.7)$$

where \hat{j} is chosen such that $\mathbf{u}_{\hat{j}} = -\mathbf{u}_j$ and $\hat{i} = \arg \min_{l \in \mathcal{X}_{i\hat{j}}} (\|\mathbf{v}_l - \mathbf{v}_i\|)$ with $\mathcal{X}_{i\hat{j}} = \{l \in \{1, 2, \dots, m\} | c_{l\hat{j}} = 0 \wedge \exists n \in \mathbb{N}_0 : (\mathbf{v}_l = \mathbf{v}_i + n\mathbf{u}_{\hat{j}})\}$. The effect of this is that with communication enabled, information can be passed between pairs of faces which have anti-parallel

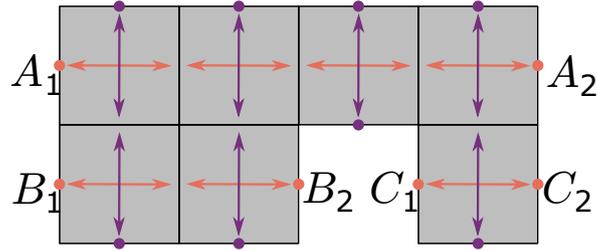


Figure 3.3 One of the proposed controllers allows modules to communicate. Illustration of the communication paths for a 2-D MFP robot. Communication between faces A_1 and A_2 takes place over the full width of the robot, as information can be passed from one module to the next. The row below this has a gap in the robot structure. Faces B_1 and B_2 can communicate as information is passed between the two modules. Faces C_1 and C_2 can communicate as they are part of the same module.

and colinear normals, and which lie on each end of an unbroken line of modules (shown in Figure 3.3).

3.2 Static 2-D force and torque analysis

Fundamental to an MFP robot is its ability to self-propel effectively. Existing systems which use analogue thrusters can produce forces and torques with a high resolution, and therefore move about with a large degree of accuracy. It is not immediately apparent that a system using only binary actuators could also produce high resolution torques and forces. The purpose of this section is to demonstrate the potentially high accuracy with which an MFP system can move, by deriving the number of unique forces and torques a 2-D convex MFP robot can produce. Due to the convexity, we have $\mathcal{B}(\mathcal{R}) = \mathcal{R}$. The robot is of rectangular shape, with dimensions l_1 and l_2 .

We make the following assumptions about the actuator force model:

- Only actuators on external module faces can be active ($c_{ij} = 1 \Rightarrow a_{ij} = 0$).
- Each active actuator $a_{ij} = 1$ applies a force of magnitude ξ and direction $-\mathbf{u}_j$ on the robot at position \mathbf{h}_{ij} .
- Forces due to fluid flow within the robot are negligible.

The total force caused by robot's actuators is given by

$$\mathbf{f} = - \sum_i \sum_j a_{ij} \xi \mathbf{u}_j. \quad (3.8)$$

For conciseness, and with a slight abuse of notation, we extend the 2-D vectors into 3-D. For example (x_1, x_2) becomes $(x_1, x_2, 0)$. We can then write the total torque acting on the robot as

$$\begin{aligned} \tau &= - \sum_i \sum_j a_{ij} \xi |\mathbf{h}_{ij} \times \mathbf{u}_j| \\ &= - \sum_i \sum_j a_{ij} \xi |\mathbf{v}_i \times \mathbf{u}_j|. \end{aligned} \quad (3.9)$$

Throughout this section for simplicity we set $\xi = 1$ and $s = 1$. Additionally, we assume we can freely choose the actuator states.

3.2.1 Number of possible forces

Theorem 1. The number of unique total forces that can be generated by a 2-D convex MFP robot is $(2l_1 + 1)(2l_2 + 1)$.

Proof. Figure 3.4 depicts the robot. From Equation 3.8, we obtain that the magnitude of the force applied along the x_1 axis is $f_1 = \sum_i (a_{i1} - a_{i2})$. The maximum of f_1 is realized if the l_2 external actuators with $j = 1$ are active, while all other actuators with $j \in \{1, 2\}$ are inactive (as shown in Figure 3.4). Therefore, the maximum of f_1 is l_2 . Any value $k \in \{1, 2, \dots, l_2\}$ can be realized by activating $k = 1, 2, \dots, l_2$ of the external actuators with $j = 1$, while all other actuators with $j \in \{1, 2\}$ are inactive. Similarly, one can obtain that the minimum of f_1 is $-l_2$, and that any value $k \in \{-l_2, -l_2 + 1, \dots, -1\}$ can be realized. Moreover, $f_1 = 0$ can be realized by deactivating all actuators with $j = 1, 2$. Every actuator activation, $\{a_{ij}\}$, realizes a value $k \in \{-l_2, -l_2 + 1, \dots, l_2\}$. Therefore the number of possible unique forces along the x_1 axis is $2l_2 + 1$. By symmetry, the number of unique forces along the x_2 axis is $2l_1 + 1$. As the horizontal and vertical force components are assumed to be independent, the

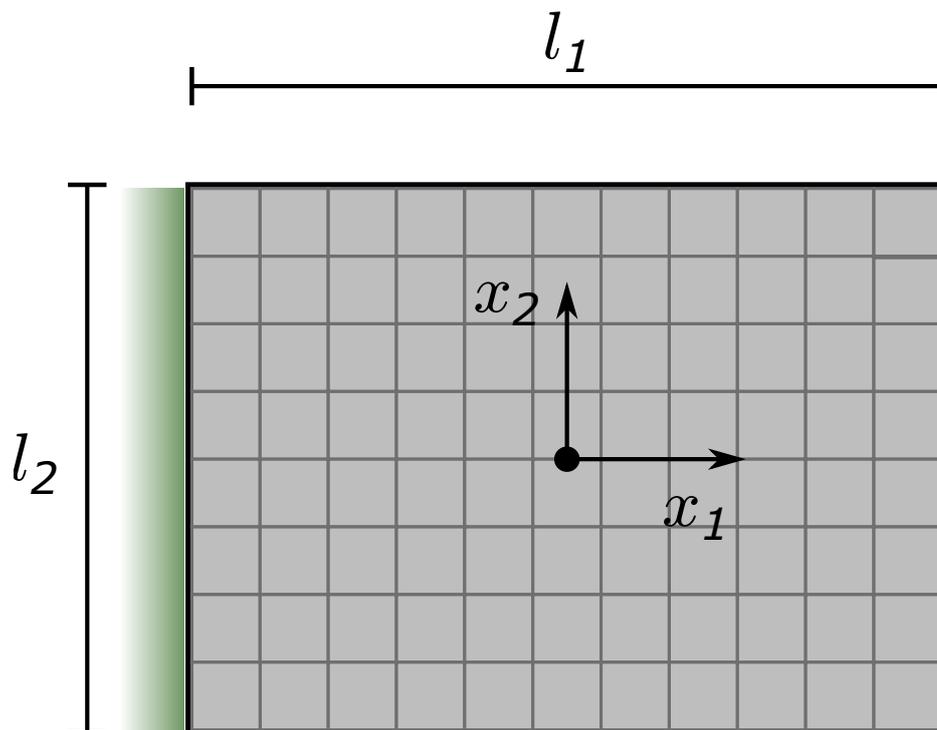


Figure 3.4 An l_1 by l_2 robot firing its actuators (shown in green) to achieve maximum force along the x_1 axis.

number of possible unique total forces that can be generated is

$$(2l_1 + 1)(2l_2 + 1). \quad (3.10)$$

□

3.2.2 Number of possible torques

Consider a 2-D convex robot. We first treat the case for which the robot has two odd side lengths, i.e, $l_1 \in 2\mathbb{Z}^+ - 1$ and $l_2 \in 2\mathbb{Z}^+ - 1$ (the other cases are presented in Appendix A).

Torque is maximized when

$$a_{ij} = \begin{cases} 1, & |\mathbf{v}_{ij} \times \mathbf{u}_j| < 0 \wedge c_{ij} = 0; \\ 0, & \text{otherwise.} \end{cases} \quad (3.11)$$

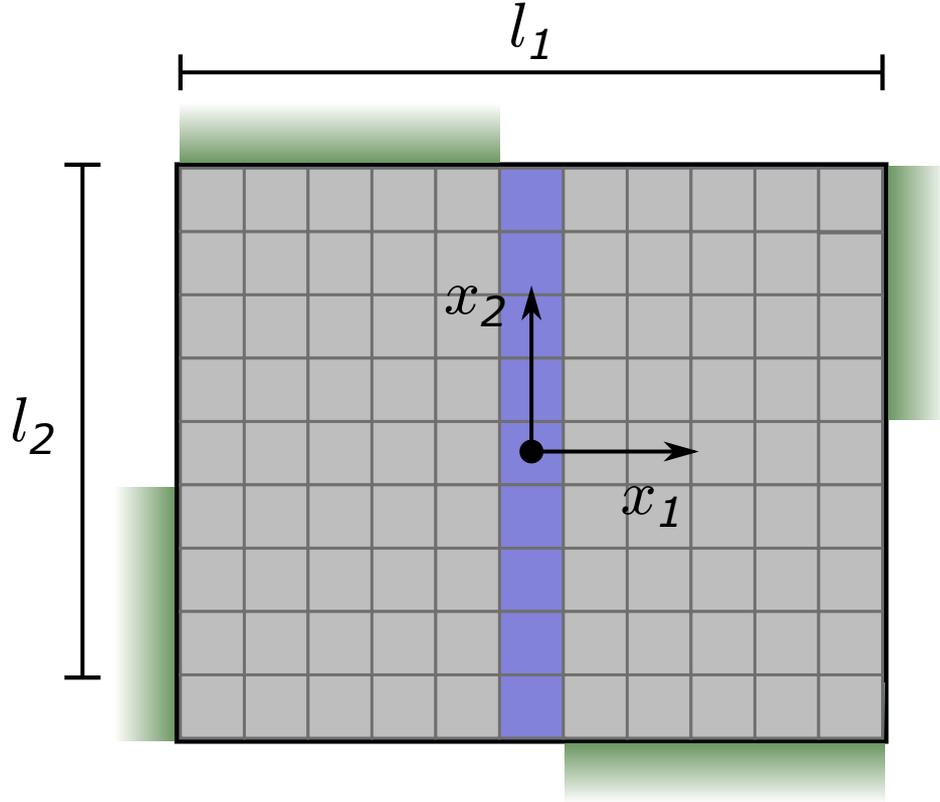


Figure 3.5 An l_1 by l_2 robot firing its actuators (shown in green) to achieve maximum torque. The blue modules highlight the $l_1 = 1$ case.

This actuator firing configuration is illustrated in Figure 3.5. From Equation 3.9 this gives a net torque of:

$$\begin{aligned}
 \tau_M(l_1, l_2) &= 2\left(1 + 2 + \dots + \frac{l_1 - 1}{2}\right) \\
 &\quad + 2\left(1 + 2 + \dots + \frac{l_2 - 1}{2}\right) \\
 &= \frac{(l_1 - 1)(l_1 + 1)}{4} + \frac{(l_2 - 1)(l_2 + 1)}{4} \\
 &= \frac{l_1^2 + l_2^2 - 2}{4}.
 \end{aligned} \tag{3.12}$$

Similarly, the minimum achievable torque is $-\tau_M(l_1, l_2)$.

Lemma 1. Every integer torque $\tau \in \{-\tau_M(1, l_2), \dots, \tau_M(1, l_2)\}$ can be produced for the case that $l_1 = 1$ and $l_2 \in 2\mathbb{Z}^+ - 1$.

Proof. We prove by induction that this holds.

Base case: Assume $l_2 = 1$. The robot consists of a single module; no torque can be obtained (Equation 3.9), which is consistent with $\tau_M(1, 1) = 0$.

Inductive step: We assume that the statement holds true for any $l_2 = 1, 3, \dots, k$, for some $k \in 2\mathbb{Z}^+ - 1$. Adding modules symmetrically about the center of mass does not change the positions of existing modules (and actuators). Therefore any torque that can be produced for $l_2 = k$ can still be produced when $l_2 = k + 2$. That is to say, all integer torque values $\tau \in \{-\tau_M(1, k), \dots, \tau_M(1, k)\}$ can be produced.

Now consider the actuator firing configuration which produces the maximum torque $\tau_M(1, k+2) = \frac{(k+2)^2-1}{4}$. This torque can be reduced by any $\Delta \in \{1, 2, \dots, \frac{k+1}{2}\}$ by deactivating a single actuator a_{ij} with $\mathbf{h}_{ij} \in \{(\frac{1}{2}, 1), (\frac{1}{2}, 2), \dots, (\frac{1}{2}, \frac{k+1}{2})\}$. In addition, by deactivating actuator a_{ij} with $\mathbf{h}_{ij} = (-\frac{1}{2}, -\frac{k+1}{2})$ the torque can be reduced by any $\Delta \in \{1 + \frac{k+1}{2}, 2 + \frac{k+1}{2}, \dots, k+1\}$.

Given that the difference between $\tau_M(1, k+2)$ and $\tau_M(1, k)$ is $k+1$, every integer torque $\tau \in \{\tau_M(1, k+2) - \tau_M(1, k), \dots, \tau_M(1, k+2)\}$ is possible. By symmetry, every integer torque $\tau \in \{-\tau_M(1, k+2), \dots, -\tau_M(1, k+2) + \tau_M(1, k)\}$ is possible. Therefore every integer torque $\tau \in \{-\tau_M(1, k+2), \dots, \tau_M(1, k+2)\}$ is possible and the inductive step is complete. \square

Theorem 2. The number of unique net torques that can be generated by a 2-D convex MFP robot with $l_1 = 1$ and $l_2 \in 2\mathbb{Z}^+ - 1$ is $N_\tau(1, l_2) = \frac{l_2^2+1}{2}$.

Proof. The maximum and minimum possible torque is $\tau_M(1, l_2)$ and $-\tau_M(1, l_2)$ respectively. From Lemma 1 every integer torque between these bounds can be produced. Given that each actuator can only apply either an integer torque, no other net torques can be produced. The number of unique torques is therefore

$$N_\tau(1, l_2) = 2\tau_M(1, l_2) + 1 = \frac{l_2^2 + 1}{2}. \quad (3.13)$$

\square

Lemma 2. Every integer torque $\tau \in \{-\tau_M(l_1, l_2), \dots, \tau_M(l_1, l_2)\}$ can be produced for the case $l_1 \in 2\mathbb{Z}^+ - 1$ and $l_2 \in 2\mathbb{Z}^+ - 1$.

Proof. We prove by induction that this holds.

Base case: Assume $l_1 = 1$. All torques $\tau \in \{-\tau_M(1, l_2), \dots, \tau_M(1, l_2)\}$ are possible as shown in Lemma 1.

Inductive step: We assume that the statement holds true for any $l_1 = 1, 3, \dots, k$ for some $k \in 2\mathbb{Z}^+ - 1$. Any torque that can be produced for $l_1 = k$ can still be produced when $l_1 = k + 2$. That is to say, all integer torque values $\tau \in \{-\tau_M(k, l_2), \dots, \tau_M(k, l_2)\}$ can be produced. Consider the actuator firing configuration which produces the maximum torque of $\tau_M(k + 2, l_2) = \frac{(k+2)^2 + l_2^2 - 2}{4}$. This torque can be reduced by any $\Delta \in \{1, 2, \dots, \frac{k+1}{2}\}$ by deactivating a single actuator a_{ij} with position $\mathbf{h}_{ij} \in \{(-1, \frac{l_2}{2}), (-2, \frac{l_2}{2}), \dots, (-\frac{k+1}{2}, \frac{l_2}{2})\}$. In addition, by deactivating actuator a_{ij} with $\mathbf{h}_{ij} = (\frac{k+1}{2}, -\frac{l_2}{2})$ the torque can be reduced by any $\Delta \in \{1 + \frac{k+1}{2}, 2 + \frac{k+1}{2}, \dots, k + 1\}$.

Given that the difference between $\tau_M(k + 2, l_2)$ and $\tau_M(k, l_2)$ is $k + 1$, every integer torque $\tau \in \{\tau_M(k + 2, l_2) - \tau_M(k, l_2), \dots, \tau_M(k + 2, l_2)\}$ is possible. By symmetry, every integer torque $\tau \in \{-\tau_M(k + 2, l_2), \dots, -\tau_M(k + 2, l_2) + \tau_M(k, l_2)\}$ is possible. Therefore every integer torque $\tau \in \{-\tau_M(k + 2, l_2), \dots, \tau_M(k + 2, l_2)\}$ is possible and the inductive step is complete. \square

Theorem 3. The number of unique net torques that can be generated by a 2-D convex MFP robot with $l_1 \in 2\mathbb{Z}^+ - 1$ and $l_2 \in 2\mathbb{Z}^+ - 1$ is $N_{\tau_1} = \frac{l_1^2 + l_2^2}{2}$.

Proof. The maximum and minimum possible torque is $\tau_M(l_1, l_2)$ and $-\tau_M(l_1, l_2)$ respectively. From Lemma 2 every integer torque between these bounds can be produced. Given that each actuator can only apply an integer torque, no other net torques can be produced. The number of unique torques is therefore

$$N_{\tau_1} = 2\tau_M(l_1, l_2) + 1 = \frac{l_1^2 + l_2^2}{2}. \quad (3.14)$$

\square

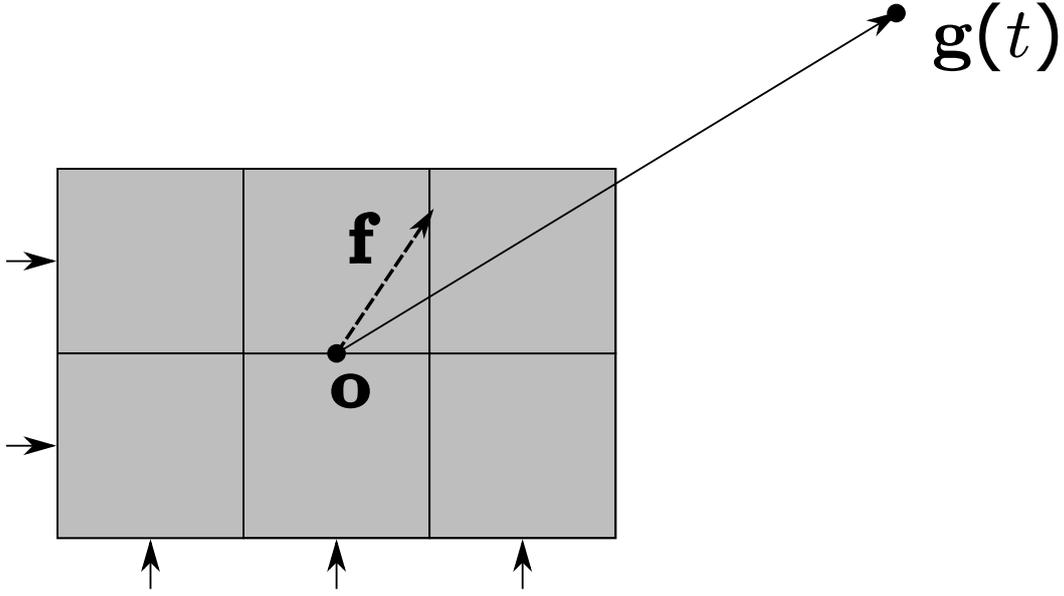


Figure 3.6 The occlusion-based motion controller *dec* demonstrated with a convex MHP robot. Both the robot center of mass \mathbf{o} and the target point $\mathbf{g}(t)$ are indicated. The small black arrows indicate the individual forces applied by occluded faces. The dashed arrow indicates the net force \mathbf{f} .

3.3 Directed locomotion task and controllers

To validate the MFP concept we consider a locomotion task which requires an MFP robot to move to the goal point. We introduce a set of decentralized controllers, adapted from the literature, to solve the task.

3.3.1 Task definition

The objective of the robot is to reach a distance of g_T , or less, from the goal $\mathbf{g}(t)$. Formally the objective is satisfied if and only if:

$$\exists T > 0 : \mathcal{B} \cap \{\mathbf{x} \in \mathbb{R}^N : |\mathbf{g}(T) - \mathbf{x}| \leq g_T\} \neq \emptyset. \quad (3.15)$$

Note that the robot's initial position is such that:

$$\mathcal{B} \cap \{\mathbf{x} \in \mathbb{R}^N : |\mathbf{g}(0) - \mathbf{x}| \leq g_T\} = \emptyset. \quad (3.16)$$

Table 3.1 Sensor to actuator mapping of the occlusion-based motion controller (dec)

c_{ij}	d_{ij}	a_{ij}
0	0	1
0	1	0
1	0	0
1	1	0

3.3.2 Occlusion-based motion controller (dec)

The occlusion-based motion controller, referred to as dec, is adapted from an occlusion-based cooperative transport controller, which was used for controlling swarms of mobile robots [9]. It is a decentralized controller; each module face executes an identical copy.

In keeping with the ethos of the MHP system, the adapted controller is of extreme simplicity. It requires no communication, no run-time memory, and no arithmetic computation. It maps the sensor state of a given face directly to the state of the corresponding actuator on that face (see Table 3.1). If face j of module i is connected to another module ($c_{ij} = 1$), the corresponding actuator is set as inactive ($a_{ij} = 0$). In this way all actuators on internal module faces are inactive. When face j is not connected to any module ($c_{ij} = 0$), the actuator is set as active ($a_{ij} = 1$) if the goal is not visible ($d_{ij} = 0$), otherwise, the actuator is set as inactive ($a_{ij} = 0$). Therefore all external module faces that are occluded from the goal have active actuators, and those that are not occluded have inactive actuators. The net result is that the robot moves towards the goal (Figure 3.6). This is proved to be correct for robots of general morphology under certain constraints in Section 3.4.

As mentioned earlier, it is possible for the structure of an MFP robot to enclose a section of the environment. For a 2-D robotic platform, such as the surface-water MHP robot (see Section 4.2), this presents no problem. For a 3-D robotic platform, however, the fluid would have nowhere to move other than into the robot. For 3-D pneumatic implementations, all valves surrounding the enclosed section would open, which is acceptable (although a waste of fuel). For 3-D hydraulic implementations, however, all pumps surrounding the enclosed section would fire while the fluid would not move, causing a large load.

Table 3.2 Sensor to actuator mapping of the occlusion-based motion controller with communication (dec-com)

c_{ij}	d_{ij}	\hat{d}_{ij}	a_{ij}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

3.3.3 Communication-enabled motion controller (dec-com)

An alternative controller variant, referred to as dec-com, allows for communication among the modules' faces. Formally, the controller that is being executed on face j of module i , has access to the goal sensor reading d_{ij} of a paired face as described in Section 3.1.4.

Table 3.2 describes the controller's mapping. As with the dec controller, when face j is connected to another module ($c_{ij} = 1$), the corresponding actuator is set as inactive ($a_{ij} = 0$). When face j is not connected to any module ($c_{ij} = 0$), the actuator is set as active ($a_{ij} = 1$) if the goal is visible only from the paired face ($\hat{d}_{ij} = 1$), but not from face j itself ($d_{ij} = 0$), otherwise, the actuator is set as inactive ($a_{ij} = 0$). This prevents faces on opposite sides of the robot firing at the same time and expending energy for no net gain in force or torque.

3.3.4 Obstacle avoidance motion controller (dec-obs)

The third controller variant, referred to as dec-obs, is designed for the case that the environment contains obstacles.

Table 3.3 describes the controller's mapping. As with the other two controllers, when face j is connected to another module ($c_{ij} = 1$), the corresponding actuator is set as inactive ($a_{ij} = 0$). When face j is not connected to any module ($c_{ij} = 0$), the actuator is set as active ($a_{ij} = 1$) if the goal is not visible ($d_{ij} = 0$) or if an obstacle is detected ($r_{ij} = 1$), otherwise, the actuator is set as inactive ($a_{ij} = 0$).

Table 3.3 Sensor to actuator mapping of occlusion-based motion controller with obstacle avoidance (dec-obs)

c_{ij}	d_{ij}	r_{ij}	a_{ij}
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

3.4 Controller analysis

In this section we analyze the use of the dec controller with 2-D MFP robots in an obstacle-free environment. We prove that a robot of arbitrary morphology always moves towards the goal point up to a certain morphology-dependent distance.

3.4.1 Preliminaries

Figure 3.7 provides an overview of the situation. As our analysis exhibits four-fold rotational symmetry, we only need to treat the case in which the goal point lies in the area defined by $x_1 > c_1 + l_1/2$, $x_2 \geq c_2 - l_2/2$. We consider this area as two separate regions, A ($x_2 > c_2 + l_2/2$) and B ($x_2 \leq c_2 + l_2/2$), as shown in Figure 3.7.

Lemma 3. For a given axis there are an equal number of external module faces with normal vectors which lie parallel to the axis and external module faces with normal vectors which lie anti-parallel to the axis.

