



This is a repository copy of *Modular Hydraulic Propulsion: A Robot that Moves by Routing Fluid Through Itself*.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/108979/>

Version: Accepted Version

Proceedings Paper:

Doyle, M., Xu, X., Gu, Y. et al. (3 more authors) (2016) Modular Hydraulic Propulsion: A Robot that Moves by Routing Fluid Through Itself. In: 2016 IEEE International Conference on Robotics and Automation. 2016 IEEE International Conference on Robotics and Automation (ICRA 2016), 16-21 May 2016, Stockholm, Sweden. IEEE , pp. 5189-5196.

<https://doi.org/10.1109/ICRA.2016.7487725>

Reuse

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



eprints@whiterose.ac.uk
<https://eprints.whiterose.ac.uk/>

Modular Hydraulic Propulsion: A Robot that Moves by Routing Fluid Through Itself

Matthew J. Doyle, Xinyu Xu¹, Yue Gu¹, Fernando Perez-Diaz, Christopher Parrott and Roderich Groß

Abstract—This paper introduces the concept of Modular Hydraulic Propulsion, in which a modular robot that operates in a fluid environment moves by routing the fluid through itself. The robot’s modules represent sections of a hydraulics network. Each module can move fluid between any of its faces. The modules (network sections) can be rearranged into arbitrary topologies. We propose a decentralized motion controller, which does not require modules to communicate, compute, nor store information during run-time. We use 3-D simulations to compare the performance of this controller to that of a centralized controller with full knowledge of the task. We also detail the design and fabrication of six 2-D prototype modules, which float in a water tank. Results of systematic experiments show that the decentralized controller, despite its simplicity, reliably steers modular robots towards a light source. Modular Hydraulic Propulsion could offer new solutions to problems requiring reconfigurable systems to move precisely in 3-D, such as inspection of pipes, vascular systems or other confined spaces.

I. INTRODUCTION

Robots are becoming increasingly adept at performing tasks more effectively than humans, or performing tasks that humans are unable to. However, most robots are only suited for a limited range of tasks; new tasks are likely to require different robots. Modular robots [1], [2] offer an alternative approach. They comprise individual modules that can be mechanically configured to perform a certain task, and then be reconfigured (or self-reconfigure) to perform a different task. The advantage of modular robots is thus their flexibility.

Two of the challenges in modular robotics are [3]:

- to effectively propel connected structures in three dimensions (3-D);
- to miniaturize the building blocks, that is, modules to sub-millimetre scale.

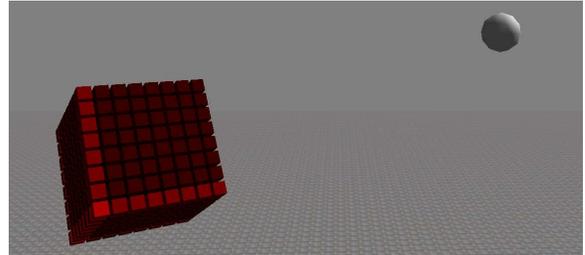
Working towards these challenges may lead to effectively propelling programmable matter [4], [5].

In this paper we propose the concept of *Modular Hydraulic Propulsion* (MHP). Each module of an MHP robot represents a section of a fluid network. The modules (and network sections) are mechanically linked. They can be reconfigured into arbitrary network topologies, either manually—as considered in this paper—or through self-assembly. Each module can route fluid between any of

*This work was funded by the Engineering and Physical Sciences Research Council (EPSRC) through scholarship support (M. Doyle) and grant no. EP/K033948/1.

The authors are with Sheffield Robotics and the Department of Automatic Control and Systems Engineering, The University of Sheffield, UK, e-mail: {matthew.doyle, fernando.perez.diaz, c.parrott, r.gross}@sheffield.ac.uk

¹These authors contributed equally.



(a)



(b)

Fig. 1. Modular Hydraulic Propulsion: (a) 3-D simulation of a robot, comprising 512 cubic modules, approaching a target in a fluid environment. Each module can route fluid between any of its faces. By deciding how to route the fluid through the structure, both translation and rotation can be controlled. (b) 2-D physical prototype floating in a water tank. Each module is square shaped and contains four micro-pumps, one per side. When a pump is off, it acts as a bypass.

its network interfaces—which lead either to a neighboring module or the environment. The fluid motion, resulting from the modules’ routing protocols, causes the MHP robot to translate and rotate.

We describe the design of an MHP robot, which consists of cubic modules (see Fig. 1a). Each module’s face has a connection sensor and a light detection sensor. We propose a decentralized motion controller (routing protocol) that steers the MHP robot towards a light source (but has no control over orientation). The controller is reactive—it simply maps a module’s binary sensor readings onto the module’s binary actuators. The controller does not require modules to communicate, perform arithmetic computation, nor store memory during run-time. Its performance is compared against a centralized controller, adopted from [6]. The latter can control both translation and rotation.

