

# Coordinating Collective Locomotion in an Amorphous Modular Robot

Chih-Han Yu<sup>\*†</sup>  
chyu@fas.harvard.edu

Justin Werfel<sup>†</sup>  
justin.werfel@wyss.harvard.edu

Radhika Nagpal<sup>\*†</sup>  
rad@eecs.harvard.edu

<sup>\*</sup>School of Engineering and Applied Sciences

<sup>†</sup>Wyss Institute for Biologically Inspired Engineering  
Harvard University, Cambridge, MA, USA

**Abstract**—Modular robots can potentially assemble into a wide range of configurations to locomote in different environments. However, designing locomotion strategies for each configuration is often tedious and has generally relied on a priori known connection geometry. Here we present a framework for 2D modular robots made of square modules assembled with arbitrary geometry, which achieve collective and directed locomotion with no centralized controller. Individual modules communicate locally and *provably* achieve consensus in coordinating movement in a common travel direction. In experiments with simulations and hardware prototypes, we show that robots achieve effective locomotion, irrespective of the number of modules and their connectivity which can be highly asymmetric.

## I. INTRODUCTION

Modular robots, built from many identical and interchangeable components, have many potential advantages over traditional robots in flexibility, robustness, and adaptiveness. However, their reconfigurability raises a control problem: how can modules find a controller appropriate for any given configuration? Designing a control strategy for each possible configuration can be tedious and difficult, and even if this can be achieved, autonomously determining the proper controller can remain a challenge for the robot. Typically such robots are put together by hand with known geometry [1], [2], or determine their geometry and choose a corresponding set of module controllers from a fixed repertoire [3], [4], [5]. In many cases the controller is centralized, with a leader module that directs the actions of the others [4]. Furthermore, it is often hard to generalize controllers from one configuration to another.

On the other hand, many natural systems consisting of many interchangeable agents have no predetermined geometry and no centralized leader [6]. Cellular slime molds, a classic example, have a stage in their life cycle in which many individual amoebae aggregate into single multicellular slugs of variable morphology, which can then crawl as though a single coordinated unit [7]. Such examples lead us to consider modular robots with units connected in any arbitrary geometry (i.e., amorphous robots), and distributed controllers that allow for coordinated movement in a direction arrived at by the group as a whole.

In this work we describe hardware and control schemes for an amorphous modular robot whose unit modules are 2D squares made from telescoping links and compliant connectors [8], [9], [10] (Fig. 1). Each single module can

independently travel in any of the four cardinal directions, using a simple periodic actuator (discrete CPG). When multiple modules assemble to form an aggregate robot, local communication suffices for the group to agree on a common direction of travel and to coordinate their actuation cycles to travel in that direction.

A key feature of our approach is that the proposed algorithm is completely *decentralized*—there is no leader that determines coordination or direction. Instead the algorithm uses a strictly local and decentralized consensus process to coordinate among modules. The proposed approach *does not rely on prior knowledge of module connectivity* and can be easily applied to any configuration of modules. We analytically prove that all modules will converge to a coordinated locomotion pattern and a common direction to move, regardless of connectivity. This makes the approach robust and adaptive: modules are capable of performing locomotion however they are assembled.

In addition to our theoretical results, we also present hardware and simulation experiments. Using real robots, we show that modules can achieve coordinated and directed locomotion, even with initially conflicting goal directions among modules. Using physics-based simulations, we show that our coordination method can generate efficient locomotion in complex assemblies. Coordination is essential for effective locomotion: robots composed of ineffectively coordinated modules trying to travel in the same direction can barely move, while those using our method can travel at least as fast as individual modules. We also show our algorithms are scalable with robot size, with convergence time for direction agreement and locomotion coordination being proportional to connectivity graph diameter and number of modules, respectively.

## II. RELATED WORK

One major advantage of modular robots is they can potentially transform to different shapes and locomotion patterns when encountering different environments. The associated challenge is that for each configuration, the robot needs to find an effective locomotion pattern, determining how to actuate each module such that the system as a whole can locomote efficiently.

Several groups have proposed gait look-up table approaches: the modules first determine their configuration, then choose a corresponding set of controllers from a library

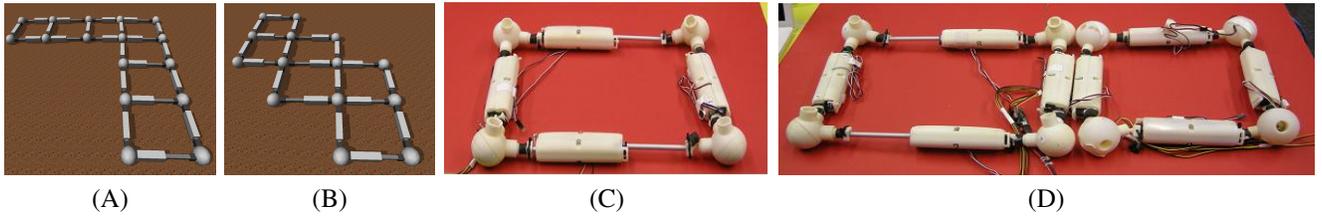


Fig. 1. Modules of different configurations can achieve efficient locomotion with our framework: (A) L-shape configuration (B) Ladder-like configuration (C) Single hardware module (D) Two linked hardware modules.

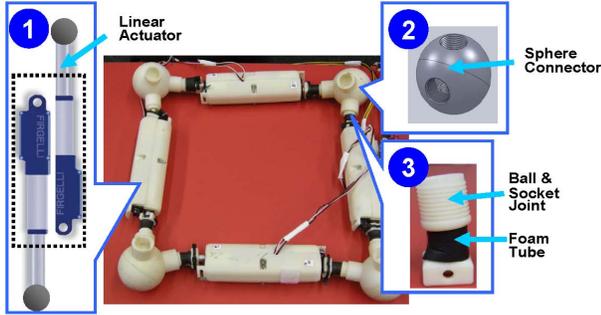


Fig. 2. A single robot module hardware prototype. Each link consists of a pair of linear actuators (1). Each actuator is connected to an interfacing sphere (2) via a flexible connector (3), which consists of a ball-and-socket joint inside a foam tube that gives it restoring force to maintain a preferred position.

