The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Design of Prismatic Cube Modules for Convex Corner Traversal in 3D

Michael Philetus Weller, Brian T Kirby, H Benjamin Brown, Mark D Gross and Seth Copen Goldstein

***Abstract*—The prismatic cube style of modular robot is a promising design for realizing self-reconfigurable 3D lattices. Cubic lattices with prismatic transitions simplify many aspects of the hardware and planning control needed for reconfiguration. Despite much research on how cubic modules can coordinate to reconfigure, until now these transitions have not been fully demonstrated in hardware. We describe our movement primitives for both orthogonal and convex corner transitions with prismatic cube modules. We discuss the design of a hardware module capable of performing these transitions, as well as assess the performance of this hardware in an initial demonstration of these transitions.**

I. INTRODUCTION

This paper describes a hardware system that can realize self-reconfiguring 3D lattices. This system can serve as a testbed to evaluate promising reconfiguration algorithms that until now could only be demonstrated in simulation [1]–[4]. There have been numerous research efforts aimed at developing self-reconfigurable hardware [5], with the result of achieving successful systems with robust docking and actuation in 3D [6], [7]. These systems generally have few degrees of freedom and rely on revolute actuators, and thus lattice reconfiguration requires coordinating many modules in complex planning spaces. In situations where there are too few neighbors, reconfiguration may not even be possible (for example see [8]). Due to the time and expense involved in building prototype hardware systems, ensembles have generally been limited to fewer than one hundred modules [5]. The combination of relatively small ensembles and the need for large numbers of modules to coordinate in reconfiguration has limited the utility of these systems as testbeds for control algorithms.

To provide a system that can serve as a testbed for self-reconfiguring lattices with a relatively small number of modules, we have pursued a module design that is simple to control and requires coordinating relatively few neighbors to reach any nearby lattice cell. Our *prismatic cube* modules (Figure 1) adopt the morphology established by the Telecube [9]; a cube with six independently controlled faces that extend to more than double the breadth of a module. In our system each face is fitted with an electrostatic latch [10] that provides passive alignment and an electronically controlled rigid mechanical connection.

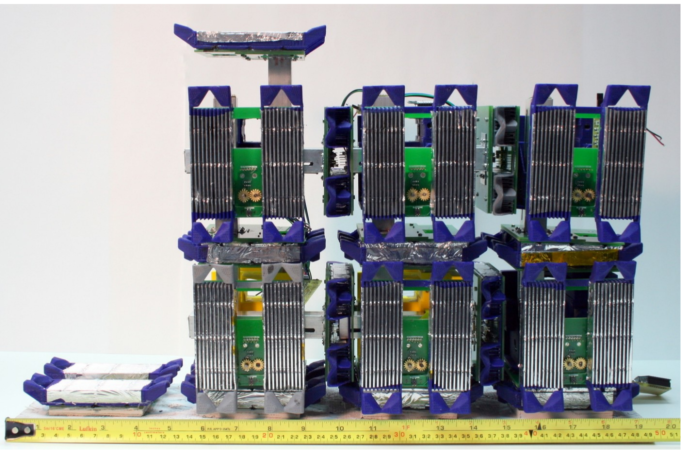


Fig. 1. A $126mm$ lattice of six prismatic cube modules. Video can be seen at http://www.cs.cmu.edu/∼claytronics/iros09/prismaticcubes.

In the following section we discuss related work and its bearing on our design. We then describe in greater detail several guiding considerations for our prismatic cube modules. Then, we describe the hardware implementation and capabilities. Finally we assess the performance of our module hardware in demonstrations of both orthogonal and convex diagonal transitions, and discuss directions for future work.

II. RELATED WORK

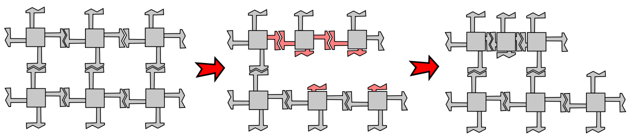
Our design decisions have been informed by previous research on producing hardware modules for self-reconfiguring 3D lattices. Here we discuss some previous results that have led us to investigate prismatic cube style hardware.