We present the design and fabrication of a two dimensional (2-D) MHP robotic system (see Fig. 1b). The robot floats in a water tank. Experiments show that MHP robots of different topologies reliably move towards a light source.

The paper is organized as follows. Section II overviews

related work. Section III details the MHP concept. Section IV presents the two motion algorithms—one centralized and one decentralized. Section V presents the 3-D simulations. Section VI details the physical prototypes and experiments. Section VII concludes the paper.

II. RELATED WORK

Most modular robotic systems operate on the ground. They are essentially confined to 2.5-D [7], [8], [9]. Truly reaching into the third dimension would require either a supporting structure or an excessive amount of modules. For only a few systems, modules operate freely in 3-D environments, such as in fluids [10], [11], [12], [13], [14], [15], [16] or in the air [17]. The modular structures they form, however, are either 1-D [10], [13], [16], 2-D [17], or not self-propelling [11], [12], [14], [15]. To our knowledge, at present no modular robot platform supports 3-D structures that self-propel freely in 3-D. The MHP concept, if implemented in 3-D, would offer a route to overcoming this.

MHP is related to the concept of collective actuation [18]. In the latter, a modular robot deforms through local changes in its topology, producing force in a similar way to muscles. This force can lead to changes in position and orientation. The concept of anatomy-based organization [19] can be considered as collective actuation too; here a modular robot is hierarchically organized into components representing muscles, bones, joints etc. In both collective actuation and MHP, modules act collectively to achieve propulsion. In MHP however, individual modules are self-propelled too.

Lipson’s group designed several systems in which individual modules move passively in a 3-D oil tank [11], [12]. The modules are externally propelled by agitating the fluid. They receive power by docking to an immobile support structure. While they can form 3-D structures, these are immobile, as any attempt to undock from the support structure will result in a loss of power. One of the systems described in [11] uses fluid, which gets drawn through the modules by an external pump, to facilitate the reconfiguration process. This design has inspired MHP. Note however, that unlike MHP, the modules presented in [11] have no motive capabilities of their own. In [14] and [15], further systems are described that undergo stochastic reconfiguration. Again the modules are unpowered and not self-propelling.

Several modular underwater robotic systems utilising self-propelled modules have been developed. Vasilescu et al. [10], [13] present the AMOUR platform, a set of reconfigurable modules, which helps deploy and monitor a static sensor network. In [6], a centralized motion controller is presented that can be applied to robots with arbitrary configurations of thrusters. The algorithm is tested using a robot with configurable thrusters; translation and rotation are both achieved. This work is followed up in [20], which proposes an algorithm to learn the thruster configuration of the robot. In [21], a system of underwater modular robots is described; their body shapes and controllers are co-evolved. The robots swim by using actuated joints to oscillate limbs in a similar manner to that of humans. Mintchev et al. [16] present a chain-type

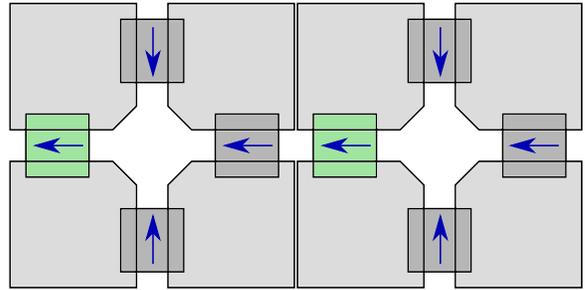


Fig. 2. 2-D cross-section of two connected modules. Boxes represent routing units, which can be dynamically configured in three ways. Green boxes indicate a routing unit configured as an active pump. Dark grey boxes indicate a routing unit configured as a bypass—fluid can freely pass in either way. Further options (not shown, and not used here) are to configure a routing unit to block the flow or use bidirectional pumps. Blue arrows indicate water flow direction.

modular system with two modes of locomotion. When acting individually, modules use propellers to translate and rotate. When linked in a chain together, they perform anguilliform swimming movements. In [22], a surface water modular robot is presented, simulating a collection of containers, which can form structures such as dynamic bridges and landing platforms.

Note that none of the modular robotic systems reported in the literature propel by routing fluid through themselves.

III. MODULAR HYDRAULIC PROPULSION (MHP)

For the remainder of this paper, a *robot* is defined as a connected configuration of one or more *modules*. We consider a robot to exhibit Modular Hydraulic Propulsion if its movement results from the routing of fluid through a modular hydraulics network. The network is distributed among the modules, each representing a section of it. Each module can route fluid between any of its faces, which incorporate dedicated interfaces. These interfaces mechanically link the modules (and network sections), leading either to a neighboring module or to the environment. In the latter case, the interface acts as a source or sink to the MHP robot.

The modules can be connected into different network topologies. This happens either manually—as considered in this paper—or automatically. For example, separate modules could self-assemble.