Proof. Let $n_1^\epsilon = \sum_i (1 - c_{i1})$ and $n_2^\epsilon = \sum_i (1 - c_{i2})$ be the number of external module faces with normal vectors parallel and anti-parallel to the x_1 -axis, respectively. Define $n_1^l = \sum_i c_{i1}$ and $n_2^l = \sum_i c_{i2}$ as the number of internal module faces with normal vectors parallel and anti-parallel to the x_1 -axis, respectively. Let $n_1 = n_1^l + n_1^\epsilon$ and $n_2 = n_2^l + n_2^\epsilon$. Every module has two opposing faces parallel and anti-parallel to the x_1 -axis, giving $n_1 = n_2$ and therefore $n_1^l + n_1^\epsilon = n_2^l + n_2^\epsilon$. A connection between modules involves two opposing internal faces, and

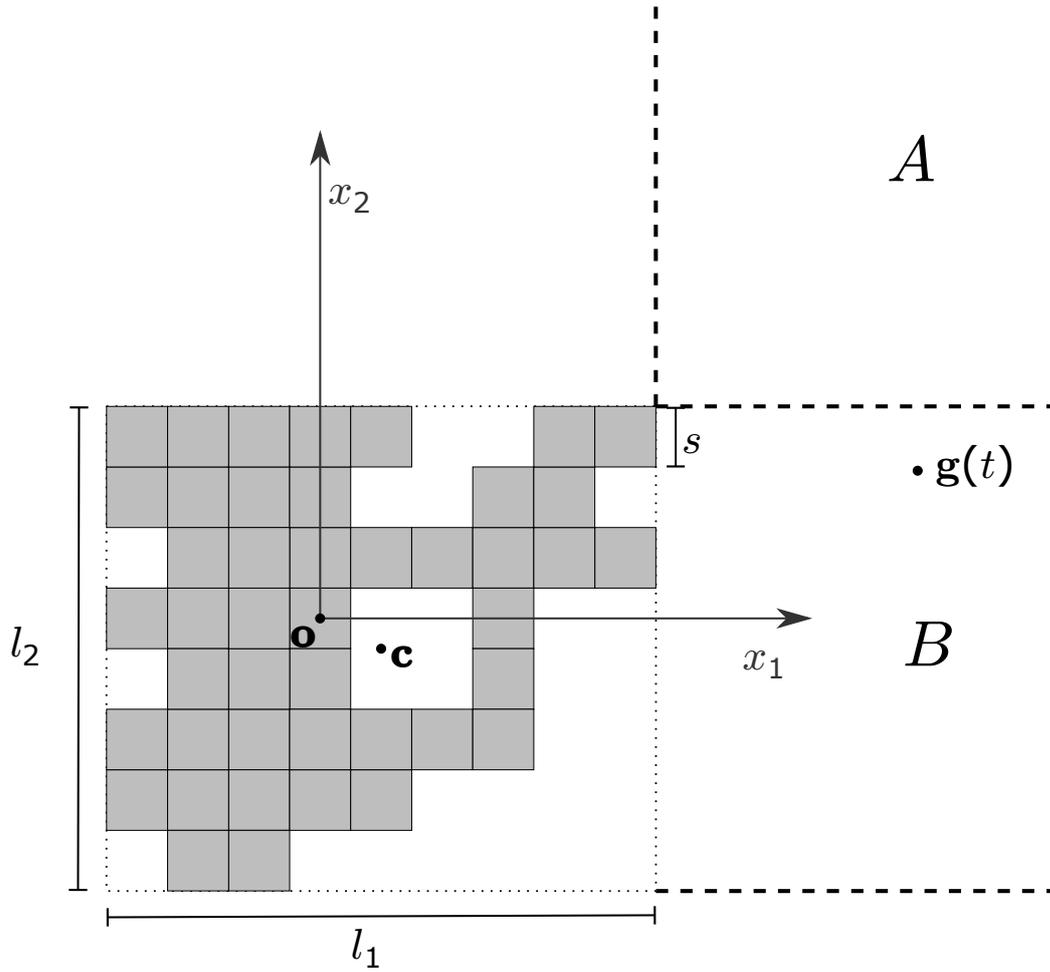


Figure 3.7 A 2-D MHP robot consisting of modules of side length s . The origin of the coordinate system, \mathbf{o} , is the robot center of mass. The Axis-Aligned Bounding Box of the robot, indicated with a dotted box, has size \mathbf{l} and geometrical center \mathbf{c} . The goal position $\mathbf{g}(t)$ is indicated. Analysis regions A and B are delineated with dashed lines.

only faces that are involved in connections are internal faces, therefore $n_1^l = n_2^l$. It follows that $n_1^\varepsilon = n_2^\varepsilon \equiv N_1$. The same argument holds for the x_2 -axis, with $n_3^\varepsilon = n_4^\varepsilon \equiv N_2$. \square

3.4.2 Robot dynamics

We describe how the robot moves in response to the total force acting upon it. We assume that the robot has a small thrust force and that the environment is highly viscous. We therefore neglect inertia and treat the robot motion as quasi-static. In effect, one can consider that, in the environment frame, the robot immediately reaches terminal velocity once actuator force

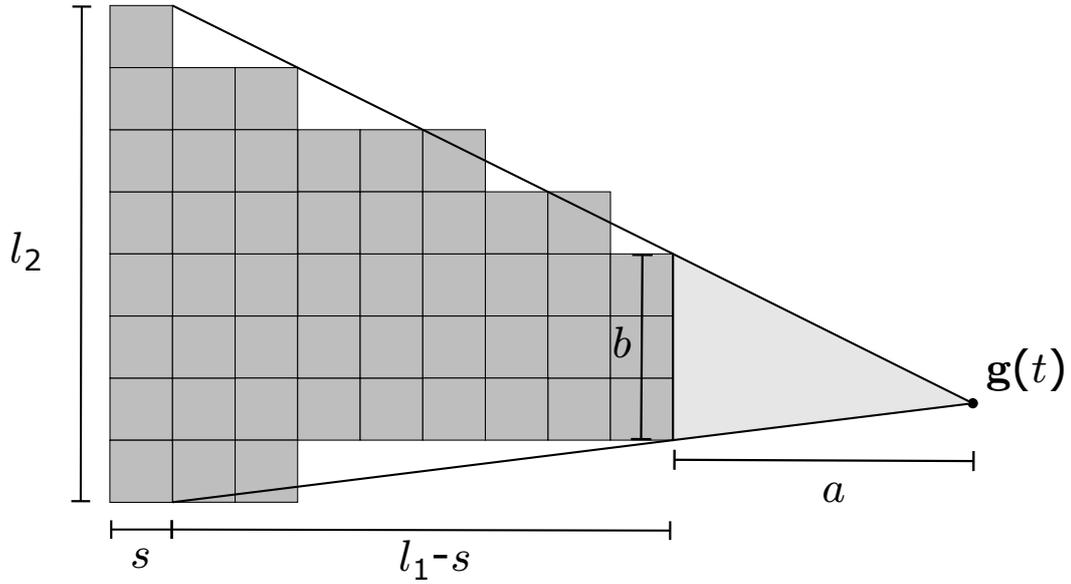


Figure 3.8 Casting rays from $\mathbf{g}(t)$ over the height of the bounding box to determine minimum force in the x_1 direction.

is applied, and immediately stops once actuator force is removed. In addition, drag can be treated as linear. We assume that the drag coefficient is independent of robot morphology or direction of motion. Formally:

$$\dot{\mathbf{q}}' = k_f \mathbf{f}', \quad (3.17)$$

where k_f is a positive constant and $\dot{\mathbf{q}}'$ and \mathbf{f}' are the robot velocity and actuator force in the environment reference frame.

Using Equation 3.3 and considering that $\mathbf{x}' = \mathbf{q}'$, $\dot{\mathbf{x}}' = \dot{\mathbf{q}}'$, $\ddot{\mathbf{q}}' = 0$, the actuator force transforms as:

$$\mathbf{f} = \mathbf{R}'^T \mathbf{f}'. \quad (3.18)$$

3.4.3 Occlusion model

Lemma 4. When the goal lies in region B the net actuator force along the x_1 -axis is limited by $f_1^B \geq \frac{l_2 a \xi}{s(l_1 - s + a)}$.

Proof. Consider rays cast from $\mathbf{g}(t)$ across the full height of the bounding box (Figure 3.8). In the worst case (lowest f_1 , most faces with $j = 1$ occluded), all of the rays are intercepted

by modules on the right hand side of the bounding box. In this minimum f_1 case, the force is:

$$\begin{aligned} f_1 &= \frac{b\xi}{s} \\ &= \frac{l_2 a \xi}{s(l_1 - s + a)}, \end{aligned} \quad (3.19)$$

where use has been made of the relationship, by similar triangles, $\frac{l_2}{b} = \frac{l_1 - s + a}{a}$. □

Lemma 5. When the goal lies in region A the net actuator force along each axis is limited by $f_1^A \geq \xi$ and $f_2^A \geq \xi$.

Proof. When the goal lies in region A, $c_{i1} = 0$ implies $d_{i1} = 0$, and hence $a_{i1} = 1$. Additionally, $c_{i3} = 0$ implies $d_{i3} = 0$, and hence $a_{i3} = 1$. Of those faces with $c_{i2} = 0$ there are $z_1 \geq 1$ with $d_{i2} = 1$, and hence $a_{i2} = 0$. This is because at least one module is placed flush with the “right” edge of the axis-aligned bounding box. Of those faces with $c_{i4} = 0$ there are $z_2 \geq 1$ with $d_{i4} = 1$, and hence $a_{i4} = 0$. This is because at least one module is placed flush with the “top” edge of the axis-aligned bounding box. The resulting forces are

$$\begin{aligned} f_1^A &= n_1^\varepsilon \xi - (n_2^\varepsilon - z_1) \xi \geq (n_1^\varepsilon - n_2^\varepsilon) \xi + \xi \\ f_2^A &= n_3^\varepsilon \xi - (n_4^\varepsilon - z_2) \xi \geq (n_3^\varepsilon - n_4^\varepsilon) \xi + \xi. \end{aligned} \quad (3.20)$$

From Lemma 3, we have $n_1^\varepsilon = n_2^\varepsilon$ and $n_3^\varepsilon = n_4^\varepsilon$, giving the limits:

$$\begin{aligned} f_1^A &\geq \xi \\ f_2^A &\geq \xi. \end{aligned} \quad (3.21)$$

□

Lemma 6. When the goal lies in region B the net actuator force along the x_2 -axis is limited by $|f_2| \leq m\xi$.

Proof. The net actuator force is limited by the total number of modules (m) in the robot. At most, one actuator per module can make a net contribution to f_2 . Therefore:

$$|f_2| \leq m\xi. \quad (3.22)$$

□

3.4.4 Distance condition

Theorem 4. The robot moves towards the goal point $\mathbf{g}(t)$ while $\max(|g_1|, |g_2|) > 4ms$.

Proof. Consider $g(t) = |\mathbf{g}(t)|$, the Euclidean distance of the goal point $\mathbf{g}(t)$ from the robot center of mass \mathbf{o} . Whenever $g(t)$ is decreasing, that is $\dot{g}(t) < 0$, the robot is approaching the goal point.

$$\begin{aligned} \dot{g}(t) &= \frac{d}{dt} \sqrt{g_1^2 + g_2^2} \\ &= \frac{\dot{g}_1 g_1 + \dot{g}_2 g_2}{\sqrt{g_1^2 + g_2^2}}. \end{aligned} \quad (3.23)$$

From Equations 3.1, 3.2, and using $\dot{\mathbf{g}}' = \mathbf{0}$, $\dot{\mathbf{g}}$ is given by

$$\begin{aligned} \dot{\mathbf{g}} &= \dot{\mathbf{R}}'^T (\mathbf{g}' - \mathbf{q}') - \mathbf{R}'^T \dot{\mathbf{q}}' \\ &= \dot{\mathbf{R}}'^T \mathbf{R}' \mathbf{g} - \mathbf{R}'^T \dot{\mathbf{q}}'. \end{aligned} \quad (3.24)$$

In 2-D, $\mathbf{R}' = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$, where θ is the rotation angle of the robot in the environment reference frame, giving

$$\begin{aligned} \dot{\mathbf{g}} &= \dot{\theta} \begin{pmatrix} -\sin \theta & \cos \theta \\ -\cos \theta & -\sin \theta \end{pmatrix} \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} g_1 \\ g_2 \end{pmatrix} \\ &\quad - \mathbf{R}'^T \dot{\mathbf{q}}' \\ &= \dot{\theta} \begin{pmatrix} g_2 \\ -g_1 \end{pmatrix} - \mathbf{R}'^T \dot{\mathbf{q}}'. \end{aligned} \quad (3.25)$$

Using Equations 3.17 and 3.18 gives:

$$\begin{aligned} \dot{\mathbf{g}} &= \dot{\theta} \begin{pmatrix} g_2 \\ -g_1 \end{pmatrix} - k_f \mathbf{R}'^T \dot{\mathbf{f}}' \\ &= \dot{\theta} \begin{pmatrix} g_2 \\ -g_1 \end{pmatrix} - k_f \dot{\mathbf{f}} \\ &= \begin{pmatrix} \dot{\theta} g_2 - k_f f_1 \\ -\dot{\theta} g_1 - k_f f_2 \end{pmatrix}. \end{aligned} \quad (3.26)$$

Therefore Equation 3.23 becomes

$$\dot{g}(t) = \frac{-f_1 k_f g_1 - f_2 k_f g_2}{\sqrt{g_1^2 + g_2^2}}. \quad (3.27)$$

The robot approaches the goal point while

$$f_1 g_1 + f_2 g_2 > 0. \quad (3.28)$$

First consider the goal point lying in region A. From Lemma 5 $f_1^{AB} > 0$ and $f_2^A > 0$ therefore Equation 3.28 holds.

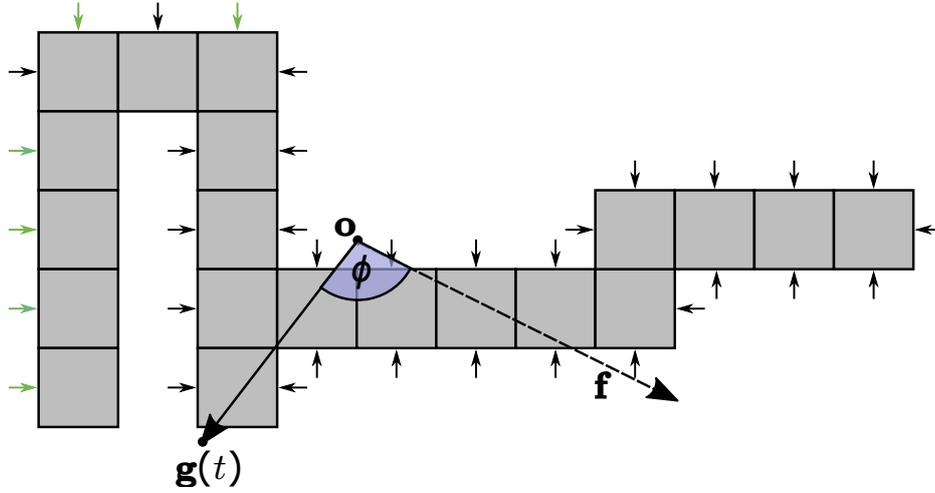


Figure 3.9 A 2-D MFP robot composed of 20 modules. The robot center of mass \mathbf{o} and the goal point $\mathbf{g}(t)$ are indicated. The small arrows indicate activated actuators. Green arrows indicate the actuators which make a net contribution to the total force acting on the robot. The angle ϕ between the total force vector \mathbf{F} and the vector $\mathbf{g} - \mathbf{o}$ is greater than $\pi/2$, hence Equation 3.28 does not hold. Note that while this means the robot does not move towards the goal point at this instant, it does not imply that the robot will never reach the goal. In this example, once the force is applied to the robot and the goal position moves, a different set of actuators will be activated which do provide $\phi \leq \pi/2$.

Now consider region B. Equation 3.28 does not hold generally (see Figure 3.9 for a counter example). Instead, we rearrange it to produce a condition on g_1 :

$$g_1 > -\frac{f_2 g_2}{f_1}. \quad (3.29)$$

By considering that $|g_2| < l_2$, $|f_2^B| \leq m\xi$ (Lemma 6) and $f_1^B \geq \frac{l_2 a \xi}{s(l_1 - s + a)}$ (Lemma 4):

$$\begin{aligned} g_1 &> \frac{ms(l_1 - s + a)}{a} \\ &> ms\left(\frac{l_1 - s}{a} + 1\right). \end{aligned} \quad (3.30)$$

If we assume that $a \geq l_1$ then:

$$g_1 > 2ms. \quad (3.31)$$

It is guaranteed that the assumption $a \geq l_1$ will hold if:

$$g_1 > 2ms + 2l_1. \quad (3.32)$$

Considering that $l_1 \leq ms$ gives:

$$g_1 > 4ms. \quad (3.33)$$

The symmetry of the problem gives the corresponding conditions:

$$\begin{aligned} g_1 &< -4ms \\ g_2 &> 4ms \\ g_2 &< -4ms. \end{aligned} \quad (3.34)$$

Concatenating the four conditions gives:

$$\max(|g_1|, |g_2|) > 4ms. \quad (3.35)$$

The robot moves towards the goal point while Equation 3.35 holds. \square

Corollary 1. The robot successfully completes the task if the required minimum distance to the goal point satisfies $g_T > 4\sqrt{2}ms$.

Proof. Condition 3.35 describes a square boundary around the robot of side length $8ms$, centered on $o = \mathbf{0}$. For any $\varepsilon > 0$, there exists a time at which the goal point is either within this boundary, or at most ε away from some point of it. At this time the goal point lies either within the smallest circle which encloses this boundary, or at most ε away from some point of the circle. The radius of this circle is given by $4\sqrt{2}ms$. As \mathbf{o} must be within \mathcal{B} , if \mathbf{g} is within $4\sqrt{2}ms + \varepsilon$ of \mathbf{o} , then there must exist at least one point within \mathcal{B} that is within $4\sqrt{2}ms + \varepsilon$ of \mathbf{g} . As ε can be chosen to be arbitrarily small, the task is completed successfully if $g_T > 4\sqrt{2}ms$. \square

Corollary 2. A convex robot, such that $\mathcal{B}(\mathcal{R}) = \mathcal{R}$, moves towards the goal point $\mathbf{g}(t)$ until $\mathbf{g}(t) \in \mathcal{B}$.

Proof. For a convex robot the limits on the force along each axis when the goal lies in region B are given by $f_1^B \geq \xi$ and $f_2^B = 0$. The limits in region A as given by Lemma 5 continue to apply. Given that both components of \mathbf{f} are always non-negative, Condition 3.28 always holds and the robot will approach the target point until $\mathbf{g}(t) \in \mathcal{B}$. Note that once this condition is achieved the goal point is within the robot, and none of the external module faces will be able to detect it. All the actuators will activate and the robot will not move. \square

3.5 Assumptions and limitations

Throughout this chapter several assumptions were made about the MFP system and its environment. Here, those assumptions are discussed and analyzed in more detail. In addition, limitations of the system in its current form are considered.

Section 3.1.4 introduces the presence of a goal point in the robot's environment and assumes that it is always visible to at least one of the robot's sensors. The case in which there is no goal in the environment, or the goal is obscured from or out of range of all of the robot's sensors, has not yet been explicitly considered. In this situation, when using the dec controller, all actuators on the robot will fire, expending potentially a large amount of energy for no net movement. The use of the dec-com controller overcomes part of this problem, by ensuring that no actuators will fire and no energy will be expended. In order to search for an undetected goal a new controller would be necessary. This could take the form of a random walk generator, or a controller which moves the robot in a fixed direction until a goal is discovered.

Section 3.2 describes the forces caused by the active actuators of an MFP robot. It is assumed that each active external actuator imparts a constant force on the robot. Additionally, fluid routing within the robot is not considered. In a real system the flow rates (and therefore forces) through the external ports would be a function of the fluid network topology, as well as the state of all actuators (including internal ones). Appendix B presents a simple model of

fluid flow within an MFP robot, that provides flow rates throughout the robot as a function of the activation state of all actuators (both internal and external). The next step from this model would be to develop a link between the flow rates and the resultant forces acting on the robot.

Section 3.4.2 introduces the dynamics model for the MFP robot. The robot motion is assumed to be quasi-static — the velocity of the robot proportional to the force applied. However this is only a realistic assumption in situations where the robot is moving slowly in a viscous environment. For systems such as those described in Chapter 4 this assumption does not hold strongly and so the model would be assumed to fit poorly. A more complex treatment would require the formulation of a function which returns the activation state of all actuators given the relative position of the goal, and the piecewise solution of the equations of motion.

Although the analysis in this chapter concerns 2-D systems the MFP concept also covers 3-D implementations. For such implementations which move around in a fluid (i.e. not a vacuum), control of buoyancy must be considered. Without explicit buoyancy control, an MFP robot would naturally rest at the depth at which it is neutrally buoyant. To occupy another position along the vertical axis would require active control of buoyancy. This could take the form of permanent activation of actuators, which would not require any hardware adaptations from the existing concept but would be inefficient. An alternative would be the use of “swim bladders”, actuated compartments that can be increased or decreased in size to change the amount of water displaced by the system, and therefore its resting depth. Each module could be fitted with such a device, or, to save space, specialist modules could be used that include a swim bladder instead of fluid actuators.

The existing dec controller implicitly controls depth as it directs the robot towards a goal which may lie on a different horizontal plane. Moving vertically could be less efficient than moving in the horizontal plane as the robot moves away from its natural resting depth. Furthermore, reaching a very deep (or very shallow) goal may be impossible if the robot has a shallow (or deep) natural resting depth and insufficiently strong actuators. Explicit control over depth could be achieved if the robot included pressure sensors. This could be achieved in a distributed manner if each module included a pressure sensor and swim bladder.

Alternatively, a small number of specialized modules containing both a pressure sensor and a swim bladder could be used to control the depth of the entire robot.

3.6 Summary

This chapter presented the MFP concept for a robot that moves by routing fluid through itself. The aims of the concept are to provide a good reconfiguration space, a high level of scalability and effective propulsion in any configuration. The concept uses simple binary actuators which engender scalability and future miniaturization. The actuator layout and fluid routing gives the system both a lattice structure and the ability to move as a connected unit. The concept could be applied to either fluidic or microgravity environments.

Despite using binary actuators a wide range of motions are available to an MFP robot. The number of unique forces that can be generated by a 2-D convex MFP robot of size $l_1 \times l_2$ is $(2l_1 + 1)(2l_2 + 1)$. The number of unique torques that can be generated by the same robot (assuming l_1 and l_2 are odd) is $\frac{l_1^2 + l_2^2}{2}$. An 11×11 robot for example can produce more than 400 unique forces and 100 unique torques.

A decentralized controller is adapted from the literature to solve a directed locomotion task. The controller requires neither run-time memory nor arithmetic computation, and uses binary inputs and outputs. This further adds to the scalability of the system. Proposed variants of the controller allow obstacle avoidance and communication between modules.

Under certain assumptions the controller is guaranteed to move a convex MFP robot to the goal. Furthermore, the controller is proved to move a generic 2-D MFP robot with m modules each of side length s to within a distance of $4\sqrt{2}ms$ from the goal.

Chapter 4

Validating the MFP concept

Chapter 3 introduced the MFP concept and proved that under certain conditions a generic 2-D MHP robot will move towards a goal point up until a certain minimum distance. However it was not proved that the robot would move all the way up to the goal to contact it. Additionally, the proof relies on some fairly strict assumptions. The robot must be small and moving slowly through a viscous fluid, and subject to a constant drag force regardless of morphology or direction of motion.

In order to investigate the system in more varied environments and under different conditions a 3-D simulator is developed. This chapter firstly describes the simulator, and the simulations conducted. These include evaluating the success rate and speed of all of the decentralized controllers, as well as the energy efficiency and obstacle collision rate of the dec-com and dec-obs controllers respectively. The decentralized controllers are also compared to a centralized controller taken from the literature [20]. Furthermore the resistance of the system to errors is tested.

Although simulations help to analyze the MFP system the reality gap [44] that inevitably exists between simulation and reality means that the results may not fully represent a real world system. To address this a 2-D hardware prototype of the MHP system is developed. Experiments are performed to measure the speed and success rate of the system. Feedback from the experiments is used to tune the simulator to the hardware, and further simulations of a 2-D system are performed.

This chapter is laid out as follows. Section 4.1 introduces the 3-D simulator and describes the investigations undertaken and their results. Section 4.2 presents a set of prototype hardware modules developed by two master's students, Yue Gu and Xinyu Xu, under the supervision of the thesis author. Section 4.3 reports on experiments conducted with the prototype system. Section 4.4 analyzes the effect of starting orientation on the time taken for a robot to reach the goal. Section 4.5 summarizes the findings of the chapter.

4.1 Simulations

In this section we study MFP systems using physics simulations. In Sections 4.1.2, 4.1.3 and 4.1.4 we study the ability of MFP robots to move towards the goal, save energy, and avoid obstacles. These simulations provide a basic validation of the hardware and controllers, and investigate whether our simulated robots can successfully carry out their task and if the controller variants improve performance. In particular, we seek to answer the technical challenge identified in Section 1.2, of whether MFP robots can successfully propel to the goal independent of their configuration. These simulations also serve to benchmark the performance of our controllers against a state-of-the-art centralized controller. The simulated systems can move in 3-D through fluidic environments under more realistic conditions than assumed in Chapter 3.

4.1.1 General setup

The simulator is based on the Open Dynamics Engine (ODE) [85], an open source 3-D physics library. Modules are modelled as solid cubes of density 1000 kgm^{-3} (the density of water) with a side length of 8 cm. Each module face incorporates an actuator, goal sensor and obstacle sensor at its center. Unless otherwise stated active actuators apply a force of 6.4 mN to the center of the module face, antiparallel to the face normal. The net resultant force and torque acting on a robot is integrated by ODE. As none of the controllers activates actuators of internal module faces, only the actuators of external module faces are considered. The routing of fluid throughout the robot is not simulated. Each active actuator expends

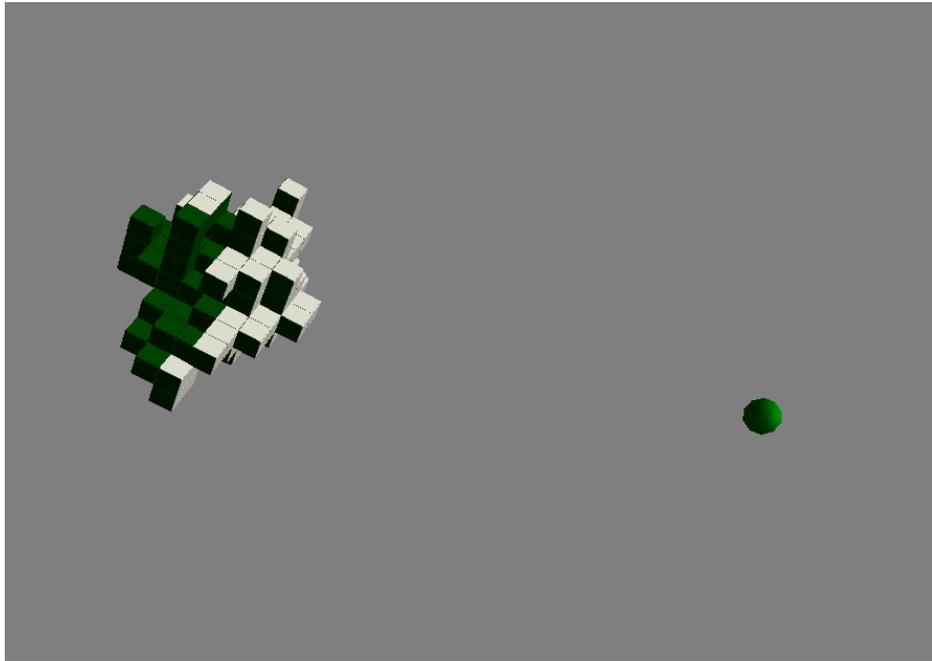


Figure 4.1 An image taken from the simulator showing a randomly generated robot moving towards the target position on the right (green sphere). Green and white module faces represent active and inactive actuators respectively.

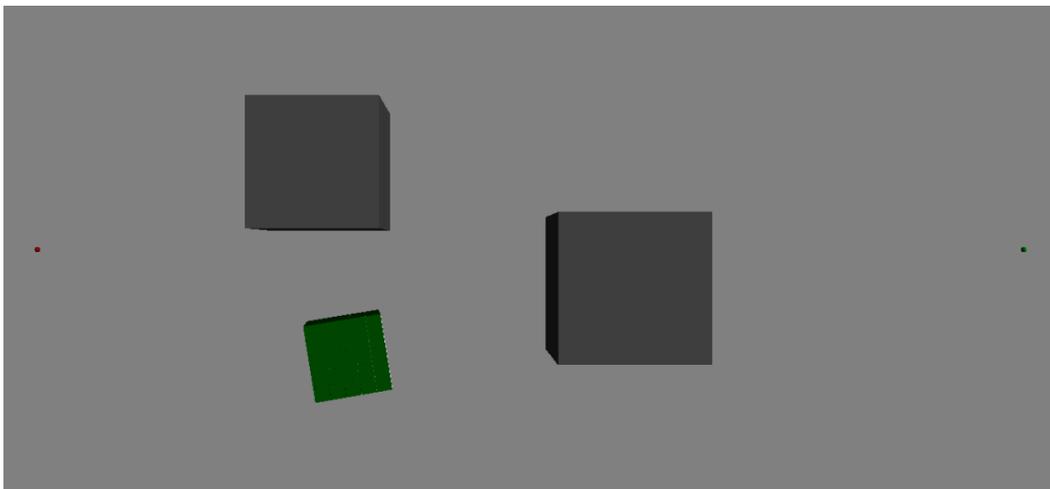


Figure 4.2 An image taken from the simulator showing a 5^3 cubic robot in the slalom environment. The robot is using the dec-obs controller and moving from its starting position on the left (red point) to the target position on the right (green point). The grey cubes are the slalom obstacles.

one simulated energy unit (seu) per time step (0.1 s). Inactive actuators expend no energy. Additionally, the energy demands of the processing and sensing are negligible.

