Decentralized Pose Control of Modular Reconfigurable Robots Operating in Liquid Environments

João V. Amorim Marques[1], Anıl Özdemir[1], Matthew J. Doyle[1], Daniela Rus[2], and Roderich Groß[1,2]

***Abstract*— Modular reconfigurable robots are touted for their flexibility, as their bodies can assume a wide range of shapes. A particular challenge is to make them move efficiently in 3D without compromising the scalability of the system. This paper proposes decentralized and fully reactive controllers for pose control of 3D modular reconfigurable robots. The robots operate in liquid environments, and move by routing fluid through themselves. Each module uses only two bits of sensory information per face. Additionally, the modules can use up to five bits of information that are exchanged via shared power lines. We prove that robots of convex shape are guaranteed to reach a goal object with a preferred orientation. Using computer simulations of Modular Hydraulic Propulsion robots, all controllers are assessed for different environments, system sizes and noise, and their performances compared against a centralized controller. Given the simplicity of the solutions, modules could be realized at scales below a millimeter-cube, where robots of high spatial resolution could perform accurate movements in 3D liquid environments.**

I. INTRODUCTION

Modular reconfigurable robots can take different shapes and therefore cope with a variety of tasks and conditions [1], [2]. Proposed applications include search and rescue [3], inspection of underwater environments [4] and construction of temporary structures [5]. In these applications, the ability of robots to control their pose is crucial. A robot may need to navigate a narrow passage, or inspect or manipulate objects [6]. Whereas these abilities have been demonstrated with individual modules, realizing them at the level of a large ensemble remains a challenge. In addition, through miniaturization, the modular resolution of robots of a given size can be increased, enabling them to perform more accurate movements. Miniaturized modular robots could also be used in novel applications, such as micro-medicine. However, this requires the modules to have exceedingly low hardware resources, making the design of controllers a challenge.

To engender scalability, we seek modular systems that are decentralized, and use simplistic hardware and software. Current solutions for pose control are either centralized or require the use of complex sensors or controllers. Tactically Expendable Marine Platform [7] and Roboat [8] modules are capable of 3 degrees of freedom (DoF) motion and can self-reconfigure into temporary 2D structures on the surface of water. However, control is centralized and requires an external camera or GPS to obtain the pose of each module. The AMOUR robot [9] is capable of 6 DoF motion underwater, yet can only reconfigure in 1D. Control is centralized, but works with any thruster configuration. ModQuad [10] is an aerial system capable of 4 DoF motion. While each module computes its own control inputs, the desired pose is provided by a central planner. The Distributed Flight Array (DFA) [11], [12] is another aerial system capable of decentralized pose control. It requires the use of external sensing to determine its horizontal position and yaw angle. Both ModQuad and DFA are limited to a 2D reconfiguration space. The Modular Hydraulic Propulsion (MHP) concept [13] proposes a robot with a cubic lattice capable of 6 DoF motion in a liquid environment. MHP robots are modular networks that propel by routing fluid through themselves. In previous work, a decentralized 2 DoF motion controller was proposed for translation towards a goal. The physical MHP platform [13] is limited to 2D.

In this paper, we propose decentralized 5 DoF pose controllers for convex-shaped MHP robots that are fully autonomous. The controllers solve the problem of approaching a goal with a preferred orientation. They use only simple binary pumps and sensors, and require no run-time memory. To the best of our knowledge, they are the first solutions achieving 5 DoF motion of modular robots in a fully autonomous, reactive and decentralized way. The simplicity of the solutions allows future miniaturization of the system, whereas the lattice structure gives rise to a large reconfiguration space.

II. PROBLEM FORMULATION

Consider an unbounded, liquid environment $\mathcal{E} = \mathbb{R}^N$, $N \in \{2, 3\}$ of density ρ, which contains a static goal at $\mathbf{g} \in \mathcal{E}$ and a robot, but is otherwise obstacle-free. The robot's body has a rigid shape, $\mathcal{A} \subset \mathbb{R}^N$. Let $h_t(\mathbf{x}) : \mathbb{R}^N \to \mathcal{E}$ denote a distance-preserving transformation from the robot's local reference frame to the global reference frame at time t. The robot then occupies

$$h_t(\mathcal{A}) = \{h_t(\mathbf{x}) \in \mathcal{E} | \mathbf{x} \in \mathcal{A}\}. \quad (1)$$

The robot is made of *modules* that are N-dimensional hyper-cubes of density ρ and unit side length, that is, the modules are neutrally buoyant. A module can be physically linked to other modules with each of its $2N$ faces. By doing so, robots of different shapes (i.e., N-dimensional lattice configurations) can be built. In the following, we assume the shape to be a hyper-rectangle, $\mathcal{A} = [-a_1/2, a_1/2] \times$

[1]J. V. Amorim Marques, A. Özdemir, M. J. Doyle, and R. Groß are with the Department of Automatic Control and Systems Engineering, The University of Sheffield, Mappin St, Sheffield, S1 3JD, UK {jvcmarques1,a.ozdemir,matthew.doyle, r.gross}@sheffield.ac.uk

[2]D. Rus is and R. Groß has been with the Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, 32 Vassar St, Cambridge, MA 02139, USA rus@mit.edu

$\cdots \times [-a_N/2, a_N/2] \subset \mathbb{R}^N$, where $a_j \in \mathbb{Z}_+$. In these configurations, the robot comprises $n = \prod_{j=1}^{N} a_j$ modules.

We assume that all modules of a robot are oriented in a consistent way, thereby having a common sense of orientation within the robot's own reference frame. Faces $2j-1$ and $2j$, $j = 1, \ldots, N$, have their outward normal vectors anti-parallel and parallel to axis X_j^{local}, respectively.