A. Lattice-Constrained Modules

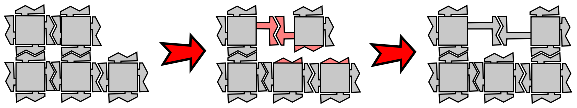
Polybot [11] demonstrated that unconstrained modules can determine their relative position in space and then generate a plan to bring the modules into mechanical alignment so that they can dock. However this is a complex and time-consuming strategy. Many systems [6], [7], [9] simplify docking by restricting module positions to a regular lattice. The modules already in position then mechanically constrain new neighbors; which helps to maintain the lattice. Assuming that modules are constrained to this regular lattice also greatly simplifies the problem of localization. A passive alignment mechanism potentially enables lattice-constrained modules to dock with neighbors without requiring any active alignment.

M P Weller and M D Gross are with the Computational Design Lab, Carnegie Mellon University {philetus,mdgross}@cmu.edu; B T Kirby and S C Goldstein are with Claytronics, Carnegie Mellon University, {bkirby, seth}@cs.cmu.edu; H B Brown is with Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA, hbb@cs.cmu.edu.

978-1-4244-3804-4/09/$25.00 ©2009 IEEE

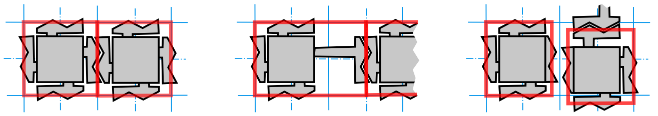


(a) In an open lattice cubes move by contracting into a single lattice cell and then expanding again.



(b) In a closed lattice cubes can generally move across one lattice cell all at once.

Fig. 2. Open-lattice (top) vs closed-lattice (bottom) actuation.



(a) With faces in closed state cubes pack in a tight lattice.

(b) Opening a face of a cube extends the width by half of a cell.

(c) With faces contracted a cube can pass by a neighboring cube.

Fig. 3. The three extension states required to support closed-lattice actuation.

B. Revolute vs Prismatic Actuation

Several of the most successful three-dimensional lattice systems have reconfigured using revolute degrees of freedom. An advantage of this strategy is that individual modules can have as few as one [6], [12] or two [7], [13] degrees of freedom, simplifying construction so that relatively large numbers of modules can be manufactured. However a critical disadvantage with revolute actuation is that lattice movements are complex to plan and require many steps to perform. Modules tend to move much as a knight jumps across a chess board, and it often takes several intermediate movements for a module to reach even an immediately adjacent cell. Revolute motion causes a module to sweep through a volume larger than itself, and if there are obstacles or bottlenecks in the local lattice, it may not be possible to reach adjacent lattice cells [8]. These movement constraints can confound planning methods such as gradient descent.

C. Open-lattice vs Closed-lattice Actuation

The first prismatic-cube-style system, the Crystalline Atom, arranged modules in an open lattice [14] as shown in Figure 2a. This arrangement suited the Atoms hardware as it only moved in 2D on a tabletop, but it is ill-suited for 3D lattices as even simple orthogonal movements involve large moment arms (i.e. the length of two fully extended modules).

The developers of the Telecube recognized this limitation and proposed that modules could instead be arranged in a closed lattice [15], as shown in Figure 2b. Closed-lattices are structurally more robust, but place additional demands on the hardware. In an open-lattice system faces can all extend and retract together [14]. However, practical closed-lattice reconfiguration requires independently actuated faces. As we discuss in Section III below, faces must also support an extended range of motion.

D. Control Abstractions

It is generally desirable to insulate higher-level planning from the complexities of coordinating groups of hardware modules to move between lattice cells. To this end, many reconfiguration algorithms create plans at the metamodule level [1], [2], [4], [14], [16], rather than individual hardware modules. One popular example is the *sliding cubes* metamodule introduced in [16]. A sliding cube metamodule can move laterally or diagonally into any adjacent lattice cell. The advantage of the sliding cube abstraction is that it simplifies planning, which has fostered the development of several interesting control algorithms [1], [2]. The disadvantage is that realizing a single sliding cube can require a large number of modules, making it impractical as a control scheme for smaller ensembles. For example, the Crystalline Atom's metamodule requires 64 modules (4x4x4) and has only been demonstrated in 2D; ATRON is one of the few systems to have realized a sliding cube in hardware, but requires 12 modules for each metamodule [8].

