

# Miche: Modular Shape Formation by Self-Dissassembly

Kyle Gilpin, Keith Kotay, and Daniela Rus

**Abstract**—We describe the design, implementation, and experimentation with a collection of robots that, starting from an amorphous arrangement, can be assembled into arbitrary shapes and then commanded to *self-dissassemble* in an organized manner. Each of the 28 modules in the system is implemented as a 1.8-inch autonomous cube-shaped robot able to connect to and communicate with its immediate neighbors. Two cooperating microprocessors control each module's magnetic connection mechanisms and infrared communication interfaces. When assembled into a structure, the modules form a system that can be virtually sculpted using a computer interface. We report on the hardware design and experiments from hundreds of trials.

## I. INTRODUCTION

We present a modular robotic system that behaves as programmable matter. The approach to realizing programmable matter uses *self-dissassembly* as the fundamental operation to achieve shape formation. The function of self-dissassembling modular robots can be thought of as analogous to sculpting. We start with a large block made of individual modules. The initial structure is transformed into the desired shape by eliminating the unnecessary modules from the structure in a controlled fashion. Much like a sculptor would remove the extra stone from a block of marble to reveal a statue, our self-dissassembling system eliminates modules to form the goal structure.

### A. System Functionality

We developed a self-dissassembling system made of robotic modules collectively named *Miche* (see Figure 1). Shape formation with *Miche* begins with an initial amorphous shape that is assembled by hand (e.g. see Figure 1(a)). After the modules in the initial structure use local communication to establish their location, they cooperatively distribute a user-defined goal configuration using neighbor-to-neighbor communication. Once all modules know whether or not to remain a part of the system, the unnecessary modules disconnect from the system and drop off to create the desired shape (e.g. see Figure 1(b)). In this paper we describe the *Miche* hardware that we designed and built. Each module is a cube whose faces are the PCBs used for the electronics and control of the system. Each cube has on-board computation, a power supply, point-to-point IR communication with its immediate neighbors, and three switchable permanent magnets. These magnets provide the connection between adjacent units and

have the feature of activating or deactivating depending on their orientation. Three small motors capable of rotating the magnets provide the disconnection actuation in the system. We have built a system consisting of 28 *Miche* modules.

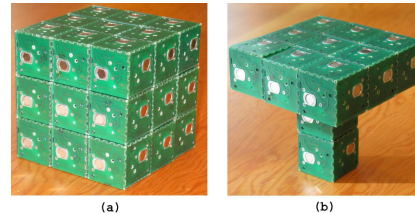


Fig. 1. A self-dissassembling system can transform from an initial uniform assembly of identical modules, (a), into a more interesting and functional assembly in (b).

### B. Advantages of Self-Dissassembly

Creating robotic systems and smart objects by self-dissassembly has one main advantage over existing approaches by self-assembly. Self-dissassembling systems entail a simple actuation mechanism to disconnect which is generally easier, faster, and more robustly achievable than actively seeking and making connections. The trade-off is two-fold. First, self-dissassembling systems must start from a pre-assembled structure of modules. In our work, this block is assembled manually, but this process can be automated using mechanical fixtures. Second, external forces must be employed to remove unwanted material from the system. Often, these forces can be found in the surrounding environment. For our experiments, we used gravity to pull unnecessary modules away from the final structure.

Additionally, modular robots that can self-dissassemble provide a simple and robust approach toward the goal of smart structures and digital clay. A collection of millions of modules, if each were small enough, could form a completely malleable building material that could solidify and then disassemble on command. Most types of objects and shapes could be created this way. The applications of self-dissassembling systems include all the applications of self-assembling systems. The added flexibility of removing specific components from the assembly ensures that our approach is especially well suited to tasks requiring temporary supporting structures. For example, self-dissassembling material could be applied as an active scaffolding to help heal severely broken bones that would otherwise require the use of permanent steel plates or pins. In addition to disassembling as the bone regrows, the scaffolding could provide valuable

This work was supported by Intel and Magswitch.

K. Gilpin, K. Kotay, and D. Rus are with the Computer Science and Artificial Intelligence Lab, MIT, Cambridge, MA, 02139, kwgilpin@mit.edu, kotay@csail.mit.edu, rus@csail.mit.edu.

medical status information to doctors. In such a scenario, the bloodstream could carry away extra modules.