Each module contains one binary contact sensor per face, which detects whether another module is attached to that face:

$$c_i = \begin{cases} 1, & \text{face } i \text{ is linked with another module;} \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

Modules with $\exists i \in \{1, \ldots, 2N\} : c_i = 0$ are referred to as *boundary* modules. All other modules are referred to as *interior* modules.

Each module contains one binary goal sensor per face. The sensor detects whether the goal is visible, that is, not occluded by the robot. It is mounted in the face center. Let $\mathbf{s}_i(t) \in \mathcal{E}$ denote the position of goal sensor i at time t. Then,

$$d_i = \begin{cases} 1, & h_t(\mathcal{A}) \cap \{\alpha \mathbf{s}_i(t) + (1-\alpha)\mathbf{g}| \\ & \alpha \in [0,1)\} = \varnothing; \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

All modules of a given robot share M binary power lines. The state of the k^{th} power line, b_k, is

$$b_k = \begin{cases} 1, & \text{at least one module activates line } k; \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

In other words, $b_k = 0$ if and only if no module actives power line k.

Each module contains a reservoir containing the same liquid as the environment. The reservoir is connected to all of the module's faces. Connected modules form a liquid network. Each face of a module contains a binary pump. When turned on ($p_i = 1$) the pump routes liquid from the reservoir through the face into the adjacent module, if present, or the environment, otherwise. When turned off ($p_i = 0$) the liquid can freely move in either direction.

A. Objective

We assume that all modules are aware of a preferred orientation, $O \in \{1, \ldots, 2N\}$, with respect to the robot's local reference frame, and that $O = 1$.[1] The objective of the robot is to reach the goal, in finite time, with orientation O. In particular, the face of the robot that is oriented towards O (hereafter referred to as the preferred face) needs to make the initial physical contact with the goal. Formally, let $\mathcal{A}_O$ denote the set of points of the modules' external faces that correspond to preferred orientation O, $\mathcal{A}_O = \{-a_1/2\} \times [-a_2/2, a_2/2] \times \cdots \times [-a_N/2, a_N/2] \subset \mathcal{A}$. The objective is satisfied if and only if $\exists T : (\forall t < T : h_t^{-1}(\mathbf{g}) \notin \mathcal{A}) \wedge (h_T^{-1}(\mathbf{g}) \in \mathcal{A}_O)$.

[1]The controllers can be adapted to any preference, or, alternatively, be generalized, if preference O is provided as input.

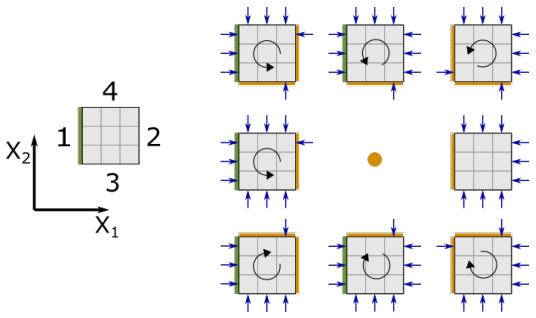

Fig. 1. Decentralized pose control without communication (`2D-0SP` controller). Shown are eight scenarios of a robot that needs to touch the goal (orange point) with its preferred face (face 1). The goal is visible from any module face marked with orange. The pumps in all other faces are active, causing the robot to translate towards the goal. Up to two additional pumps are active, causing the robot to rotate.

We assume that at time $t = 0$, $|h_0(\mathbf{0}) - \mathbf{g}| > \sqrt{(\frac{a_1}{2})^2 + \cdots + (\frac{a_N}{2})^2}$. In other words, in the initial position, the robot can freely rotate without touching the goal.

III. DECENTRALIZED CONTROL

Every module uses an identical controller. It activates power line k if an activation policy, $f_{b,k}(\mathbf{c}, \mathbf{d})$, is evaluated as 1.[2] It sets the states of its pumps using a control policy, $f_p(\mathbf{c}, \mathbf{d}, \mathbf{b})$. Both policies are reactive—they do not need run-time memory. While a module can take into account its local connectivity ($\mathbf{c}$), it is unaware of further positional information within the modular robot.

A. Pose control in 2D environments

We present three controller solutions to the aforementioned problem. The first requires no communication between the modules. The other two require that the modules exchange, respectively, 1 or 2 bits of information via the shared power line.

1) No shared power line (`2D-0SP` controller): This controller does not require a shared power line (i.e., $M = 0$). The control policy $f_p(\mathbf{c}, \mathbf{d})$ is given by

$$p_1 = \bar{c}_1 \bar{d}_1 \vee \bar{c}_1 \bar{c}_3 d_3 \vee \bar{c}_1 \bar{c}_4 d_4, \quad (5)$$

$$p_2 = \bar{c}_2 \bar{d}_2 \vee \bar{c}_2 \bar{c}_4 \bar{d}_4, \quad (6)$$

$$p_3 = \bar{c}_3 \bar{d}_3 \vee \bar{c}_3 \bar{c}_2 d_3, \quad (7)$$

$$p_4 = \bar{c}_4 \bar{d}_4 \vee \bar{c}_4 \bar{c}_2 d_4. \quad (8)$$

The first term of p_i is $\bar{c}_i \bar{d}_i$. In other words, a pump gets activated if it belongs to an external face ($\bar{c}_i$) from which the goal is not visible ($\bar{d}_i$), due being occluded by the robot itself. For convex robots, this occlusion-based strategy causes pure translation towards the goal [14], [13]. If the robot is correctly aligned towards the goal (i.e., $d_2 = d_3 = d_4 = 0$), it only translates (see Fig. 1). Otherwise, rotation is needed. As no communication is permitted, the modules contribute to rotation only if they have sufficient local information about

[2]Note that the activation policy does not take into account the state of actuators or of the power line itself. This is to prevent the power line from serving as a form of memory.

the global scenario to guarantee that their actions are in agreement. For this reason, only modules at the "corner" of a configuration (e.g., $\bar{c}_1\bar{c}_3$) contribute to rotation. Fig. 1 depicts all eight possible global scenarios, and shows for each one the subset of the corner modules that contribute to rotation.