[3], [4], [5]. This approach has been shown to allow specific robot configurations to locomote effectively in environments with obstacles. However, designing gaits manually for all robot configurations is laborious.

Other groups have studied locomotion strategies for chain-based robots based on central pattern generators (CPGs) [2], [11]. Environmental feedback can be incorporated into a CPG controller, such that the robot autonomously adjusts its gait while encountering new environments. One major drawback to the CPG approach is the dependence on the robot’s geometry; the controller designed for a specific configuration is generally hard to translate to other configurations.

Our approach is inspired by behaviors of cellular systems. In these systems, each cell acts adaptively based on simple rules, and the whole group can nevertheless exhibit coherent and efficient locomotion, as in cellular slime molds [7]. Based on this observation, our control law for each module is based on simple feedback from neighbors and can be applied to arbitrary module connectivity. By making a connection with distributed consensus algorithms [12], [13], we can further prove its correctness. Our approach is also inspired by Shimizu et al. [14]; the major difference is that their work focuses on achieving collective locomotion from modules that are incapable of independent movement, while we focus on coordinating movement of many linked self-mobile modules. We further address that modules can naturally have different initial objectives, e.g. intended traveling direction; our approach allows modules to arrive at a coherent decision and coordinate locomotion efficiently.

### III. MODULE OVERVIEW

In this section, we provide an overview of a single robot module in our system, including discussion of its mechanical design, communication, and control algorithm.

#### A. Mechanical Design and Communication of a Single Unit

A single module consists of four telescoping links connected in a square configuration by four nodes (Fig. 2). Compliant connections at the corner nodes let the frame deform and return to its original shape. Four spherical connectors at the corners of the square connect telescoping links together. Each link consists of a pair of linear actuators, bound together and pointing in opposite directions so as to achieve approximately a 3:1 extension capability if both actuators are extended simultaneously. This node/link design is inspired by previous deformable modular robots that have used such components to form other architectures such as tetrahedra [8], [9], [10].

As is typical for modular robotic systems [2], [8], we assume each module can exchange messages with its physically connected neighbors. While cells in multicellular systems may not necessarily communicate with arbitrary real-valued messages, this capability allows robot modules to achieve coordinated locomotion efficiently, as we show below.

#### B. Single Module Directed Locomotion Algorithm

The asymmetry available by extending only one actuator or the other in a link opens up the possibility of using such elements for locomotion in a desired direction. The following sequence of actions for the top and bottom links moving in unison produces net movement to the right, for instance (Fig. 3 (B)): extend the right arm only (1); retract the right arm while simultaneously extending the left one (2); retract the left arm (3). The cycle starts and ends with a configuration in which all actuators are retracted (4). At each step, elements pushed to the right are fewer and correspondingly lighter than those pushed to the left, and so net motion is to the right. Movement in any other cardinal direction can be achieved analogously.

Here we describe the controller formally. Each link is identified by a numerical index, which also corresponds to a robot-relative direction (Fig. 3 (A)). The actuation state (length) of each of the two linear actuators in a link is denoted by  $x_{i,j}$ , where  $i$  is the index number of the link and  $j$  is the index of the neighboring link. This index scheme allows us to uniquely specify the state of each linear actuator.

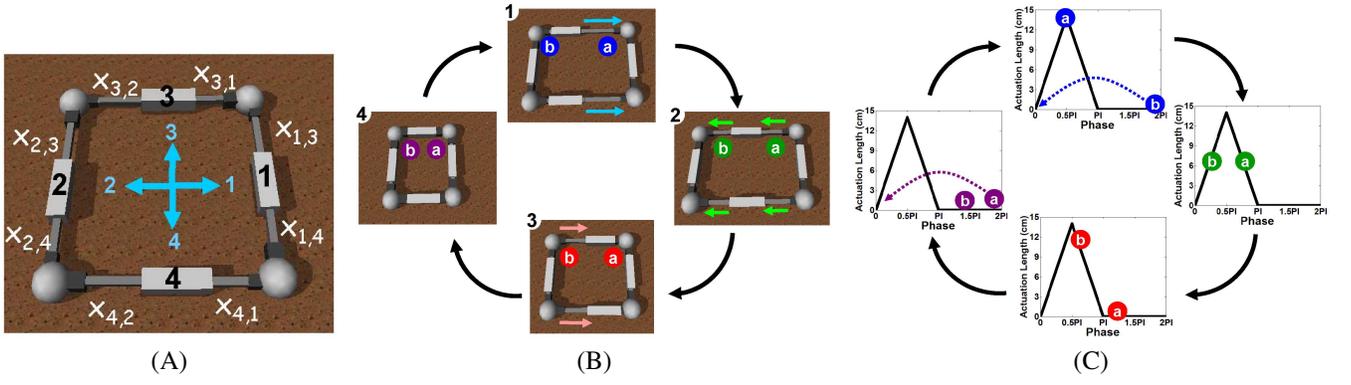


Fig. 3. (A) Indexing scheme. The four links, and corresponding cardinal directions, are indexed numerically. Linear actuators within each link are indexed by the labels of their own link and the adjacent link. (B) Rightward movement steps: Locomotion in any of the four cardinal directions can be achieved by a cycle of actuations like this one for rightward movement. (C) The corresponding phases of linear actuators in each link. X axis indicates phase and Y axis indicates the length of actuator extension.