### C. Related Work

Our work draws on prior and ongoing research in modular and distributed robotics [1]–[11] and self-assembling systems [12], [13]. Yoshida et. al. present a self-reconfiguring system which uses shape memory alloy (SMA) springs to connect neighboring modules [14], [15]. A novel system for self-assembly and reconfiguration is presented by White [16] which uses fluid flow to bind individual modules together. In [1], Pillai et. al. simulate using thousands of mechanically passive modules to construct digital representations of three-dimensional objects. The CHOBIE robot developed by Koseki [17] is unique in its mechanical design. The modules in the CHOBIE system, which are also rectangular, are able to locomote by sliding in two planes relative to one another. The work presented in this paper has a similarity with this prior work in the goal of accomplishing shape creation using modular robots. However, our strategy to achieve this goal is focused on actuation by self-disassembly—a significant departure from the methods employed in previous work.

## II. MICHE HARDWARE



Fig. 2. Each module in the system is a cube which measures 1.77 inches on each side and weighs 4.5oz. Each module is completely autonomous and can operate for several hours under its own power.

Figure 2 shows a Miche module prototype. Each module contains the resources necessary for autonomous operation: processing capabilities, actuation mechanisms, communication interfaces, and power supplies. The modules are built from six distinct printed circuit boards that interlock to form a rigid structure. When completely assembled, each cubic module is 1.8 inches on a side and weighs 4.5oz. As shown by an open module in Figure 3, all electronic components are surface mounted on the top side of the boards so that when assembled into cubes, all components reside on the

inside. The only pieces of the system mounted externally are three steel plates that form half of the magnetic connection mechanism, presented in detail below.

### A. Connection Mechanism

Individual modules bind to each other using switchable permanent magnet assemblies, hereafter referred to as Magswitches. These assemblies are produced by Magswitch Technology, Inc. [18]. Figure 4 shows an example Magswitch. Three of the faces of each cubic module contain Magswitches. Like all other components, they are mounted on the inside of the cubes and pass through similarly sized holes in the printed circuit boards. The other three cube faces of each cube are covered by steel plates. When multiple cubes are assembled into a structure, the Magswitches always attach to the steel plates of a neighboring cube, not one of the other cube's Magswitches. As a result, the modules can only attract one another. They do not repel but, instead, depend upon gravity or user intervention to clear unused modules from any final structure. A single Magswitch connected to a neighbor's steel plate can support over 4.5 lbs.—the combined weight of 17 other modules hanging vertically.

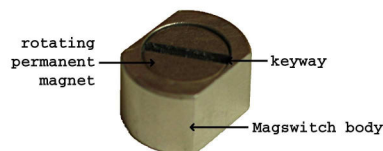


Fig. 4. Each Magswitch consists of two permanent magnets stacked on top of each other inside of a metal housing. The bottom magnet is fixed while the top one contains a keyway and is free to rotate. As the top magnet is rotated through  $180^\circ$ , the entire device switches from on to off or vice versa.

The Magswitch assemblies in Figure 4 act as switchable permanent magnets. As the traditional permanent magnet with the keyway is rotated, the Magswitch as a whole is energized, (attracts other ferromagnetic materials), and subsequently de-energized, (releases its hold). The advantage of the Magswitches is that they only consume power while changing states. Once a Magswitch is on or off, it remains in that state indefinitely. This is invaluable for the battery life of the modules.

A miniature pager-sized motor with an integrated planetary gear box drives each Magswitch. The entire Magswitch assembly is illustrated in Figure 5. When driven with 4.1V, the voltage of a freshly charged lithium-polymer battery, the motor requires approximately 1.3 seconds to switch a deactivated Magswitch on and back off again. An analog Hall Effect sensor is used to detect the state of each Magswitch. The Hall Effect sensor is placed such that it is sensitive to the magnetic field produced by the Magswitch. As the Magswitch rotates, the Hall Effect sensor produces a voltage that approximates a sine wave, so the motor can be stopped when the Magswitch is fully switched.