The simulation environment (shown in Figures 4.1 and 4.2) is continuous and unbounded. It is filled with a fluid of constant density 1000 kgm^{-3} (meaning the robots are neutrally buoyant). It contains a goal point and a single robot. The latter starts at a distance of 500 cm from the goal point, with a random orientation (using [1]). The origin of the world coordinate system lies at the starting point of the robot. The x_1 axis lies in the direction of the goal point. The *open* environment contains nothing else. The *slalom* environment additionally contains a pair of obstacles situated between the target point and the robot. Each obstacle is cubic with a side length of 50 cm. The obstacles are fixed in place—they cannot be moved by the robots. The positions of each obstacle are (125, 0, 0) cm and (375, 0, 0) cm, plus a random shift of up to 25 cm along each axis. The trial is deemed successful if the robot reaches the goal point within 250 s. If the robot does not reach the target within 250 s then the trial is classified as unsuccessful.

A simplified model of fluid drag is used in order to avoid a full, computationally expensive, fluid dynamics treatment. The drag force on each external module face is calculated individually without reference to the overall shape of the robot. The drag force is assumed to follow the quadratic drag equation for turbulent flow. This means the drag force for each external module face is:

$$\mathbf{f}_D = \begin{cases} -\frac{1}{2}\rho C_D s^2 \dot{\mathbf{q}} \cdot \mathbf{u} \dot{\mathbf{q}}, & \dot{\mathbf{q}} \cdot \mathbf{u} > 0 \\ 0, & \text{otherwise,} \end{cases} \quad (4.1)$$

where \mathbf{f}_D is the drag force acting on the face, ρ is the density of the surrounding fluid, $C_D = 0.8$ is the drag coefficient, s is the module side length, $\dot{\mathbf{q}}$ is the velocity of the face center and \mathbf{u} is the face normal.

The module's goal sensors use the ray casting functionality of ODE to check line of sight between the module face center and the goal point. Line of sight can be occluded by the body of the robot itself.

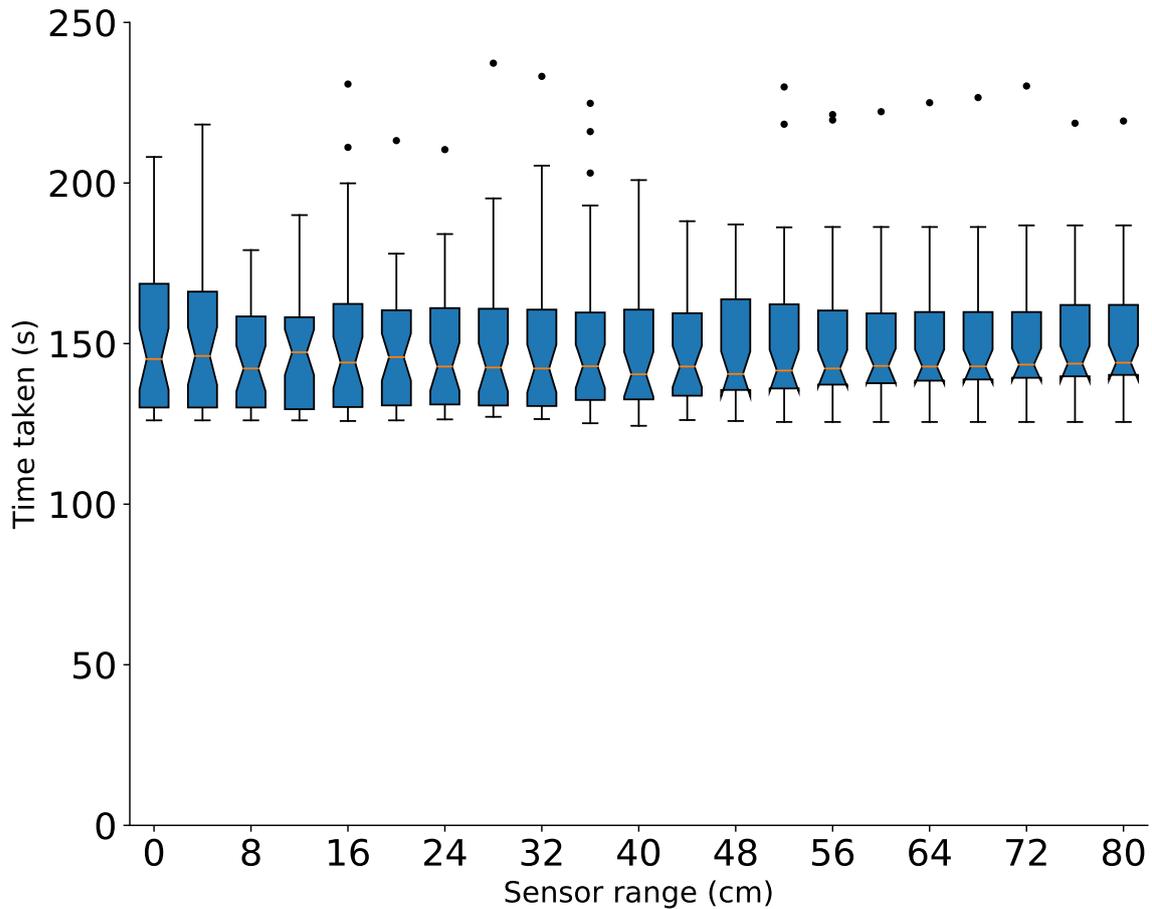


Figure 4.3 The distribution of times taken in the slalom environment using the dec-obs controller with different sensor lengths (σ). Each boxplot is composed of 40 values.

When using controller dec-obs, the obstacle sensors can be used. Similar to the target sensors the obstacle sensors also use ray casting. A ray of length σ is cast from the center of the face, parallel to the face normal. If the ray intersects with an obstacle, the sensor returns `false`. Else, it returns `true`. To calibrate the ray length 20 different lengths were tested in the slalom environment, from 4 cm to 80 cm. The time taken and number of collisions for each are shown in Figure 4.3 and Figure 4.4 respectively. Each ray length was tested over a set of 40 trials, with each set using the same set of random starting orientations and random obstacle positions. Each range had a 100% success rate. A range of $\sigma = 40$ cm was chosen due to it having the best combination of low median collision percentage and low maximum collision percentage.

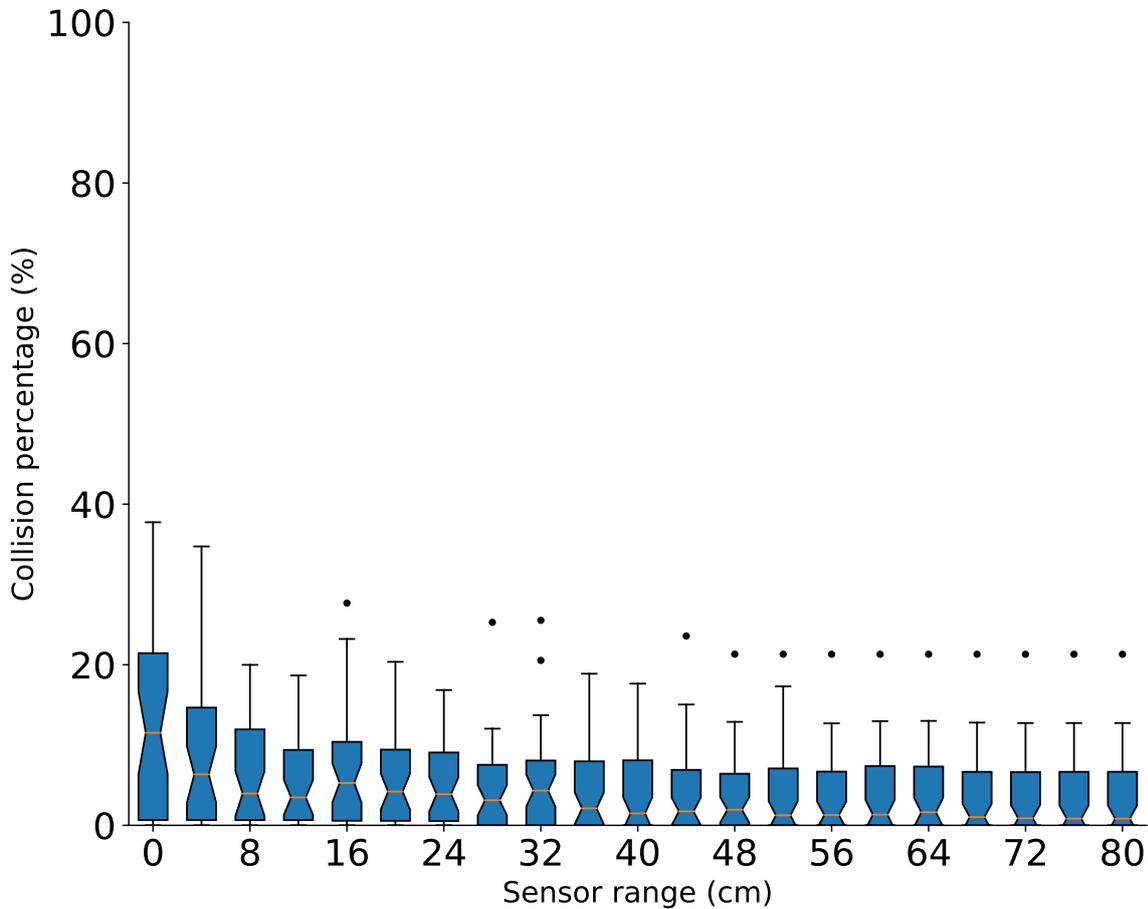


Figure 4.4 The distribution of number of collisions in the slalom environment using the dec-obs controller with different sensor lengths (σ). Each boxplot is composed of 40 values.

When using controller dec-com, a module's face may communicate with a paired face as described in Section 3.3.3. The transmission of information throughout the robot is not simulated directly, each face simply knows the required information from its paired face. As a result communication delays between the faces are not simulated; it is assumed that this would be small compared the time scale on which the robot moves.

Two different types of robot configuration are used throughout the simulations, *cubic* and *random*. Cubic robots have a convex, cubic module configuration and we define them by side length. For example a 3^3 robot is as cubic robot of 27 modules. Random robot configurations are generated in the following manner: The robot configuration is initialized with a single module. Additional modules are added one at a time. An attachment face is chosen randomly

Table 4.1 Mean time taken and energy required in the open environment

Controller	Cubic		Random	
	Time (s)	Energy (eu)	Time (s)	Energy (eu)
dec	125.1	114500	127.5	263000
dec-com	125.1	56100	127.5	80310
cen	108.8	74210	102.1	137300

from all available external module faces on the robot. A new module is added, connecting to that face. This repeats until the robot consists of the desired number of modules.

In addition to the decentralized controllers we use a centralized controller taken from the literature [20]. This controller solves equation systems to generate errors for each individual thruster, and can handle both translation and rotation. The controller requires the positions and orientations of every thruster as input. The output errors are given to PID controllers that generate the thruster output. The original controller uses individual PID values calibrated for each thruster. For our implementation we use the same values for each thruster, calibrated on a 5^3 cubic robot via grid search (shown in Figure 4.5). These same values were also used with different robot morphologies (and therefore thruster configurations). This could result in a lower than optimal performance. In order to maximize the performance of the controller both translation and rotation are controlled. The controller rotates the robot into an orientation that allows a higher rate of linear acceleration at the same time as translating towards the goal.

Throughout this section we use the two-tailed Mann-Whitney test. Each test uses a base alpha level of 0.01. Given that each set of data has two tests performed upon it, we use a Bonferroni adjusted alpha level of 0.005. We report the a posteriori p -values obtained by the tests.

4.1.2 Time to completion

This section evaluates the success rate of, and the time taken by, simulated MHP robots to reach the goal.

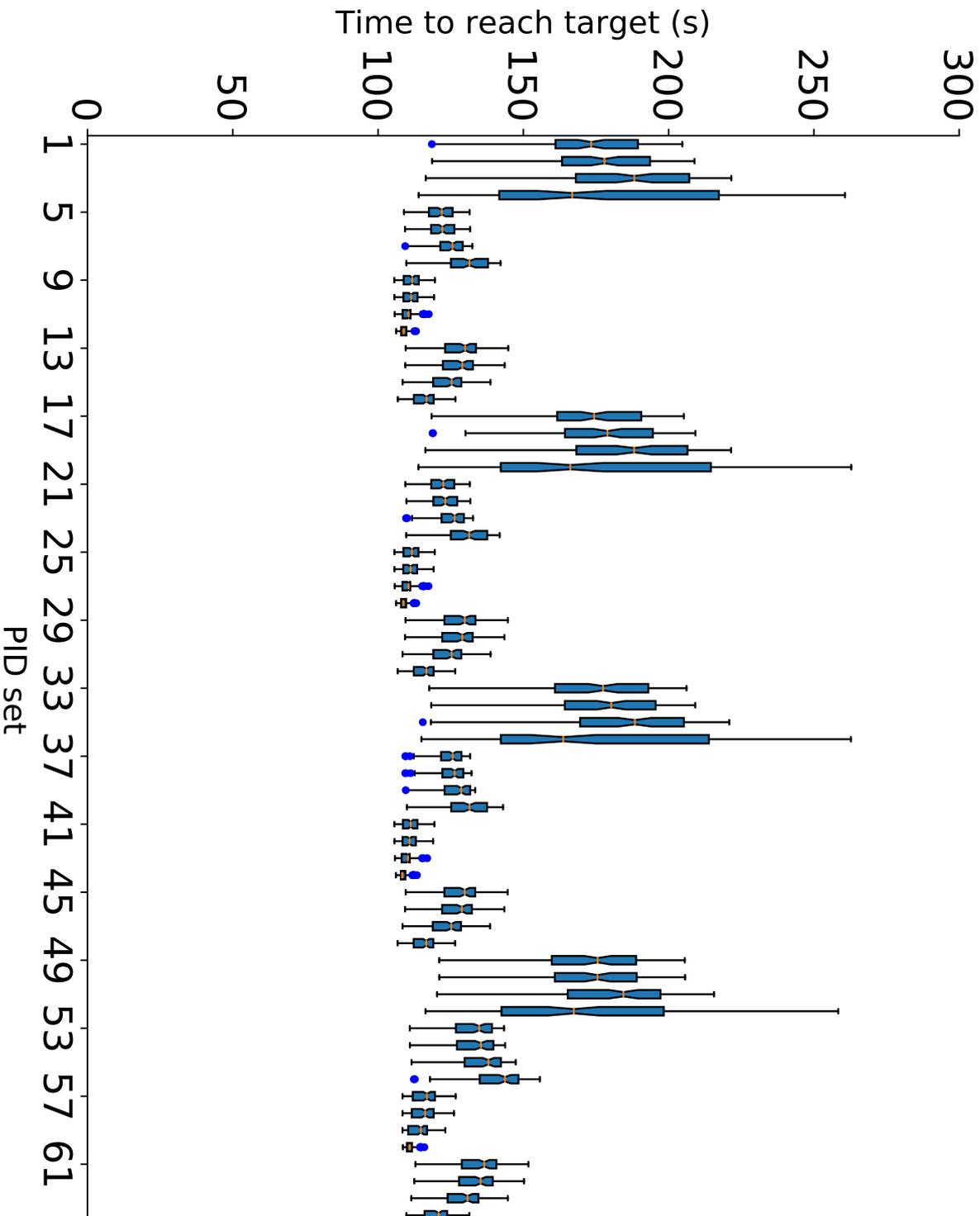


Figure 4.5 Boxplots showing calibration of the centralized controller. The i th PID set corresponds to values of $P_T = 1$, $D_T = 5^{(i/16)\%4} / 10$, $I_T = 0$, $P_R = 5^{(i/4)\%4} / 10$, $D_R = 5^{i\%4} / 10$ and $I_R = 0$ (where % is the modulo operator, P_T is the proportional translation component, D_T is the derivative translation component, I_T is the integral translation component, P_R is the proportional rotation component, D_R is the derivative rotation component and I_R is the integral rotation component).

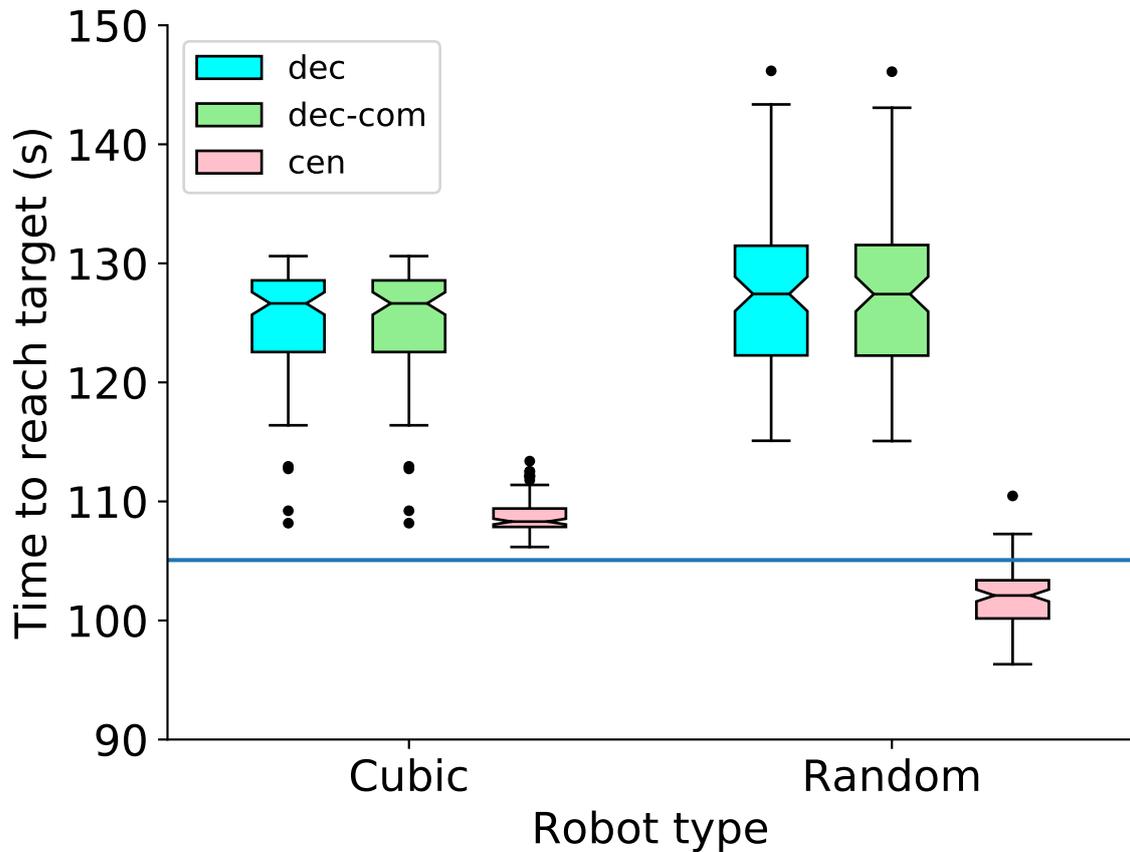


Figure 4.6 Distribution of times taken for cubic robots and random robots to reach the target position. Each box plot is composed of 100 trials. The dec, dec-com and cen controllers are shown. The horizontal blue line represents the theoretical minimum time for a 5^3 robot to reach the target position. Note that the y axis is truncated for readability.

Setup

We ran six sets of trials in the open environment, one per combination of robot configuration (*cubic* and *random*) and controller (dec, dec-com and cen). Each set consisted of 100 trials, with the same set of 100 random starting orientations.

Results

Figure 4.6 shows the times taken to complete each set of trials for robots of cubic and random shape. Table 4.1 shows the mean times. Every trial was completed successfully. The horizontal line across the graph indicates the theoretical smallest time that a 5^3 cubic

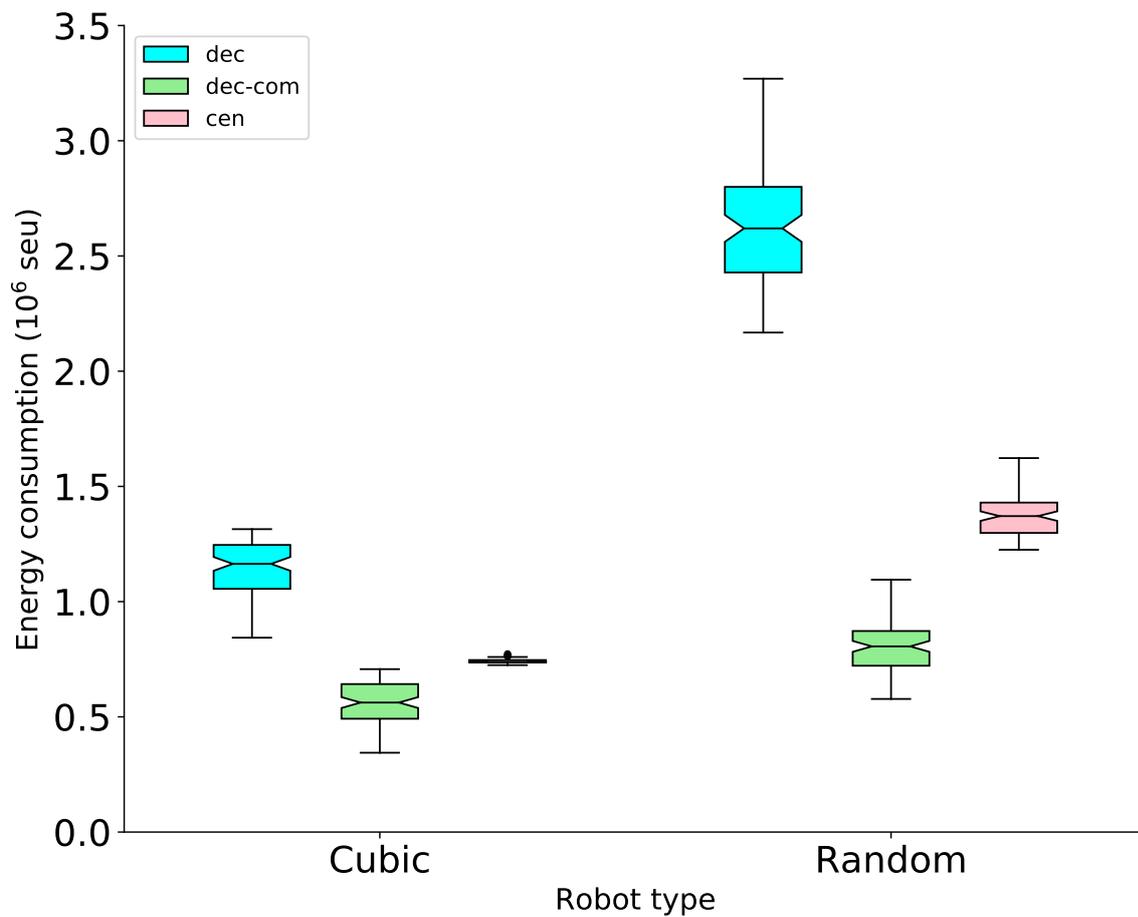


Figure 4.7 Distribution of energy required for cubic robots and random robots to reach the target position. Each box plot is composed of 100 trials. The dec, dec-com and cen controllers are shown.

robot could take to reach the goal, independent of controller. Controllers dec and dec-com produce identical distributions. This is because the effect of allowing communication is only to turn off pairs of actuators that would otherwise both be firing and producing no net force and torque. Therefore controllers dec and dec-com give rise to exactly the same motion of the robot.

Controller cen outperforms dec for both cubic robots ($p < 0.001$) and random robots ($p < 0.001$). The dec controller does not perform significantly worse with random robots than with cubic robots ($p = 0.0251$). Somewhat surprisingly random robots outperform cubic robots ($p < 0.001$) when using controller cen. This is partly explained by the fact that

the optimum time is lower for a robot of random morphology than for a cubic robot. It also suggests that only calibrating the PID values for a single robot shape does not noticeably hamper the controller in our scenarios.

4.1.3 Energy expenditure

This section evaluates the performance of simulated MHP robots with regard to energy expenditure.

Setup

We analyze the same set of simulation runs reported in 4.1.2. To calculate the energy consumption of the robot, the total number of actuators active during each time step is recorded, and then summed over every time step.

Results

Figure 4.7 shows a box plot of the cumulative energy consumption during the trials. Table 4.1 shows the mean energy consumption for each set of trials. For each controller cubic robots perform better than random robots ($p < 0.001$). This is explained by the potential concavity of random robots which results in more actuators firing for a net zero effect. The dec-com controller outperforms the dec controller for both cubic robots ($p < 0.001$) and random robots ($p < 0.001$). The largest improvement is shown for random robots, as communication helps to counter the effect of the concavity. For both types of robot the cen controller outperforms the dec controller ($p < 0.001$ in both cases) and is outperformed by the dec-com controller ($p < 0.001$ in both cases). In particular the dec-com controller reduces energy required by 51% for cubic robots and 69% for random robots compared to the dec controller.

4.1.4 Negotiating obstacles

We compare the dec, dec-obs and cen controllers in an environment with obstacles. We evaluate their performance with regards to success rate, time taken and collision rate.

Table 4.2 Mean time taken and number of collisions in the slalom environment

Controller	Cubic	
	Time (s)	Collision percentage (%)
dec	171.9	10.5
dec-obs	144.1	2.47
cen	159.9	23.5

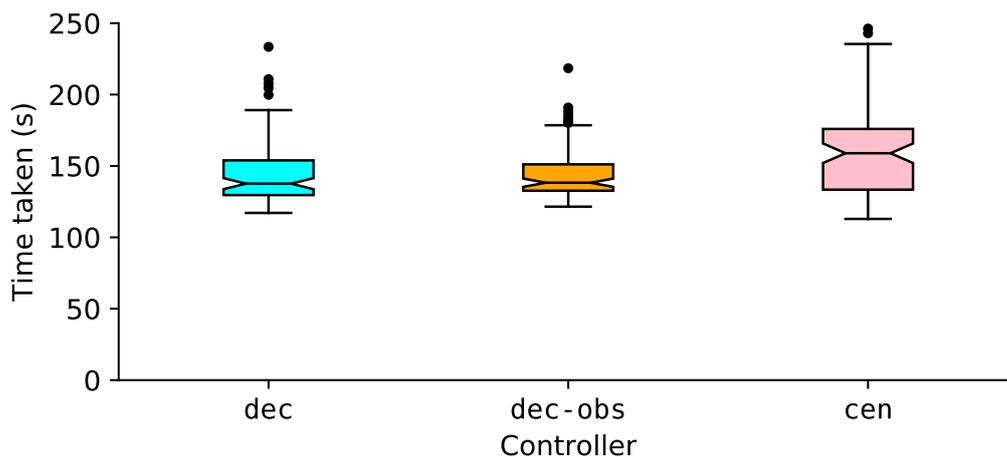


Figure 4.8 Distribution of time taken for cubic robots using the dec, dec-obs and cen controllers. The boxplots are composed of 100, 100 and 95 trials respectively.