2) 1-bit shared power line (`2D-1SP` *controller):* This controller requires a single shared power line (i.e., $M = 1$). The activation policy, $f_{b,1}(\mathbf{c}, \mathbf{d})$, is given by

$$b_1 = d_2 \vee d_3 \vee d_4. \tag{9}$$

In other words, the power line is active if and only if the goal is perceived by a non-preferred face. The control policy, $f_p(\mathbf{c}, \mathbf{d}, \mathbf{b})$, is given by

$$p_1 = b_1\bar{c}_1\bar{c}_3, \tag{10}$$
$$p_2 = b_1\bar{c}_2\bar{c}_4 \vee \bar{b}_1\bar{c}_2, \tag{11}$$
$$p_3 = b_1\bar{c}_3\bar{c}_2, \tag{12}$$
$$p_4 = b_1\bar{c}_4\bar{c}_1. \tag{13}$$

In other words, if the power line is active ($b_1 = 1$), the robot rotates, otherwise it translates.

The controller allows for only counter-clockwise rotation. As all corner modules contribute to the torque that creates the rotation, no undesired translation is produced. During translation, only the modules that belong to the face opposite to the preferred face are active.

3) 2-bit shared power line (`2D-2SP` *controller):* This controller requires two shared power lines (i.e., $M = 2$). Their activation policies, $f_{b,1}(\mathbf{c}, \mathbf{d})$ and $f_{b,2}(\mathbf{c}, \mathbf{d})$, are given by

$$b_1 = d_2 \vee d_3, \tag{14}$$
$$b_2 = d_4. \tag{15}$$

The control policy, $f_p(\mathbf{c}, \mathbf{d}, \mathbf{b})$, is given by

$$p_1 = b_1\bar{b}_2\bar{c}_1\bar{c}_3 \vee b_2\bar{c}_1\bar{c}_4, \tag{16}$$
$$p_2 = b_1\bar{b}_2\bar{c}_2\bar{c}_4 \vee b_2\bar{c}_2\bar{c}_3 \vee \bar{b}_1\bar{b}_2\bar{c}_2, \tag{17}$$
$$p_3 = b_1\bar{b}_2\bar{c}_3\bar{c}_2 \vee b_2\bar{c}_3\bar{c}_1, \tag{18}$$
$$p_4 = b_1\bar{b}_2\bar{c}_4\bar{c}_1 \vee b_2\bar{c}_4\bar{c}_2. \tag{19}$$

The control policy allows for counter-clockwise rotation ($b_1\bar{b}_2 = 1$), clockwise rotation ($b_2 = 1$), and translation ($\bar{b}_1\bar{b}_2 = 1$).

B. Pose control in 3D environments

We present three memory-less control strategies that are extensions of the 2D variants (see Section III-A). Note that when adding the third dimension, the problem becomes more complex, as in general the robot will have to choose from multiple axes of rotation.

1) No shared power line (`3D-0SP` *controller):* This controller does not require a shared power line (i.e., $M = 0$). The control policy $f_p(\mathbf{c}, \mathbf{d})$ is given by

$$p_1 = \bar{c}_1\bar{d}_1 \vee \bar{c}_1\bar{c}_3d_3 \vee \bar{c}_1\bar{c}_4d_4 \vee \bar{c}_1\bar{c}_5d_5 \vee \bar{c}_1\bar{c}_6d_6, \tag{20}$$
$$p_2 = \bar{c}_2\bar{d}_2 \vee \bar{c}_2\bar{c}_4\bar{d}_4, \tag{21}$$
$$p_3 = \bar{c}_3\bar{d}_3 \vee \bar{c}_3\bar{c}_2d_3, \tag{22}$$
$$p_4 = \bar{c}_4\bar{d}_4 \vee \bar{c}_4\bar{c}_2d_4, \tag{23}$$
$$p_5 = \bar{c}_5\bar{d}_5 \vee \bar{c}_5\bar{c}_2d_5, \tag{24}$$
$$p_6 = \bar{c}_6\bar{d}_6 \vee \bar{c}_6\bar{c}_2d_6. \tag{25}$$

The policy allows for clockwise and counter-clockwise rotation along two axes—X_2^{local} and X_3^{local}. Rotations along the two axes can occur simultaneously, effectively generating two further axes of rotation. For $d_5 = d_6 = 0$, the policy is identical to `2D-0SP` [see Eqs. (5)–(8)]. Note, however, that rather than only the corner modules, all modules belonging to the same edge of the robot contribute to the rotation.

2) 2-bit shared power line (`3D-2SP` *controller):* This controller requires two shared power lines (i.e., $M = 2$). The activation policies $f_{b,1}(\mathbf{c}, \mathbf{d})$ and $f_{b,2}(\mathbf{c}, \mathbf{d})$ are given by

$$b_1 = d_2 \vee d_3 \vee d_4, \tag{26}$$
$$b_2 = d_5 \vee d_6. \tag{27}$$

Note that Eq. (26) is the same as for `2D-1SP` [Eq. (9)].