Let  $d$  indicate the index of the desired traveling direction and  $d'$  indicate the opposite direction.  $\Gamma$  is a set of two links that enables the module to move while being actuated. For instance, if the desired traveling direction is 1 ( $d = 1$ ), then  $d'$  is 2 and  $\Gamma$  contains two links that are parallel to the traveling direction ( $\Gamma = \{3, 4\}$ ). The links that are not in  $\Gamma$  remain static. The actuation state function for links in  $\Gamma$  can be written as follows:

$$\forall i \in \Gamma, x_{i,d}(t) = f(\theta \bmod 2\pi) \quad (1)$$

$$x_{i,d'}(t) = f((\theta - \pi/2) \bmod 2\pi) \quad (2)$$

where  $\theta = \omega \cdot t$  is the phase of the actuator;  $\omega$  is a parameter determined by actuation speed; and actuation function  $f$  (Fig 3 (C)) is formally defined as follows:

$$f(z) = \begin{cases} \lambda \cdot z & \text{if } z \leq \frac{\pi}{2} \\ \lambda \cdot (\pi - z) & \text{if } \frac{\pi}{2} < z \leq \pi \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where  $\lambda = 2L/\pi$ , with  $L$  the maximal actuation length. The two actuators in a link go through the same cycle with phase difference  $\pi/2$ . Fig. 3 (C) shows the sequence of phases corresponding to the locomotion cycle described above. In §V, we show real robot experiments and simulations to validate this mode of locomotion.

Module's direction of travel may be chosen based on sensor feedback; for instance, a module equipped with light sensors may exhibit phototaxis by choosing the direction with highest light level.

### C. Simulation Environment

In addition to the hardware prototype, we also construct a physics-based simulation (shown in Fig. 3) with Open Dynamics Engine (ODE). This simulator allows us to examine the behavior of our control approach for large numbers of modules. The dimensions and mass of each simulated component are set based on the real robot. Several environmental parameters, e.g., parameters for the ground contact model, are optimized based on real robot behavior to provide high fidelity simulation.

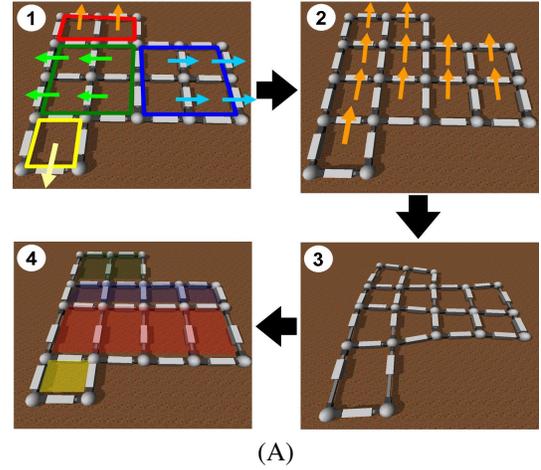


Fig. 4. 1: An asymmetric assembly of modules, with different initial preferred traveling directions. 2: Modules communicate with local neighbors and reach a global agreement on a traveling direction. 3: Modules start traveling toward the agreed direction with uncoordinated movement cycles. 4: Modules achieve coordinated movement and locomote efficiently. Color blocks represent locomotion phase of the modules. Each module maintains the same phase as its horizontally-connected neighbors; phase in adjacent rows differs by  $\pi$ .

## IV. ASSEMBLIES OF MODULES

In this section, we describe how modules coordinate their movements when they are aggregated together. A key aspect of the approach is that *it does not assume any specific connectivity*. Initially, each module has some preferred traveling direction (Fig. 4 (1)). Next, modules communicate with local neighbors to reach agreement on a common direction (Fig. 4 (2)). Finally, modules start coordinating their movement cycles until they achieve an efficient traveling wave (Fig. 4 (3, 4)). In this work, we do not address how modules come together and attach to form aggregate robots, which is itself an important topic of research [15], [16]; instead we focus on coordinating the resulting collective: how does an arbitrary assembly of modules act as a single unified unit?

In the following subsections we present our approaches

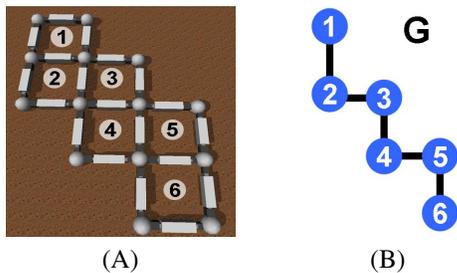


Fig. 5. Graph representation (B) of a robot assembly configuration (A).

associated with the above coordination process: assumptions about connections (§IV-A), agreement regarding common direction of travel (§IV-B), and coordination of movement cycles (§IV-C).

### A. Module linkages

An aggregate robot is composed of modules attached edge-to-edge, so as to effectively occupy sites on a square grid to form a contiguous structure of arbitrary geometry. In our simulations, we implement a shared edge as a single link with doubled mass, rather than doubling the links along each edge by literally putting two modules side by side. Our framework does not rely on modules to assemble into specific configurations, and allows arbitrary robot assembly configurations to achieve coordinated movements.

Each module in the aggregate robot maintains an independent controller,<sup>1</sup> and is able to communicate real values bi-directionally with its physically connected neighbors. We assume that the two controllers of a pair of modules that share an edge can agree on a common value for extension of the corresponding linear actuators. We use an undirected graph  $G$  to represent such communication topology by representing a module with a node and a communication link as an edge (Fig. 5).

### B. Choosing a direction

In order for the robot to locomote effectively, all modules must agree on a single direction. Modules may have their own initial preferences, e.g., based on sensor feedback as above; and the goal for the aggregate robot may be to travel in the direction of the most intense stimulus sensed by any unit, so that the robot exhibits a behavior like collective phototaxis or chemotaxis based on distributed sensing.