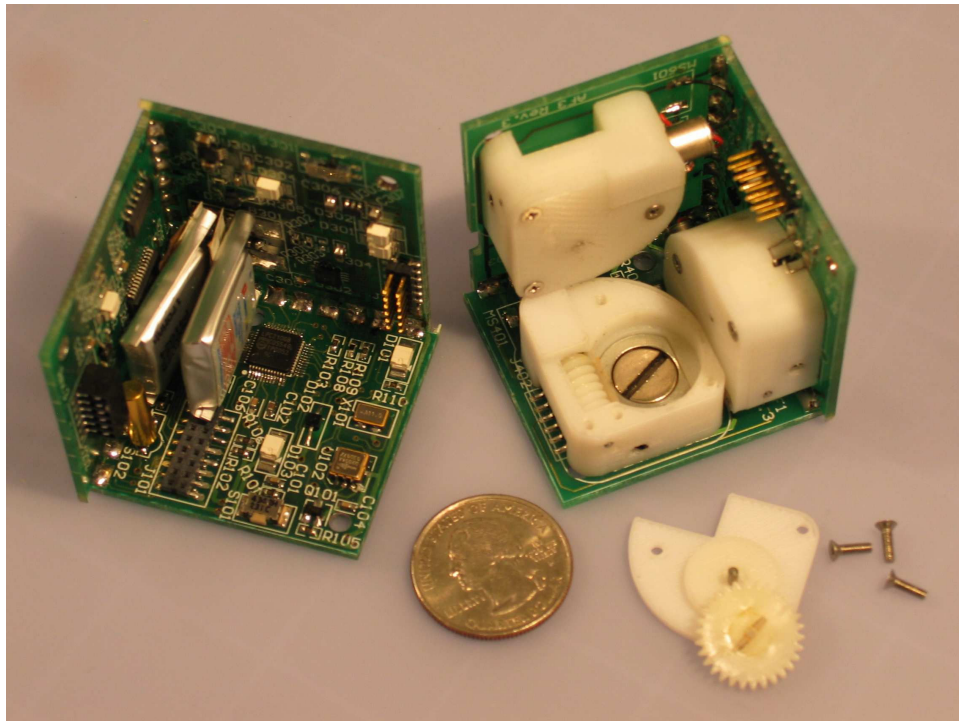


Fig. 3. An open module shows all of its major components. Each contains two microprocessors, connection mechanisms, infrared emitters and detectors, an accelerometer, a tilt switch, and batteries. Each cube is totally self-sufficient.

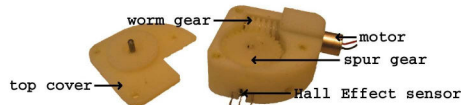


Fig. 5. A worm gear attached to the output shaft of a miniature DC motor turns a spur gear that mates with the keyway in the Magswitch. In the figure, the Magswitch is obscured by the spur gear, and the removed top cover of the entire assembly is shown on the left.

### B. Processors

Each module contains two microprocessors that perform different tasks. The primary microprocessor is a Philip's 32-bit ARM7 processor. It is responsible for all of the high-level disassembly algorithms. The second processor is an 8-bit programmable system on a chip (PSoC) that is manufactured by Cypress Microsystems. The PSoC handles the low-level functions that would otherwise occupy the ARM. In particular, it implements six serial receive ports, one for each face of the module. This allows a single module to receive messages from all its neighbors simultaneously. The ARM and PSoC communicate using the I<sup>2</sup>C protocol [19].

### C. Communication Interface

Communication between modules is performed using infrared (IR) light. Optical communication was chosen for its forgiving nature which permits some misalignment between the transmitter and receiver. Each of the six cube faces contains an infrared LED and an infrared sensitive photodiode.

Together, these allow bidirectional communication between neighboring cubes at 9600 bits per second (bps). While higher bit rates were achievable, 9600bps proved adequate.

Because the infrared LED and photodiode cannot occupy the same physical space and must be placed off-center on each face, every module has only one valid orientation in a composite structure. Otherwise, the LEDs and photodiodes of neighboring cubes would not align. However, because any self-disassembling structure must be assembled manually, this restriction does not affect the functionality of the system.

### D. Sensing

Each module is able to detect its absolute three-dimensional orientation by using a two-axis accelerometer and a binary tilt switch that are connected to the ARM microprocessor. A tilt switch is needed in addition to the accelerometer in order to detect a module's orientation when neither of the two accelerometer axes is experiencing acceleration due to gravity. This information, combined with the more precise data from the accelerometer, is enough to determine which side of a module is facing down.