In principle, the modules could support the formation of 1-D, 2-D or 3-D structures. Modules for 2-D structures could be square shaped, as depicted in Figure 2. They would have four thrusters and network interfaces located at the middle of each side. Modules for 3-D structures could be cubic shaped. They would have six thrusters and network interfaces located at the center of each face.

Regardless of their morphology, modules have a central reservoir of fluid, linked with the environment only via the modules’ interfaces. Thrusters have two possible states, active or inactive. When active, thrusters pump fluid from the reservoir into the environment or neighboring modules. When inactive, thrusters allow fluid to be passed through freely, in any direction. This means that the fluid pumped

out by the active thrusters from an MHP robot (i.e. its internal network of reservoirs) is instantly replaced by fluid drawn in from the environment through passive thrusters. By this method of environment-module and module-module fluid routing, the robot will propel itself.

The actuators could optionally be bidirectional, and/or have the ability to close, thereby preventing fluid flow in and/or out. In the examples demonstrated in this paper however, these options are not used.

Each module of an MHP robot has connection sensors, one per face (side). This sensor reports whether or not the corresponding face (side) is connected to a neighboring module.

IV. CONTROL ALGORITHMS

In this paper, two motion controllers are considered: one decentralized and one centralized. Both controllers allow MHP robots to move to a target location such as a light source or other point of interest within the environment.

For the decentralized motion controller, additional binary target sensors are added at each module's face (side). They signal whether the target is detected or not. The sensor is assumed to point along the normal to the face (side), have a 180° field of view, and a range that is practically infinite (larger than the size of the environments considered here). For the centralized motion controller, we assume that the relative target position and the robot's 3-D posture are known.

A. Decentralized motion controller

The decentralized motion controller operates at the level of a module's face (side). It maps the sensors' state directly onto the binary state of the corresponding thruster: if both the connection sensor and target sensor return false, the thruster fires; otherwise, the thruster is off. As a consequence, only thrusters located on the external faces of the MHP robot may get activated, but only if they do not detect the target. In other words, all thrusters that are part of the robot's surface that is occluded with respect to the target location provide thrust. They pump fluid directly into the environment. Simultaneously, fluid is drawn into the MHP robot from the non-occluded parts of its surface, and passively routed through the structure.

The motion controller is inspired by an occlusion-based coordination algorithm for cooperative transport [23]: A swarm of mobile robots cooperatively pushes a tall convex object towards a target location. The robots push the object only if their view of the target is occluded by the object to be moved.

The validity of this controller can be proven for a 3-D convex¹ robot assuming quasi-static motion. The controller operation is shown in Figure 3. Let B , a convex surface, be the boundary of such a robot. Let C be the center of mass of the robot, p_i the center of each of its faces and T the target location. The occluded points of the boundary,

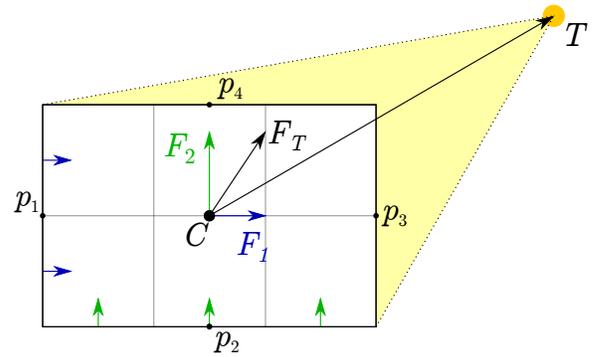


Fig. 3. A six module robot with center of mass C and a target point T , highlighted in orange. Faces one and two are occluded from the target point by the body of the robot, and therefore all thrusters on those faces fire. The short blue and green arrows denote the firing thrusters. Faces three and four are not occluded and so do not fire. The firing of thrusters on face one and face two produces forces \mathbf{F}_1 and \mathbf{F}_2 respectively. The resultant of these forces, \mathbf{F}_T , has a positive component along \mathbf{CT} , and so the robot moves towards the target position.

$O \subset B$, are those which do not have direct line of sight to the target. For a convex surface, the inwards normal at any point X in O has a positive scalar product with vector \mathbf{XT} . For cubic modules the only convex shapes that can be formed are cuboids. Therefore, a given robot face can either be totally occluded or totally visible from T . Given that the thrusters are evenly distributed over each robot face, no torque is produced. Moreover, the translational force exerted by the thrusters can be summed over each robot face i and applied on C . Following the above, the force, \mathbf{F}_i on face i has a positive component in the $\mathbf{p}_i\mathbf{T}$ direction, though not necessarily positive along \mathbf{CT} . The geometry of the robot guarantees that if the force on an occluded face has a non-positive component along \mathbf{CT} , there will be an equal and opposite force acting on a parallel face. Therefore, if all the occluded thrusters are firing, they will exert a net force $\mathbf{F}_T = \sum_i \mathbf{F}_i$ with a positive component along \mathbf{CT} , bringing the robot towards the target.