To perform interesting self-reconfiguration experiments on ensembles of up to 100 ATRON hardware modules, the developers promulgated the idea of an emergent metamodule [17]. Rather than dividing the structure into a lattice of metamodules, when a module wants to move it opportunistically recruits neighboring modules to form a temporary metamodule; once the movement is completed the metamodule is dissolved.

III. Design Considerations

In the design of our prismatic cube modules we sought to satisfy three criteria: (1) docking and undocking should require as little planning as possible; (2) modules should be able to move to all adjacent lattice cells, and require as few neighbors as possible to do so; and (3) the module should be as simple as possible; in particular it should have as few, simple mechanical subsystems as possible.

As our aim is to run higher-level planning algorithms on a relatively small ensemble, we have given the greatest weight to the first two criteria. To this end we chose a prismatic cube form-factor designed for a closed lattice. Prismatic cubes simplify docking (criterion one) and make it possible to move modules to adjacent lattice cells (criterion two) with the assistance of only a few neighbors. Unfortunately, this also requires the implementation of six independently actuated faces, running somewhat counter to criterion three. Even so, our use of electrostatic rather than mechanical latches keeps the total mechanical degrees of freedom relatively low, satisfying the third criteria to a limited extent.

To establish the parameters of our hardware modules we began by designing the control abstraction that would serve as an interface to higher-level planning systems. The series of movements the hardware must perform to realize these atomic transitions described below constrain our design.

(a) A single prismatic cube moving laterally.

(b) A pair of modules can move diagonally around a convex corner.

Fig. 4. A single prismatic cube hardware module can move vertically or laterally by coordinating with a single neighbor (left); a group of six modules coordinates to transition two modules around a convex corner (right).

A. *Face Extension States*

As mentioned in Section II, to support closed-lattice actuation the faces of a module must be independently actuated and must reach three different lengths of extension. As shown in Figure 3 these states are: a *closed* state where a face latches to the face of a module in the neighboring lattice cell; an *open* state in which two neighboring modules can latch across a one-cell-wide gap in the lattice; and a *contracted* state which allows one module to move past another in an adjacent cell without interference. For example, in Figure 2b the bottom face of the moving module as well as the top faces of the neighbors it is moving over must all be contracted during the transition to avoid interference. Supporting three extension states makes our hardware modules more complex in violation of our third criterion above. However our experience suggests that the alternative, open-lattice actuation, is untenable in 3D due to the moment-arms that would have to be resisted.

B. *Movement Primitives for 3D Reconfiguration*

We have designed our prismatic cube modules to reconfigure by recruiting neighbors to perform movement primitives, a scheme similar to the temporary metamodules used to reconfigure ATRON modules [17]. Prismatic cubes utilize two key movement primitives for general reconfiguration: orthogonal translation and convex corner traversal (Figure 4). While planning with movement primitives is somewhat more complex than planning with a lattice of uniform metamodules, it allows finer-grained control and reconfiguration on smaller ensembles. As these two movement primitives provide access to immediately adjacent lattice cells (satisfying our second criterion) we believe that applying higher-level planning algorithms will remain straightforward as the configuration space does not grow in complexity during intermediate states.

The most significant strain on the hardware occurs when traversing a convex corner against gravity (Figure 4b). This transition is illustrated in Figure 5. Prominently, in the first and third steps the system must cantilever two modules with a large moment-arm supported by a single module. The forces that must be overcome to accomplish these transitions establish the parameters of our design. In the following section we describe how we developed our hardware modules to execute this series of motions.

IV. HARDWARE

The 3D prismatic cubes have been designed to perform the convex corner motion primitive against normal gravitational loading. It utilizes six electrostatic latches mounted on arms actuated to move in and out of a central core (a partially exploded rendering is shown in Figure 6). To minimize deflection we attempted to minimize the weight of the system and the size of the lattice, and maximize the rigidity of the structural components and intermodule connections. By limiting module weight and the size of the lattice we reduced the torques that our latches must withstand. This avoided a vicious circle of ballooning mechanical requirements.