### E. Power

Each module is equipped with two rechargeable lithium-polymer batteries connected in parallel. These batteries supply power to the module's electronics and motors. Their combined 340mAh capacity allows a module to continuously transmit messages on all six faces for over 6 hours if the motors are not activated. The electrical connection to recharge the batteries is provided through two of the metal

faces that adorn the outside of the cubes. Current to recharge each module's batteries flows from the metallic sides of a specially designed trough, through the metal faces and conductive epoxy to solder mask-free contacts on the back of the printed circuit boards. Starting or stopping the charging process is achieved by simply placing the modules in, or removing the modules from, the charging trough. Up to 15 modules can be recharged simultaneously in one trough.

### III. COMMUNICATION AND CONTROL

To support the algorithms that allow Mice to disassemble, we have implemented a series of low-level functions that control the hardware in each module. These routines place an abstraction barrier between the localization, shape distribution, and disassembly algorithms and the complex hardware contained in each module. This separation facilitates the rapid implementation and modification of the high-level concepts which are responsible for the system's visible behavior. The high-level algorithms do not have to contend with the specifics of basic tasks such as exchanging messages or activating a Magswitch.

Once a module has the ability to transmit and receive messages, the low-level operation reduces to the simple process illustrated in Figure 6. After initializing, a module loops forever, simply receiving and transmitting messages to its neighbors. The interesting behavior responsible for the system's self-disassembly is governed by how the high-level algorithms for shape aggregation respond to received messages. The most important messages include "acknowledge," "ping," "localization," "include module," "reflection," "disconnect single magswitch," and "disconnect all magswitches."

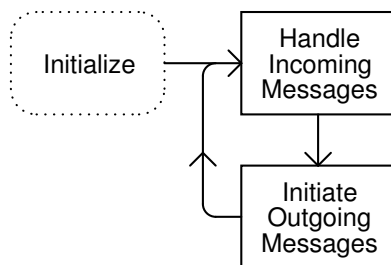


Fig. 6. The message processing loop executing on each module is simple. First, modules initialize all their peripherals. Then, they loop infinitely, receiving and sending inter-module messages. How a module changes its internal state in response to received messages and what messages it transmits in return, dictate the system's high-level abilities.

These basic messages are used to drive the high-level control algorithm for self-disassembly which is divided into five phases and shown in Figure 7. The first phase, neighbor discovery, commences after the modules are reset. Modules are added manually to the initial assembly one at a time. During this phase, modules use ping messages to detect any neighbors in close proximity. When a neighboring module is detected, the magswitch on that face is commanded to rotate to the *on* position, and the two modules test whether they are

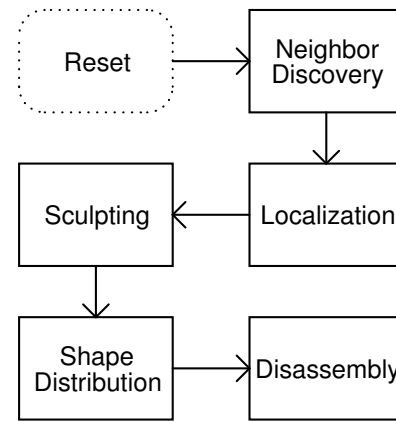


Fig. 7. The entire self-disassembly process consists of five phases: neighbor discovery, localization, sculpting, shape distribution, and disassembly.

able to communicate. At the end of the neighbor discovery phase, all the modules in the structure are connected as a solid structure.

During the localization phase, which follows neighbor discovery, modules discover their positions within the structure and use reflection messages to transmit their positions back to a MATLAB program (see Figure 8) running on the user's desktop computer. Using the regional information contained in the localization messages, all modules in the structure are able to determine their relative coordinates without any concept of the structure as a whole.

Once each module has transmitted its position to the user's computer, the MATLAB program displays a 3D model of the system using a GUI. Using this model, the user virtually sculpts the desired final configuration of the modules; the user selects which modules should be included in the final shape. After this sculpting process is complete, the program generates a sequence of inclusion messages.

Next, during the shape distribution phase, the MATLAB program transmits the inclusion messages to a single root module in the structure. The modules then propagate these inclusion messages to their proper destinations. As with the localization process, the inclusion messages do not contain a complete representation of the final structure. To conserve space, they only contain local information.