B. Centralized motion controller

The centralized motion controller also considers only thrusters on the MHP robot's surface, that is, on its external faces.

We adopted the centralized algorithm reported in [6]. This algorithm calculates Moore-Penrose pseudoinverse matrices to solve equation systems for translation and rotation of a robot. It then feeds the resultant thruster error vectors into separate translational and rotational proportional-integral-derivative (PID) controllers. The resulting PID outputs are then summed and fed to the thrusters. A small but important distinction is our use of binary rather than continuous thruster outputs. We set the thrusters to fire if the summed PID output is above zero, and to not fire, otherwise.

Choosing good parameters for the PID controllers is not trivial. A variety of module positions and orientations are considered in the experiments, and parameters that are appropriate for one modular structure may not be suitable

¹Robots with concave shapes are analyzed in simulation [see Section V.B.3)].

for others. One approach would be to determine empirically a set of parameters for each structure. This would however prove computationally expensive. Therefore, we calibrate the parameters for a certain morphology, a 3 by 3 by 3 cube.

The method for the initial calibration is similar to that in [6]. All PID terms are set to zero, and the proportional constant increased until the robot can navigate to the target. The differential constant is then increased until there is minimal overshoot. As in [6] the integral constant is kept at zero, as no steady-state offsets in the robot position are observed in simulation.

V. 3-D SIMULATION

A. Implementation

The 3-D simulator used here is based on the open-source Open Dynamics Engine (ODE) library [24]. The robots operate in an unbounded continuous underwater environment, and are assumed to be neutrally buoyant.

The robots are a 3-D implementation of the MHP concept. They are configured at the beginning of each trial, and do not change shape thereafter. Figure 1(a) shows a robot made of 512 modules. Also included in the environment is a target location.

The motion controllers considered in Section IV use only thrusters on the robot-environment boundary. Active thrusters are assumed to provide a constant force, and each of these forces is integrated by ODE. For simplicity the internal fluid flow is not modelled.

As the robots work in an underwater environment, the drag forces caused by the water need to be simulated. A full fluid dynamics treatment would prove too computationally expensive [11]. Instead, the drag force for each module is calculated, assuming a drag coefficient of 0.8, with ODE then integrating all the forces to provide the net drag. This adds a basic form of resistance to the motion of the robot without taking into account its overall shape.

Unless otherwise stated, the robots are cubic with a total side length of 8 cm, and a total force per side of 6.4 mN. In order to achieve this with different numbers of modules, the module size and the thruster strength are scaled accordingly (e.g. the one module used in a 1^3 module robot has a side length of 8 cm and its thrusters have a force of 6.4 mN; the eight modules used in a 2^3 module robot have a side length of 4 cm and thrusters of force 1.6 mN). We refer to robots by their total number of modules (e.g. a 4^3 module robot is a cubic robot with 64 modules in total, and with 16 modules on each face).

B. Results

In the following, we describe a number of simulation experiments. For each set of parameters, 100 independent trials were performed, with each trial starting with a single modular robot centered on the origin. Its 3-D starting orientation is uniformly random distributed using the method described in [25]. A target position is set at a fixed distance of 100 cm from the center of the robot. The robot is given 50 seconds to reach the target position. A robot is considered to

have reached the target position if its centroid comes within 6 cm of the target. This value was chosen to ensure that the target position is always outside of the robot itself. Note that if the target position resided within its boundaries, the robot would be unable to sense it.²

For each set of trials, the success rate, that is, the percentage of robots that reach the target within 50 seconds was measured. In addition, for successful trials, the average velocity was determined.

1) *Thruster inaccuracies*: In this section, we model slight manufacturing defects, or damage sustained in the course of operation. Each thruster's firing direction is permanently offset by a vector that is randomly generated at the beginning of the trial. The random length of the error vector has a uniform distribution between 0 and some specified positive real number. The error vector is set to be parallel to the corresponding face with a random orientation about the face normal. For both controllers, robots with 1^3 , 2^3 , 4^3 , 8^3 and 16^3 modules were used.

The results from the first set of runs are shown in Figure 4. In the case with no induced error, the distributions for the decentralized controller are similar for all side lengths. This is to be expected as the net force per face is the same across the different sizes of robot. In addition to this, all the sensors on one face will return the same value, meaning each of the robots will receive the same net input, and produce the same net output. The centralized controller displays better overall velocity at zero error, with similar distributions for each side length as in the decentralized case.

Thruster errors produce a reduced performance for robots with a small number of modules using the decentralized controller. Although the median velocity of the successful robots does not appreciably decrease, the success rate drops and the variance of the velocity increases. As the number of modules increases the success rate again reaches 100%, showing the increased resilience of the system to errors. The change in median velocity with module number is not as conclusive, however small increases can be noticed in the 8^3 and 16^3 module case. Along with this, the variance also decreases back to levels comparable to those in the no error case. This is important as it makes the system more predictable, and therefore easier to control. The centralized controller also shows initial large drops in success rate, which returns to 100% with an increase in module number. The median velocity remains higher than that of the decentralized controller except in the 2^3 module case with up to 45° error.