Our final weight for a complete robot was $773g$. Our final lattice size of $126mm$ was limited primarily by latch strength and the interactions shown in Figure 7. In the contracted state, as illustrated, the entire latch assembly (from fins to the opposite end of the arm) must fit within the lattice cell to avoid interference with neighboring modules.

An innovation that allows our latch to be relatively deeper than the Telecube's is that we allow our arm to pass through a void in the opposite face (shown in Figure 8) when both are contracted. Without our pass-through face design the length of the arm would have to fit entirely within the core when fully contracted.

A. *Latching*

Docking between modules uses electrostatic (ES) latching. This mechanism requires no moving parts to latch or unlatch. Shown in Figure 8, the latch is a pair of vertical comb structures overlaid with a thin film of aluminized mylar. Pairs of combs are required to allow charge accumulation using independent voltage sources [10]. The latching force is proportional to the surface area of the mylar as well as the excitation voltage. Our experience with the drive electronics indicated that the practical excitation voltage is limited to $1000V$. To further increase latch strength in a given volume we increase the surface area by varying comb pitch and depth.

An advantage of our latch design is that actuation and docking are axis-aligned so that we can use one larger motor for both. Other designs that use a secondary motor to drive a pin [11], key [14] or claw [6] must make trade-offs between the power of the main actuation motors, the secondary latching motors that fight intermodule binding, and the overall size and weight of the module.

In keeping with the size considerations illustrated in Figure 7, the latch must be as shallow as possible while still having enough strength to prevent unwanted separations. The depth is important as it directly affects the minimum lattice spacing. The latch depth in Figure 7 is a strong constraint, adding 2mm to the lattice size for each 1mm of depth.

The latches also feature a passive self-alignment mechanism consisting of four male pins on one side and four female slots on the opposite latch. This is designed to correct for both lateral and angular misalignment that is introduced into the system under gravity due to deflection of components and less than completely rigid inter- and intra-module connections. In

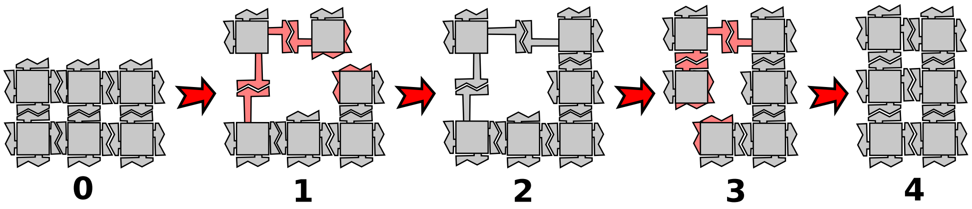


Fig. 5. Series of movements to translate a pair of modules around a convex corner.

particular, while lateral misalignment can be corrected in some situations through the coordination of neighbors, angular misalignment cannot. Thus the self-alignment mechanism must be able to correct for significant angular misalignment of up to thirty degrees.

With comb structures, binding can be an issue when attempting to undock. This is caused in certain orientations when module loading encourages the fins to pinch together rather than remaining parallel. By making the individual fins thinner and more compliant we have been able to minimize this effect. Our passive self-alignment features, necessary to ensure the fins are properly aligned before insertion, also had to be carefully designed to prevent binding during undocking.

Our final latch design, shown in Figure 8, withstands 300N in dead-lift conditions. Using nineteen pairs of overlapping fins on each comb, we achieve a total surface area of $39600mm^2$. This is achieved within an overall latch dimension of $80mm$ x $60mm$ x $18mm$.

B. Arms

The arms provide a rigid connection between the latch and the core while permitting the length to vary as needed for reconfiguration. Due to the constraints illustrated above (Figure 7), the overall travel is actually longer than can be housed within the core. To maintain simplicity and rigidity our arms are each a single piece of extruded aluminum.

Arms must maintain rigidity across relatively long distances, and it is critical to minimize sag and rotational play. We made the cross-sectional area of the arm as large as possible to resist the developed moments, and used a square tube to resist torsion. With such large internal dimensions, we consumed most of the available space within the core and thus found it most effective to place the linear actuator mechanism and control circuitry within the hollow expanse of the arm. A custom extrusion made it possible to achieve our desired dimensions while minimizing the wall thickness (and weight).