The final phase is disassembly. During the disassembly phase, the modules not designated to be in the final structure disconnect from their neighbors to reveal the shape the user sculpted previously. This happens instantaneously and in parallel around the structure.

Each of the phases of self-disassembly is dependent on a distributed, localized message passing algorithm executing on each module. Since the algorithms are communication intensive, the main optimization point is to minimize the size and number of required messages for each phase of the disassembly process. Due to space limitation in this submission, the details and analysis of these algorithms could not be included. However, these technical details are available in [20] and a companion theoretical paper [21].

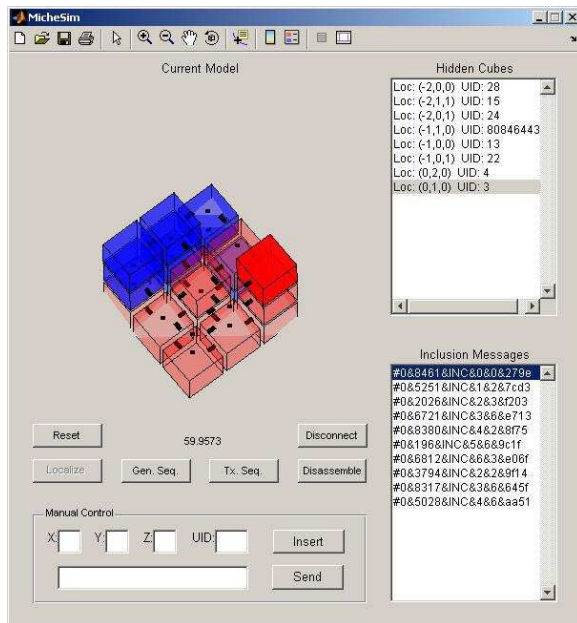


Fig. 8. A graphical user interface (GUI) was used to virtually sculpt the initial configuration of modules into a more interesting configuration. Here, modules that will be included in the final structure are shown in red, and those that will not be included are shown in blue. The list box in the lower right displays the sequence of messages that will be transmitted to the root module (the darkest red cube) for distribution in the structure.

#### IV. MICHE EXPERIMENTS

Figure 9 shows an example shape sculpted by the Miche system. This shape was generated out of a single-layer block of 15 modules. A 3-by-5 square of modules was assembled in the plane. The root module was located in the center of both axes. The final shape was the humanoid robot shape shown in Figure 9. The experiment was repeated 26 times. In all but two cases, every module that was not supposed to be a part of the final humanoid structure disconnected. The two errors were traced to a single Magswitch that was jammed and unable to deactivate. After it was fixed, there were no additional problems. For 10 of the 26 tests, we suspend the initial 3-by-5 sheet of modules horizontally to see if the unnecessary modules would fall away after they disconnected. In seven of these experiments, all extra modules fell away from the structure. In the other three cases, a single scrap module became wedged between two of its three neighbors while it was falling away from the structure. Based on 15 of the humanoid experiments, the average time required to create the structure was 90 seconds. This total consists all phases of self-disassembly (see Figure 7) except the sculpting phase, whose length is determined by the user.

Any solid 3D shape that can be constructed from 28 modules is achievable in this system. Specifically, we have generated a 15 module dog out of 27 modules, a flower, and many abstract geometries. Each of these experiments has been tried multiple times. In particular, we generated the dog, as shown in Figure 10, five times. In each case, the self-disassembly algorithm operated correctly and all the



Fig. 9. The final robot-like shape we “self-disassembled” using an initial 3-by-5 rectangle of modules. From start to finish, the self-disassembly process required a total of approximately 90 seconds excluding any time spent modeling the desired final shape.

appropriate modules knew whether to disconnect or remain a part of the final configuration. Because the dog model is a three-dimensional structure, it is impossible for all of the extra modules to fall away, even if the dog is suspended in the air. In each of the five experiments, there were two scrap modules that disconnected, but could not fall away because they rested on top of modules that were intended to be a part of the final structure. On average, there were an additional 2.4 modules that disconnected from their neighbors, but became wedged in the structure when they were supposed to fall away. By rotating the structure or lightly tapping these modules, we were able to achieve the desired final configuration.