2) *Thruster or sensor failure*: In this section, we model mechanical or electrical problems that cause a thruster or sensor to fail without the robot being aware of the failure. Having specified a probability of failure, each thruster (sensor) will independently fail with such probability. If a thruster fails it will not fire for the duration of the trial. If a sensor fails it will return a constant value for the duration of the trial. Only the decentralized controller is used in the experiments

²Note that the centralized controller would still be capable of reaching the target exactly, assuming its position is known.

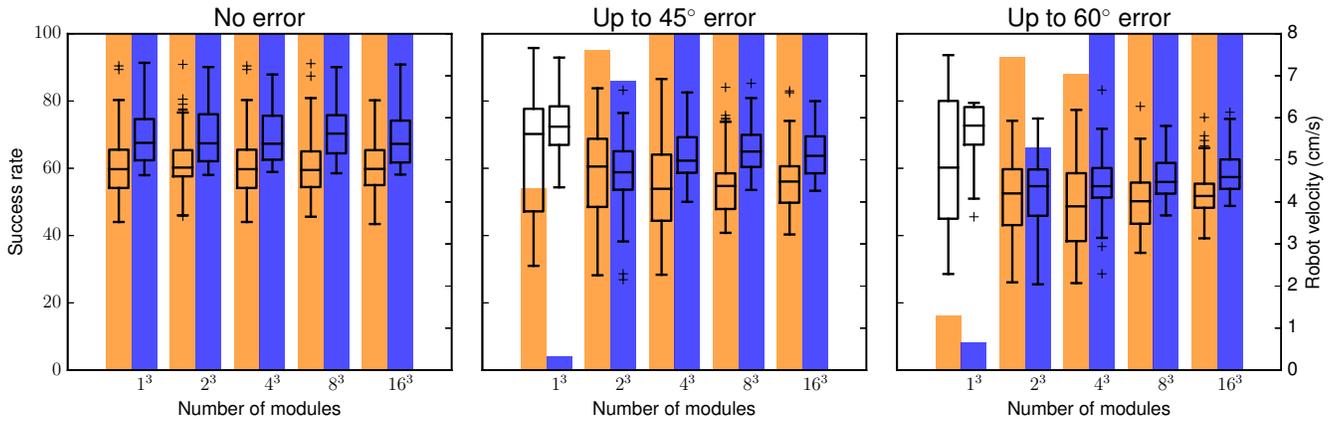


Fig. 4. The bars show the success rate of the robots (the number of trials that succeeded out of 100) for the decentralized (orange) and centralized (blue) controllers. The overlaid boxplots show the velocities of successful robots when subjected to permanent errors in their thruster directions for the decentralized (orange) and centralized (blue) controllers. The top and bottom boundaries of the boxplots represent the 3rd and 1st quartiles respectively, and the middle line the median. The top (bottom) whiskers represents the highest (lowest) datum within 1.5 inter-quartile range of the upper (lower) quartile. Outliers are also indicated.

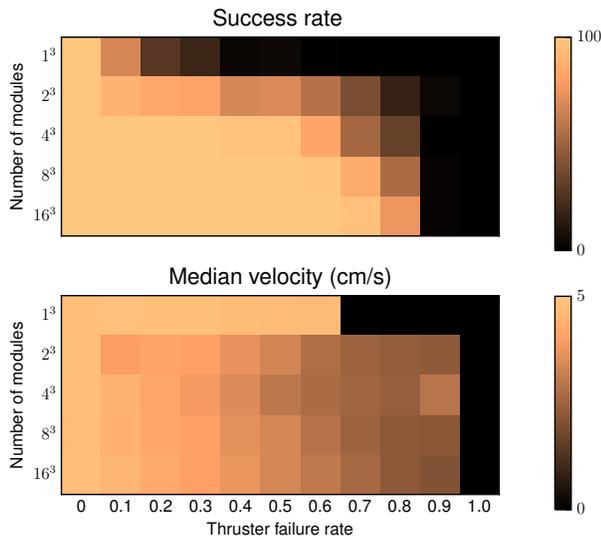


Fig. 5. Color maps showing the success rates (top) and median velocities (bottom) of robots using the decentralized controller when subjected to permanently failing thrusters. The velocities are averaged over successful trials only. Where there is a success rate of zero, the velocity is also recorded as zero.

henceforth.

The results from the experiments with thruster failure are shown in Figure 5. It is apparent that if the number of modules is sufficiently high, success rate degrades more gracefully. As the number of modules tends towards infinity, the probability that an entire face fails tends towards zero. The largest robots tested, the 16^3 module robots, exhibit a success rate of 0.97 at a thruster failure probability of 0.7.