In such a case, each module will have an initial preferred direction with associated confidence level. A simple algorithm can be used to reach global agreement: each module repeatedly updates its direction and confidence to match that of the highest-confidence unit in the set consisting of itself and its neighbors. If a module has the same confidence as a neighbor but a different preferred direction, it can break the tie by adding a small random value to its own confidence (with mean 0 so that either direction is equally likely to win). Such an algorithm will reach global agreement

<sup>1</sup>The hardware prototype actually simulates a decentralized set of independent controllers using a single centralized controller.

in a number of iterations equal to the maximum distance between the highest-confidence non-outlier module and any other module—i.e., worst-case the graph diameter, best-case half the graph diameter. If updates are asynchronous, fewer iterations may be required, as a value may be able to propagate through multiple modules in a single iteration if the updates happen to take place in the right order.

Because sensed quantities of interest are generally likely to change slowly over space compared to the distance between modules, neighboring modules will typically start with confidence values not too dissimilar. Thus a value much higher than any neighboring values is likely to reflect a sensor malfunction rather than a highly localized phenomenon of interest, and should accordingly be discarded. This can be implemented as follows: if a module finds that its confidence value is much higher than any of its neighbors, it ignores its own state and adopts that of its highest-confidence neighbor; if a module finds that exactly one of its neighbors has a very high confidence that may indicate an error, it waits one update cycle before adopting its neighbor’s state, to see whether that state persists or is discarded.

An implementation of the above algorithm, omitted here for space reasons, is given online [17]. While its correctness is tedious to prove formally, it is supported empirically by its success in thousands of trials (§V).

### C. Coordinating movements

Here we present the algorithm that allows modules to achieve coordinated movement after agreeing on a common traveling direction. We first consider the following two-module locomotion case: Consider a “passive” edge shared by two modules, that is, one perpendicular to the direction of locomotion, that does not move during the locomotion cycle. It is clear that for best coordination and minimal interference between the two modules, the cycles should be coordinated so that one module pulls this edge forward while the other pushes it in the same direction (Fig. 6 (A)).

Thus, the two modules’ locomotion phases need to be offset by  $\pi$  (front module being ahead of back module) if the modules are connected along their traveling direction. We modify the link actuation function (Eqs. 1 and 2) by replacing  $\theta$  with  $\theta + \phi_i$ , where  $\phi_i$  is module  $i$ ’s phase variable:

$$\forall i \in \Gamma, x_{i,d}(t) = f((\theta + \phi_i) \bmod 2\pi) \quad (4)$$

$$x_{i,d'}(t) = f((\theta + \phi_i - \pi/2) \bmod 2\pi) \quad (5)$$

We define  $\phi_{ij}^*$  as the desired phase offset between module  $i$  and its neighbor  $j$ . If two modules  $i$  and  $j$  are connected along their traveling direction (Fig. 6 (A)), then  $\phi_{ij}^* = \phi_j - \phi_i = \pi$ . Modules that are connected perpendicular to their traveling direction should maintain synchronized locomotion phases ( $\phi_{ij}^* = 0$ ). Fig. 6 (B) shows an example of the desired relationships between a single module and its neighbors.

The challenge is for modules to coordinate to set  $\phi_i$  such that all  $\phi_j - \phi_i = \phi_{ij}^*$  are satisfied, when *each module has only a local view and its choice of  $\phi_i$  can potentially affect all other modules*. We denote the set of module  $i$ ’s neighbors as  $N_i$ . Alg. 1 shows a fully decentralized approach for each

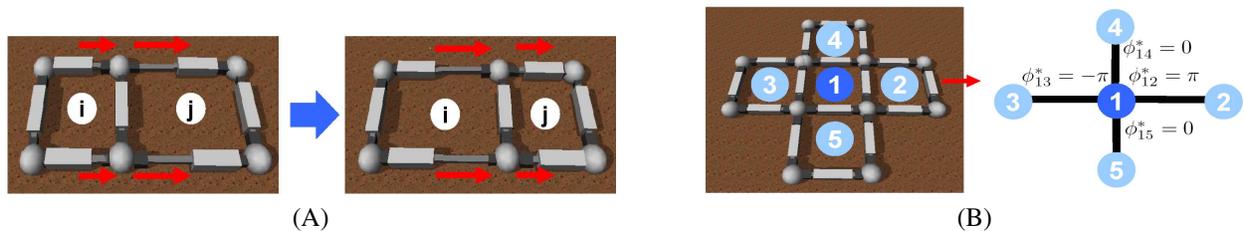


Fig. 6. (A) The cycles of two linked modules need to be offset so that both move the “passive” edge they share at the same time. (B) The desired phase differences between module 1 and its neighbors. Red arrow indicates the robot’s traveling direction. Module 1 needs to achieve  $\phi_{12}^* = \pi$  with its front neighboring module,  $\phi_{13}^* = -\pi$  with its back neighboring module, and  $\phi_{14} = \phi_{15} = 0$  with other modules.

---

**Algorithm 1** Pseudocode for modules to achieve coordinated movement. Initially, each module  $i$  has a random  $\phi_i$  and starts by identifying the desired phase offsets  $\phi_{ij}^*$  between it and its neighbors. Each module then iteratively modifies its phase  $\phi_i$  until all  $\phi_{ij}^*$  are satisfied. In line 11,  $\alpha$  is a damping factor and needs to be set as:  $0 < \alpha < \frac{1}{\max_i |N_i|}$ .