The critical component in the performance of the Miche system is point-to-point message transmission. All of the steps in the self-disassembly algorithm rely upon robust message transmission. Ignoring the fact that modules can become wedged while trying to fall away from the structure, all the observed errors were due to message transmission or reception failures. The rest of the section details our experimental characterization of messaging.

We tested the neighbor discovery process with modules that had anywhere from one to six neighbors. When the IR LED and photodiode on neighboring modules were properly aligned, the neighbor discovery process worked correctly: each module detected the other’s presence and they latched together. Sometimes, if a module was poorly aligned with a neighbor, it failed to detect it. Shifting either of the modules slightly tends to fix this problem. Even if two modules are poorly aligned, and cannot communicate directly, it is unlikely to compromise the overall reliability of a large system. In general, every module has several neighbors, so communication and structural links are redundant. (Note that articulation points in the initial configuration are critical to the system’s reliability. If communication through such a

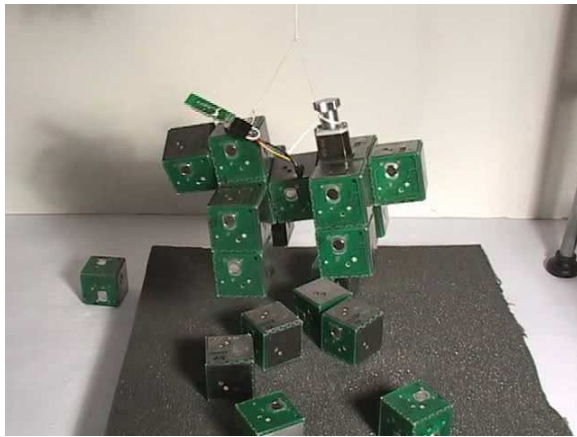


Fig. 10. A 15 module Dog was constructed out of a 27 module block. Note that during self-disassembly the original block was suspended using a magnet. The excluded blocks fell off the structure.

point is impossible, communication with every node past the articulation point will also be impossible.) In our initial experiments, every module could communicate with at least one of its neighbors. It was never the case that a module communicated with none of its neighbors. In all of the 191 tests that we recorded, every single module successfully connected to the structure through at least one point and prepared for localization. That means the neighbor discovery process operated correctly over 1,200 times.

We could observe and time the algorithm's progress in two ways. First, we were able to monitor the time required for all modules to localize by monitoring a LED on each module. Second, we were able to measure the time required for the GUI to realize that all modules had localized: the time required for all reflection messages to propagate back to the GUI. We measured these times for linear, square, and cubic structures of different size. We were not able to measure the amount of time required for all modules in a cubic structure to localize because not all of the LEDs were visible. We were still able to record the time required for all modules of a cubic structure appeared in the GUI.

In one particular set of experiments, we measured the amount of time required for a line of modules to localize. We chose a line because it is the worst-case configuration for the localization algorithm: one link failure affects all modules downstream of the root module. We recorded the localization times for chains of modules that were one, four, seven, and nine units long when the root module was at the end of the chain. We performed 16, 15, and 15 experiments for the 4-, 7-, and 9-module cases, respectively. Figure 11 illustrates the mean and standard deviation of the time required for all user LEDs to begin blinking.

## V. SCALABILITY

While the 28 modules that we constructed were sufficient to test our hardware and algorithms, it is important that the system scales well. Miche's hardware and algorithms should continue to perform well as the number of modules

in the system is increased dramatically. From the hardware perspective, we chose Magswitches as the connection mechanism for their simplicity and excellent strength to weight ratio. As noted above, a single module is able to support 17 others. So long as completed structures avoid long, skinny appendages, their structural integrity should increase with their size. Additionally, the Magswitch-to-steel plate connection does not require fine alignment. Consequently, the affect of any accumulated misalignment in a large structure does not affect the structure's integrity. With refinement of the manufacturing process, the existing variation of the module size could be reduced thereby improving the reliability of the communication links. Powering an expanded system is also simple because the modules' rechargeable nature and the fact it is easy to recharge many modules simultaneously.