The control policy, $f_p(\mathbf{c}, \mathbf{d}, \mathbf{b})$ is given by

$$p_1 = b_1\bar{b}_2\bar{c}_1\bar{c}_3 \vee \bar{b}_1b_2\bar{c}_1\bar{c}_6, \tag{28}$$
$$p_2 = b_1\bar{b}_2\bar{c}_2\bar{c}_4 \vee \bar{b}_1b_2\bar{c}_2\bar{c}_5 \vee \bar{b}_1\bar{b}_2\bar{c}_2, \tag{29}$$
$$p_3 = b_1\bar{b}_2\bar{c}_3\bar{c}_2 \vee b_1b_2\bar{c}_3\bar{c}_5, \tag{30}$$
$$p_4 = b_1\bar{b}_2\bar{c}_4\bar{c}_1 \vee b_1b_2\bar{c}_4\bar{c}_6, \tag{31}$$
$$p_5 = \bar{b}_1b_2\bar{c}_5\bar{c}_1 \vee b_1b_2\bar{c}_5\bar{c}_4, \tag{32}$$
$$p_6 = \bar{b}_1b_2\bar{c}_6\bar{c}_2 \vee b_1b_2\bar{c}_6\bar{c}_3. \tag{33}$$

The control policy allows for counter-clockwise rotation around either of three axes—X_1^{local}, X_2^{local} and X_3^{local}. For $d_5 = d_6 = 0$ (implying $b_2 = 0$), the policy is identical to `2D-1SP` [see Eqs. (10)–(13)].

3) 5-bit shared power line (`3D-5SP` *controller):* This controller requires five shared power lines (i.e., $M = 5$). The activation policy $f_{b,i}(\mathbf{c}, \mathbf{d})$ is given by $b_i = d_i$, though without any change in behavior, b_1 can be omitted. The control policy is provided in [15], due to the length of the Boolean expressions. It allows for clockwise and counter-clockwise rotation along two axes—X_2^{local} and X_3^{local}. Rotations along the two axes can occur simultaneously, effectively generating two further axes of rotation.

IV. MATHEMATICAL ANALYSIS

This section provides an analysis of the decentralized controllers `2D-1SP` and `3D-2SP` for convex-shaped robots. We make the following assumptions: (i) Time t is continuous, instantaneous information is obtained from sensors ($\mathbf{c}$ and $\mathbf{d}$) and power lines ($\mathbf{b}$), and changes in pump ($\mathbf{p}$) and power line ($\mathbf{b}$) activation take immediate effect; (ii) the robot is subject to translational and rotational drag forces. It accelerates instantaneously, emulating quasi-static motion. It is thus either at rest or moving with terminal velocity.

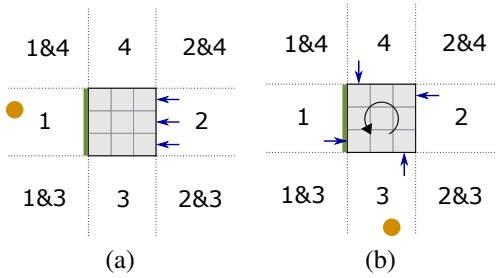

Fig. 2. Illustration of a modular robot that is tasked to approach a goal (orange point) with a preferred orientation (green face, using the `2D-1SP` controller. In (a), the robot is correctly oriented, and only needs to translate towards the goal. It does so by activating the pumps (blue arrows) on the face that is opposite to the preferred face. In (b), the robot rotates counter-clockwise, by activating four pumps (blue arrows) of its corner modules, until the preferred orientation is reached.

Theorem 1. *A goal that has not been reached is detected by all goal sensors of at least one of the robot's faces.*

Proof. We assume that the opposite was true, that is, $\exists t : \mathbf{g} \notin h_t(\mathcal{A}) \wedge (\nexists i : \bar{c}_i \Rightarrow d_i)$. Consider the position of the goal at time t in the local coordinate system of the robot, $\mathbf{g}^{\text{local}} = h_t^{-1}(\mathbf{g}) \in \mathbb{R}^N$. It follows that $\mathbf{g}^{\text{local}} \in \mathbb{R}^N \setminus \mathcal{A} = \mathbb{R}^N \setminus [-a_1/2, a_1/2] \times \cdots \times [-a_N/2, a_N/2] = \cup_j \mathcal{H}_j$, where $\mathcal{H}_j = \{(x_1, \ldots, x_N) \in \mathbb{R}^N | (x_j < -a_j/2) \vee (a_j/2 < x_j)\}$. Therefore, $\exists j : \mathbf{g}^{\text{local}} \in \mathcal{H}_j$. There exists at least one module with goal sensor $2j-1$ at $\mathbf{s}^{\text{local}}_{2j-1}[j] = -a_j/2$ and with $\bar{c}_{2j-1} = 1$, and at least one module with goal sensor $2j$ at $\mathbf{s}^{\text{local}}_{2j}[j] = a_j/2$ and with $\bar{c}_{2j} = 1$. Hence, $\mathcal{A} \cap \{\alpha \mathbf{s}^{\text{local}}_{2j-1} + (1-\alpha)\mathbf{g}^{\text{local}} | \alpha \in [0,1)\} = \varnothing$ or $\mathcal{A} \cap \{\alpha \mathbf{s}^{\text{local}}_{2j} + (1-\alpha)\mathbf{g}^{\text{local}} | \alpha \in [0,1)\} = \varnothing$ must hold true. As h_t is bijective, it follows from Eq. (3) that either $d_{2j-1} = 1$ or $d_{2j} = 1$, which violates our assumption. □

A. Pose control in 2D environments

Consider the `2D-1SP` controller presented in Section III-A.2. When executed on an interior module ($\forall i : c_i = 1$), if follows from Eqs. (10)–(13) that no pump is activated ($p_1 = p_2 = p_3 = p_4 = 0$); fluid can freely flow in any direction through the module. When executed on a boundary module, pumps on faces linked to other modules are not activated either [Eqs. (10)–(13)]. Therefore, the following analysis focuses on pumps on the robot's external faces. The robot has $2(a_1 + a_2)$ such pumps.