---

```

// Each module  $i$  initializes desired phase offsets with neighbors
for all  $j \in N_i$  do
  if module  $j$  is in front of  $i$  in the traveling direction
  then
     $\phi_{ij}^* = \pi$ 
5:  else if module  $j$  is at back of  $i$  then
     $\phi_{ij}^* = -\pi$ 
  else
     $\phi_{ij}^* = 0$ 
// Each module  $i$  iteratively communicates with its neighbors and updates its phase  $\phi_i$ 
10: loop
     $\phi_i \leftarrow \phi_i + \alpha \cdot \sum_{j \in N_i} (\phi_j - \phi_i - \phi_{ij}^*)$ 

```

---

module to iteratively modify  $\phi_i$  based on feedback from neighbors such that all  $\phi_{ij}^*$  are eventually satisfied.

At every time step, each module computes the differences between current and desired phase offsets to its neighbors and acts proportionally to the summation of all such differences. In fact, the update function of  $\phi_i$  (Line 11) shares similarity with distributed consensus (DC) algorithms in multi-agent systems. Our case is that of the subclass called biased consensus, in which  $\phi_{ij}^* \neq 0$  for some  $i$  and  $j$ . By establishing this connection, we can construct a convergence guarantee for Alg. 1.

*Theorem 1:* Let  $\phi_i$  be the phase of module  $i$  and  $\phi_{ij}^*$  be the desired phase offset between  $i$  and its neighbor  $j$ . If module  $i$  updates  $\phi_i$  based on Alg. 1 and the communication graph  $G$  is connected,  $\phi_i$  will eventually converge to a value such that  $\phi_j - \phi_i = \phi_{ij}^*$  for all modules  $i$  and  $j \in N_i$ .

*Proof:* see Appendix. ■

Our movement coordination algorithm applies not only to the “traveling wave” locomotion strategy that we describe in this section, but can potentially be generalized to other coordinated locomotion patterns that can be formulated as inter-module relationships. This class of algorithms has also

been demonstrated previously for static modular robot applications, such as environmentally-adaptive structure formation [13]. Here we have shown that the approach can be extended to dynamic tasks, such as coordination of periodic actuation.

## V. EXPERIMENTAL RESULTS

In this section, we present various experimental results from applying this framework in both real and simulated robots:

- We use hardware robots to test the efficiency of our locomotion strategy with one and two modules. In the two-module case, we also compare locomotion efficiency with and without coordinated phase offsets.
- We use physics-based simulators to test our coordinated locomotion algorithm in several complex and irregular assemblies, evaluating: (a) how coordination scheme affects locomotion efficiency; (b) how shape of the robot affects locomotion speed; (c) how locomotion efficiency changes with number of modules.
- We test the convergence speed of both direction consensus and module movement coordination algorithms as the number of modules increases.

Our results show that our approach allows modules of various different connectivities to achieve efficient locomotion in both simulated and real robots. In addition, the module movement coordination algorithm is scalable to the number of modules in the system. When more modules are aggregated together, locomotion efficiency in aggregated modules is as or more efficient than individual modules.

### A. Real Robot Experiments

Here we use the hardware prototype we previously described in Section III. The length of each link is 16 cm when it is fully contracted. We set linear actuators’ speed to their maximal speed 2.3 cm/s. We use traveling speed as a metric to evaluate modules’ locomotion efficiency. To define a measure that is invariant to the scale of the modules, we define our speed as percentage of a module’s body length that the robot can travel in each locomotion cycle. To evaluate the effectiveness of our algorithm, we first assemble two modules together to compare their locomotion capability with and without phase coordination. The modules start with random desired traveling directions, and they reach agreement on a common direction in two communication cycles. After

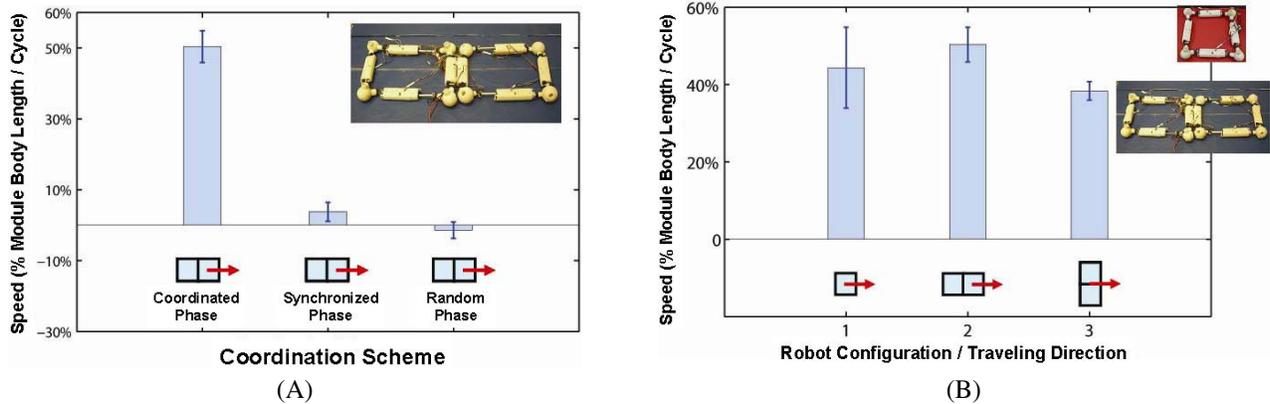


Fig. 7. Hardware experiments: (A) Comparison of robot locomotion speed with different module phase coordination schemes. Locomotion speed is significantly greater when modules achieve phase coordination with Alg. 1 (left). When two modules have synchronized phases or maintain random phases, the modules can barely move. (B) The robot's locomotion speed becomes more consistent if a second module is attached to the first (reduced variance, left vs middle and right bars). We can also see from this result that an additional module attached along the traveling direction can also help to increase the robot's locomotion speed (middle vs right bar). Bottom: A sequence of steps in a locomotion cycle with coordinated phase offsets for a two-module robot.