Setup

We ran one set of trials for each controller. Each set consisted of 100 trials, using the same set of 100 randomly generated starting orientations. In each trial we recorded the number of time steps in which the robot is in contact with an obstacle.

Results

Figures 4.8 and 4.9 respectively show distributions of time taken and collision percentage for each set of trials. For the dec and dec-obs controllers all trials were completed successfully. For the cen controller 95 trials were completed successfully. Table 4.2 shows the mean times and collisions rates for successful trials. The dec-obs controller is not significantly faster than the dec controller ($p = 0.55$). The cen controller is not significantly slower than the dec controller ($p = 0.013$), but is significantly slower than the dec-obs controller

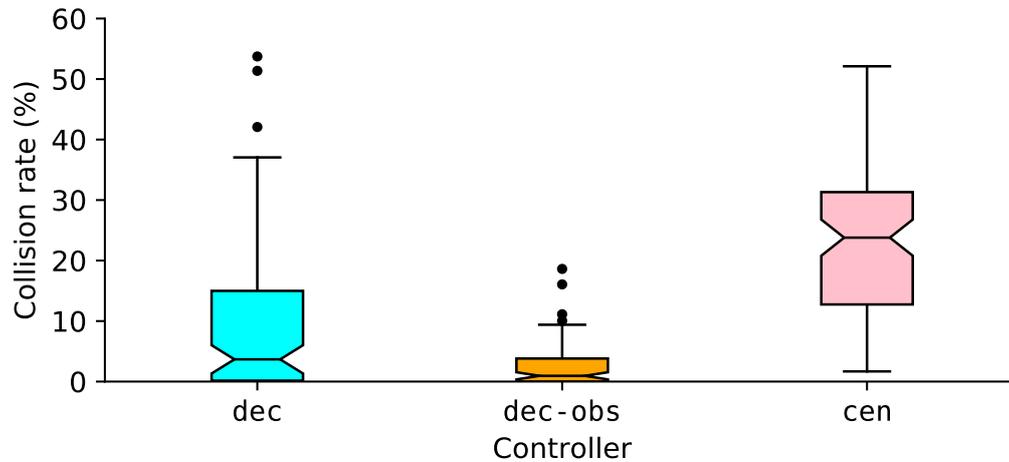


Figure 4.9 Distribution of number of collisions for cubic robots using the dec, dec-obs and cen controllers. The boxplots are composed of 100, 100 and 95 trials respectively.

($p = 0.002$). Figure 4.9 shows the percentage of time that the robot was in physical contact with an obstacle. The dec-obs controller shows significantly less collisions than the dec controller ($p < 0.001$). A possible interpretation is that the additional time taken for robots using the dec-obs controller to navigate around obstacles is approximately the same as the time lost by robots using the dec controller colliding with the obstacles. The cen controller shows significantly more collisions than both the dec ($p < 0.001$) and dec-obs ($p < 0.001$) controllers. The relatively poor performance of the cen controller in this scenario can be explained by the fact that the controller causes the robot to move in an almost direct line towards the goal. This maximizes the number of encounters the robot has with the obstacles, increasing collisions and reducing speed.

4.1.5 On normality

We tested the normality of the results of Sections 4.1.2, 4.1.3 and 4.1.4 using the Shapiro-Wilk test, at a significance level of $\alpha = 0.01$. All sets of trials using random robots produced normal distributions. Those using cubic robots produced significantly skewed distributions. This suggests that future studies might be best conducted using robots of random configurations, rather than preset configurations such as cubes. One possible interpretation of the discrepancy

is that the interaction of the uniformly randomized starting orientation with the cube shape provides many trials in which a robot might approach the goal slowly (if it is forced into a curved trajectory), and few trials in which the robot has an “ideal” starting orientation (for example face-on or vertex-on towards the goal) and can approach the goal quickly. Random robots tend to have a more rounded shape, and therefore are potentially less effected than cubic robots by the starting orientation.

4.1.6 Actuator errors and varying resolution

Whereas the simulations of the previous sections sought to provide basic validation of the concept, the purpose of this section is to test the robustness of the system and evaluate how well it can handle faults, an important characteristic of a modular system. Furthermore, the scalability of the system (one of the requisite properties identified in Section 1.2) is tested, in terms of both size and number of modules. We look at situations in which some actuators fail entirely for the duration of trial, and cases in which actuators have a misalignment in terms of the direction of the force they produce. For this section we use a starting goal distance of 100 cm and a maximum time of 50 s. This helps ease simulation time constraints when using robots with large numbers of modules. Note also that for this section the centralized controller was calibrated for a 3^3 robot. Furthermore, each set of trials does not contain the same set of random starting orientations.

Setup

We performed 30 sets of trials to investigate the effect of misaligned actuators. Each set consists of 100 trials. This models manufacturing defects in the actuators, or damage sustained in the course of operation. Each actuator’s firing direction is permanently offset by some angle from the normal, randomly generated up to some maximum value. Both the *dec* and *cen* controllers were tested. For each controller five sets of trials had no actuator error, five sets of trials used a maximum error of 45 degrees and five sets of trials used a maximum error of 60 degrees.

We performed 50 sets of trials to investigate the effect of failed actuators. Each set consists of 100 trials. All trials used the dec controller. At the outset of a trial each actuator is permanently deactivated with a certain probability. Note that this also models the situation in which the goal sensors are permanently stuck returning 1. The 50 sets of trials cover error probabilities ranging from 0.1 to 1.0.

For both types of error five different robot sizes were used, from 1^3 to 16^3 . The module size was scaled such that the total side length of the robot was 8 cm. For example the module in a 1^3 robot had a side length of 8 cm. Each module in a 4^3 robot had a side length of 2 cm. Correspondingly the actuator force is scaled such that the total force produced by one robot face, if all the actuators on that face fire, is 6.4 mN. Therefore a 4^3 robot would have a force per module of 0.4 mN. This means that all robots tested have the same overall size and maximum force, differing only in the modular resolution.

4.1.7 Results

Figure 4.10 shows the success rates and median velocities of the robot when subjected to permanently failing actuators. There is a general tendency for robots of higher resolution to maintain a higher success rate when subjected to a larger error. The highest resolution robots tested, the 16^3 robots, exhibit a success rate of 95% at an actuator failure probability of 0.7. Meanwhile the success rate of 1^3 robots drops to less than 5% at an actuator failure rate of 0.3. Above a certain error level the success rate stops improving even with higher resolution robots. This is partly explained by the fact that with a high resolution the variation in the number of non-failed actuators per robot face is small, and the total force per robot face can be estimated as number of actuators \times error rate \times force per actuator (or 6.4 mN \times error rate). Thus, independent of the starting orientation, at a high error rate the robots cannot generate enough force to reach the target position within the time limit, even if they are moving in the correct direction. For a 1^3 robot the variation in the number of non-failed actuators per robot face is high. Even with a high error rate, if the starting orientation is such that one or more of the occluded faces has non-failed actuators, the robot can still potentially achieve a high thrust.

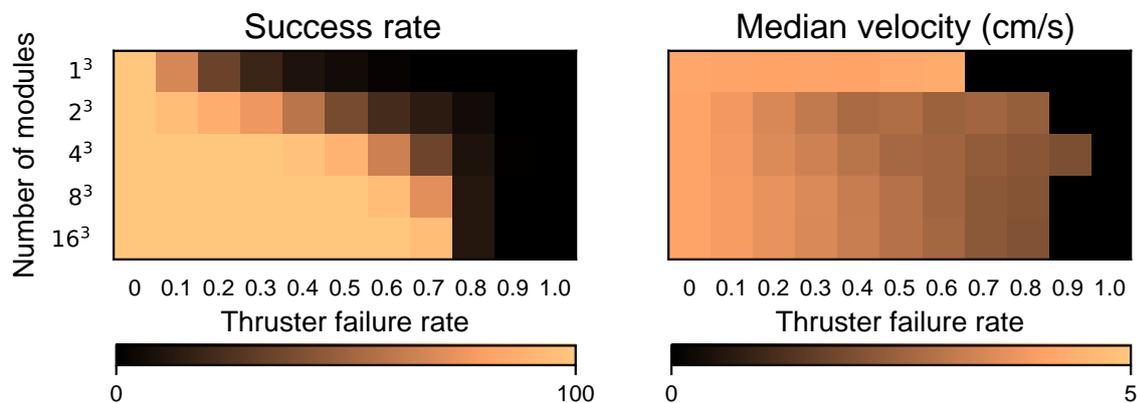


Figure 4.10 Color maps showing the success rates and median velocities of robots using the dec controller when subjected to permanently failing actuators. The velocities are averaged over successful trials only. Where there is a success rate of zero, the velocity is also recorded as zero.

Figure 4.11 shows the success rates and velocity distributions of the robot when subjected to misaligned thrusters. In the presence of no error all trials are successful. In addition, for each controller, the velocity distributions at all resolutions are similar. This is to be expected due to the scaling of the actuator forces and module sizes. Along with the fact that at any time all goal sensors on a given robot face must be occluded or unoccluded means that at different resolutions the robot is functionally identical. As before, the cen controller outperforms the dec controller with no error present. When error is introduced the median velocities drop for all controllers and resolutions. In addition, the success rates of the low resolution robots decrease. These effects are more pronounced with a higher error. Additionally, with a low resolution the variation of the robot velocity increases substantially. This is partly explained by the lower number of successful trials. At higher resolution the variation decreases to become similar to the no error case. With error the cen controller outperforms the dec controller at high resolutions. However at low resolutions the dec controller performs similarly or slightly better than the cen controller.

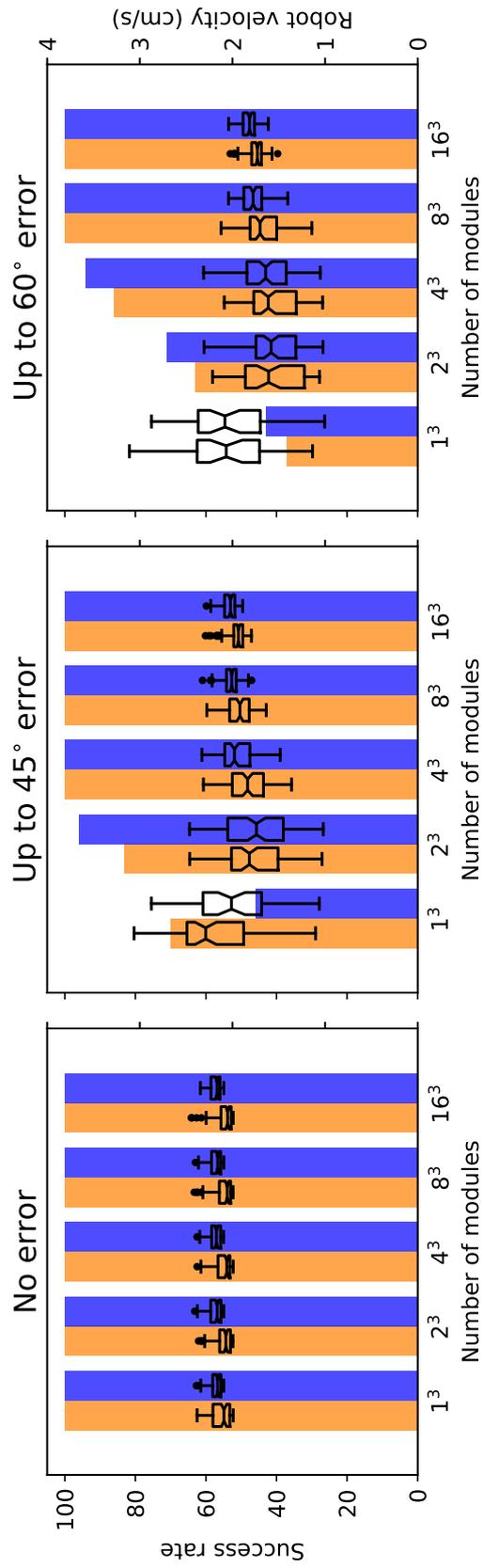


Figure 4.11 The results of MHP robots working with misaligned actuators. The bars show the success rate of the robots (out of 100) for the dec (orange) and cen (blue) controllers. The overlaid boxplots show the distributions of velocities for successful trials.

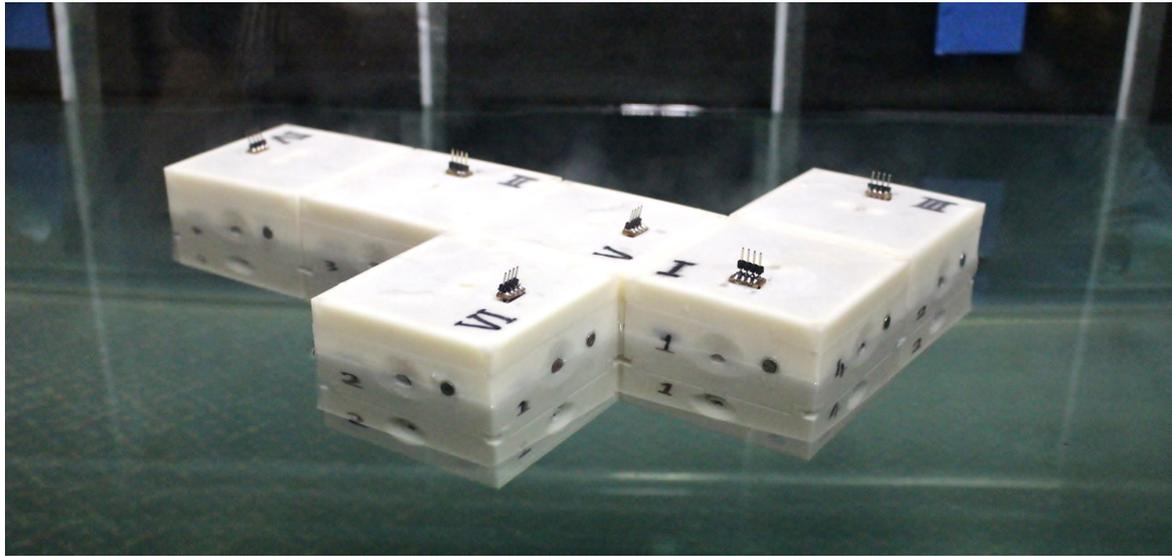


Figure 4.12 The hardware prototype of the MHP system floating on water in a tank. Each module is square shaped and contains four micro-pumps and an internal reservoir, four light sensors, a battery, and a microcontroller.

4.2 Hardware

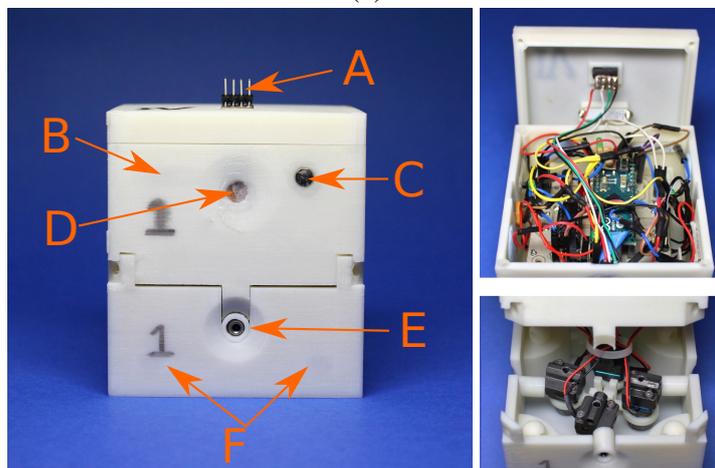
Thus far only theoretical or simulated results have been produced. In order to test an embodied version of the MHP concept, a set of six prototype modules (shown in Figure. 4.12) has been developed and tested. Floating on water, these represent a 2-D implementation of MHP. These hardware modules allow a primary validation of a physically implemented system, and also provide the means to align the simulator with a real world system.

4.2.1 Chassis

The module is a cuboid of dimensions 80 x 80 x 90 mm (L x W x H) as shown in Figure 4.13. The 3-D printed chassis consists of three sections: a *lower section*, an *upper section* and a *lid*. The lower section houses the thrusters—fluid micropumps—and is filled with water when in operation. The upper section contains the electronics, sensors and battery, and is enclosed and waterproofed. The top is covered by the lid, which contains a port for programming the microcontroller, and a magnetic switch to allow the module to be turned on and off. The modules weigh approximately 500 g each, and are balanced such that during operation the



(a)



(b)

Figure 4.13 Prototype module: (a) Assembled module. (b) Internal circuitry in the opened upper section (top right), micropumps in situ in the bottom section (bottom right) and annotated module (left). A: External programming port. B: Magnetic switch (mounted internally). C: The corresponding magnet. D: LDR mounted behind a clear barrier. E: Micropump output port. F: Connection magnets (mounted internally).

water level will be approximately 64 mm high. Modules can connect to one another using pairs of permanent magnets on each module's face, and can be reconfigured manually.

Sensors and actuators

One light dependent resistor (LDR) is mounted on each lateral face of the module pointing outwards. They are used to detect the target—an external light source. In order to translate each analog LDR input into a binary value, a threshold is set. If a calculated background value subtracted from the LDR reading is less than the threshold then the sensor input is treated as 0. Otherwise, it is treated as 1.

The thrusters are submersible centrifugal micropumps, with dimensions of 29 x 16 x 16 mm, a maximum flow rate of approximately 11 ml/s and a current draw of 500 mA. They are mounted in the lower section such that the output ports point outward from the horizontal center of each face. The ports are wrapped with a plastic sleeve to improve water tightness.

Each module maintains an internal reservoir of water that is linked to the environment through the fluid pumps. The pumps allow the water in these reservoirs to be passed between modules or between module and environment. As water is pumped out of a module, the resultant pressure difference causes water be drawn in through any pump which is not turned on.

For neighbor detection, the modules use a permanent magnet and magnetic switch pair mounted in each face. When two modules connect, the corresponding permanent magnets will trip the magnetic switch on the connected faces. In this way, modules can tell which of their faces are connected to neighboring modules.

4.2.2 Electronics and control

The processing capability of the modules is provided by an Arduino Micro. The microcontroller drives all pumps using a quadruple half H-bridge motor driver, and can receive analog inputs from the LDRs and digital inputs from the magnetic switches. A custom PCB is used to implement the circuit, which is powered by a 9 V Alkaline battery. Four green

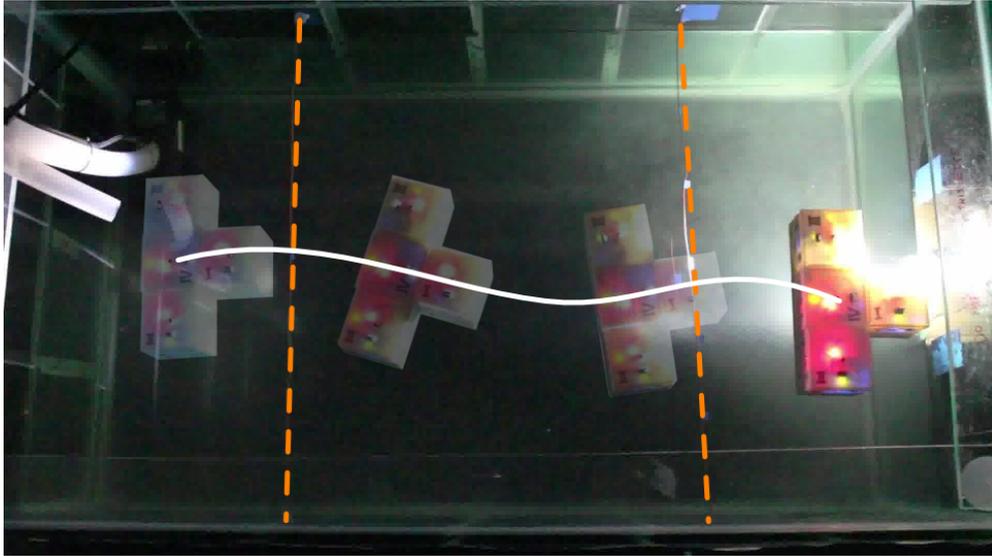


Figure 4.14 The experimental environment and an example path taken by a four module configuration. Each module of the robot executes the decentralized motion controller. The left hand and right hand dashed orange lines show the start and finish lines respectively. The white line shows the path the robot took from left to right. The torch is located outside the water tank on the right hand side.

LEDs display the activation status of the pumps. One further red LED is used for debugging purposes.

The modules each run a modified version of the dec controller, which takes advantage of the fact that each module has centralized control over its four faces. If a situation occurs in which two opposite faces of the same module were to be active, the controller will automatically render both inactive (the pumps will be deactivated). This reduces the overall time which pumps are active and so reduces stress on the system. Note that as the modules cannot communicate with one another (other than via the connection sensors) the full dec-com controller is not implemented.

4.3 Experiments

4.3.1 Setup

The experimental environment consists of a 126 by 68 cm water tank, shown in Figure 4.14. The water level is sufficiently high to prevent modules from coming into contact with the bottom. The robot has to move in the direction of a static target light.

The light source is a 210 lumens electric torch. It is positioned outside of the water tank on the right hand side (as seen in Figure 4.14), pointing inwards. A start and finish line are marked such that the distance between the lines is 50 cm, and the distance from the finish line to the torch is approximately 20 cm. Each trial starts with the robot just beyond the start line with a random orientation. As the robot's center of mass passes under the start line a timer is started. The timer stops when the robot center of mass passes underneath the finish line, and the time taken is recorded.¹

4.3.2 Results

We performed trials using all nine possible configurations of four modules, that is, configurations which cannot be transformed into another configuration by either rotation or mirroring. These configurations are shown pictorially in Figure 4.15, along with the centers of mass of each configuration. Figure 4.14 shows an example of a path taken by one of the robots. Three trials were performed for each configuration, that is, 27 trials in total. All configurations successfully completed all their trials with the maximum time taken being 13.0 s, and the minimum time taken being 6.5 s. The average time taken over all trials of all configurations was 9.6 s, corresponding to a velocity of just under 5 cm s^{-1} . We also performed a small number of trials with six modules, which were successfully completed.

Robots were allowed to continue moving after they passed the finish line, until they reached the side of the water tank. We noted that robots with multiple modules hit the wall

¹Upon being released into the water, the robot would slightly drift in random directions. However each configuration took less than five seconds to cross the start line.

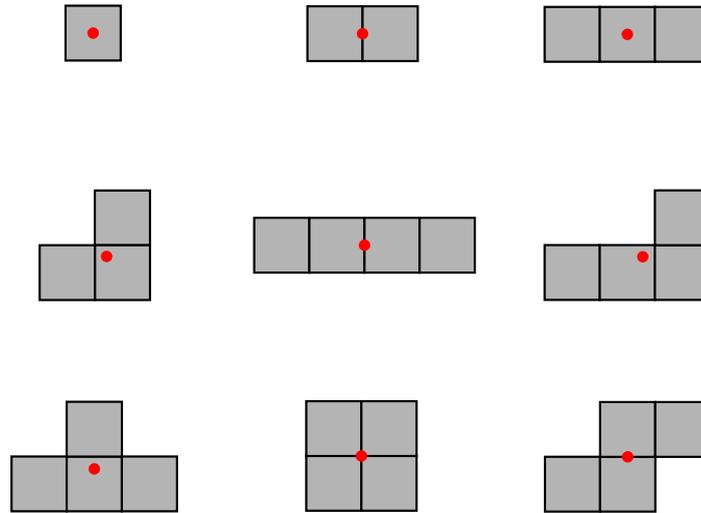


Figure 4.15 All 9 distinct robot configurations using four modules. These are configurations which cannot be transformed into another configuration by rotation or mirroring. The red dots indicate 2D centers of mass.

much closer to the center (the position at which the torch was mounted) than robots consisting of a single module.

4.3.3 Comparison with theory

Compared to the theory presented in Chapter 3 the hardware prototype and experiments show some notable differences. The goal is not a single point; the torch used was not a good approximation for a light source with isotropic emission, and so the light received by a given sensor was highly dependent on the distance and angle between the sensor and the torch. This provides a similar situation to that tested in Section 4.1.6, wherein inaccurate sensors give incorrect responses. Moreover, the light sensors themselves do not accurately represent the idealized sensors considered in the theory. The theoretical sensors have infinite range and a hemispherical field of view. The detection strength of the physical sensors drops as distance to the light source increases, and the sensors have an inconsistent field of view that is less than a hemisphere.

The fluid pumps also exhibit differences to the theoretical actuators. The pump outputs can have small misalignments, such that the internal fluid is not expelled directly away from

the module, along the face normal. This can cause unwanted torques to be applied to the system, as well as reducing the translational force applied. Additionally, the flow rate of each pump is not exactly the same.

In accordance with the MHP concept, water should be expelled only at specified points (by active thrusters) and drawn in only at specified points (sites of inactive thrusters). However the physical fluid network formed by the prototypes was not perfect and prone to leakage. This could result in water not properly circulating through the network. The environment is also different to that dealt with by the theory. As discussed in Section 3.5 the robot's motion in water is not quasi-static, with inertia playing a large role. In addition to this, minor disturbances in the water cause the robot to drift around even when unpowered, and could have caused trajectories to be more curved than they otherwise would have been. The success rate of the system indicates its ability to function well despite these faults.

The primary technical challenge identified in Chapter 1 was to develop a reconfigurable modular robot with a 3-D lattice structure that can propel effectively in any configuration. This physical prototype partially fulfils this challenge. The system is only a 2-D implementation. In addition to requiring fewer connectors, sensors and actuators than a 3-D implementation, this also avoids dealing with problems of buoyancy and depth control, water-proofing, and the use of sensors and communication underwater. However the system fulfils the fundamental challenge of self-propelling effectively in a wide range of configurations.

In Section 3.2 we derived a formula for the number of forces and torques that can be created by 2-D convex robots. Using our experimental results we can provide a rough estimate for the robot actuator force and thereby calculate the range of forces and torques available to the system.

Adapting Equation 4.1, the equation of motion for a single module moving face-on in a straight line in 1-D is:

$$M\ddot{x} = \xi - \frac{1}{2}\rho C_D s^2 \dot{x}^2, \quad (4.2)$$

where ξ is the (single) actuator force and x is the robot position. During the experimental trials the robots reach terminal velocity before they pass the start line. At terminal velocity:

$$\xi = \frac{1}{2} \rho C_D s^2 \dot{x}_T^2, \quad (4.3)$$

where \dot{x}_T is the terminal velocity. Substituting the experimental value of $\dot{x} = 5 \text{ cms}^{-1}$ (for the trials using a single module) and using $s^2 = 51.2 \text{ cm}$ gives $\xi = 6.7 \text{ mN}$. From Appendix A this means that a 2×3 robot (the largest convex robot that we tested) can produce 13 discrete torques between -1.6 mNm and 1.6 mNm . A 10×10 robot would be able to produce 201 discrete torques between -26.8 mNm and 26.8 mNm .