Lemma 1. *An $a_1 \times a_2$ robot that faces the goal with only the preferred face, that is, $\bigvee_{i \neq 1} d_i = 0$, completes the task in finite time.*

Proof. From $\bigvee_{i \neq 1} d_i = 0$ and Eq. (9), it follows that $b_1 = 0$. Using Eqs. (10)–(13), we obtain $p_1 = p_3 = p_4 = 0$ and $p_2 = \bar{c}_2$. Hence all modules that have an external face with a normal anti-parallel to the preferred face's normal activate the corresponding pump. Each active pump produces a constant thrust force, $f_p > 0$, which is anti-parallel to the preferred face's normal. As all active pumps (and hence forces) are arranged symmetrically, no torque is produced. The robot undergoes a pure translation towards the goal (see Fig. 2a.) The direction of movement is parallel to the two lines that define, respectively, the half-plane sensing regions of goal sensors 3 and 4. As a consequence, $\bigvee_{i \neq 1} d_i = 0$ for as long as the robot does not touch the goal. The robot has a_2 active pumps giving a net force of $a_2 f_p$ along the preferred face's normal. This is a positive constant and, from the quasi-static motion assumption, produces a positive constant velocity. As the distance between the robot and goal is finite and decreasing at a constant rate, the goal is reached in finite time. □

Theorem 2. *An $a_1 \times a_2$ robot (with $a_j \geq 2$ for some j) completes the task in finite time.*

Proof. According to Lemma 1, we only need to consider the case $\bigvee_{i \neq 1} d_i = 1$. From Eq. (9), it follows that $b_1 = 1$. As the second term $(\bar{b}_1 \bar{c}_2)$ in Eq. (11) disappears, Eqs. (10)–(13) assume a symmetric form, causing the corner modules of the rectangular configuration to activate four pumps in total (see Fig. 2b). As the activated pumps face in opposing directions, no translation occurs. The geometric center of the robot coincides with the center of mass. The pumps firing parallel and anti-parallel to the X_1 axis produce a torque of $2(\frac{a_2}{2} - \frac{1}{2})f_p = (a_2 - 1)f_p$, whereas the pumps firing parallel and anti-parallel to the X_2 axis produce a torque of $(a_1 - 1)f_p$. The combined torque is $(a_1 + a_2 - 2)f_p \geq f_p$, which is positive constant. As the moment of inertia of the robot is constant, the torque results in a positive, constant angular velocity (assuming quasi-static motion). The robot hence rotates counter-clockwise until $\bigvee_{i \neq 1} d_i = 0$, in which case Lemma 1 applies. The time to rotate is bounded by the time for a full revolution; the latter is constant given the angular velocity is constant. □

Corollary 1. *A 1×1 robot is not guaranteed to complete the task in finite time.*

Proof. As can be seen from the proof of Theorem 2, a single module is unable to rotate, as the torque is $a_1 + a_2 - 2 = 0$. The reason for this is the rotation and reflection symmetry of the module, where each pump's force and position vectors are anti-parallel to each other. □

B. Pose control in 3D environments

Consider the `3D-2SP` controller presented in Section III-B.2. As for the 2D post control analysis, only pumps in the robot's external faces need to be considered. The robot has $2(a_1a_2 + a_1a_3 + a_2a_3)$ such pumps.

Lemma 2. *An $a_1 \times a_2 \times a_3$ robot that faces the goal with only the preferred face, that is, $\bigvee_{i \neq 1} d_i = 0$, completes the task in finite time.*

Proof. The proof is omitted, as a straight-forward extension of the proof of Lemma 1. □

Theorem 3. *An $a_1 \times a_2 \times a_3$ robot (with $a_k, a_l \geq 2$ for some $k \neq l$) completes the task in finite time.*

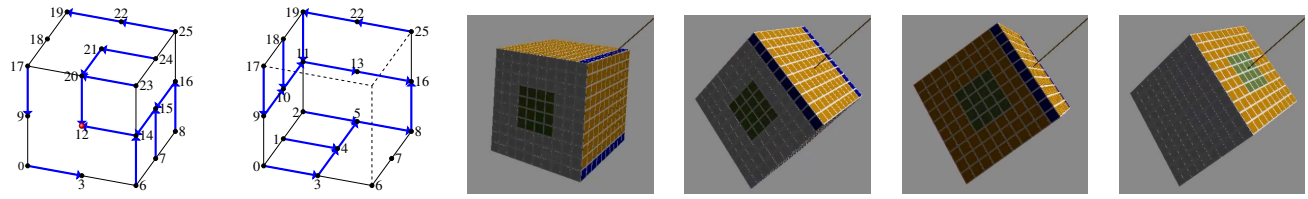

Fig. 3. Visualization of the graph representing the 26 global sensing states (nodes) of a convex-shaped 3D robot with respect to the goal. Each node represents a region (corner point, edge without corner points, face without edges) that contains the robot's closest point to the goal. Transitions between nodes (directed edges) are those chosen by the decentralized `3D-2SP` controller, which has only partial information about the global state. For any given node, the controller follows a unique path that is guaranteed to terminate in the preferred state (node 12) after at most eight transitions. In (a) the front, right and top of the graph are represented. In (b) the back, left and bottom of the graph are represented. The screenshots show a robot as it transitions (c) from 24 to 21, (d) from 21 to 20, (e) from 20 to 12, and (f) in final state 12. Lines indicate the direction of the goal from the robot's center. Blue faces indicate activated pumps. Yellow faces indicate non-occluded faces.