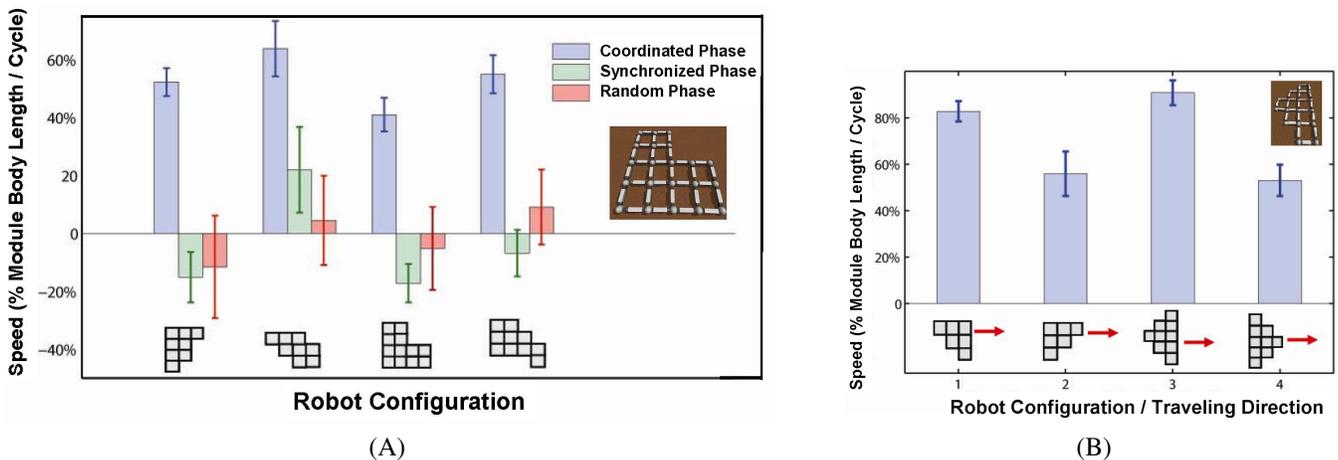


Fig. 8. Simulation experiments: (A) Comparison between different ways of setting phase offsets between modules. If phase offsets are computed based on Alg. 1 (coordinated phase), it leads to much more efficient locomotion for the robot. On the other hand, the robot can move very limited distances with synchronized and random phase offsets. (B) Different shapes and traveling directions vs speed: If the robot's center of mass is at the front, locomotion is faster (first and third bars).

modules decide on their phases, they start locomoting in the agreed traveling direction.

Fig. 7 (A) shows a comparison of different phase offsets between modules. The modules' locomotion speed is averaged over 12 locomotion cycles, and the error bars show standard deviation of the speed among cycles. When modules use Alg. 1 to compute their phase offsets, they can locomote effectively (left bar). When modules have the same phases or maintain random phase offsets (the data is averaged from 6 random phases), the modules show very limited locomotion abilities, even though each executes an actuation cycle that would let it move effectively if on its own.

We also evaluate speed of locomotion with one and two modules and in different traveling directions. Fig. 7 (B)

shows that a single module's locomotion speed has higher variability (error bars) compared to two-module configurations. The intuition behind this is that an additional module can help the robot stabilize its movement. We can also see that if two modules are attached along their traveling direction, the robot's locomotion speed is also higher than the speed of travel in the orthogonal direction.

### B. Arbitrary Connectivities

With the ODE simulator, we investigate modules' locomotion efficiency in a wide range of robot configurations. Again, we compare modules' locomotion efficiency with and without our movement coordination (Fig. 8 (A)). For coordinated and synchronized phase experiments, results show

mean and standard deviation from 40 locomotion cycles. For random phase experiments, the graph shows results from 5 different random initializations and 20 locomotion cycles in each case (100 cycles in total). The most efficient locomotion results when all modules compute their phases based on Alg. 1. This is consistently true for different irregular module connectivities.

### C. Efficiency vs Shape

From Fig. 8 (A), we can also see that the robot’s moving speed varies with different shapes. This inspired us to further explore how shape might potentially affect the robot’s locomotion. Fig 8 (B) shows how varying robot shape and traveling direction affects locomotion efficiency. In these experiments, when front rows of the robot have more modules than back rows, the robot travels faster (bars 1 and 3). This suggests that a robot’s locomotion speed may improve when its center of mass is closer to the front end.

### D. Efficiency vs Module Aggregation

The independent modules that make up an aggregate robot have the potential to act in conflict, interfering with their collective locomotion, as Fig. 7 (A) makes clear. We thus next investigate how the attainable traveling speed is affected as the number of modules increases. Ideally, one would like for self-mobile modules not to lose speed when they join a larger robot.

Fig. 9 shows the speed of robots comprising one to four modules with coordinated phases, for both real and simulated robots. Simulations tested varying levels of ground friction to explore different terrain conditions.

In all cases, the speed of robots with two or more modules was at least as great as that of single modules. Simulations showed that under some conditions, adding a second module significantly increased the speed of the robot; even in the high-variance cases where the increase was not significant, the data suggest an overall increase rather than decrease. The mechanism appears to involve the fact that some components can slide backwards during parts of the movement cycle; the second module provides increased stability and reduces sliding away from the travel direction. Adding modules beyond the second had no significant further effect on speed.

While Alg. 1 leads to effective locomotion, there may well exist locomotion patterns that produce still greater speed, potentially attainable only with larger assemblies or with heterogeneity among module controllers. In future work we intend to investigate the possibility of such patterns.

### E. Scalability Experiments

Here we explore how the direction agreement and movement coordination algorithms of §IV-B and §IV-C scale with the size of the system.