It is worth noting that the robots in both our experiments and simulations reach all the way up to the goal. The distance condition derived in Section 3.4 represents a limit up to which an MFP robot is guaranteed will approach the goal. Although we do not prove that the MFP robots are guaranteed to approach the target beyond this limit, we expect that in all, or almost all cases they would. The results here provide examples of systems that do so, and we are yet to find examples of systems that do not.

In Section 4.1 we investigated a 3-D MHP system in simulation. Using the value derived experimentally for the thrust force we present a 2-D simulation of the robot hardware described in Section 4.2. In addition to modifying the thrust force, drag force and target distance correspondingly to match those in Section 4.3, errors are introduced into the actuation and sensing to model those observed in the hardware. Instead of a hemispherical detection cone the target sensors detect in a cone of total angle α , where α is randomly generated between 120 degrees and 180 degrees for each sensor at the outset of each trial. Additionally, the force vector for each actuator is offset from the corresponding face normal by angle β , where β is randomly generated between 0 degrees and 60 degrees for each actuator at the outset of each trial. As in Section 4.3 all possible robots consisting of up to four modules were trialled. Three simulation trials were conducted for each shape, for a total of 27 trials. All trials were successful, with the maximum time taken being 16.8 s, and the minimum time taken being 7.6 s. The average time taken over all trials of all configurations was 10.2 s, corresponding to a velocity of 4.9 cms^{-1} .

4.4 Orientation analysis

In this chapter we studied the ability of MFP robots to successfully reach the goal, and their robustness to errors in actuators and sensors. However our study into how long it takes a robot to successfully reach the target was limited. Viscosity of the environmental fluid, mass of the robot and strength of the actuators could all be expected to have straightforward effects on the speed with which a robot reaches the goal. A more interesting analysis might be undertaken on the overall robot shape. One important note is that many combinations of robot shape and initial starting orientation will cause the robot to rotate, making an analysis on the effects of the robot shape more complex. Robots which have symmetrical actuator activation about the robot-goal vector will not rotate (unless due to noise, as in the hardware experiments). Is it therefore most straightforward to analyze these types of robots (with corresponding appropriate starting orientations).

One might expect that more streamlined robots shapes can reach higher velocity. For a physical system this would be the case. However given our highly simplified drag model (Equation 4.1), in which drag force is proportional to the cross sectional area of the robot, this is not the case. Figures 4.16a and 4.16b show an example. Both robots experience the same drag force as they have the same cross-sectional area (distance); the overall shape of the robot is not considered.

The orientation of a robot does however have an effect on the time taken to reach the goal. Consider a square robot that in one trial is moving towards the goal face-on, and in another is moving towards the goal vertex-on (Figures 4.16c and 4.16d). Returning to the equation of motion for a robot moving in a straight line in 1-D, at terminal velocity:

$$f = \frac{1}{2}\rho C_D A \dot{x}_T^2, \quad (4.4)$$

where \dot{x}_T is the terminal velocity, f is the total actuator force in the direction of the goal and A is the cross sectional area. Solving for \dot{x}_T :

$$\dot{x}_T = \sqrt{\frac{2f}{\rho C_D A}}. \quad (4.5)$$

The cross sectional area A of the robots in Figures 4.16c and 4.16d scales from $5s$ to $5\sqrt{2}s$, a factor of $\sqrt{2}$. However the total thrust force f scales by the same factor, so the terminal velocity \dot{x}_T is the same in both cases. If the robots start a trial moving at terminal velocity they will reach the goal at the same time in either orientation. In our simulation trials however the robots start at rest. In this situation the robots in each orientation would not take the same amount of time to reach the goal. Solving the equation of motion with the initial conditions $x = 0$ at $t = 0$ and $\dot{x} = 0$ at $t = 0$ gives

$$t = \frac{\sqrt{2} \cosh^{-1} e^{\frac{\rho C_D A x}{2}}}{\sqrt{f \rho C_D A}}. \quad (4.6)$$

Using the same values as in Section 4.1.2 (converted for a 2-D robot), a 5^3 robot of module size 0.08 m, density 1000 kgm^{-3} and actuator force 6.4 mN, moving face-on towards the goal from a distance of 500 cm would take 354 s to reach the target. The same robot moving vertex-on towards the goal from the same distance would take 250 s to reach the target. The discrepancy is caused by the fact that although both robots will reach the same terminal velocity, the robot with the highest actuator force (vertex-on) reaches the terminal velocity more quickly. Therefore for a small robot-goal distance a vertex-on robot will perform substantially better than a face-on robot. As the initial robot-goal distance increase, the disparity between the two orientations will decrease. Note that in physical experiments the robots experience significant random rotation, and therefore the difference caused by the initial starting orientation could be difficult to perceive.

4.5 Summary

This chapter validated the MFP concept through the use of computer simulations and experiments with a hardware prototype.

A 3-D physics simulator was developed to simulate MFP robots of generic shape. Both cubic robots and robot of random shape, consisting of 125 modules, were able to successfully complete the directed locomotion task. Robots using the dec-com controller save on average 51% and 69% energy when compared to the dec controller, for cubic and random robots

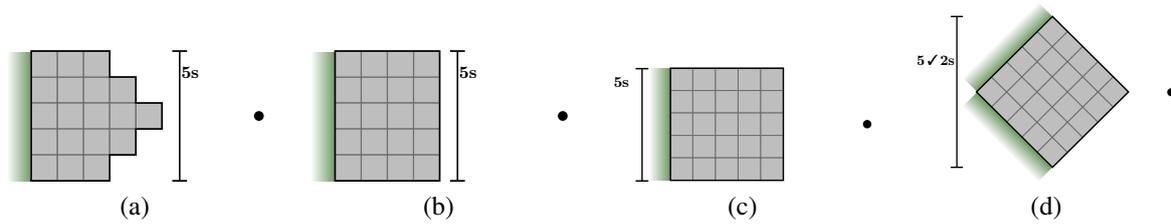


Figure 4.16 Streamlined or rotated 2-D MFP robots. Each robot is moving to the the right, towards the indicated goal point. Due to the symmetry of the situation none of the robots will rotate. (a) An MFP robot with a streamlined shape. (b) An MFP robot with a less streamlined shape, that will nevertheless experience the drag force as the one in (a). (c) A 5^2 robot moving face-on towards the goal. (d) The same 5^2 robot moving vertex-on towards the goal.

respectively. Both controllers performed competitively when compared to a centralized controller taken from the literature. The dec-obs controller did not significantly improve task completion time in an environment with obstacles. However it did significantly reduce the collision rate. When subjected to misaligned or failed actuators robots with higher resolution tended to show better success rate and lower variance in time taken.

A set of prototype hardware modules were developed. The modules move in 2-D on the surface of water. Each module is equipped with four actuators, four light sensors and four connection sensors, allowing the system to implement the dec controller. Experiments with the prototype system show that a robot composed of up to four modules can complete the directed locomotion task in any possible configuration.

Chapter 5

Validating the evo-bot concept

Chapters 3 and 4 developed and analyzed a system that can propel through fluids. Two implementations are proposed, one that operates in liquids and one that operates in a microgravity environment. While the liquid-based system (MHP) is analyzed in detail, the microgravity-based system (MPP) is not. In particular no hardware experiments are performed.

This chapter investigates the evo-bots, modular self-reconfigurable robots which float in 2-D on top of an air table, simulating a microgravity environment. In addition to providing an opportunity to investigate modular robots in a microgravity environment, the system also serves as an example of how self-reconfiguration can enhance modular robots as discussed in Chapter 1. Propulsion of the robots is brought about by a stop-start motion mechanism. Although similar types of motion have been studied before in simulation [14], this chapter reports on the first experimental demonstration of such a mechanism.

The evo-bots were designed as a physically embodied artificial life system which can evolve in morphology and adapt to a changing environment. The ultimate goal of the system is to fulfil one of the grand challenges of evolutionary robotics [23]—to create ecosystems of physically evolving robots. In pursuit of this goal this chapter experimentally demonstrates the individual functions of the evo-bots. These are:

1. Response to environmental stimuli. This allows the evo-bots to detect sources of energy.

2. Charging from environmental energy sources. This allows evo-bots that are suitably configured to draw energy from the environment and sustain themselves.
3. Sharing energy between modules. This allows harvested energy to be distributed between every module in an evo-bot, keeping the entire robot powered.
4. Stop-start motion control. Moving around randomly allows the evo-bots to physically interact with one another to precipitate growth or replication and stochastically search for energy sources. Being able to stop moving allows evo-bots to charge from energy sources that they have detected.
5. Inter-module communication and self-assembly. Modules which collide and connect can decide whether to stay connect or disconnect. This allows for self-assembly: the creation of new evo-bots from individual modules in the environment and the growth of existing evo-bots.

Functions 1 and 4 are demonstrated in an experiment that requires modules to stop in position under a light source. Functions 2 and 3 are demonstrated by tests of the charging capabilities of the evo-bot modules. Function 5 is demonstrated in an experiment which requires individual modules to self-assemble into linear polymers.

This chapter is laid out as follows. Section 5.1 describes the evo-bot concept. Section 5.2 details modifications made to the module design (Section 5.2.1) and experimental environment (Section 5.2.2) to improve successful connection and floating of the evo-bots. Section 5.3 reports on experiments with the evo-bots, demonstrating motion control (Section 5.3.1), energy management (Section 5.3.2) and simple self-assembly (Section 5.3.3). Section 5.4 summarizes the chapter.

5.1 Concept

The evo-bots were first introduced by Groß et al. [37], who describe the basic concept and present simulations of the system. The concept was further described by Escalera et al. [26, 24]. The thesis author summarizes the concept here.

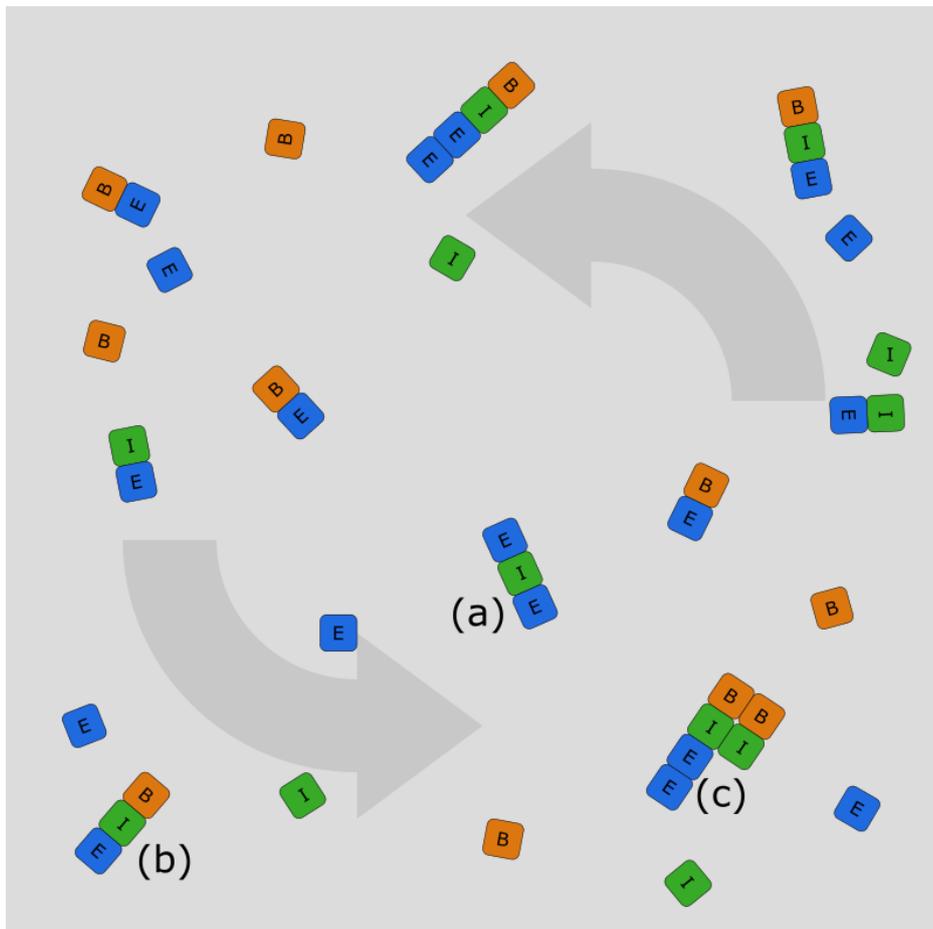


Figure 5.1 The evo-bot environment containing a “primordial soup” of modules. Along with individual modules, growing polymers (a), completed organisms (b) and replicating organisms (c) are present. The arrows indicate the general direction of motion induced by the external propulsion

Evo-bot modules have no explicit form of self-propulsion. Instead they are induced into semi-random motion by an external propulsion mechanism in their environment (shown in Figure 5.1). The modules are square, with a connector on each face, and can connect in a lattice pattern to form structures in 2-D. Connection between modules occurs as they randomly collide with each other while moving through their environment. Upon connection, modules can communicate, and decide whether to stay connected or detach from one another.

Modules are classed as one of three types, energy, interaction and boundary, as detailed in Table 5.1. Each type has certain unique capabilities. Modules follow rules which determine what kind of structures they form (shown in Figure 5.2). Individual modules permit

Table 5.1 Evo-bot module functionality

	<i>Energy</i> modules have the ability to harvest (light) energy from the environment and store it. This energy can be shared with connected modules.
	<i>Interaction</i> modules have an anchor mechanism which can be activated or deactivated when necessary. When active, this mechanism increase the friction between the module and the environment, preventing the module (and any structure it is connected to) from moving. When inactive, the module (and structure) moves around semi-randomly as normal.
	<i>Boundary</i> modules define organisms. A polymer that accretes a boundary module stops growing and starts replicating.

connections on any of their faces (Figure 5.2a). Once a pair of individual modules connects, each module only allows further modules to connect such that a linear *polymer* of modules is formed, with no branches (Figure 5.2b). Once a polymer accretes a boundary module it becomes an *organism* and stops growing (Figure 5.2c).

Organisms are capable of self-replication, as shown in Figure 5.3, using unconnected modules in the environment. This is done on a module-by-module basis starting with the organism's boundary module. Each module accretes a module of the same type as itself, thereby forming a replica of the organism. Once this process is complete, the original and the copy separate from one another.

Polymers and organisms depend on energy to survive. This can be harvested by energy modules from areas of light within the environment. The harvested energy is then shared throughout the evo-bot structure. Structures consume energy at a rate dependent on the number of modules they are composed of—a structure with more modules consumes energy at a faster rate. If a structure runs out of energy, it “dies” and splits into its constituent modules. These modules can then be reused in the formation or replication of new structures.

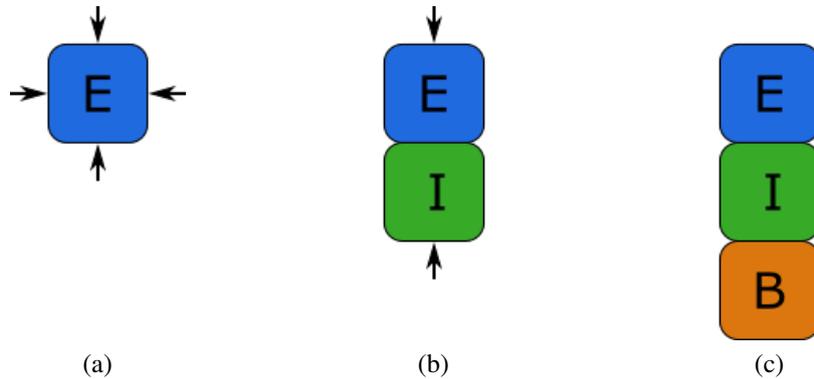


Figure 5.2 An example of the evo-bot growth process. Black arrows indicate module faces which are accepting new connections. (a) An individual energy module allows an additional module to connect on any face. (b) An interaction module connects to form a 2 module polymer. New connections are only accepted on either end of the polymer. (c) A boundary module connects to form a 3 module organism, which stops growing.

Energy can be distributed throughout the environment in different ways. Depending on its modular composition, in terms of the numbers of interaction and energy modules, an organism may or may not be capable of surviving in a given environment long term. This means that unsuccessful organisms will expire, while successful ones survive and propagate their morphology.

Escalera et al. [26] argue that the evo-bot concept encompasses at least one definition of life—that of Koshland Jr. [52]. Koshland Jr. proposes seven pillars that an organism must adhere to in order to be considered as a living being. They are program, improvisation, compartmentalization, energy, regeneration, adaptability and seclusion. The evo-bots can be claimed to embody these concepts as follows. Each organism encodes the information of its constituent modules, their functionality and how they interact (program). The modules are able to alter said program between generations (improvisation). Each organism defines its own space, which is protected by a boundary, wherein internal processes run protected from external disturbances (compartmentalization). Organisms are able to harvest, store and distribute energy from the environment (energy). Organisms possess the capacity of self-replication (regeneration). They can vary their behavior according to external stimuli

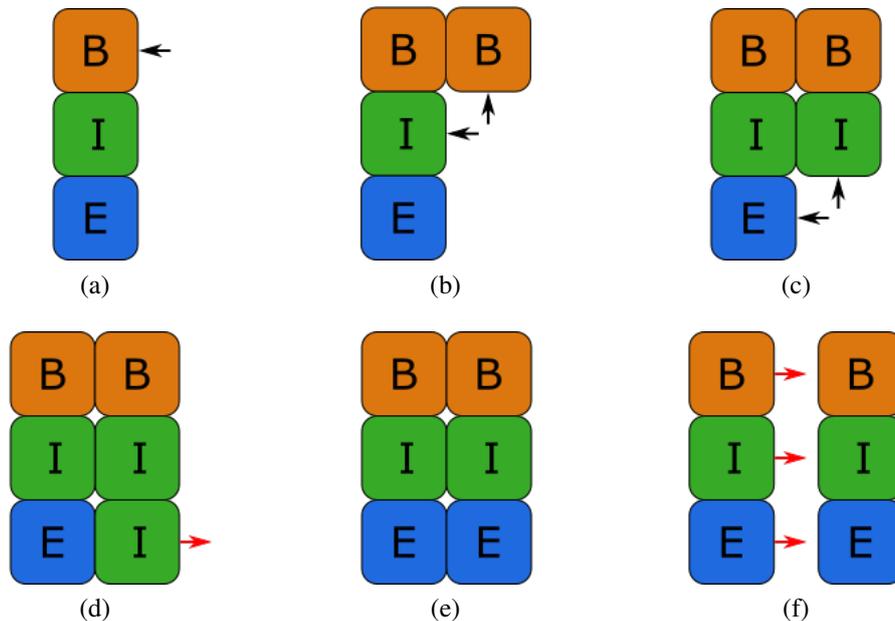


Figure 5.3 An example of the evo-bot replication process. Black arrows indicate module faces which are accepting new connections. Red arrows indicate the disconnection of connected modules. (a) An organism boundary module accepts new connections on one of its perpendicular faces. (b) A new boundary module connects. The next module in the original organism now accepts new modules on the same side. (c) A new interaction module connects. The final module in the original organism now accepts new connections on the same side. (d) A new interaction module connects. This is the wrong type of module, so it is rejected. (e) A new energy module connects and the replicated organism is complete. (f) The original organism and the replicated organism disconnect from one another.

(adaptability). Finally, the information within each organism is protected from external agents (seclusion).

5.2 System modifications

As modular robots the ability of the evo-bots to successfully form connections between modules is paramount. A connection is successful if the modules involved can both communicate and share energy, and if the connected structure can float well on the air table. In their original state the evo-bot modules had a fairly low connection success rate. In particular connected structures of more than two modules had problems floating, and frequently became inadvertently stuck on the air table surface and unable to move. This section qualitatively



Figure 5.4 The evo-bot hardware. (a) A single evo-bot module, showing chassis (A), Connectors (B), solar cells (C), servo motors (D) and PCB (E). (b) Six evo-bot modules, five of which are connected.

analyzes the reasons for the poor floating and details efforts to improve both the module design and the experimental environment.

5.2.1 Modules

Groß et al. [37] developed the first evo-bot hardware prototype. A second version was developed by Ding et al.[17]. The current version of the hardware (Figure 5.4) was developed by Escalera et al. [26, 24]. Although conceptually there are three different types of modules each hardware module is functionally identical. Modules can be configured to perform any of the three roles during a given experiment. The modules are approximately square with a side length of 6 cm and height 3.5 cm (Figure 5.4a-A). The connection mechanism on each module side consists of a fixed permanent magnet and a rotatable pair of magnets mounted on a servo motor (Figure 5.4a-D). The module environment is an air table. Each module incorporates a stop-start mechanism, which takes the form of an anchor device actuated by a fifth servo motor. Each module has an array of solar cells on its top (Figure 5.4a-C) and an internal LiPo battery. The module PCB (Figure 5.4a-E) is mounted beneath the solar cells. Spring loaded contacts (Figure 5.4a-B) on each side of the modules provide electrical connectivity between connected modules.

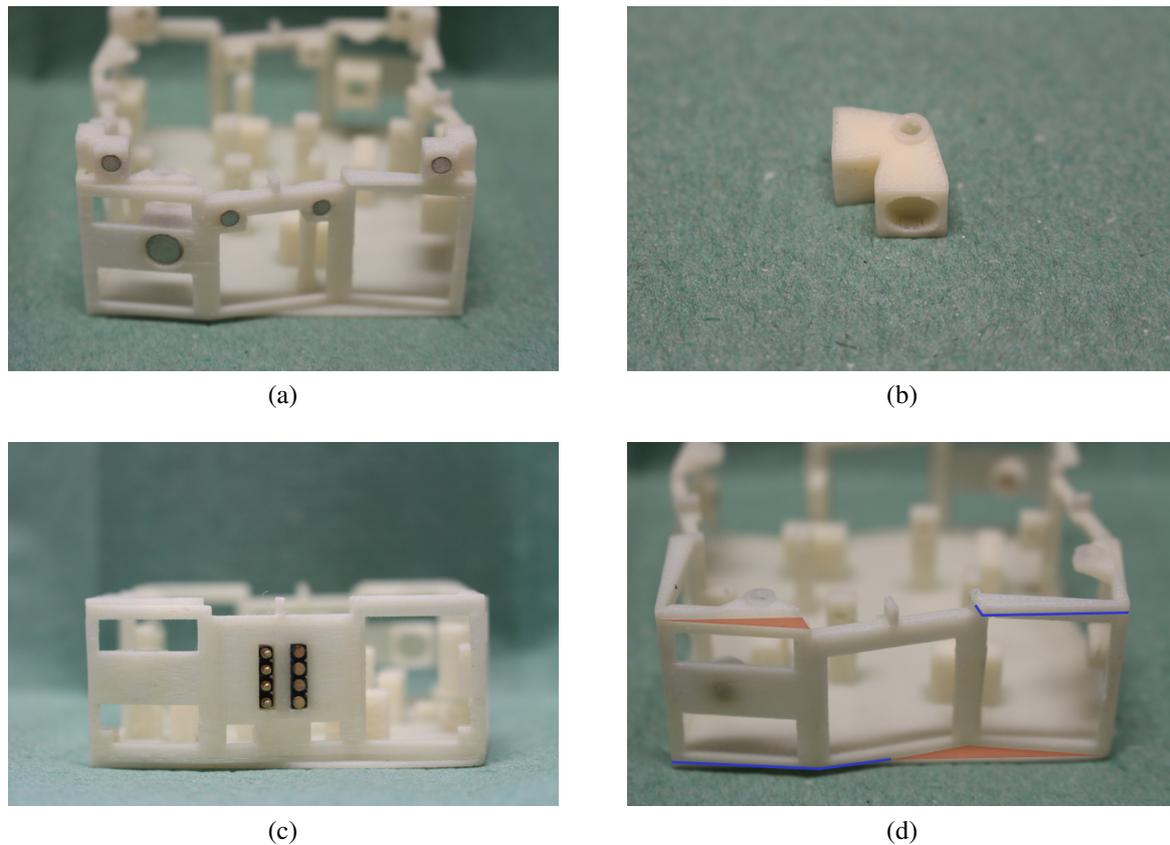


Figure 5.5 Modifications made to the evo-bot chassis design to improve successful connection rate. (a) Additional permanent magnets are mounted on top of the module. In addition the main connection magnet has been replaced with a larger one. (b) A new servo horn designed to hold a larger magnet. (c) An alternate connector design with the electrical connectors mounted directly into the face of the chassis. (d) Overlapping jaws highlighted with red triangles and blue lines.

As evo-bot modules connect they can be subject to three types of misalignment, referred to as yaw, pitch and roll misalignment. Yaw misalignment does not effect floating, although it can cause problems with the electrical connections. Although the size of the misalignment is generally quite small, the fine balance of the module floating means that a small disturbance is enough to ground a structure. In addition, the effect of the misalignments can be amplified if a structure is composed of multiple modules. In order to decrease misalignment a series of alterations were made to the chassis design, shown in Figure 5.5.

One set of alterations was designed to increase the attractive force between modules. Two approaches were considered, adding additional permanent magnets (Figure 5.5a) and

increasing the size of the existing servo-mounted magnets. The latter also required a redesign of the servo horns (Figure 5.5b). In both cases the floating ability of connected structures was improved. Unfortunately the ability of connected modules to disconnect was greatly impaired, as the servos were not strong enough to overcome the larger magnetic force. To incorporate larger, more powerful servos would not only have required a full redesign of the module but would also increase its weight, further hampering the floating.

Each of the four electrical connectors on an evo-bot module are connected directly to the same PCB. When constructing a module each of the connectors has to be aligned correctly. If a connector protrudes too far from the chassis it can cause a pitch misalignment with a connected module. If the connector does not protrude far enough then a stable electrical connection may not be formed. The fact that all four connectors are physically coupled via the PCB makes it difficult to have all four correctly aligned at the same time. In order to correct this issue the chassis and PCB was redesigned so that the electrical connectors were mounted directly in each face (Figure 5.5c), and were attached to the PCB via cables. Although this made it easier to align each connector individually, it also increased the time taken to construct each module and would have required a redesign of both the module chassis and the PCB.

In order to correct for roll misalignment the chassis design was modified to include interlocking jaws (Figure 5.5d). Although they increased the floating ability of connected modules, the additional friction that they caused significantly hampered connection and disconnection between modules. For the remainder of this chapter the original module design from [26, 24] is used.