Similar to the case of thruster direction error, higher numbers of modules do not seem to have an effect on the rate of degradation of median velocity. Note that the median velocity is only calculated over successful trials.

Results from experiments with sensor failure resemble

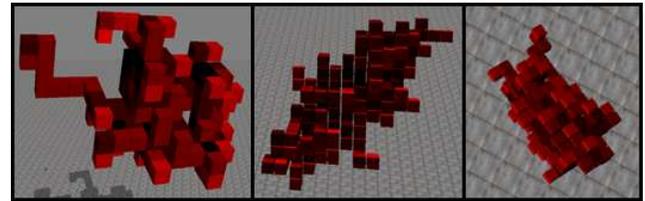


Fig. 6. Examples of randomly generated robots, moving towards a target using the decentralized motion controller.

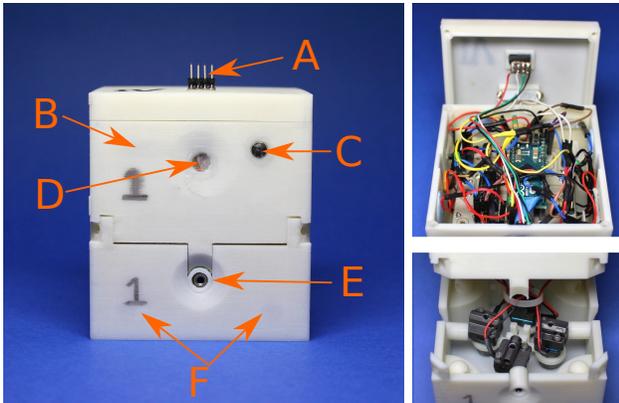
those in Figure 5. This can be explained by the fact that a sensor returning the wrong value is equivalent to a thruster firing (or not firing) at the wrong time.

3) *Random morphologies*: Robots with random morphologies were also generated and analyzed. These robots are composed of 100 modules each. Examples are shown in Figure 6. The robots are generated in an iterative manner, starting from a single module. One of the existing modules is randomly selected, and then one of its neighborless faces, if any, is selected. A new module is then affixed to this selected face. This process continues until all 100 modules are in place. The ability of the resultant robot to reach the target is then tested using the decentralized algorithm. We generated 100 such robots, all of which successfully reached the target with an average time of 18.5 seconds. The shortest time taken was 12.8 seconds and the longest time taken was 24.3 seconds.

4) *Larger structures*: The largest robots we simulated were 64^3 module robots, consisting of 262144 modules and 24576 externally facing thrusters and target sensors. Due to the prohibitive length of time needed to simulate these robots (on an office computer), we only completed five trials. The robots were simulated without induced errors, and all successfully reached the target (at a distance of 100 cm).



(a)



(b)

Fig. 7. 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).

VI. 2-D PHYSICAL PLATFORM

In order to test an embodied version of the MHP concept, a set of six prototype modules has been developed and tested. Floating on water, these represent a 2-D implementation of MHP.

A. Hardware Design

1) *Chassis*: The module is a cuboid of dimensions 80 x 80 x 90 mm (L x W x H) as shown in Figure 7. 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

are balanced such that during operation the 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.

2) *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 *false*. Otherwise, it is treated as *true*.

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.

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

The modules each run the decentralized motion controller. An alteration to the controller is to prevent pumps in a single module from working against each other. Whilst the controller in Section V treats each face independently, the controller here treats all four together. If a situation occurs in which two opposite faces of the same module were to be active, the controller will automatically render both inactive. This has the same net effect, but is more efficient. As in Section V, if a module’s face is connected to a neighbor, the controller will disable the corresponding pump.

B. Experiments

1) *Setup*: The experimental environment consists of a 126 by 68 cm water tank, shown in Figure 8. The water level is sufficiently high to prevent modules from coming

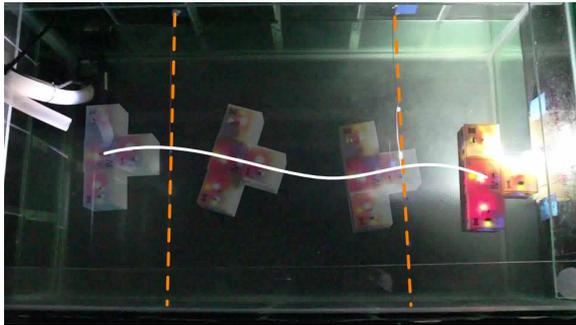


Fig. 8. The experimental environment and an example path taken by a four module configuration. Each module of the robot executes the decentralized motion controller. The left hand and right hand dashed orange lines show the start and finish lines respectively. The white line shows the path the robot took from left to right. The torch is located outside the water tank on the right hand side.