(a) Consensus direction: Fig. 10 shows the number of iterations required for randomly generated, randomly initialized robots to come to global agreement on direction using an implementation [17] of the algorithm outlined in §IV-B, with asynchronous updating. Tie-breaking random values were

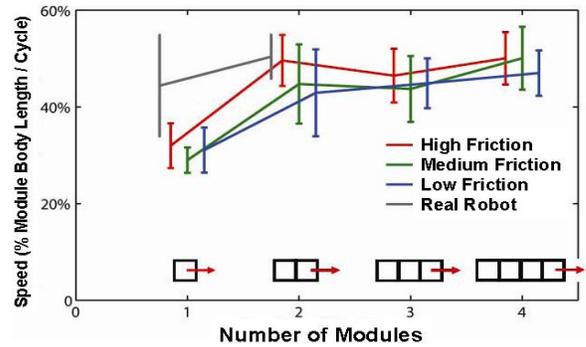


Fig. 9. Number of modules vs. robot speed. Adding a second module to the first can increase net travel speed; adding further modules has no significant effects.

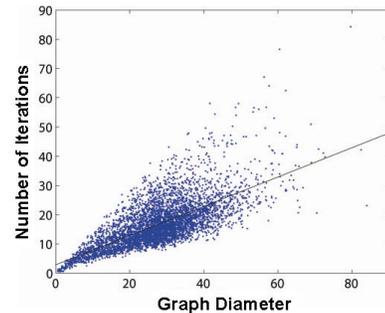


Fig. 10. Number of iterations  $n$  required for randomly generated robot to reach global agreement on direction, as a function of diameter  $d$  of its connectivity graph. Results show 5000 trials using asynchronous updating, with best-fit line minimizing squared error  $n = 0.50d + 2.8$ .

chosen to be small compared to the discretization of confidence values. In 5000 out of 5000 trials, the robot reached global agreement on the direction associated with the highest non-outlier confidence. The typical number of iterations  $n$  was proportional to the graph diameter  $d$ , with best-fit line  $n = 0.50d + 2.8$ . When modules execute the algorithm synchronously, convergence to the appropriate direction is still reliably correct but somewhat slower ( $n = 0.86d + 2.5$ ), since unlike the asynchronous case values can propagate at most one module per time step.

(b) Movement coordination: We define  $\epsilon(t) = \sum_i \sum_{j \in N_i} |\phi_{ij}(t) - \phi_{ij}^*|$  as a measure of convergence to coordinated movements, where  $\phi_{ij}(t)$  is the phase offset between modules  $i$  and  $j$  at time  $t$ . The number of iterations required for convergence is defined as  $t_{\min} = \min_t \frac{\epsilon(t)}{\epsilon(0)} < 5\%$ . Fig. 11 shows averaged  $t_{\min}$  versus number of modules in configurations with square connectivity (100 different initializations) and random connectivity (100 different connectivities and initializations). We can see  $t_{\min}$  is linearly proportional to size of the system. Further, square connectivities on average require fewer iterations than random connectivities. Since square connectivities generally have smaller diameters than random connectivities, these results coincide with theories presented in [13].

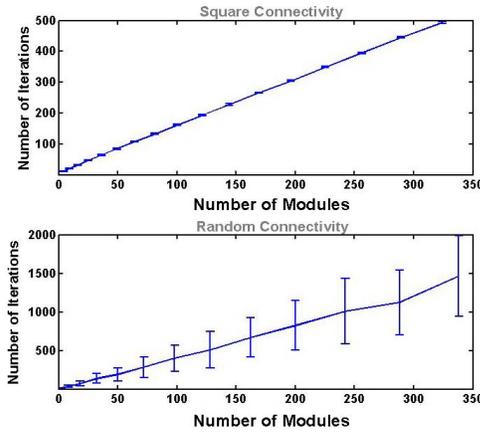


Fig. 11. Convergence speed vs. number of modules. We sequentially increase the number of modules in both square (top) and random (bottom) connectivities. Each point is computed from 100 random initializations. The average number of iterations required for convergence is linearly proportional to the number of modules.

## VI. CONCLUSIONS

In this work we have presented distributed coordination schemes by which the modules of an amorphous modular robot can agree on a travel direction and synchronize their movement to achieve effective group locomotion. No central controller, global knowledge, or a priori known connectivity is required. We have investigated the correctness and effectiveness of these schemes analytically and experimentally, in both simulation and hardware.

The biological inspiration for this work is the cellular slime molds that spend part of their life locomoting as individual cells and part as a multicellular slug [7]. For future applications, we envision developing a decentralized robotic system of self-mobile modules which can both act independently (e.g., for effective coverage of a large area) and join together into an aggregate robot at need (e.g., for more effective movement). Such a system would require the capabilities we have discussed in this paper, and this work represents a first step toward that larger goal.

## VII. ACKNOWLEDGEMENT

This research is supported by an NSF EMT Grant (CCF-0829745) and Wyss Institute for Bio-inspired Engineering.

### APPENDIX: PROOF OF THEOREM 1

Let  $\phi_i(t)$  be module  $i$ 's phase offset variable at time step  $t$ . We first collect all  $\phi_i(t)$  into a column vector:

$$X(t) = [\phi_1(t), \phi_2(t), \dots, \phi_n(t)]'$$

The update equation in Alg. 1 can be rewritten as:

$$\phi_i(t+1) = \phi_i(t) + \alpha \cdot \sum_{j \in N_i} (\phi_j(t) - \phi_i(t) - \phi_{ij}^*)$$