5.2.2 Environment

The experimental environment is shown in Figure 5.6. It consists of a square air table of side length 85 cm. The surface is a 10 mm thick acrylic sheet with holes drilled in a square pattern with a spacing of 10 mm. The main upward force is supplied by four industrial air blowers positioned below the table. The environment has foam boundaries of height 30 mm,

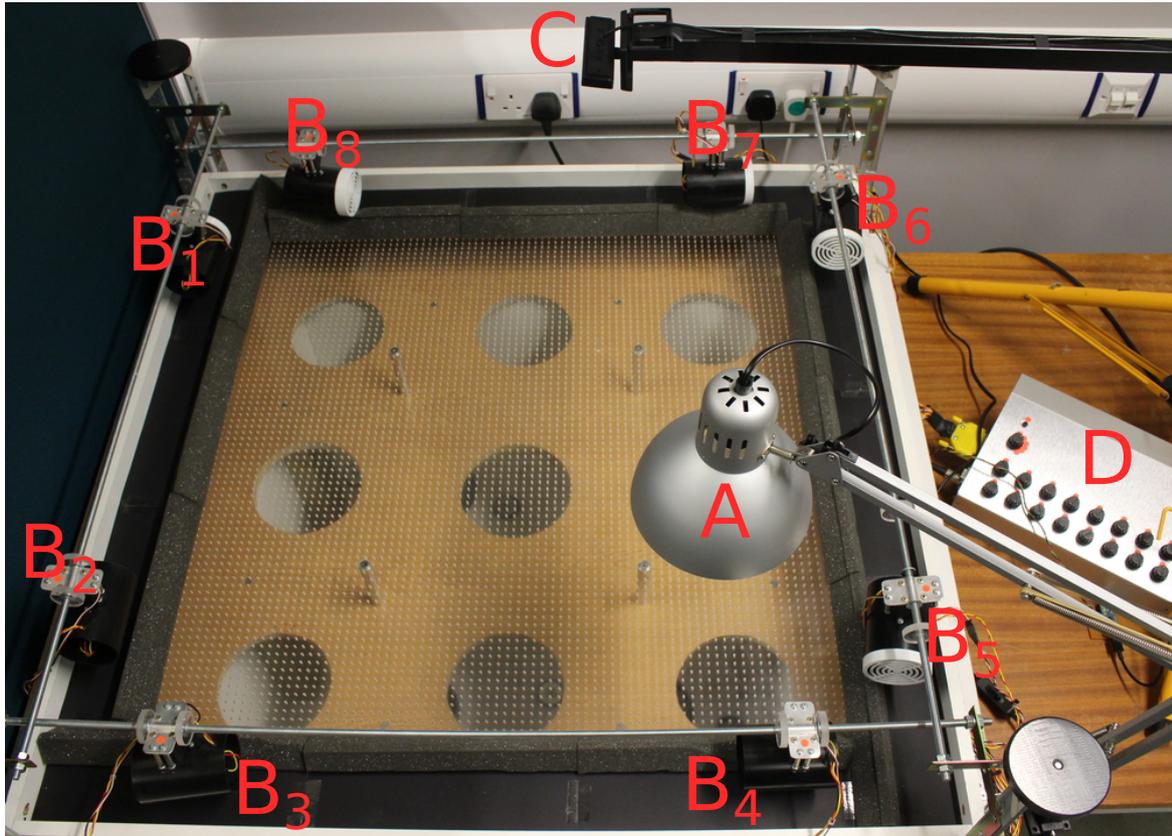


Figure 5.6 Overview of the experimental environment, an air table, including the overhead lamp (A), the side fans (B₁ to B₈), the overhead camera (C), and the fan control box (D).

which protect the modules from damage as they contact the borders. To improve the floating of the evo-bots several changes were made.

To improve the air flow and therefore increase the lift on the evo-bots the diameter of the holes in the air table surface was increased from 1 mm to 2.5 mm. In addition the total surface area was reduced from $100 \times 100 \text{ cm}^2$ to $85 \times 85 \text{ cm}^2$. These changes had a large positive effect upon the floating ability of the evo-bots.

Another change was the implementation of side fans around the perimeter of the air table. Eight side fans are used (two per side), mounted on rails around the table, pointing inwards. The fans serve a dual purpose: (i) to provide some horizontal force to help to move robots that are struggling to float, (ii) to increase the speed at which all the robots move around the air-table, and increase their coverage of the area, thereby increasing the number of physical interactions between robots. Each fan can rotate in the horizontal plane up to 180 degrees.

Algorithm 1 Stopping upon detection of light

Precondition: V_0 is a threshold value for light intensity measured by the solar cells

```
1: procedure STOPPING( $V$ )
2:   activate_anchor
3:   wait 3
4:   deactivate_anchor
5:   while true do
6:      $V \leftarrow$  solar_cell_output
7:     if  $V \geq V_0$  then
8:       activate_anchor
9:       return
10:    end if
11:  end while
12: end procedure
```

The rotation and speed of each fan can be set via a control box, which allows the fans to be activated in preset patterns throughout experiments. The fans were particularly effective for robots that became stuck near the perimeter of the environment. However robots near the center of the environment were relatively unaffected.

In addition to these changes an 18.5 W LED lamp with a 25 degrees beam angle was set up. This was used for the experiments in Section 5.3 unless otherwise noted. Mounted on a lamp stand it can be positioned arbitrarily over the air table. Additionally, an overhead camera was mounted above the table to track the modules throughout the experiments for post-analysis.

5.3 Experiments

In this section we report experiments that demonstrate the key features of the evo-bots. Videos of the experiments are available [25].

5.3.1 Light detection and stop-start motion control

For an evo-bot structure to maintain a constant supply of energy it must be able to locate an energy source. To do this, an evo-bot module must be capable of detecting whether it is

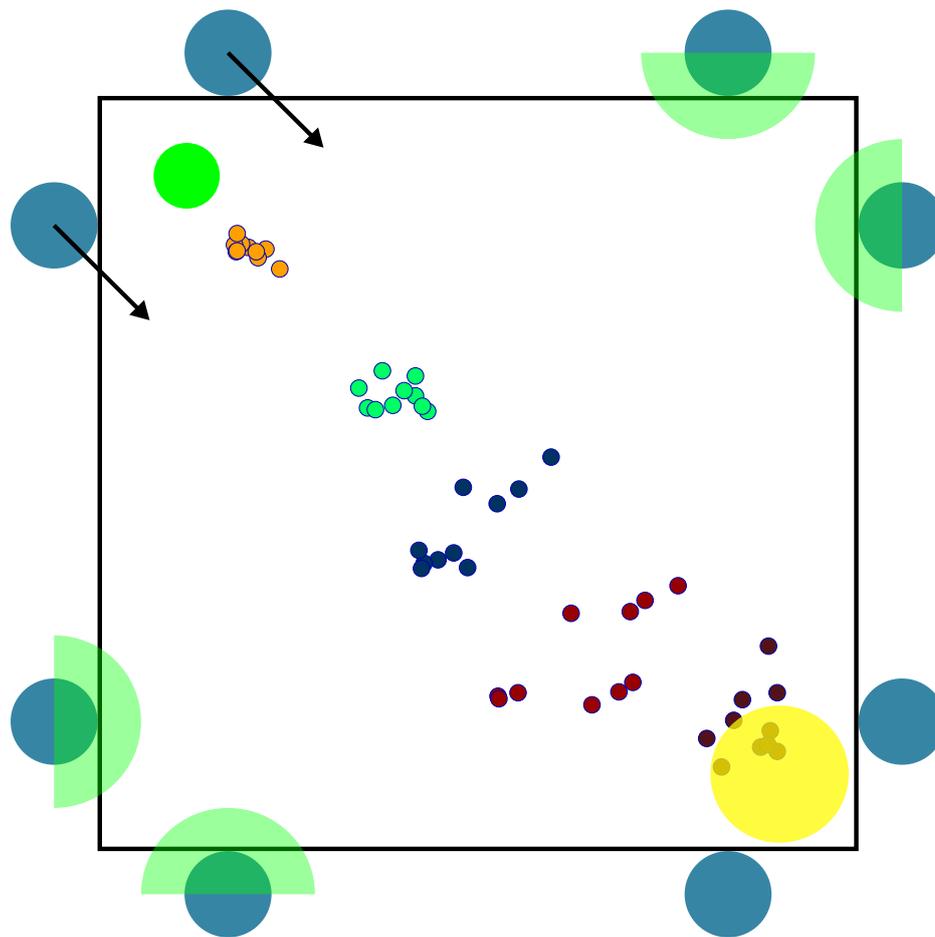


Figure 5.7 Results from the light detection and motion control experiment. The side fans are indicated by large blue circles. The yellow circle in the bottom right indicates the position of the light source, and the green circle in the top left the module starting position. The colored dots represent the final positions of the modules (10 trials per color), from top left to bottom right: orange—0.3 V, teal—0.4 V, blue—0.5 V, red—0.8 V, brown—2.0 V.

located beneath an energy (light) source, and of activating its anchor mechanism to stop and charge. We test this capability here. The specific set up is shown in Figure 5.7 (right). An evo-bot module was placed in the top left corner of the table. The light source was positioned 12 cm above the surface of the table in the bottom right corner, facing downwards vertically. For this experiment a bulb with a 360 degrees beam angle was used, in order to provide a smooth light gradient over the air table. The two side fans in the top left corner were pointed in the direction of the bottom right corner. The bottom left and top right side fans were set to oscillate over a 180 degrees range. The bottom right side fans were not used. The fans

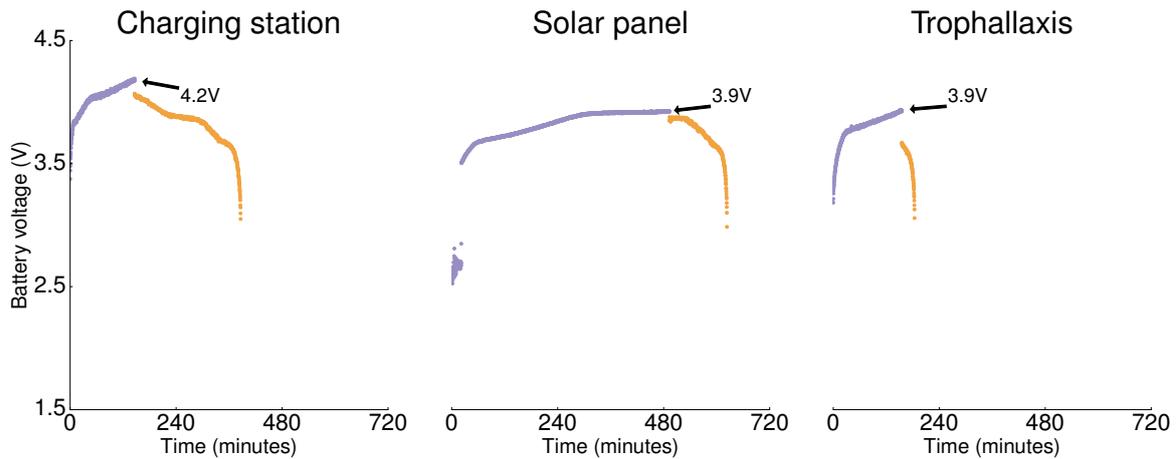


Figure 5.8 Graphs showing the charging and subsequent discharging of an evo-bot module. Charging via (left) charging station, (center) solar panel, or (right) another evo-bot module.

were set up in this manner in order to push the evo-bot module towards the light source. A pseudocode description of the algorithm used by the module is shown in Algorithm 1. The evo-bot module was given a certain threshold for the intensity of light incident on its solar panel (determined by measuring the voltage across the solar panels). Upon being powered on the module had its anchor activated and so did not move. After 3 seconds the anchor deactivated. The module was blown towards the light source and activated its anchor once the threshold was exceeded. Its final position was recorded. Trials lasted for 20 seconds or until the module stopped.

We conducted 50 trials: 10 trials each for 5 different voltage thresholds: 0.3 V, 0.4 V, 0.5 V, 0.8 V and 2.0 V. Figure 5.7 (right) shows the recorded positions at the end of each trial. All modules stopped successfully. As can be seen, the final positions of the module are stratified depending on their thresholds—modules with higher thresholds get closer to the light source before stopping. This confirms that the module interacts with its environment as intended.

5.3.2 Energy harvesting and sharing

We examine the ability of an evo-bot module to be charged via charging station (at 4.9 V), solar panel, or another module (trophallaxis). In each case, the module started fully discharged.

It was then charged for a certain period, and then discharged to evaluate its operating time. During the charging period only the micro-processor was running. During the discharging period, in addition, one servomotor was activated every 2 seconds in order to simulate the typical energy expenditure of a module during experimentation. This value was chosen based on the other experiments we conduct in this thesis. However less time-sensitive experiments may not require such frequent activity, reducing energy expenditure and increasing operating time.

The charging/discharging curves are shown in Figure 5.8. Via charging station the module charging period lasted until the battery was charged to 4.2V. This took 140 minutes, and provided 240 minutes of operating time. Due to the slow rate of charge via solar panel, charging up to 4.2 V was not feasible. Instead, the charging period was set to 500 minutes. This charged to 3.9 V and provided 120 minutes of operating time. For trophallactic charging, the donor module started fully charged (4.2 V). The receiving module was then charged by the donor module until the donor module was fully discharged. The receiving module charged to 3.9 V over 160 minutes, subsequently providing 30 minutes of operating time. An artificial ecosystem could see these charging methods combined. All organisms could feed from a limited number of charging stations, but demand would be high. Those organisms with *e*-modules would be able to locate and charge from a distribution of light sources.

5.3.3 Polymer formation

We examine the ability of the evo-bots to form linear polymers. A pseudocode description of the algorithm used by the module is shown in Algorithm 2. The algorithm worked as follows. Individual modules were able to form a connection on any of their faces. Once a module connected to another, it refused connections with new modules on faces orthogonal to its existing connection(s). In this way only linear polymers could form permanently (other configurations could form temporarily until incorrectly attached modules were rejected). Modules periodically closed and opened their connections in order to free themselves if connected in the wrong position. At the outset of each trial four evo-bot modules were positioned at the corners of a square of side length 34 cm, concentric with the air table. The

Algorithm 2 Self-assembly into linear polymers

```

1: procedure ASSEMBLY
2:   activate_anchor
3:    $c1 \leftarrow 0$                                 ▷ Index of the first face to connect
4:    $c2 \leftarrow 0$                                 ▷ Index of the second face to connect
5:   for  $i \leftarrow 1$  to 4 do                       ▷  $i$  is a cyclical index of the 4 module connectors
6:     open_connector  $i$ 
7:   end for
8:   wait 3
9:   deactivate_anchor
10:  while true do
11:    for  $i \leftarrow 1$  to 4 do
12:      if face_ $i$ _message_received then
13:        if  $c1 = 0$  then
14:           $c1 \leftarrow i$ 
15:          close_connector  $i + 1$ 
16:          close_connector  $i - 1$ 
17:        else if  $c1 = i + 2$  and  $c2 = 0$  then
18:           $c2 \leftarrow i$ 
19:        end if
20:      end if
21:      if  $c1 = 0$  or ( $c1 = i + 2$  and  $c2 = 0$ ) then
22:        refresh_connector  $i$ 
23:      end if
24:    end for
25:  end while
26: end procedure

```

fifth module was placed at the center of the table. The rotation of the modules was arbitrary. The modules were not given any specific type designation (*e*-module, *i*-module, *b*-module) for the purpose of this experiment.

A total of ten trials were performed. The average length of time needed to form a 5-module polymer was 188 seconds. The longest time taken was 407 seconds. The shortest time taken was 62 seconds. Figure 5.9 shows stills from an example trial.

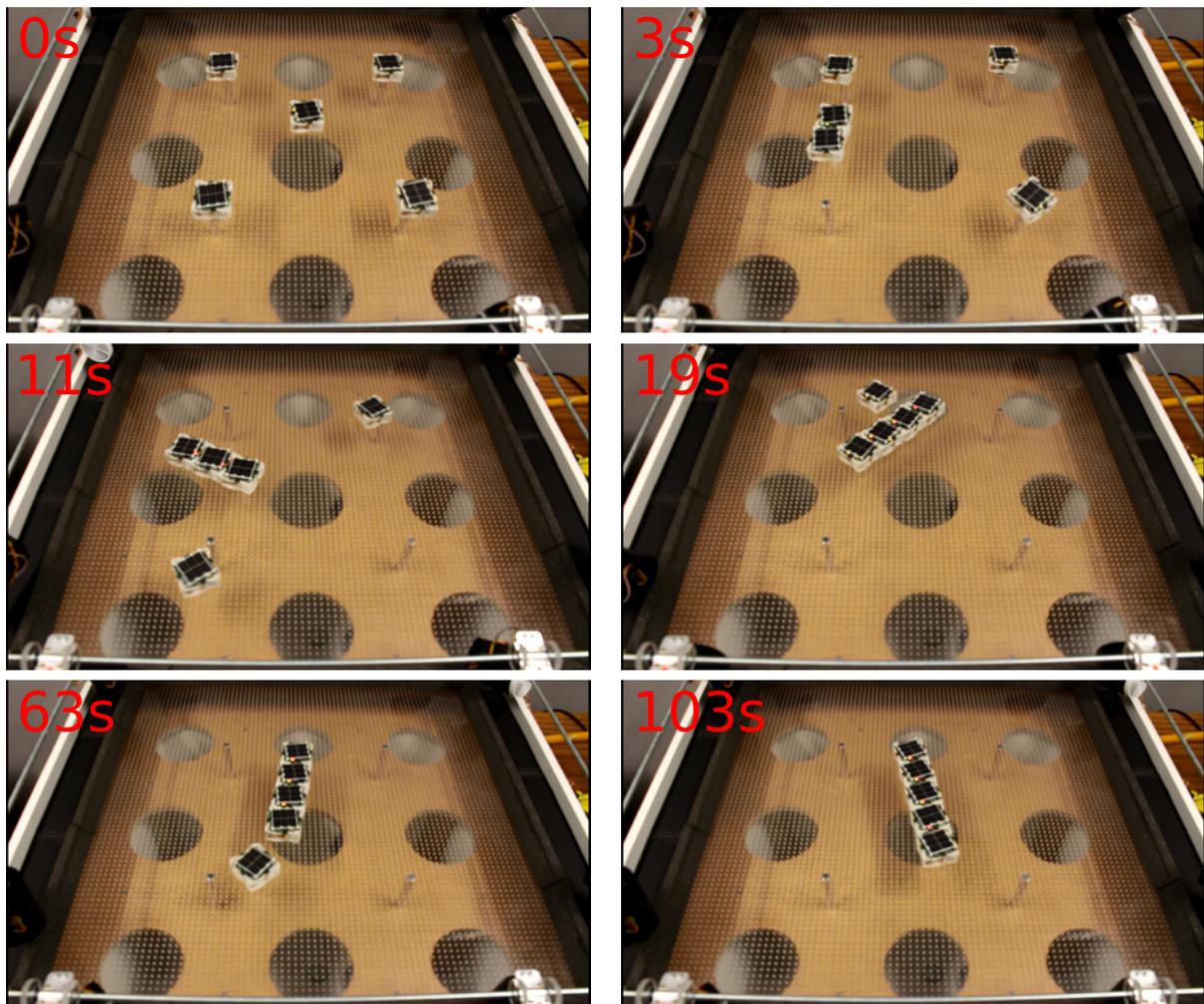


Figure 5.9 A sequence of stills taken from a video of five evo-bot modules forming a linear polymer.

5.4 Summary

This chapter described modifications made to the evo-bot system to improve the ability of the modules to connect and float, and therefore prepare the system for experimentation. The experimental environment was modified to increase vertical air flow. In addition, side fans were installed to increase the mixing of the robots and to help counteract the issues the robots had with floating. Multiple modifications were made to the module design to try and improve connection success and floating. This included re-siting the connectors, adding stronger

magnets and adding interlocking jaws. Although these modifications helped to improve the floating, due to their drawbacks the original module design was used for experiments.

Experiments were performed to demonstrate the individual functions of the evo-bots. The stop-start motion system was shown to allow modules to interact with their environment in a simple manner. The ability of the evo-bots to detect, harvest and share energy was shown. Finally, self-assembly into simple linear structures was performed. These represent the first steps towards realizing the evo-bot system as a physically embodied artificial life system.

Chapter 6

Conclusion

The objectives of this thesis concerned the investigation of modular systems which operate in fluidic or microgravity environments. Of particular concern was the way in which such environments could allow the use of novel propulsion mechanisms. Two such systems were investigated. The MFP system, using a novel fluid routing mechanism, was developed from concept through to hardware prototype. Theory, simulations and experiments confirmed that the system could be successful in fluidic environments. The evo-bot system, which also has the ability to self-reconfigure, was shown to be capable of using a novel stop-start propulsion mechanism to interact with its environment.

Chapter 3 introduced the MFP concept. Each MFP module contains fluid actuators, a reservoir, and fluid routing elements. Connected modules form a fluid network which can self-propel by routing fluid through itself. Two potential implementations of the concept were discussed: MHP robots operate in liquid environments, using the environmental medium as a routing agent. MPP robots can operate in microgravity environments, without the need for any atmosphere, by using an internal store of fuel. We demonstrated that 2-D convex MFP systems can generate a wide range of unique torques and forces. A decentralized controller (dec) was adapted from a swarm robotics controller in the literature. Using this controller it was proved that under certain conditions an MHP robot will move towards a goal point up to a minimum distance. Furthermore, convex MHP robots will move towards a goal

point until touching it. Additionally, two variants of the controller were developed to handle communication (*dec-com*) and obstacle avoidance (*dec-obs*).

Chapter 4 aimed to validate the MFP concept via simulation and experiment. A 3-D physics simulator was used to investigate MHP robots of cubic or random morphology, consisting of 125 modules. In both an open environment and an environment with obstacles, the decentralized controllers were shown to be successful. The *dec-com* controller reduced the rate of energy consumption while maintaining the same time required to reach the goal. The *dec-obs* controller did not decrease the time taken for robots to reach the goal, but did reduce the rate at which robots collided with obstacles. When subjected to misaligned or failed actuators, robots with higher resolution tended to show better success rate and lower variance in time taken. The decentralized controller was also compared to a centralized controller taken from the literature. Although the centralized controller produced significantly faster completion times, the decentralized controller was competitive. To further validate the MFP concept a hardware prototype was developed. Consisting of six modules, the prototype can be configured in 2-D and floats on the surface of water. Using fluid pumps, light sensors and an onboard microcontroller, the prototype was used to carry out the directed locomotion task, using a light source as the goal. Experiments showed that a robot consisting of four modules would always successfully reach the goal in any possible configuration.

Chapter 5 details the work done on the evo-bot system. This included preparing the experimental environment (an air table) and refining the module design. In order to improve floating and connection success rate of the evo-bots the air table was reduced in size and the hole diameter was increased. A number of alterations to the module chassis were attempted including overlapping flanged edges, different sizes and placement of magnets, and the physical decoupling of the four connectors. Ultimately the original chassis design was used for experiments. The experiments undertaken include the first demonstration of the novel stop-start propulsion mechanism of the evo-bots. Individual modules were shown to be able to respond to an environmental stimulus in the form of light. Additionally, energy harvesting and transfer between modules was demonstrated, along with the self-assembly of linear polymers.

6.1 Discussion

This section discusses the achievements made in pursuing the thesis objectives, and the limitations of the work.

6.1.1 Achievements

- **Novel propulsion via fluid routing** - The MFP concept takes advantage of the nature of its intended environment to self-propel via the internal routing of fluid. This means that the modules require no external actuators which could potentially hamper reconfiguration, and allows for a 3-D lattice structure that does not block the flow of the routing fluid and keeps the module design compact. This allows the system to produce a wide range of fast and effective motions while having a high degree of reconfigurability. In addition the simple, binary actuators allow for future versions of the modules to be scaled down in size.
- **Decentralized motion control** - A swarm controller taken from the literature was adapted and repurposed for the MFP system. In its original form the controller directed a swarm of robots to push a solid object towards a goal [10]. With the MFP system the robot itself acts as the obstacle being pushed by the actuators. The controller is decentralized and requires no communication, run-time memory or arithmetic computation. This makes the controller, along with the binary sensors and actuators, very simple, and means that as the hardware is miniaturized the controller can continue to be applied even if the available processing and memory capabilities are very low. The decentralized nature of the controller also means that it scales well across a large number of modules.
- **Theoretical analysis** - The range of torques and forces that a 2-D MFP system can generate was analyzed, showing that despite using binary actuators MFP robots can develop a wide range of possible motions. This suggests that such systems could be used for tasks which require a high degree of maneuverability. The ability of an MFP

robot to complete a directed locomotion task is proved. Although the proof is made under relatively tight conditions it provides a theoretical grounding for the system.

- **Simulation of a 3-D system** - The use of simulations allows for analysis of the MFP concept across a wider array of environments and conditions than with the theoretical proof. In addition, robots with large numbers of modules and a wide variety of configurations were tested. Every variant of the decentralized controller was analyzed, demonstrating that the robot could complete the task successfully in a wide variety of configurations, that the *dec-com* controller saved energy without compromising on speed, and that the *dec-obs* controller reduce the rate of collisions with obstacles. Testing the decentralized controller against a centralized one taken from the literature [20] was important to show that, while more complex controllers do provide better solutions, the decentralized controller is competitive. Furthermore, simulations with robots subjected to thruster errors demonstrated that robots with a higher modular resolution tend to be more resilient to failure.
- **Experiments with a 2-D hardware prototype** - Although simulations can study a variety of scenarios the assumed reality gap means that hardware experiments are an important tool for validation. The development of a set of prototype modules provided insight into the challenges that would be faced to construct a full version of the concept. Experiments demonstrated that the MFP concept could be successfully applied in reality.
- **Evo-bot stop-start motion demonstration** - A stop-start propulsion mechanism was demonstrated experimentally for the first time. Using just a single binary actuator an evo-bot module can choose to move about randomly or remain fixed in position. This very simple form of motion control allows evo-bots to respond to stimuli in their environment.

6.1.2 Limitations

- **No MFP self-reconfiguration** - Self-reconfiguration can greatly increase the utility of a modular system, however the MFP system must be manually reconfigured. This is due to the thesis focus being on providing effective propulsion while being in (and having the capability to be in) a wide variety of configurations. Furthermore many self-reconfigurable systems are already present in the literature. In order to improve the real world utility of the MFP system and make it into a more complete concept, self-reconfiguration could be considered.
- **2-D MFP prototype** - Although a 3-D MFP system was analyzed in simulation, the hardware prototype was only capable of moving and being configured in 2-D. This was chosen in order to avoid the cost and complexity of designing and building a submersible system which would introduce concerns such as full waterproofing. In addition, moving from square modules in 2-D to cubic modules in 3-D increases the number of connectors, actuators and sensors by 50%, while potentially also requiring a larger microprocessor. Although these concerns make a 3-D system more complex and costly to build, the core fluid routing concept should transition straightforwardly.
- **Limited number of MFP prototype modules** - The MFP prototype consisted only of six modules, and rigorous experiments were only performed using four. The limited size of the experimental environment and the finite budget for developing the system meant that it was not realistic to construct more modules. However, to further investigate the scalability of the system, future experiments should be conducted with a larger number of modules.
- **Lack of task diversity** - Despite modular robots promising ubiquity of problem solving the MFP system is only tested with one task, that of directed locomotion. A real world application would likely require the solution of multiple, complex tasks, such as the ability to navigate along a passageway or move along paths with multiple nodes. Additionally, while two slightly different environments are tested (one with-

out obstacles and one with obstacles) this could be extended to model more varied situations.