The performance of Miche's algorithms should also scale favorably with the size of the system. We designed the algorithms to minimize the size of every inter-module message. This also means that messages never carry global information. They remain a fixed size even as the system grows. Therefore, as the system scales, the computational and storage facilities required by each module do not. The algorithms were also designed to minimize the number of irrelevant messages that each module receives. Messages are never simply broadcast to all modules. Every message has a specific destination and travels along the shortest possible path to reach that destination. This ensures that each message passes through the minimum number of other modules. For a full analysis of the scalability of the Miche algorithms, please see [20] or [21].

## VI. DISCUSSION

This paper details the design, implementation, and experimentation with the 28 module Miche robot we developed in our lab. This system is capable of realizing goal-directed shapes using a process of self-disassembly. Each module

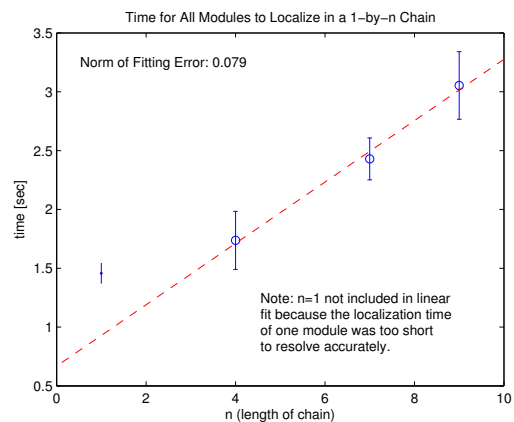


Fig. 11. The time required for a chain of modules to localize is linear in the length of the chain. The circles represent the average time required for all modules to localize. For each different experiment, the whiskers span two standard deviations. When fitting the line to the data, the one-module case was ignored because it was too difficult to resolve the time required to localize one module.

uses distributed localized message-driven algorithms. Our extensive experimentation with this system has shown that the concept and algorithms are very robust. The system is based on a simple concept and minimizes the dependency on actuation. Making objects by virtually sculpting uses disconnection as the basic actuation mechanism. Although this operation requires an external force to remove extra modules, it is simpler and therefore more reliable than making connections because it reduces to “letting go” rather than seeking a connector and making a robust connection to it.

There were several hurdles associated with designing the software that controls the modules. We had to develop communication algorithms that were able to reliably pass messages from one cube to another. When these failed due to mechanical misalignment, we had to ensure that the high-level algorithms that control the disassembly process were not affected. An important goal was to have a distributed algorithmic solution with performance guarantees for self-disassembly. This leads to solutions for module localization, shape distribution, and disassembly coordination that distribute the data and do not use (or represent) global information.

In the process of implementing our system, we faced numerous hardware design, control, and coordination challenges. When dealing with large self-organizing systems, there are many unsolved problems. We need to deal with the problem of disconnected modules becoming wedged between their neighbors. Empirically, gravity alone is insufficient to complete the self-disassembly process in all cases. Alternative forces such as vibration or rotation of the structure may be necessary. Additionally, if one wished to replace the three steel faces of each module with three additional Magswitches, the modules could be made to repel their neighbors when attempting to extricate themselves from the structure. Another possibility is to perform the detachment process in a layered fashion instead of simultaneously. When all the unused modules disconnect at once, the remaining structure twists and shears enough to prevent some modules from falling away. Releasing modules in a controlled fashion may eliminate this problem.

We also do not yet understand the most efficient way to communicate with all of the modules in a system. Future work should attempt to balance the communication load across all modules. We should also refine our strategy for dealing with malfunctioning units. Because modules often rely on their neighbors’ Magswitches to remain attached to the structure, it should be possible to systematically include or exclude malfunctioning units from the final assembly in some cases. Finally, studying self-organization presents thought provoking problems because it requires vast systems and robust algorithms capable of scalable, adaptive, task-driven distributed coordination and control. Better methods of rapidly fabricating, programming, and debugging such systems are needed.

## VII. ACKNOWLEDGMENTS

The authors would like to thank Iuliu Vasilescu, Marsette Vona, and Carrick Detweiler for extensive discussions and help with the design and construction of this system. We are grateful to the Magswitch and Intel for supporting this work.