Aggregating all modules' update equations, we can write the collective dynamics of phase offset variable updates in matrix form:

$$X(t+1) = A \cdot X(t) + \tilde{b} \quad (6)$$

where  $A = [a_{ij}]$ , an  $n \times n$  matrix with element  $a_{ij}$  defined by:

$$a_{ij} = \begin{cases} \alpha & \text{if } j \in N_i \text{ and } i \neq j \\ 1 - \alpha \cdot |N_i| & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

and where  $\tilde{b}_i = \alpha \cdot \sum_{j \in N_i} \phi_{ij}^*$  is a bias vector. We note that  $A$  is a stochastic matrix since each row sums to 1. In addition,  $\sum_i \tilde{b}_i = 0$ , since  $\phi_{ij}^* = -\phi_{ji}^*$  for all  $i, j$ . We can write the optimality condition (when all  $\phi_j - \phi_i = \phi_{ij}^*$ ) as:

$$X^* = A \cdot X^* + \tilde{b} \Rightarrow \alpha \cdot L \cdot X^* = \tilde{b}$$

where  $L = 1/\alpha(I - A)$  is a Laplacian matrix with  $\text{rank}(L) = n - 1$  and  $\text{null}(L) = 1$ . Since  $\sum_i \tilde{b}_i = 0$  and  $A$  is a stochastic matrix, the summation of state variables is conserved over time:  $\sum_i \phi_i(t) = C, \forall t$ . This additional constraint falls in the nullspace of  $L$ . We define  $L'$  as the matrix whose first  $n$  rows are  $L$  and whose last row is the new constraint; we can then show  $\text{rank}(L') = n$  and there exists a unique solution  $X^*$  for  $\alpha L' X^* = \tilde{b}$ . Therefore, there is a unique optimal solution  $X^*$  (satisfying  $\phi_j - \phi_i = \phi_{ij}^*, \forall i$  and  $j \in N_i$ ) for any initial condition  $X(0)$ . Subtracting the optimality condition  $X^* = AX^* + \tilde{b}$  yields the following dynamics:

$$X(t+1) - X^* = A \cdot (X(t) - X^*) \Rightarrow Y(t+1) = A \cdot Y(t) \quad (7)$$

Eq. 7 has the same form as average consensus [12]; we can then follow the proof procedure in [12]. Let  $X^* = [\phi_1^*, \phi_2^*, \dots, \phi_n^*]$  with  $\phi_j - \phi_i = \phi_{ij}^*$  for all  $i$  and  $j \in N_i$ . We define Lyapunov function  $V(t) = \sum_i (\phi_i(t) - \phi_i^*)^2 = (X(t) - X^*)^T (X(t) - X^*)$ . Following the same procedure as Theorem 4 in [13], we can show:

$$V(t+1) < (\mu_2(A))^2 V(t)$$

where  $\mu_2(A)$  is the 2nd largest eigenvalue of  $A$ . Using the theory of graph Laplacians, we have  $0 \leq \mu_2(A) < 1$  if the module communication graph  $G$  is *connected*. We can then show  $\lim_{t \rightarrow \infty} V(t) = 0$  and  $X(t)$  converges to  $X^*$  with exponential rate.

## REFERENCES

- [1] W.-M. Shen, M. Krivokon, H. Chiu, J. Everist, M. Rubenstein, and J. Venkatesh, "Multimode locomotion for reconfigurable robots," *Autonomous Robots*, vol. 20, no. 2, pp. 165–177, 2006.
- [2] A. Kamimura, H. Kurokawa, E. Yoshida, S. Murata, K. Tomita, and S. Kokaji, "Distributed adaptive locomotion by a modular robotic system, M-TRAN II," in *Proc. IROS*, 2004.
- [3] B. Salemi and W.-M. Shen, "Distributed behavior collaboration for selfreconfigurable robots," in *Proc. ICRA*, 2004.
- [4] M. Park, S. Chitta, and M. Yim, "Isomorphic gait execution in homogeneous modular robots," in *RSS Workshop on Self-Reconfigurable Modular Robots*, 2006.
- [5] M. Yim, C. Eldershaw, Y. Zhang, and D. G. Duff, "Limbless conforming gaits with modular robots," in *Proc. ISER*, 2004.
- [6] I. Couzin, J. Krause, N. Franks, and S. Levin, "Effective leadership and decision making in animal groups on the move," *Nature*, vol. 433, 2005.
- [7] J. T. Bonner, *The Social Amoebae*. Princeton University Press, 2009.
- [8] A. Lyder, R. Garcia, and K. Støy, "Mechanical design of Odin, an extendable heterogeneous deformable modular robot," in *Proc. IROS*, 2008.
- [9] C. Yu and R. Nagpal, "Self-adapting modular robotics: A generalized distributed consensus framework," in *Proc. ICRA*, 2009.
- [10] S. Curtis et al., "Tetrahedral robotics for space exploration," *IEEE Aerospace and Electronic Systems Magazine*, 2007.
- [11] A. Ijspeert, D. Crespi, Ryczko, and J.-M. Cabelguen, "From swimming to walking with a salamander robot driven by a spinal cord model," *Science*, vol. 315, no. 5817, pp. 1416–1420, 2007.
- [12] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," in *Proc. IEEE*, 2007.
- [13] C.-H. Yu and R. Nagpal, "Sensing-based shape formation tasks on modular multi-robot systems: A theoretical study," in *Proc. AAMAS*, 2008.
- [14] M. Shimizu, A. Ishiguro, and T. Kawakatsu, "Slimebot: A modular robot that exploits emergent phenomena," in *Proc. ICRA*, 2005.
- [15] R. Groß, M. Bonani, F. Mondada, and M. Dorigo, "Autonomous self-assembly in swarm-bots," in *IEEE Transactions on Robotics*, 2006.
- [16] S. Murata, K. Kakomura, and H. Kurokawa, "Docking experiments of a modular robot by visual feedback," in *Proc. IROS*, 2006.
- [17] Online Appendix URL: <http://www.fas.harvard.edu/~chyu/icra10-appendix.pdf>.