- **Incomplete model of MFP dynamics** - Throughout most of the theoretical analysis of the MFP system in Chapter 3 the internal fluid flow and actuators of the MFP system are neglected and each active actuator on an external face is assumed to produce a constant force acting on the robot. A more accurate model should account for the internal routing of fluid through the structure, and allow the effects of active internal actuators to be predicted. This could be used to inform the design of future hardware implementations.
- **No full evo-bot experiments** - The original purpose of the evo-bot system was to serve as an ecosystem of physically embodied virtual organisms. While this is alluded to within this thesis, the experiments performed only go so far as to validate the individual functions of the evo-bot system, namely the stop-start motion, light detection, energy harvesting and sharing, and basic self-assembly. The environments are simple and static, using a only a single fixed light source. Demonstrating the ability of the evo-bots to self-replicate linear polymers, as well as to sustain themselves indefinitely by harvesting energy from more dynamic experimental environments would go further to realizing the original purpose of the system.

6.2 Future work

In view of the limitations discussed above, there are several areas in which the work in this thesis could be furthered. An ideal way to validate the MFP concept is via physical experiments and so the hardware is one possible focus for future work. Developing a system that can both move and self-reconfigure in 3-D is the ultimate goal. This would require a redesign of the chassis to handle six connectors/actuators/sensors. A watertight compartment inside the structure would be needed to accommodate the battery and control circuitry. Additionally, some form of buoyancy control could be required. To incorporate self-

reconfiguration an active connection mechanism would be required instead of the permanent magnets that are currently used. This could be achieved using electro-permanent magnets, which only require energy to transition between latched or unlatched states. Alternatively, the fluid routing actuators themselves could be used to disconnect modules connected by permanent magnets. This would require careful consideration of whether the structural integrity of the robot as a whole was strong enough to withstand normal propulsion. In either case the actuators could be used to maneuver reconfiguring modules between connection points, potentially in a similar manner to the M-Blocks [78]. Given the simple binary nature of the hardware and control, miniaturization could be another area for improvement. As well as moving towards the ideal of programmable matter, and opening up environments with tighter space constraints, miniaturization would also allow robots with larger numbers of modules to be tested without needing a larger experimental environment.

Another angle for future improvement is with regards to the MFP control. There is more work that could be done on motion control. Thus far only translation has been explicitly controlled. Developing pose controllers could allow for docking between modules, facilitating higher level controlled self-assembly or self-reconfiguration. In addition, future controllers could be designed to handle more complex environments and paths with multiple nodes. This could also require the use of more elaborate sensors.

As described in the previous section there is no detailed theoretical model of the MFP dynamics. Developing such a model could inform the future design of both hardware and control and help create faster and more efficient robots.

Although the individual functions of the evo-bots have been validated the system still has a long way to go to fulfil its original purpose. Further experiments could be conducted to validate the capability of the robots to perform linear polymer replication, and to be able to harvest enough energy to survive indefinitely in the experimental environment. In order to achieve this both the robots and the environment need refining. The robot floating could be improved, and either a stronger light source and more efficient solar cells, or a different charging method used. In addition, to produce a stable ecosystem of evolving robots, a larger number of modules (and a larger environment) may be required. Simulation work could be

conducted to inform on the number of modules needed. Further to this, the system could be expanded to incorporate more complex interactions between robots. Additional types of modules could be implemented. For example, by exploiting the trophallactic capabilities of the system, “predator” and “defence” modules might allow robots to harvest energy from one another and protect themselves against such harvesting respectively.

Appendix A

Static torque analysis of a 2-D MFP robot

In Section 3.2 we derived the number of unique torques and forces that a 2-D convex MFP robot can develop. We showed the derivation for the number of torques for the case $l_1 \in 2\mathbb{Z}^+ - 1$ and $l_2 \in 2\mathbb{Z}^+ - 1$ (both side lengths are odd). Here we show the derivation for the other two cases (even-even side lengths and odd-even side lengths).

Two even side lengths

Consider a 2-D convex robot with even side lengths, $l_1 \in 2\mathbb{Z}^+$ and $l_2 \in 2\mathbb{Z}^+$. Torque is maximized when

$$a_{ij} = \begin{cases} 1, & |\mathbf{v}_{ij} \times \mathbf{u}_j| < 0 \wedge c_{ij} = 0; \\ 0, & \text{otherwise.} \end{cases} \quad (\text{A.1})$$

From Equation 3.9 this gives a net torque of:

$$\begin{aligned} \tau_M(l_1, l_2) &= 2\left(\frac{1}{2} + \frac{3}{2} + \dots + \frac{l_1 + 1}{2}\right) \\ &\quad + 2\left(\frac{1}{2} + \frac{3}{2} + \dots + \frac{l_2 + 1}{2}\right) \\ &= \frac{l_1^2}{4} + \frac{l_2^2}{4}. \end{aligned} \quad (\text{A.2})$$

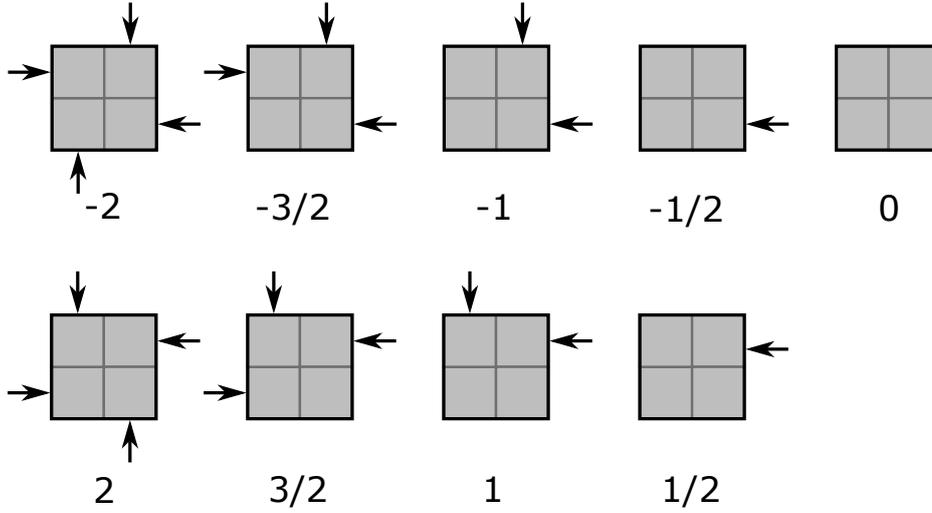


Figure A.1 All unique torques of a 2 by 2 robot.

Similarly, the minimum achievable torque is $-\tau_M(l_1, l_2)$.

Lemma 7. Every half-integer torque $\tau \in \{-\tau_M(2, l_2), \dots, \tau_M(2, l_2)\}$ can be produced for the case that $l_1 = 2$ and $l_2 \in 2\mathbb{Z}^+$.

Proof. We prove by induction that this holds.

Base case: Assume $l_2 = 2$. The robot consists of four modules. $\tau_M(2, 2) = 2$, therefore we must show that every half-integer $\in \{2, \dots, 2\}$ can be produced. Figure A.1 shows that all of these torques can be produced.

Inductive step: We assume that the statement holds true for any $l_2 = 2, 4, \dots, k$, for some $k \in 2\mathbb{Z}^+$. Adding modules symmetrically about the center of mass does not change the positions of existing modules (and actuators). Therefore any torque that can be produced for $l_2 = k$ can still be produced when $l_2 = k + 2$. That is to say, all half-integer torque values $\tau \in \{-\tau_M(2, k), \dots, \tau_M(2, k)\}$ can be produced.

Now consider the actuator firing configuration which produces the maximum torque $\tau_M(2, k + 2) = 1 + \frac{(k+2)^2}{4}$. This torque can be reduced by any $\Delta \in \{\frac{1}{2}, \frac{3}{2}, \dots, \frac{k+1}{2}\}$ by deactivating a single actuator a_{ij} with $\mathbf{h}_{ij} \in \{(1, \frac{1}{2}), (1, \frac{3}{2}), \dots, (1, \frac{k+1}{2})\}$. If, in addition, the actuator a_{ij} with $\mathbf{h}_{ij} = (-1, -\frac{1}{2})$ is deactivated the maximum torque can be reduced by any $\Delta \in \{1, 2, \dots, \frac{k+2}{2}\}$.

Again from the configuration which produces the maximum torque, by deactivating a single actuator a_{ij} with $\mathbf{h}_{ij} \in \{(1, \frac{1}{2}), (1, \frac{3}{2}), \dots, (1, \frac{k+1}{2})\}$ along with deactivating the actuator a_{ij} with $\mathbf{h}_{ij} = (-1, -\frac{k+1}{2})$ the torque can be reduced by any $\Delta \in \{\frac{1}{2} + \frac{k+1}{2}, \frac{3}{2} + \frac{k+1}{2}, \dots, k+1\}$. If, in addition, the actuator a_{ij} with $h_{ij} = (-1, -\frac{1}{2})$ is deactivated the maximum torque can be reduced by any $\Delta \in \{1 + \frac{k+1}{2}, 2 + \frac{k+1}{2}, \dots, k + \frac{3}{2}\}$.

Given that the difference between $\tau_M(2, k+2)$ and $\tau_M(2, k)$ is $k+1$, every half-integer torque $\tau \in \{\tau_M(2, k+2) - \tau_M(2, k), \dots, \tau_M(2, k+2)\}$ is possible. By symmetry, every half-integer torque $\tau \in \{-\tau_M(2, k+2), \dots, -\tau_M(2, k+2) + \tau_M(2, k)\}$ is possible. Therefore every half-integer torque $\tau \in \{-\tau_M(2, k+2), \dots, \tau_M(2, k+2)\}$ is possible and the inductive step is complete. \square

Theorem 5. The number of unique net torques that can be generated by a 2-D convex MFP robot with $l_1 = 2$ and $l_2 \in 2\mathbb{Z}^+$ is $N_\tau(2, l_2) = l_2^2 + 5$.

Proof. The maximum and minimum possible torque is $\tau_M(2, l_2)$ and $-\tau_M(2, l_2)$ respectively. From Lemma 7 every half-integer torque between these bounds can be produced. Given that each actuator can only apply either an integer or a half-integer torque, no other net torques can be produced. The number of unique torques is therefore

$$N_\tau(2, l_2) = 4\tau_M(2, l_2) + 1 = l_2^2 + 5. \quad (\text{A.3})$$

\square

Lemma 8. Every half-integer torque $\tau \in \{-\tau_M(l_1, l_2), \dots, \tau_M(l_1, l_2)\}$ can be produced for the case $l_1 \in 2\mathbb{Z}^+$ and $l_2 \in 2\mathbb{Z}^+$.

Proof. We prove by induction that this holds.

Base case: Assume $l_1 = 2$. All torques $\tau \in \{-\tau_M(2, l_2), \dots, \tau_M(2, l_2)\}$ are possible as shown in Lemma 7.

Inductive step: We assume that the statement holds true for any $l_1 = 2, 4, \dots, k$ for some $k \in 2\mathbb{Z}^+$. Any torque that can be produced for $l_1 = k$ can still be produced when $l_1 = k+2$. That is to say, every half-integer torque $\tau \in \{-\tau_M(k, l_2), \dots, \tau_M(k, l_2)\}$ can be

produced. Consider the actuator firing configuration which produces the maximum torque of $\tau_M(k+2, l_2) = \frac{(k+2)^2 + l_2^2 - 2}{4}$. This torque can be reduced by any $\Delta \in \{\frac{1}{2}, \frac{3}{2}, \dots, \frac{k+1}{2}\}$ by deactivating a single actuator a_{ij} with position $\mathbf{h}_{ij} \in \{(-\frac{1}{2}, \frac{l_2}{2}), (-\frac{3}{2}, \frac{l_2}{2}), \dots, (-\frac{k+1}{2}, \frac{l_2}{2})\}$. If, in addition, the actuator a_{ij} with $h_{ij} = (\frac{1}{2}, -\frac{l_2}{2})$ is deactivated the maximum torque can be reduced by any $\Delta \in \{1, 2, \dots, \frac{k+2}{2}\}$.

Again from the configuration which produces the maximum torque, by deactivating a single actuator a_{ij} with $\mathbf{h}_{ij} \in \{(-\frac{1}{2}, \frac{l_2}{2}), (-\frac{3}{2}, \frac{l_2}{2}), \dots, (-\frac{k+1}{2}, \frac{l_2}{2})\}$ along with deactivating the actuator a_{ij} with $\mathbf{h}_{ij} = (\frac{k+1}{2}, -\frac{l_2}{2})$ the torque can be reduced by any $\Delta \in \{\frac{1}{2} + \frac{k+1}{2}, \frac{3}{2} + \frac{k+1}{2}, \dots, k+1\}$. If, in addition, the actuator a_{ij} with $h_{ij} = (\frac{1}{2}, -\frac{l_2}{2})$ is deactivated the maximum torque can be reduced by any $\Delta \in \{1 + \frac{k+1}{2}, 2 + \frac{k+1}{2}, \dots, k + \frac{3}{2}\}$.

Given that the difference between $\tau_M(k+2, l_2)$ and $\tau_M(k, l_2)$ is $k+1$, every half-integer torque $\tau \in \{\tau_M(k+2, l_2) - \tau_M(k, l_2), \dots, \tau_M(k+2, l_2)\}$ is possible. By symmetry, every half-integer torque $\tau \in \{-\tau_M(k+2, l_2), \dots, -\tau_M(k+2, l_2) + \tau_M(k, l_2)\}$ is possible. Therefore every half-integer torque $\tau \in \{-\tau_M(k+2, l_2), \dots, \tau_M(k+2, l_2)\}$ is possible and the inductive step is complete. \square

Theorem 6. The number of unique net torques that can be generated by a 2-D convex MFP robot with $l_1 \in 2\mathbb{Z}^+$ and $l_2 \in 2\mathbb{Z}^+$ is $N_{\tau_1} = l_1^2 + l_2^2 + 1$.

Proof. The maximum and minimum possible torque is $\tau_M(l_1, l_2)$ and $-\tau_M(l_1, l_2)$ respectively. From Lemma 8 every half-integer torque between these bounds can be produced. Given that each actuator can only apply an integer or a half-integer torque, no other net torques can be produced. The number of unique torques is therefore

$$N_{\tau_1} = 4\tau_M(l_1, l_2) + 1 = l_1^2 + l_2^2 + 1. \quad (\text{A.4})$$

\square

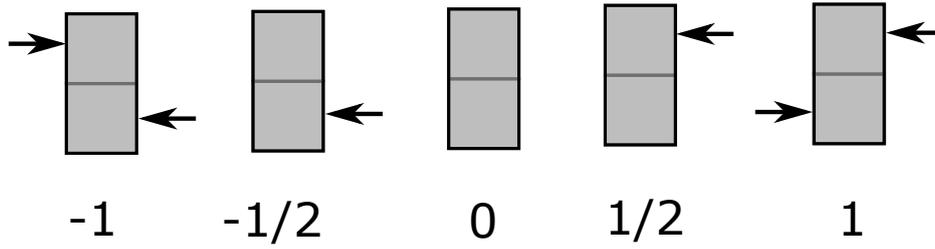


Figure A.2 All unique torques of a 1 by 2 robot.

One odd and one even side length

Consider a 2-D convex robot with even side lengths, $l_1 \in 2\mathbb{Z}^+ - 1$ and $l_2 \in 2\mathbb{Z}^+$. Torque is maximized when

$$a_{ij} = \begin{cases} 1, & |\mathbf{v}_{ij} \times \mathbf{u}_j| < 0 \wedge c_{ij} = 0; \\ 0, & \text{otherwise.} \end{cases} \quad (\text{A.5})$$

From Equation 3.9 this gives a net torque of:

$$\begin{aligned} \tau_M(l_1, l_2) &= 2(1 + 2 + \dots + \frac{l_1 - 1}{2}) \\ &\quad + 2(\frac{1}{2} + \frac{3}{2} + \dots + \frac{l_2 + 1}{2}) \\ &= \frac{l_1^2 - 1}{4} + \frac{l_2^2}{4}. \end{aligned} \quad (\text{A.6})$$

Similarly, the minimum achievable torque is $-\tau_M(l_1, l_2)$.

Lemma 9. Every half-integer torque $\tau \in \{-\tau_M(2, l_2), \dots, \tau_M(2, l_2)\}$ can be produced for the case that $l_1 = 2$ and $l_2 \in 2\mathbb{Z}^+$.

Proof. We prove by induction that this holds.

Base case: Assume $l_2 = 2$. The robot consists of four modules. $\tau_M(1, 2) = 1$, therefore we must show that every half-integer $\in \{1, \dots, 1\}$ can be produced. Figure A.2 shows that all of these torques can be produced.

Inductive step: We assume that the statement holds true for any $l_2 = 2, 4, \dots, k$, for some $k \in 2\mathbb{Z}^+$. Adding modules symmetrically about the center of mass does not change the

positions of existing modules (and actuators). Therefore any torque that can be produced for $l_2 = k$ can still be produced when $l_2 = k + 2$. That is to say, every half-integer torque $\tau \in \{-\tau_M(1, k), \dots, \tau_M(1, k)\}$ can be produced.

Now consider the actuator firing configuration which produces the maximum torque $\tau_M(1, k + 2) = \frac{(k+2)^2}{4}$. This torque can be reduced by any $\Delta \in \{\frac{1}{2}, \frac{3}{2}, \dots, \frac{k+1}{2}\}$ by deactivating a single actuator a_{ij} with $\mathbf{h}_{ij} \in \{(\frac{1}{2}, \frac{1}{2}), (\frac{1}{2}, \frac{3}{2}), \dots, (\frac{1}{2}, \frac{k+1}{2})\}$. If, in addition, the actuator a_{ij} with $h_{ij} = (-\frac{1}{2}, -\frac{1}{2})$ is deactivated the maximum torque can be reduced by any $\Delta \in \{1, 2, \dots, \frac{k+2}{2}\}$.

Again from the configuration which produces the maximum torque, by deactivating a single actuator a_{ij} with $\mathbf{h}_{ij} \in \{(\frac{1}{2}, \frac{1}{2}), (\frac{1}{2}, \frac{3}{2}), \dots, (\frac{1}{2}, \frac{k+1}{2})\}$ along with deactivating the actuator a_{ij} with $\mathbf{h}_{ij} = (-\frac{1}{2}, -\frac{k+1}{2})$ the torque can be reduced by any $\Delta \in \{\frac{1}{2} + \frac{k+1}{2}, \frac{3}{2} + \frac{k+1}{2}, \dots, k + 1\}$. If, in addition, the actuator a_{ij} with $h_{ij} = (-\frac{1}{2}, -\frac{1}{2})$ is deactivated the maximum torque can be reduced by any $\Delta \in \{1 + \frac{k+1}{2}, 2 + \frac{k+1}{2}, \dots, k + \frac{3}{2}\}$.

Given that the difference between $\tau_M(1, k + 2)$ and $\tau_M(1, k)$ is $k + 1$, every half-integer torque $\tau \in \{\tau_M(1, k + 2) - \tau_M(1, k), \dots, \tau_M(1, k + 2)\}$ is possible. By symmetry, every half-integer torque $\tau \in \{-\tau_M(1, k + 2), \dots, -\tau_M(1, k + 2) + \tau_M(2, k)\}$ is possible. Therefore every half-integer torque $\tau \in \{-\tau_M(1, k + 2), \dots, \tau_M(1, k + 2)\}$ is possible and the inductive step is complete. \square

Theorem 7. The number of unique net torques that can be generated by a 2-D convex MFP robot with $l_1 = 1$ and $l_2 \in 2\mathbb{Z}^+$ is $N_\tau(1, l_2) = l_2^2 + 1$.

Proof. The maximum and minimum possible torque is $\tau_M(1, l_2)$ and $-\tau_M(1, l_2)$ respectively. From Lemma 9 every half-integer torque between these bounds can be produced. Given that each actuator can only apply either an integer or a half-integer torque, no other net torques can be produced. The number of unique torques is therefore

$$N_\tau(1, l_2) = 4\tau_M(1, l_2) + 1 = l_2^2 + 1. \quad (\text{A.7})$$

\square

Lemma 10. Every half-integer torque $\tau \in \{-\tau_M(l_1, l_2), \dots, \tau_M(l_1, l_2)\}$ can be produced for the case $l_1 \in 2\mathbb{Z}^+ - 1$ and $l_2 \in 2\mathbb{Z}^+ - 1$.

Proof. We prove by induction that this holds.

Base case: Assume $l_1 = 1$. All torques $\tau \in \{-\tau_M(1, l_2), \dots, \tau_M(1, l_2)\}$ are possible as shown in Lemma 9.

Inductive step: We assume that the statement holds true for any $l_1 = 1, 3, \dots, k$ for some $k \in 2\mathbb{Z}^+ - 1$. Any torque that can be produced for $l_1 = k$ can still be produced when $l_1 = k + 2$. That is to say, all half-integer torque values $\tau \in \{-\tau_M(k, l_2), \dots, \tau_M(k, l_2)\}$ can be produced. Consider the actuator firing configuration which produces the maximum torque of $\tau_M(k + 2, l_2) = \frac{(k+2)^2 + l_2^2 - 1}{4}$. This torque can be reduced by any $\Delta \in \{\frac{1}{2}, \frac{3}{2}, \dots, \frac{k+1}{2}\}$ by deactivating a single actuator a_{ij} with position $\mathbf{h}_{ij} \in \{(-1, \frac{l_2}{2}), (-2, \frac{l_2}{2}), \dots, (-\frac{k+1}{2}, \frac{l_2}{2})\}$. If, in addition, the actuator a_{ij} with $h_{ij} = (1, -\frac{l_2}{2})$ is deactivated the maximum torque can be reduced by any $\Delta \in \{1, 2, \dots, \frac{k+2}{2}\}$.

Again from the configuration which produces the maximum torque, by deactivating a single actuator a_{ij} with $\mathbf{h}_{ij} \in \{(-1, \frac{l_2}{2}), (-2, \frac{l_2}{2}), \dots, (-\frac{k+1}{2}, \frac{l_2}{2})\}$ along with deactivating the actuator a_{ij} with $\mathbf{h}_{ij} = (\frac{k+1}{2}, -\frac{l_2}{2})$ the torque can be reduced by any $\Delta \in \{\frac{1}{2} + \frac{k+1}{2}, \frac{3}{2} + \frac{k+1}{2}, \dots, k + 1\}$. If, in addition, the actuator a_{ij} with $h_{ij} = (1, -\frac{l_2}{2})$ is deactivated the maximum torque can be reduced by any $\Delta \in \{1 + \frac{k+1}{2}, 2 + \frac{k+1}{2}, \dots, k + \frac{3}{2}\}$.

Given that the difference between $\tau_M(k + 2, l_2)$ and $\tau_M(k, l_2)$ is $k + 1$, every half-integer torque $\tau \in \{\tau_M(k + 2, l_2) - \tau_M(k, l_2), \dots, \tau_M(k + 2, l_2)\}$ is possible. By symmetry, every half-integer $\tau \in \{-\tau_M(k + 2, l_2), \dots, -\tau_M(k + 2, l_2) + \tau_M(k, l_2)\}$ is possible. Therefore every half-integer torque $\tau \in \{-\tau_M(k + 2, l_2), \dots, \tau_M(k + 2, l_2)\}$ is possible and the inductive step is complete. \square

Theorem 8. The number of unique net torques that can be generated by a 2-D convex MFP robot with $l_1 \in 2\mathbb{Z}^+ - 1$ and $l_2 \in 2\mathbb{Z}^+ - 1$ is $N_{\tau_1} = l_1^2 + l_2^2$.

Proof. The maximum and minimum possible torque is $\tau_M(l_1, l_2)$ and $-\tau_M(l_1, l_2)$ respectively. From Lemma 10 every half-integer torque between these bounds can be produced. Given that each actuator can only apply an integer or a half-integer torque, no other net torques can

be produced. The number of unique torques is therefore

$$N_{\tau_1} = 4\tau_M(l_1, l_2) + 1 = l_1^2 + l_2^2. \quad (\text{A.8})$$

□

Appendix B

Outline of a more detailed model of fluid flow within an MFP robot

In Section 3.2 only the activation of external actuators is considered. Additionally, all active external actuators are considered to apply the same, constant force to the robot, situated at the actuator position. The purpose of this section is to present a more detailed model of fluid flow within an MFP robot, providing flow rates throughout the network as a function of the activation state of all actuators (both internal and external). For simplicity, and in alignment with Section 3.2, a 2-D system is considered, however the model could be extended directly to 3-D.

B.1 Flow rate and pressure from conservation laws

The model calculates the volumetric flow rates through each routing pipe of an MHP robot. This is done by invoking the laws of conservation of energy and conservation of mass throughout the fluid network.

In order to satisfy the law of mass conservation, the sum of the volumetric flow rates at each internal module reservoir must be zero. This relation holds at each module within the robot.

$$\sum_{i=1}^4 Q_i = 0,$$

where the Q_i are flow rates through each pipe of a given module. The law of mass conservation therefore gives rise to N equations, where N is the number of modules in the system. The equations contain $4N - C$ variables, where C is the total number of connections between modules within the robot.

To satisfy the conservation of energy law, the change in pressure around a closed loop must be zero. This means that the difference in pressure between two neighbouring module reservoirs must be equal to any pressure increase caused by the presence of an active pump, minus pressure losses due to the fluid flowing through a pipe.

$$P_A - P_B = H_{AB} - h_{AB},$$

where P_A and P_B are the pressures at the reservoirs of two adjacent modules. H_{AB} is the pressure developed by any pumps between the two reservoirs, h_{AB} is the loss due to flow through the pipe between the reservoirs.

The pressure drop between a module reservoir and the environment can also be considered. Here, the difference between the environmental pressure and the pressure in the reservoir is equal to any pressure developed by a pump, minus pressure losses both due to the fluid flowing through a pipe, and the sudden expansion of the pipe into the environment.

$$P_A - P_e = H_{Ae} - h_{Ae} - f_{Ae},$$

where P_e is the environmental pressure. f_{Ae} is the loss due to sudden expansion and the module-environment boundary. Applying the law of energy conservation gives rise to $4N - C$ equations of N variables. Therefore the conservation laws give in total an equations system

Table B.1 System constants

Fluid density of water ρ	$1.0 \times 10^3 \text{ kgm}^{-3}$
Dynamic viscosity of water (at 20°) μ	$1.0 \text{ m}^3\text{s}^{-1}$
Pipe length L	$3.0 \times 10^{-2} \text{ m}$
Pipe radius r	$2.5 \times 10^{-3} \text{ m}$
Pump max. flow rate Q_{max}	$1.1 \times 10^{-5} \text{ m}^3\text{s}^{-1}$
Pump max. pressure (at 4V) H_C	$3.2 \times 10^4 \text{ kgm}^{-1}\text{s}^{-2}$
Pump internal hydraulic resistance k_2	$2.5 \times 10^9 \text{ kgm}^{-4}\text{s}^{-1}$

of $N + 4 - C = 5N - C$ equations and $4N - C + N = 5N - C$ variables, which is solvable for the flow rates and pressure drops through each pipe segment.

B.2 Worked example

In this section we present functions for H_{AB} , h_{AB} and f_{AB} . Real values from the hardware prototype presented in Chapter 4 are used to solve a simple example system with four modules and one active actuator.