Proof. According to Lemma 2, we need to consider only the case $\bigvee_{i\neq 1} d_i = 1$. From Eqs. (26) and (27), it follows that $b_1 \vee b_2 = 1$. As such, the third term of Eq. (29) disappears. Depending on the values of b_1 and b_2, two of Eqs. (28)–(33) return 0, whereas the remaining four equations assume a symmetric form, causing the modules belonging to the robot's edges that are parallel to an axis, X_j^{local}, to activate $4a_j$ pumps in total. This generates a 2D rotation around axis X_j^{local}. By applying the analysis for Theorem 2, one can show that a new situation, with different values b_1 or b_2, is reached in finite time. The robot can be in any of 26 global sensing states[3], which are depicted in Fig. 3a–b. The robot transitions between these states by using 2D rotations along either of the robot's reference frame axes. Fig. 3a–b shows the graph of transitions realized by the controller. The graph is acyclic. Moreover, from any starting orientation (i.e., global sensing state) there is a unique path to reach the desired orientation, which corresponds to global sensing state 12. For the example in Fig. 3c–f, the path involves three steps. As any path can be completed by at most eight steps (i.e., 2D rotations), the overall task is completed in finite time. □

V. SIMULATION STUDIES

A. Simulation setup

The simulator was built using the open-source Open Dynamics Engine (ODE) library [16]. All studies simulate a 3D, liquid environment that contains a modular robot and a goal. The modules are 1 cm cubes. The goal is a static sphere of 1 cm diameter. The liquid and modules have the same density $\rho = 1\,\text{g/cm}^3$. As a full fluid dynamics treatment would be too computationally expensive [17], drag forces are determined using a quadratic approximation. Each active pump provides the same thrust, which is normalized such that the robot moves at 1 cm/s in an aquatic environment.

By default, the study is conducted using a $10 \times 10 \times 10$ robot, that is, with a robot of 10 cm body length. The robot is placed 85.5 cm away from the goal. Its starting orientation is chosen at random from a uniform distribution. To reach the goal, the robot has to translate a distance of up to 8 body lengths. A simulation trial terminates if the robot collides with the goal, or after 1200 s, whichever comes first. The trial is considered successful if the preferred face of the robot collided with the goal.

[3]We refer to these states as global, as the modules do not have access to complete state information.

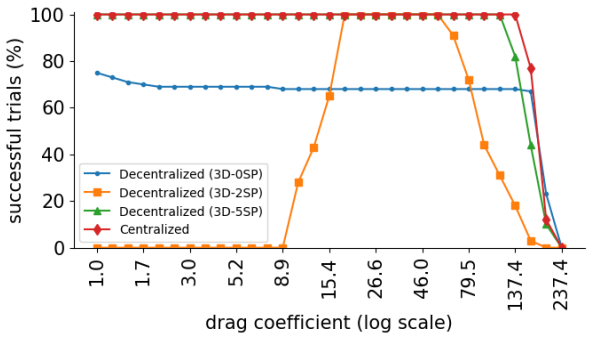

Fig. 4. Performance of a $10 \times 10 \times 10$ robot in environments with different drag coefficient (100 observations per setting). A trial is successful if the robot in its preferred orientation collided with the goal within a fixed time period.

We test the decentralized controllers presented in Section III-B: `3D-0SP`, `3D-2SP` and `3D-5SP`. The results are compared against a centralized controller from the literature [18]. The centralized controller uses the same set of binary pumps. It exploits additional knowledge, the relative positions and orientations of all the modules with respect to the goal. The parameters of the centralized controller were calibrated using a grid search.

B. Impact of drag

We tested drag coefficients $c_d = 1.2^k, k \in \{1, 2, \ldots, 30\}$. For each, 100 trials were conducted.

Fig. 4 shows the percentage of successful trials for all controllers. In general, for drag coefficients $c_d > 225$, the goal cannot be reached even when travelling at terminal velocity for the entire trial duration. The `3D-0SP` controller succeeds in about 70% of the trials for all but the highest drag coefficients. In the unsuccessful trials the robot reaches the goal with a non-preferred orientation. For this controller only a subset of the edges contribute to rotation, and this is not always sufficient to correctly align the robot prior to reaching the goal. The `3D-2SP` controller performs flawlessly for

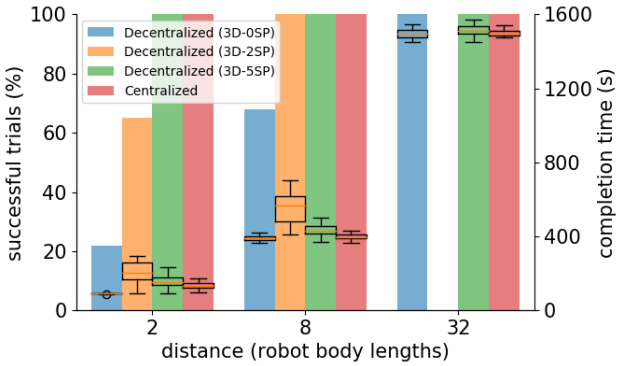

Fig. 5. Performance of a $10 \times 10 \times 10$ robot when the goal is initially a short, medium, or large distance away from the robot (100 observations per setting). The controllers are (from the left to the right): the decentralized controllers, `3D-0SP`, `3D-2SP`, and `3D-5SP`, and the centralized controller. Bars represent the percentage of successful trials. Box plots represent completion times (successful trials only).

drag coefficients in the range $18.5 \leq c_d \leq 55.2$. For lower drag coefficients, the drag is insufficient to counter inertial forces, causing over-shoot during rotation. The `3D-5SP` and centralized controllers perform flawlessly for all but the highest drag coefficients. As they can rotate counter-clockwise and clockwise, over-shoot can be corrected.