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 8), 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.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 9, along with the centers of mass of each configuration. Figure 8 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 seconds, and the minimum time taken being 6.5 seconds. The average time taken over all trials of all configurations was 9.6 seconds, corresponding to a velocity of just under 5 cm/s. We also performed a small number of trials with six modules, which were successfully completed (see accompanying video).

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 much closer to the center (the position at which the torch was mounted) than robots consisting of a single module. The torch used was not a good approximation for a light source with isotropic emission, and so the light received by the robot was unreliable. This provided a similar situation to that tested in Section V, wherein inaccurate sensors give

³Upon being released into the water, the robot would slightly drift in random directions. Each configuration took however less than five seconds to cross the start line.

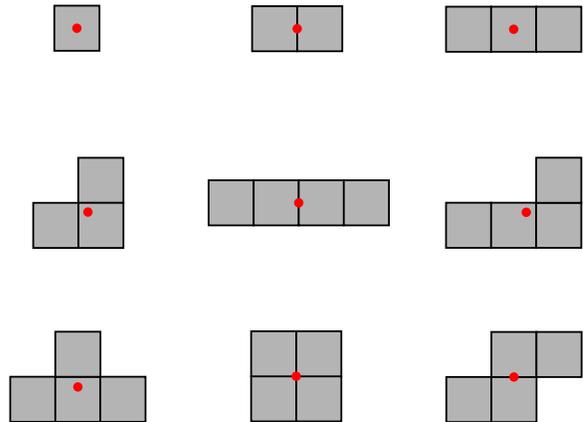


Fig. 9. All 9 distinct robot configurations using four modules. These are configurations which cannot be transformed into another configuration by rotation or mirroring. The red dots indicate 2D centers of mass.

incorrect responses.

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.

VII. CONCLUSION

We have presented the novel concept of modular robot propulsion—MHP. We have shown that a centralized controller from the literature could be applied to control the robot’s position and potentially orientation. We have also investigated a simpler, decentralized controller that, while unable to control the orientation of the robot, would have the advantage of scaling better with the number of modules used. The decentralized controller is parameter-free and does not require any communication or memory. It does not need to retain knowledge of the target position and acts only on what the sensors detect at the current time.

Chen et al. [23] presents an analysis of how a passive 2-D object (rather than a modular robot) is pushed towards a target using an infinite number of point robots with equal force. We show that the analysis naturally extends to the robots presented here, under the assumption of quasi-static motion. This makes our decentralized algorithm provably correct for 3-D convex shapes.

We have conducted 3-D simulations of an MHP robot using both motion controllers. Given a sufficient granularity, the robot was able to produce reliable movement towards a target despite vast inaccuracies or faults affecting its sensors or actuators.

We have presented a physical prototype of a 2-D MHP robot. Using the decentralized controller this robot was able to successfully navigate towards a light source in all possible configurations using up to four modules. Robots with six modules were also found to be able to accomplish the task. A video showing the robot in operation is included in the supplementary material.

In the future an interesting direction of research would be to control the rotation in a decentralized way as well. This could be achieved by coupling the occlusion algorithm with a routing protocol to simulate a fluid based reaction wheel. Another direction could be to further miniaturize the physical system and extend it to 3-D. The routing units could be realized using piezoelectric micro-pumps and/or -valves, potentially allowing such modules to be produced at small scales and in large numbers.

VIII. ACKNOWLEDGEMENTS

We would like to thank Paul Eastwood for his help with setting up the experimental environment.