The pressure drop along a pipe is given by the Hagen-Poiseuille law:

$$h_{AB} = \frac{8\mu L Q_{AB}}{\pi r^4} = k_1 Q_{AB}, \quad (\text{B.1})$$

where μ is the dynamic viscosity of the fluid, L is the length of the pipe and r is the pipe radius. The pressure drop due to a sudden expansion into a tank is given by:

$$f_{Ae} = \frac{\rho L Q_{AB}^2}{2\pi^2 r^4}, \quad (\text{B.2})$$

where ρ is the fluid density.

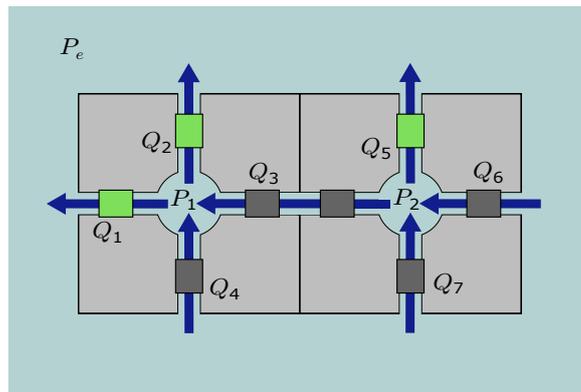


Figure B.1 An MHP robot consisting of two modules. Three pumps are active, five are inactive. Pressure nodes and assumed flow directions are indicated.

The pumps produce a corresponding, flow-rate dependent, pressure drop due to their internal structure, whether they are active or inactive. Active pumps produce a constant, positive pressure differential.

$$H_{AB} = H_C + Q_{AB}k_2,$$

where H_C is the pressure differential of an active pump, and k_2 is its internal hydraulic resistance. These values can be found from the pump manufacturer's datasheet [91]. For an inactive pump, there is no constant pressure increase, however the flow dependent pressure drop still remains.

Table B.1 shows the typical values of the system, taken or estimated from the hardware prototype. Substituting these into equations B.1 and B.2 (approximating Q as the max. pump flow rate Q_{max}) gives $h_{AB} = 1.26 \times 10^4$ and $f_{Ae} = 6.25 \times 10^1$. Given that $f_{Ae} \sim 10^{-2} \times h_{AB}$, we can safely neglect the f_{Ae} terms when calculating the flow rates.

Figure B.1 shows a an example MFP robot with three active pumps. Flow rates are indicated.

The law of mass conservation gives two equations:

$$Q_1 + Q_2 = Q_3 + Q_4$$

$$Q_3 + Q_5 = Q_6 + Q_7.$$

From the law of energy conservation:

$$P_2 - P_1 = 2(k_1 + k_2)Q_3$$

$$P_e - P_1 = H_C - (k_1 + k_2)Q_1$$

$$P_e - P_1 = H_C - (k_1 + k_2)Q_2$$

$$P_e - P_1 = (k_1 + k_2)Q_4$$

$$P_e - P_2 = H_C - (k_1 + k_2)Q_5$$

$$P_e - P_2 = (k_1 + k_2)Q_6$$

$$P_e - P_2 = (k_1 + k_2)Q_7.$$

This gives 9 equations with 9 variables ($P_1, P_2, Q_1, \dots, Q_7$). Solving these, and using the values in Table B.1, gives:

$$Q_1 = 3.2 \times 10^{-6} \text{ m}^3 \text{ s}^{-1}$$

$$Q_2 = 3.2 \times 10^{-6} \text{ m}^3 \text{ s}^{-1}$$

$$Q_3 = 1.1 \times 10^{-6} \text{ m}^3 \text{ s}^{-1}$$

$$Q_4 = 5.4 \times 10^{-6} \text{ m}^3 \text{ s}^{-1}$$

$$Q_5 = 5.4 \times 10^{-6} \text{ m}^3 \text{ s}^{-1}$$

$$Q_6 = 3.2 \times 10^{-6} \text{ m}^3 \text{ s}^{-1}$$

$$Q_7 = 3.2 \times 10^{-6} \text{ m}^3 \text{ s}^{-1}.$$

References

- [1] Arvo, J. (1992). Fast random rotation matrices. In *Graphics Gems III*, pages 117–120. Academic Press Professional, San Diego, CA.
- [2] Bishop, J., Burden, S., Klavins, E., Kreisberg, R., Malone, W., Napp, N., and Nguyen, T. (2005). Programmable parts: A demonstration of the grammatical approach to self-organization. In *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3684–3691. IEEE.
- [3] Bonani, M., Longchamp, V., Magnenat, S., Rétornaz, P., Burnier, D., Roulet, G., Vausard, F., Bleuler, H., and Mondada, F. (2010). The marXbot, a miniature mobile robot opening new perspectives for the collective-robotic research. In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4187–4193. IEEE.
- [4] Braley, S., Rubens, C., Merritt, T., and Vertegaal, R. (2018). GridDrones: A self-levitating physical voxel lattice for interactive 3d surface deformations. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, pages 87–98. ACM.
- [5] Breivik, J. (2001). Self-organization of template-replicating polymers and the spontaneous rise of genetic information. *Entropy*, 3(4):273–279.
- [6] Brodbeck, L., Hauser, S., and Iida, F. (2015). Morphological evolution of physical robots through model-free phenotype development. *PloS one*, 10(6):e0128444.
- [7] Campbell, J. and Pillai, P. (2008). Collective actuation. *International Journal of Robotics Research*, 27(3–4):299–314.
- [8] Castano, A., Behar, A., and Will, P. M. (2002). The Conro modules for reconfigurable robots. *IEEE/ASME Transactions on Mechatronics*, 7(4):403–409.
- [9] Chen, J., Gauci, M., and Groß, R. (2013). A strategy for transporting tall objects with a swarm of miniature mobile robots. In *Proceedings of the 2013 IEEE International Conference on Robotics and Automation*, pages 863–869. IEEE.
- [10] Chen, J., Gauci, M., Li, W., Kolling, A., and Groß, R. (2015). Occlusion-based cooperative transport with a swarm of miniature mobile robots. *IEEE Transactions on Robotics*, 31(2):307–321.
- [11] Chirikjian, G. S., Zhou, Y., and Suthakorn, J. (2002). Self-replicating robots for lunar development. *IEEE/ASME Transactions on Mechatronics*, 7(4):462–472.

- [12] Christensen, D., Andersen, J., Blanke, M., Furno, L., Galeazzi, R., Hansen, P., and Nielsen, M. (2014). Collective modular underwater robotic system for long-term autonomous operation. *Science*, 19(1):35–40.
- [13] Davey, J., Kwok, N., and Yim, M. (2012). Emulating self-reconfigurable robots-design of the SMORES system. In *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4464–4469. IEEE.
- [14] Demir, N. and Açıkmese, B. (2015). Probabilistic density control for swarm of decentralized on-off agents with safety constraints. In *Proceedings of the 2015 American Control Conference*, pages 5238–5244. IEEE.
- [15] Detweiler, C., Sosnowski, S., Vasilescu, I., and Rus, D. (2009). Saving energy with buoyancy and balance control for underwater robots with dynamic payloads. In *Proceedings of the 11th International Symposium on Experimental Robotics*, pages 429–438. Springer.
- [16] Diegel, O., Badve, A., Bright, G., Potgieter, J., and Tlale, S. (2002). Improved mecanum wheel design for omni-directional robots. In *Proceedings of the 2002 Australasian Conference on Robotics and Automation*, pages 117–121.
- [17] Ding, R., Eastwood, P., Mondada, F., and Groß, R. (2012). A stochastic self-reconfigurable modular robot with mobility control. In *Proceedings of the 13th Conference on Advances in Autonomous Robotics*, pages 416–417. Springer.
- [18] Doniec, M., Detweiler, C., and Rus, D. (2014). Estimation of thruster configurations for reconfigurable modular underwater robots. In *Proceedings of the 16th International Symposium on Experimental Robotics*, pages 655–666. Springer.
- [19] Doniec, M., Detweiler, C., Vasilescu, I., and Rus, D. (2010a). Using optical communication for remote underwater robot operation. In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4017–4022. IEEE.
- [20] Doniec, M., Vasilescu, I., Detweiler, C., and Rus, D. (2010b). Complete SE³ underwater robot control with arbitrary thruster configurations. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, pages 5295–5301. IEEE.
- [21] Dorigo, M., Trianni, V., Şahin, E., Groß, R., Labella, T. H., Baldassarre, G., Nolfi, S., Deneubourg, J.-L., Mondada, F., Floreano, D., et al. (2004). Evolving self-organizing behaviors for a swarm-bot. *Autonomous Robots*, 17(2-3):223–245.
- [22] Eiben, A. (2015). Evosphere: The world of robot evolution. In *Proceedings of the 4th International Conference on Theory and Practice of Natural Computing*, pages 3–19. Springer.
- [23] Eiben, A. E. (2014). Grand challenges for evolutionary robotics. *Frontiers in Robotics and AI*, 1(4):1–2.
- [24] Escalera, J. A., Doyle, M. J., Mondada, F., and Groß, R. (2016a). Evo-bots: A simple, stochastic approach to self-assembling artificial organisms. In *Proceedings of the 13th International Symposium on Distributed Autonomous Robotic Systems*, pages 373–385. Springer.

- [25] Escalera, J. A., Doyle, M. J., Mondada, F., and Groß, R. (2016b). Online supplementary material. url <http://naturalrobotics.group.shef.ac.uk/supp/2016-006/>.
- [26] Escalera, J. A., Mondada, F., and Groß, R. (2014). Evo-bots: A modular robotics platform with efficient energy sharing. In *Modular and Swarm Systems Workshop at IROS* (URL: <https://naturalrobotics.group.shef.ac.uk/publications/2014-irosws-escalera.pdf>).
- [27] Fukuda, T. (1988). Self organizing robots based on cell structures-CEBOT. In *Proceedings of the 1988 IEEE International Workshop on Intelligent Robots*, pages 145–150. IEEE.
- [28] Fukuda, T. and Nakagawa, S. (1988). Dynamically reconfigurable robotic system. In *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, pages 1581–1586. IEEE.
- [29] Furno, L., Blanke, M., Galeazzi, R., and Christensen, D. J. (2017). Self-reconfiguration of modular underwater robots using an energy heuristic. In *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 6177–6284. IEEE.
- [30] Ganesan, V. and Chitre, M. (2016). On stochastic self-assembly of underwater robots. *IEEE Robotics and Automation Letters*, 1(1):251–258.
- [31] Gilpin, K., Knaian, A., and Rus, D. (2010). Robot pebbles: One centimeter modules for programmable matter through self-disassembly. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, pages 2485–2492. IEEE.
- [32] Goldstein, S. C., Campbell, J. D., and Mowry, T. C. (2005). Programmable matter. *Computer*, 38(6):99–101.
- [33] Goldstein, S. C. and Mowry, T. C. (November 2004). Claytronics: A scalable basis for future robots. In *RoboSphere 2004*.
- [34] Griffith, S., Goldwater, D., and Jacobson, J. M. (2005). Robotics: Self-replication from random parts. *Nature*, 437(7059):636.
- [35] Griffith, S. T. (2004). *Growing machines*. PhD thesis, Massachusetts Institute of Technology.
- [36] Groß, R., Bonani, M., Mondada, F., and Dorigo, M. (2006). Autonomous self-assembly in swarm-bots. *IEEE Transactions on Robotics*, 22(6):1115–1130.
- [37] Groß, R., Magnenat, S., Kuchler, L., Massaras, V., Bonani, M., and Mondada, F. (2009). Towards an autonomous evolution of non-biological physical organisms. In *Proceedings of the 10th European Conference on Artificial Life*, pages 173–180. Springer.
- [38] Haghghat, B., Droz, E., and Martinoli, A. (2015). Lily: A miniature floating robotic platform for programmable stochastic self-assembly. In *Proceedings of the 2015 IEEE International Conference on Robotics and Automation*, pages 1941–1948. IEEE.

- [39] Haghghat, B. and Martinoli, A. (2016). Characterization and validation of a novel robotic system for fluid-mediated programmable stochastic self-assembly. In *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2778–2783. IEEE.
- [40] Haghghat, B. and Martinoli, A. (2017). Automatic synthesis of rulesets for programmable stochastic self-assembly of rotationally symmetric robotic modules. *Swarm Intelligence*, 11(3-4):243–270.
- [41] Ieropoulos, I., Taylor, B., Theodosiou, P., Støy, K., Nejatimoharrami, F., and Faíña, A. (2016). Evobot: An open-source, modular liquid handling robot for nurturing microbial fuel cells. In *Proceedings of the Artificial Life Conference 2016*, pages 626–633. MIT Press.
- [42] Inou, N., Kobayashi, H., and Koseki, M. (2002). Development of pneumatic cellular robots forming a mechanical structure. In *Proceedings of the 7th International Conference on Control, Automation, Robotics and Vision*, pages 63–68. IEEE.
- [43] Jacobson, H. (1958). On models of reproduction. *American Scientist*, 46(3):255–284.
- [44] Jakobi, N., Husbands, P., and Harvey, I. (1995). Noise and the reality gap: The use of simulation in evolutionary robotics. In *Proceedings of the 3rd European Conference on Artificial Life*, pages 704–720. Springer.
- [45] Jorgensen, M. W., Ostergaard, E. H., and Lund, H. H. (2004). Modular ATRON: Modules for a self-reconfigurable robot. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 2068–2073. IEEE.
- [46] Karagozler, M. E., Goldstein, S. C., and Reid, J. R. (2009). Stress-driven MEMS assembly + electrostatic forces = 1mm diameter robot. In *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2763–2769. IEEE.
- [47] Kernbach, S., Meister, E., Schlachter, F., Jebens, K., Szymanski, M., Liedke, J., Laneri, D., Winkler, L., Schmickl, T., Thenius, R., et al. (2008). Symbiotic robot organisms: REPLICATOR and SYMBRION projects. In *Proceedings of the 8th workshop on performance metrics for intelligent systems*, pages 62–69. ACM.
- [48] Klavins, E. (2007). Programmable self-assembly. *IEEE Control Systems*, 27(4):43–56.
- [49] Klavins, E., Ghrist, R., and Lipsky, D. (2006). A grammatical approach to self-organizing robotic systems. *IEEE Transactions on Automatic Control*, 51(6):949–962.
- [50] Komosinski, M. (2003). The Framsticks system: versatile simulator of 3d agents and their evolution. *Kybernetes*, 32(1/2):156–173.
- [51] Konidaris, G., Taylor, T., and Hallam, J. (2007). Hydrogen: Automatically generating self-assembly code for hydron units. In *Distributed Autonomous Robotic Systems 6*, pages 33–42. Springer.
- [52] Koshland, D. E. (2002). The seven pillars of life. *Science*, 295(5563):2215–2216.

- [53] Kurokawa, H., Kamimura, A., Yoshida, E., Tomita, K., Kokaji, S., and Murata, S. (2003). M-TRAN II: Metamorphosis from a four-legged walker to a caterpillar. In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2454–2459. IEEE.
- [54] Kurokawa, H., Kamimura, A., Yoshida, E., Tomita, K., Murata, S., and Kokaji, S. (2002). Self-reconfigurable modular robot (M-TRAN) and its motion design. In *Proceedings of the 7th International Conference on Control, Automation, Robotics and Vision*, volume 1, pages 51–56. IEEE.
- [55] Kurokawa, H., Tomita, K., Kamimura, A., Kokaji, S., Hasuo, T., and Murata, S. (2008). Distributed self-reconfiguration of M-TRAN III modular robotic system. *International Journal of Robotics Research*, 27(3-4):373–386.
- [56] Li, G., Gabrich, B., Saldaña, D., Das, J., Kumar, V., and Yim, M. (2019, in press). ModQuad-Vi: A vision-based self-assembling modular quadrotor. In *Proceedings of the 2019 IEEE International Conference on Robotics and Automation*. IEEE.
- [57] Lipson, H. and Pollack, J. B. (2000). Automatic design and manufacture of robotic lifeforms. *Nature*, 406(6799):974–978.
- [58] Mateos, L. A., Wang, W., Gheneti, B., Duarte, F., Ratti, C., and Rus, D. (2019, in press). Autonomous latching system for robotic boats. In *Proceedings of the 2019 IEEE International Conference on Robotics and Automation*. IEEE.
- [59] Miconi, T. (2008). Evosphere: evolutionary dynamics in a population of fighting virtual creatures. In *Proceedings of the 2008 Congress on Evolutionary Computation*, pages 3066–3073. IEEE.
- [60] Mintchev, S., Ranzani, R., Fabiani, F., and Stefanini, C. (2015). Towards docking for small scale underwater robots. *Autonomous Robots*, 38(3):283–299.
- [61] Mintchev, S., Stefanini, C., Girin, A., Marrazza, S., Orofino, S., Lebastard, V., Manfredi, L., Dario, P., and Boyer, F. (2012). An underwater reconfigurable robot with bioinspired electric sense. In *Proceedings of the 2012 IEEE International Conference on Robotics and Automation*, pages 1149–1154. IEEE.
- [62] Miyashita, S., Casanova, F., Lungarella, M., and Pfeifer, R. (2008a). Tribolon: Water based self-assembly robot with freezing connector (video). In *Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4147–4148. IEEE.
- [63] Miyashita, S., Kessler, M., and Lungarella, M. (2008b). How morphology affects self-assembly in a stochastic modular robot. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*, pages 3533–3538. IEEE.
- [64] Miyashita, S. and Pfeifer, R. (2012). Attributes of two-dimensional magnetic self-assembly. *Adaptive Behavior*, 20(2):117–130.
- [65] Murata, S., Yoshida, E., Kamimura, A., Kurokawa, H., Tomita, K., and Kokaji, S. (2002). M-TRAN: Self-reconfigurable modular robotic system. *IEEE/ASME Transactions on Mechatronics*, 7(4):431–441.

- [66] Naldi, R., Forte, F., Serrani, A., and Marconi, L. (2015). Modeling and control of a class of modular aerial robots combining under actuated and fully actuated behavior. *IEEE Transactions on Control Systems Technology*, 23(5):1869–1885.
- [67] O’Hara, I., Paulos, J., Davey, J., Eckenstein, N., Doshi, N., Tosun, T., Greco, J., Seo, J., Turpin, M., Kumar, V., et al. (2014). Self-assembly of a swarm of autonomous boats into floating structures. In *Proceedings of the 2014 IEEE International Conference on Robotics and Automation*, pages 1234–1240. IEEE.
- [68] Østergaard, E. H., Kassow, K., Beck, R., and Lund, H. H. (2006). Design of the ATRON lattice-based self-reconfigurable robot. *Autonomous Robots*, 21(2):165–183.
- [69] Oung, R. and D’Andrea, R. (2014). The Distributed Flight Array: Design, implementation, and analysis of a modular vertical take-off and landing vehicle. *International Journal of Robotics Research*, 33(3):375–400.
- [70] Oung, R. and D’Andrea, R. (2011). The Distributed Flight Array. *IEEE/ASME Transactions on Mechatronics*, 21(6):908–917.
- [71] Parrott, C., Dodd, T. J., and Groß, R. (2014). HiGen: A high-speed genderless mechanical connection mechanism with single-sided disconnect for self-reconfigurable modular robots. In *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3926–3932. IEEE.
- [72] Parrott, C., Dodd, T. J., and Groß, R. (2018). HyMod: A 3-DOF hybrid mobile and self-reconfigurable modular robot and its extensions. In *Proceedings of the 13th International Symposium on Distributed Autonomous Robotic Systems*, pages 401–414. Springer.
- [73] Paulos, J., Eckenstein, N., Tosun, T., Seo, J., Davey, J., Greco, J., Kumar, V., and Yim, M. (2015). Automated self-assembly of large maritime structures by a team of robotic boats. *IEEE Transactions on Automation Science and Engineering*, 12(3):958–968.
- [74] Penrose, L. S. and Penrose, R. (1957). A self-reproducing analogue. *Nature*, 179(4571):1183.
- [75] Pesavento, U. (1995). An implementation of von Neumann’s self-reproducing machine. *Artificial Life*, 2(4):337–354.
- [76] Piranda, B. and Bourgeois, J. (2018). Designing a quasi-spherical module for a huge modular robot to create programmable matter. *Autonomous Robots*, 42(8):1619–1633.
- [77] Romanishin, J. W., Gilpin, K., Claici, S., and Rus, D. (2015). 3D M-Blocks: Self-reconfiguring robots capable of locomotion via pivoting in three dimensions. In *Proceedings of the 2015 IEEE International Conference on Robotics and Automation*, pages 1925–1932. IEEE.
- [78] Romanishin, J. W., Gilpin, K., and Rus, D. (2013). M-Blocks: Momentum-driven, magnetic modular robots. In *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4288–4295. IEEE.

- [79] Saldana, D., Gabrich, B., Li, G., Yim, M., and Kumar, V. (2018). ModQuad: The flying modular structure that self-assembles in midair. In *Proceedings of the 2018 IEEE International Conference on Robotics and Automation*, pages 691–698. IEEE.
- [80] Sastra, J., Chitta, S., and Yim, M. (2009). Dynamic rolling for a modular loop robot. *The International Journal of Robotics Research*, 28(6):758–773.
- [81] Seo, J., Yim, M., and Kumar, V. (2016). Assembly sequence planning for constructing planar structures with rectangular modules. In *Proceedings of the 2016 IEEE International Conference on Robotics and Automation*, pages 5477–5482. IEEE.
- [82] Shen, W.-M., Salemi, B., and Will, P. (2002). Hormone-inspired adaptive communication and distributed control for conro self-reconfigurable robots. *IEEE Transactions on Robotics and Automation*, 18(5):700–712.
- [83] Sims, K. (1994a). Evolving 3D morphology and behavior by competition. *Artificial life*, 1(4):353–372.
- [84] Sims, K. (1994b). Evolving virtual creatures. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, pages 15–22. ACM.
- [85] Smith, R. et al. (2005). Open dynamics engine (<http://ode.org/>).
- [86] Spector, L., Klein, J., and Feinstein, M. (2007). Division Blocks and the open-ended evolution of development, form, and behavior. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, pages 316–323. ACM.
- [87] Sprowitz, A., Pouya, S., Bonardi, S., Van den Kieboom, J., Möckel, R., Billard, A., Dillenbourg, P., and Ijspeert, A. J. (2010). Roombots: Reconfigurable robots for adaptive furniture. *Computational Intelligence Magazine, IEEE*, 5(3):20–32.
- [88] Støy, K., Brandt, D., and Christensen, D. J. (2010). *Self-Reconfigurable Robots: An Introduction*. MIT Press.
- [89] Støy, K. and Nagpal, R. (2004). Self-repair through scale independent self-reconfiguration. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 2062–2067. IEEE.
- [90] Sutantyo, D., Buntoro, D., Levi, P., Mintchev, S., and Stefanini, C. (2013). Optical-guided autonomous docking method for underwater reconfigurable robot. In *Proceedings of the 2013 IEEE International Conference on Technologies for Practical Robot Applications*, pages 1–6. IEEE.
- [91] TCS Micropumps (2015). *M100 and M200 Micropump Datasheet*. <https://micropumps.co.uk/DATA/pdf/IM02%20-%20Instructions%20M100%20and%20M200%20REV%203.pdf>.
- [92] Tolley, M. T. and Lipson, H. (2010). Fluidic manipulation for scalable stochastic 3D assembly of modular robots. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, pages 2473–2478. IEEE.

- [93] Vasilescu, I., Detweiler, C., Doniec, M., Gurdan, D., Sosnowski, S., Stumpf, J., and Rus, D. (2010). AMOUR V: A hovering energy efficient underwater robot capable of dynamic payloads. *International Journal of Robotics Research*, 29(5):547–570.
- [94] Vasilescu, I., Varshavskaya, P., Kotay, K., and Rus, D. (2005). Autonomous modular optical underwater robot (AMOUR) design, prototype and feasibility study. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 1603–1609. IEEE.
- [95] Virgo, N., Fernando, C., Bigge, B., and Husbands, P. (2012). Evolvable physical self-replicators. *Artificial Life*, 18(2):129–142.
- [96] von Haller, B., Ijspeert, A., and Floreano, D. (2005). Co-evolution of structures and controllers for neubot underwater modular robots. In *Proceedings of the 7th European Conference on Artificial Life*, pages 189–199. Springer.
- [97] von Neumann, J. and Burks, A. W. (1966). *Theory of self-reproducing automata*. University of Illinois Press.
- [98] Wang, W., Mateos, L. A., Park, S., Leoni, P., Gheneti, B., Duarte, F., Ratti, C., and Rus, D. (2018). Design, modeling, and nonlinear model predictive tracking control of a novel autonomous surface vehicle. In *Proceedings of the 2018 IEEE International Conference on Robotics and Automation*, pages 6189–6196. IEEE.
- [99] Weel, B., Crosato, E., Heinerman, J., Haasdijk, E., and Eiben, A. (2014). A robotic ecosystem with evolvable minds and bodies. In *2014 IEEE International Conference on Evolvable Systems*, pages 165–172. IEEE.
- [100] White, P., Kopanski, K., and Lipson, H. (2004). Stochastic self-reconfigurable cellular robotics. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, volume 3, pages 2888–2893. IEEE.
- [101] White, P., Zykov, V., Bongard, J., and Lipson, H. (2005). Three dimensional stochastic reconfiguration of modular robots. In *Proceedings of the 2005 Robotics: Science and Systems Conference*, pages 161–168. MIT Press.
- [102] Yim, M. (1994a). *Locomotion with a unit-modular reconfigurable robot*. PhD thesis, Citeseer.
- [103] Yim, M. (1994b). New locomotion gaits. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 2508–2514. IEEE.
- [104] Yim, M., Roufas, K., Duff, D., Zhang, Y., Eldershaw, C., and Homans, S. (2003). Modular reconfigurable robots in space applications. *Autonomous Robots*, 14(2-3):225–237.
- [105] Yim, M., Shen, W.-M., Salemi, B., Rus, D., Moll, M., Lipson, H., Klavins, E., and Chirikjian, G. S. (2007). Modular self-reconfigurable robot systems [grand challenges of robotics]. *Robotics & Automation Magazine, IEEE*, 14(1):43–52.

-
- [106] Yim, M., Zhang, Y., Roufas, K., Duff, D., and Eldershaw, C. (2002). Connecting and disconnecting for chain self-reconfiguration with PolyBot. *IEEE/ASME Transactions on Mechatronics*, 7(4):442–451.
- [107] Zykov, V., Chan, A., and Lipson, H. (2007a). Molecubes: An open-source modular robotics kit. In *IROS 2007 Self-Reconfigurable Robotics Workshop*, pages 3–6.
- [108] Zykov, V., Mytilinaios, E., Adams, B., and Lipson, H. (2005). Robotics: Self-reproducing machines. *Nature*, 435(7039):163–164.
- [109] Zykov, V., Mytilinaios, E., Desnoyer, M., and Lipson, H. (2007b). Evolved and designed self-reproducing modular robotics. *IEEE Transactions on Robotics*, 23(2):308–319.
- [110] Zykov, V., Williams, P., Lassabe, N., and Lipson, H. (2008). Molecubes extended: Diversifying capabilities of open-source modular robotics. In *IROS 2008 Self-Reconfigurable Robotics Workshop*, pages 22–26.