The final arm design is shown in Figure 9 with its major components depicted. The linear actuator components are located within the extruded aluminum tube. The captive nut that travels along the actuator is accessible through slots in the tube. All control circuitry, including the high voltage generation and switching for the latch, is safely contained within the tube as well. The latches mount to the face plate, which is reinforced and has a void to allow the other arm to pass through in its retracted state. A linear contact track, discussed in the next section, is attached to the outer surface of the aluminum tube and engages wiper contacts mounted on the core guide surface.

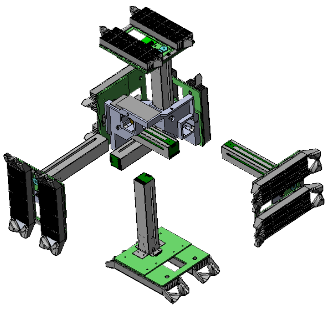


Fig. 6. Exploded view of prismatic module.

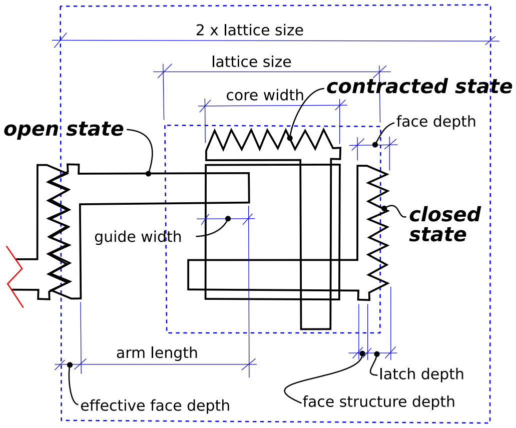


Fig. 7. Size and scaling parameters in the prismatic module design for the three states of a closed lattice system.

C. Core

The core's main purpose is to provide a rigid connection between all the arms, maintaining the proper orientations in the lattice. To minimize the lattice size, the core dimensions are essentially the total width of the three axes of arms with a minimum amount of enveloping plastic structure. Another important part of the core that has an effect on the overall lattice size are the guide surfaces. These envelope the prismatic arms and provide a bearing surface to resist moments developed as the arms extend and the latches support neighboring modules.

As the arms take up most of the working area of the

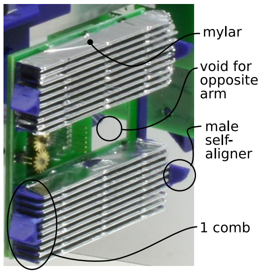


Fig. 8. Closeup view of ES latch assembly with male passive alignment.

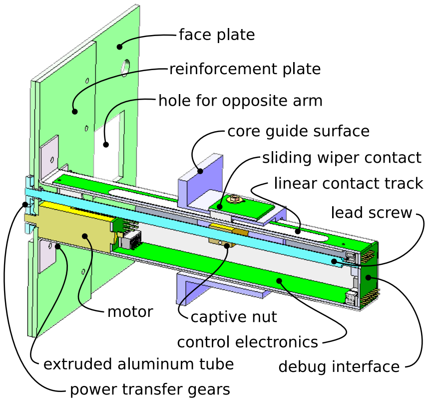


Fig. 9. Cutaway view of prismatic arm assembly with internal linear actuator and control and latch mounting face.

core, only a few small areas are free for control circuitry and batteries. To make use of commercially available batteries with an appropriate current capability, we had to slightly increase the module (and consequently, the lattice) size. Three lithium polymer batteries in series provide 11.1V nominal with a capacity of 750mAH and peak current capability of 9A, which is sufficient to actuate all six arms under load simultaneously. For testing we also make use of tethers to provide power (as seen in Figure 5).

The extreme range of motion of the arms within the tightly packed core as well as the fairly small scale of the modules made most methods of electrical connection impractical. We implemented a linear sliding contact system which allowed the arm to travel while the electrical connections on the core remain fixed. With sliding contacts, fewer connections are better. We implemented the control in a distributed fashion, with a dedicated microprocessor in each arm. Module actions are coordinated by this master controller through a single half-duplex RS-485 bus. Thus, our wiring is comprised solely of battery wires and small 4-wire cables to the stationary components of the sliding contacts. The continuous outer tracks in Figure 10 provide uninterrupted power and communications while the internal track pattern provides absolute position.