For the remainder of this paper, a drag coefficient of $c_d = 20$ is used.

C. Impact of goal distance

We study the problem of controlling the pose of the robot over short (2 body lengths), medium (8 body lengths), and long (32 body lengths) distances from the goal. The trial duration is adjusted accordingly to 300 s, 1200 s and 4800 s.

Fig. 5 shows the percentage of successful trials as well as the time it takes to reach the goal in the successful trials. The `3D-0SP` controller exhibits excellent performance if the goal is far away from the robot's starting location. At short distance, however, the performance is poor. As the robot rotates and translates simultaneously, shorter distances do not allow the robot to rotate into the desired orientation before reaching the goal. This effect is amplified by the fact that fewer modules contribute to rotation than for the other controllers. The `3D-2SP` controller fails to complete some of the trials when the goal is at short distance. This controller first rotates, and only then translates; high drag coefficients prevent the robot from reaching the goal within the time limit. For a distance of 8 body lengths, this effect disappears. For a distance of 32 body lengths, however, the robot performs poorly, as unable to sufficiently correct the over-shoot. For all travel distances, the `3D-5SP` and centralized controllers are always able to complete the task.

The `3D-0SP` controller has the best completion times, as it is always translating at maximum speed. The `3D-5SP` and centralized controllers perform comparatively well.

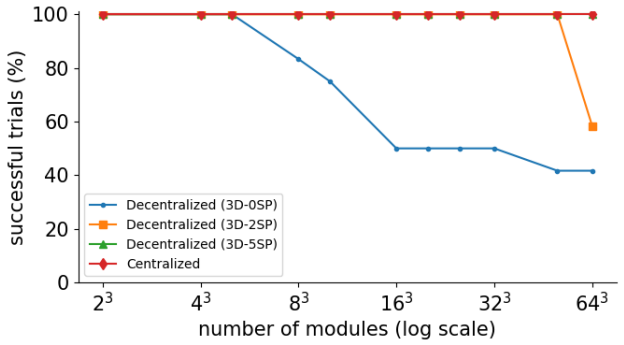

Fig. 6. Performance of robots of different modular resolution (100 observations per setting). While the number of modules and their size changes, the robot's overall dimensions remains the same. For details, see text.

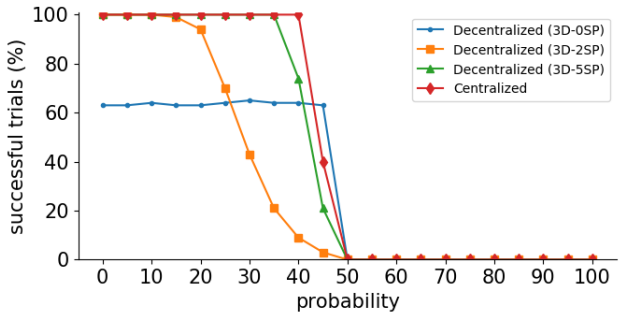

Fig. 7. Performance of a $10 \times 10 \times 10$ robot when subjected to different levels of actuation noise (100 observations per setting). For details, see text.

D. Impact of modular resolution

We study the ability of the controllers to cope with modular robots of different resolution. While the overall robot size remains the same, the number and size of modules varies. For each setup, the thrust force of the pumps is re-calibrated so that the net translational force remains the same.

Fig. 6 shows the results. The performance of the `3D-0SP` controller decreases substantially with increased modular resolution. The smaller the modules, the lower the torque. This means a more accurate, but slower, rotation. The performance of the `3D-2SP` controller is more robust, but also decays for very small module sizes (around 0.16 cm width). The `3D-5SP` performs flawlessly for the range of modular resolutions considered here, but we predict it will eventually degenerate at higher resolutions. The centralized controller performs well irrespective of the modular resolution (ignoring band-width limitations). As it has additional information (relative positions of all modules), it can activate half of externally-facing pumps of a robot to support a rotation.

E. Impact of actuation and sensor noise

First, we study modular robots exposed to actuation noise. Each pump chooses with probability P a uniformly random activation value, and, otherwise, uses the value obtained by the controller.

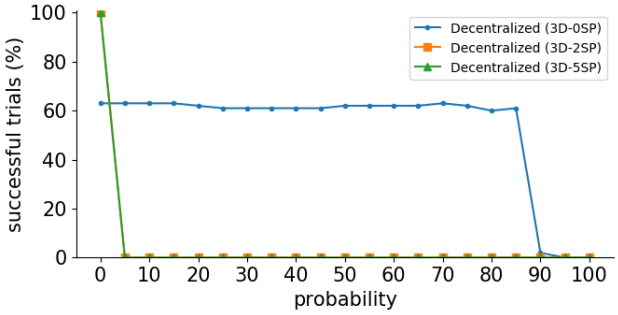

Fig. 8. Performance of a $10 \times 10 \times 10$ robot when subjected to different levels of sensory noise (100 observations per setting). For details, see text.

Fig. 7 shows the results. All controllers succeed in rejecting relatively large disturbances. Apart from the `3D-2SP` controller, the performance is not substantially affected up to a noise level of 40%. Thereafter the performance decreases rapidly. For noise levels of 50% and beyond the robot is unable to reach the goal within the time limit.

Second, we study modular robots exposed to sensor noise. Each sensor reports with probability P a uniformly random value, and, otherwise, uses the original value. The centralized controller is not included here, as it does not use sensors.