REFERENCES

- [1] M. Yim, W.-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. S. Chirikjian, "Modular self-reconfigurable robot systems," *IEEE Robot. Automat. Mag.*, vol. 14, no. 1, pp. 43–52, 2007.
- [2] K. Støy, D. Brandt, and D. J. Christensen, *Self-Reconfigurable Robots: An Introduction*. MIT Press, Cambridge, MA, 2010.
- [3] R. Groß and M. Dorigo, "Self-assembly at the macroscopic scale," *Proceedings of the IEEE*, vol. 96, no. 9, pp. 1490–1508, 2008.
- [4] S. C. Goldstein, J. D. Campbell, and T. C. Mowry, "Programmable matter," *Computer*, vol. 38, no. 6, pp. 99–101, 2005.
- [5] K. Gilpin, A. Knaian, and D. Rus, "Robot pebbles: One centimeter modules for programmable matter through self-disassembly," in *Proc. of the 2010 IEEE Int. Conf. on Robotics and Automation*, 2010, pp. 2485–2492.
- [6] M. Doniec, I. Vasilescu, C. Detweiler, and D. Rus, "Complete SE³ underwater robot control with arbitrary thruster configurations," in *Proc. of the 2010 IEEE Int. Conf. on Robotics and Automation*. IEEE, Los Alamitos, CA, 2010, pp. 5295–5301.
- [7] H. Kurokawa, K. Tomita, A. Kamimura, S. Kokaji, T. Hasuo, and S. Murata, "Distributed self-reconfiguration of M-TRAN III modular robotic system," *Int. J. Robot. Res.*, vol. 27, no. 3-4, pp. 373–386, 2008.
- [8] S. Kernbach, O. Scholz, K. Harada, S. Popescu, J. Liedke, *et al.*, "Multi-robot organisms: State of the art," in *ICRA 2010 Workshop on Modular Robotics*. IEEE, 2010, pp. 1–10.
- [9] J. W. Romanishin, K. Gilpin, and D. Rus, "M-blocks: Momentum-driven, magnetic modular robots," in *Proc. of the 2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. IEEE, 2013, pp. 4288–4295.
- [10] I. Vasilescu, P. Varshavskaya, K. Kotay, and D. Rus, "Autonomous modular optical underwater robot (AMOUR): Design, prototype and feasibility study," in *Proc. of the 2005 IEEE Int. Conf. on Robotics and Automation*. IEEE, Los Alamitos, CA, April 2005, pp. 1603–1609.
- [11] P. White, V. Zykov, J. Bongard, and H. Lipson, "Three dimensional stochastic reconfiguration of modular robots," in *Proc. of the 2005 Robotics: Science and Systems Conf.* MIT Press, Cambridge, MA, 2005, pp. 161–168.
- [12] V. Zykov and H. Lipson, "Experiment design for stochastic three-dimensional reconfiguration of modular robots," in *IROS 2007 Self-Reconfigurable Robotics Workshop*. IEEE, 2007.
- [13] I. Vasilescu, C. Detweiler, M. Doniec, D. Gurdan, S. Sosnowski, J. Stumpf, and D. Rus, "AMOUR V: A hovering energy efficient underwater robot capable of dynamic payloads," *Int. J. Robot. Res.*, vol. 29, no. 5, pp. 547–570, 2010.
- [14] M. T. Tolley and H. Lipson, "Fluidic manipulation for scalable stochastic 3D assembly of modular robots," in *Proc. of the 2010 IEEE Int. Conf. on Robotics and Automation*. IEEE, Los Alamitos, CA, 2010, pp. 2473–2478.
- [15] M. T. Tolley, M. Kalontarov, J. Neubert, D. Erickson, and H. Lipson, "Stochastic modular robotic systems: A study of fluidic assembly strategies," *IEEE Trans. Robot.*, vol. 26, no. 3, pp. 518–530, 2010.
- [16] S. Mintchev, C. Stefanini, A. Girin, S. Marrazza, S. Orofino, V. Lebastard, L. Manfredi, P. Dario, and F. Boyer, "An underwater reconfigurable robot with bioinspired electric sense," in *Proc. of the 2012 IEEE Int. Conf. on Robotics and Automation*. IEEE, Los Alamitos, CA, 2012, pp. 1149–1154.
- [17] R. Oung and R. D'Andrea, "The distributed flight array: Design, implementation, and analysis of a modular vertical take-off and landing vehicle," *Int. J. Robot. Res.*, vol. 33, no. 3, pp. 375–400, 2014.
- [18] J. Campbell and P. Pillai, "Collective actuation," *Int. J. Robot. Res.*, vol. 27, no. 3–4, pp. 299–314, 2008.
- [19] D. J. Christensen, J. Campbell, and K. Stoy, "Anatomy-based organization of morphology and control in self-reconfigurable modular robots," *Neural Comput. Applic.*, vol. 19, no. 6, pp. 787–805, 2010.
- [20] M. Doniec, C. Detweiler, and D. Rus, "Estimation of thruster configurations for reconfigurable modular underwater robots," in *Experimental Robotics*, ser. Springer Tracts in Advanced Robotics. Springer Verlag, Berlin, Germany, 2014, vol. 79, pp. 655–666.
- [21] B. von Haller, A. Ijspeert, and D. Floreano, "Co-evolution of structures and controllers for Neobot underwater modular robots," in *Proc. of the 8th European Conf. on Artificial Life*, ser. LNAI, vol. 3630. Springer Verlag, Berlin, Germany, 2005, pp. 189–199.
- [22] I. O'Hara, J. Paulos, J. Davey, N. Eckenstein, N. Doshi, T. Tosun, J. Greco, J. Seo, M. Turpin, V. Kumar, *et al.*, "Self-assembly of a swarm of autonomous boats into floating structures," in *Proc. of the 2014 IEEE Int. Conf. on Robotics and Automation*. IEEE, 2014, pp. 1234–1240.
- [23] J. Chen, M. Gauci, W. Li, A. Kolling, and R. Groß, "Occlusion-based cooperative transport with a swarm of miniature mobile robots," *IEEE Trans. Robot.*, vol. 31, no. 2, pp. 307–321, 2015.
- [24] R. Smith *et al.*, "Open dynamics engine," 2005, <http://www.ode.org>.
- [25] J. Arvo, "Fast random rotation matrices," in *Graphics Gems III*. Academic Press Professional, San Diego, CA, 1992, pp. 117–120.