## REFERENCES

- [1] P. Pillai, J. Campbell, G. Kedia, S. Moudgal, and K. Sheth, “A 3d fax machine based on claytronics,” in *International Conference on Intelligent Robots and Systems (IROS)*, October 2006, pp. 4728–4735.
- [2] M. Yim, Y. Zhang, K. Roufas, D. Duff, and C. Eldershaw, “Connecting and disconnecting for self-reconfiguration with polybot,” in *IEEE/ASME Transaction on Mechatronics, special issue on Information Technology in Mechatronics*, 2003.
- [3] A. Castano, A. Behar, and P. Will, “The conro modules for reconfigurable robots,” *IEEE Transactions on Mechatronics*, vol. 7, no. 4, pp. 403–409, December 2002.
- [4] A. Kamimura, H. Kurokawa, E. Yoshida, S. Murata, K. Tomita, and S. Kokaji, “Automatic locomotion design and experiments for a modular robotic system,” *IEEE/ASME Transactions on Mechatronics*, vol. 10, no. 3, pp. 314–325, June 2005.
- [5] D. Rus and M. Vona, “Crystalline robots: Self-reconfiguration with compressible unit modules,” *International Journal of Robotics Research*, vol. 22, no. 9, pp. 699–715, 2003.
- [6] A. Castano and P. Will, “Mechanical design of a module for reconfigurable robots,” in *Proc. of International Conference on Intelligent Robots and Systems (IROS)*, 2000, pp. 2203–2209.
- [7] A. Pamecha, I. Ebert-Uphoff, and G. Chirikjian, “Useful metrics for modular robot motion planning,” *IEEE Trans. on Robotics and Automation*, vol. 13, no. 4, pp. 531–45, 1997.
- [8] C. Ünsal and P. Khosla, “Mechatronic design of a modular self-reconfiguring robotic system,” in *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, 2000, pp. 1742–7.
- [9] M. Yim, “Digital clay,” WebSite. [Online]. Available: <http://www2.parc.com/spl/projects/modrobots/lattice/digitalclay/index.html>
- [10] P. White, V. Zykov, J. Bongard, and H. Lipson, “Three dimensional stochastic reconfiguration of modular robots,” in *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2005.
- [11] P. S. Bhat, J. Kuffner, S. Goldstein, and S. Srinivasa, “Hierarchical motion planning for self-reconfigurable modular robots,” in *IEEE International Conference on Intelligent Robots and Systems*, 2006.
- [12] R. Nagpal, “Programmable self-assembly using biologically-inspired multiagent control,” in *Proc. of International Conference on Autonomous Agents and Multiagent Systems*, 2002.
- [13] G. Whitesides and B. Grzybowski, “Self-assembly at all scales,” *Science*, vol. 295, pp. 2418–21, March 2002.
- [14] E. Yoshida, S. Murata, S. Kokaji, A. Kamimura, K. Tomita, and H. Kurokawa, “Get back in shape! a hardware prototype self-reconfigurable modular microrobot that uses shape memory alloy,” *IEEE Robotics and Automation Magazine*, vol. 9, no. 4, pp. 54–60, 2002.
- [15] E. Yoshida, S. Kokaji, S. Murata, K. Tomita, and H. Kurokawa, “Micro self-reconfigurable robot using shape memory alloy,” *Journal of Robotics and Mechatronics*, vol. 13, no. 2, pp. 212–219, 2001.
- [16] P. White, V. Zykov, J. Bongard, and H. Lipson, “Three dimensional stochastic reconfiguration of modular robots,” in *Robotics Science and Systems*. MIT, June 8-10 2005.
- [17] M. Koseki, K. Minami, and N. Inou, “Cellular robots forming a mechanical structure (evaluation of structural formation and hardware design of “chobie ii”),” in *Proceedings of 7th International Symposium on Distributed Autonomous Robotic Systems (DARS04)*, June 2004, pp. 131–140.
- [18] Magswitch, WebSite. [Online]. Available: <http://www.magswitch.com/au/>
- [19] Philips Semiconductor, *The I<sup>2</sup>C-Bus Specification, Version 2.1*, January 2000, document Order No. 9398 393 40011.
- [20] K. Gilpin, “Distributed algorithms for self disassembly for modular robots,” MIT CSAIL M.Eng. Thesis, June 2006.
- [21] K. Gilpin, I. Vasilescu, and D. Rus, “Distributed algorithms for self-dissassembly for modular robots,” in preparation (based on MENG Thesis results), October 2006.