Fig. 8 shows the results. The `3D-2SP` and `3D-5SP` controllers are highly susceptible to sensor noise. As long as one sensor belonging to a face other than the preferred face gets a false positive reading, the corresponding power line is pulled up, causing the robot to rotate. The `3D-0SP` controller, on the other hand, performs robustly with up to 85% noise. As its modules do not share any information, false readings cannot propagate.

Run-time memory, if available, could reduce the susceptibility to sensor noise. One solution could be for the modules to filter their sensor and/or power line readings. Another solution could be for them to reach consensus on the power-line state (e.g., via quorum sensing).

VI. CONCLUSION

In this paper, we proposed a set of fully reactive and decentralized controllers for modular reconfigurable robots that perform pose control in liquid environments. The control strategies enable a convex-shaped robot to reach the goal with a preferred face. The robot uses simple binary sensors and pumps. Additionally, the strategy allows the modules to access up to 5 bits of information through shared power lines. We formally proved that one of the controllers is guaranteed to succeed in both 2D and 3D environments.

We evaluated the performance of the proposed 3D controllers in computer simulation studies. The `3D-0SP` controller performed robustly in high-drag environments, whereas the other controllers performed well in all environments. All controllers coped well with actuation noise, however, only the `3D-0SP` controller performed robustly with respect to sensor noise. The performance of `3D-0SP` and `3D-2SP` controllers was sensitive to the initial goal distance. The performance of the `3D-2SP` and `3D-5SP` controllers scaled well, but we predict it will degenerate for very high modular resolutions.

Future work will validate the controllers on the physical 2D MHP platform and consider the problem of pose control of robots of non-convex shapes. Moreover, monitoring tasks, such as the tracking of a dynamic goal [19], could be considered.

REFERENCES

[1] M. Yim, W. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. S. Chirikjian, "Modular self-reconfigurable robot systems [grand challenges of robotics]," *IEEE Robotics Automation Magazine*, vol. 14, no. 1, pp. 43–52, March 2007.

[2] K. Støy, D. Brandt, and D. J. Christensen, *Self-Reconfigurable Robots: An Introduction*. Cambridge, MA: The MIT Press, 2010.

[3] M. Yim, D. Duff, and K. Roufas, "Modular reconfigurable robots, an approach to urban search and rescue," in *Proc. of 1st Int. Workshop on Human-friendly Welfare Robotics Systems*, 2000, pp. 69–76.

[4] I. Vasilescu, P. Varshavskaya, K. Kotay, and D. Rus, "Autonomous modular optical underwater robot (AMOUR): Design, prototype and feasibility study," in *2005 IEEE Int. Conf. on Robotics and Automation*, 2005, pp. 1603–1609.

[5] I. O'Hara, J. Paulos, J. Davey, N. Eckenstein, N. Doshi, T. Tosun, J. Greco, J. Seo, M. Turpin, V. Kumar, and M. Yim, "Self-assembly of a swarm of autonomous boats into floating structures," in *2014 IEEE Int. Conf. on Robotics and Automation*, 2014, pp. 1234–1240.

[6] G. Sartoretti, S. Shaw, and M. A. Hsieh, "Distributed planar manipulation in fluidic environments," in *2016 IEEE Int. Conf. on Robotics and Automation*, 2016, pp. 5322–5327.

[7] J. Paulos, N. Eckenstein, T. Tosun, J. Seo, J. Davey, J. Greco, V. Kumar, and M. Yim, "Automated self-assembly of large maritime structures by a team of robotic boats," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 958–968, July 2015.

[8] W. Wang, L. A. Mateos, S. Park, P. Leoni, B. Gheneti, F. Duarte, C. Ratti, and D. Rus, "Design. modeling, and nonlinear model predictive tracking control of a novel autonomous surface vehicle," in *2018 IEEE Int. Conf. on Robotics and Automation*, 2018, pp. 6189–6196.

[9] 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," *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 547–570, 2010.

[10] D. Saldaña, B. Gabrich, G. Li, M. Yim, and V. Kumar, "ModQuad: The flying modular structure that self-assembles in midair," in *2018 IEEE Int. Conf. on Robotics and Automation*, 2018, pp. 691–698.

[11] R. Oung and R. D'Andrea, "The distributed flight array," *Mechatronics*, vol. 21, no. 6, pp. 908–917, 2011.

[12] ——, "The distributed flight array: Design, implementation, and analysis of a modular vertical take-off and landing vehicle," *The International Journal of Robotics Research*, vol. 33, no. 3, pp. 375–400, 2014.

[13] 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 *2016 IEEE Int. Conf. on Robotics and Automation*, 2016, pp. 5189–5196.

[14] J. Chen, M. Gauci, W. Li, A. Kolling, and R. Groß, "Occlusion-based cooperative transport with a swarm of miniature mobile robots," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 307–321, 2015.

[15] J. A. Marques, A. Özdemir, M. J. Doyle, D. Rus, and R. Groß. Online supplementary material. http://naturalrobotics.group.shef.ac.uk/supp/2019-003.

[16] R. Smith. (2005) Open dynamics engine. [Online]. Available: http://www.ode.org

[17] P. White, V. Zykov, J. Bongard, and H. Lipson, "Three dimensional stochastic reconfiguration of modular robots." in *Robotics: Science and Systems*, 2005, pp. 161–168.

[18] M. Doniec, I. Vasilescu, C. Detweiler, and D. Rus, "Complete SE3 underwater robot control with arbitrary thruster configurations," in *2010 IEEE Int. Conf. on Robotics and Automation*, 2010, pp. 5295–5301.

[19] S. Martínez and F. Bullo, "Optimal sensor placement and motion coordination for target tracking," *Automatica*, vol. 42, no. 4, pp. 661–668, 2006.