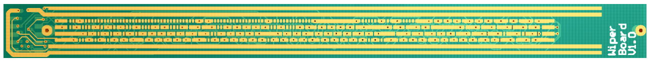


Fig. 10. Sliding linear wiper tracks provide power, communication, and absolute position.

Since the linear actuator mechanism is internal to the arm, slots are cut on two sides of the arm to allow access to the traveling nut on the lead screw. This nut is rigidly affixed to the core by inserting a pin through it and anchoring it on either side in the core body. With our electrical connections established with sliding contacts, this pin serves as our only mechanical connection to the core. This makes it a very simple process to remove and replace arms for maintenance purposes.

V. DEMONSTRATION

To validate the mechanical aspect of our design we demonstrated both the orthogonal and convex transitions in hardware by teleoperating the modules. Since the greatest mechanical strains are induced during convex corner traversal we focus primarily on the latter. In order to do this, we constructed six robots with four faces each, allowing full movement within the plane of traversal. By setting up the lattice vertically we tested the system against gravity; by constructing additional modules we will have a modular system capable of self-reconfiguration in three dimensions.

In this demonstration the individual modules are sent remote commands from the operator to either engage or disengage an electrostatic latch, or to extend or retract a face by a given distance. The diagram shown in Figure 5 illustrates the motions of a group of modules performing convex corner traversal in five stages. Figure 11 shows the same five stages as our hardware modules successfully demonstrate this movement primitive.

In performing orthogonal transitions we are generally able to merely extend a face out the proper distance and the system is rigid enough that any misalignment due to deflection is corrected by our latch's passive alignment mechanism. But during convex translation two of the most difficult steps involve multi-module cantilevers; during steps one and three active correction is required. In step one as the module on the upper right is lifted onto the next level of the lattice the misalignment due to deflection is greater than can be corrected by the passive aligner alone. The module underneath must contract both vertical faces to allow latching to succeed. In step three as the module on the upper right holds two others there is enough deflection to cause interference with the module on the bottom left. To correct for this the module on the upper right must extend its bottom face to provide additional clearance.

In the current demonstration the operator judges when it is necessary to make active corrections. To allow the modules' software to perform this active correction autonomously in the future we have included multi-axis accelerometers in each arm of each module. By sensing the angle (with accelerometers) and extension (with absolute encoders) of each face during transitions the modules will be able to determine when active corrections are necessary to avoid interference or to successfully align.

VI. DISCUSSION

Lattice modules based on prismatic rather than revolute actuation simplify the planning required for 3D self-

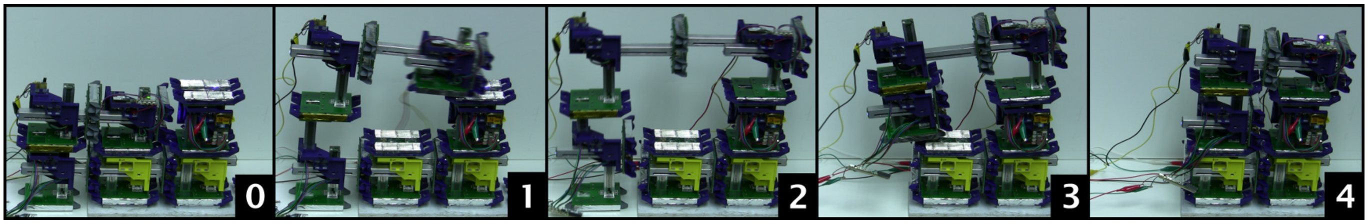


Fig. 11. Modules navigating the convex corner under gravity. Significant deflection occurs in the cantilevered module in steps 1 and 3.

reconfiguration. By using our opportunistic movement primitives modules can move orthogonally or diagonally to reach adjacent lattice cells with the coordination of small numbers of neighbors. After careful examination of the difficult convex corner traversal movement primitive we have developed a hardware module capable of completing the necessary motions in 3D under gravitational load. By successfully performing a convex translation with only six modules we have demonstrated the promise of prismatic-cube-style modules as a platform for experimenting with higher-level planning in hardware.

One of main design considerations from Section III was to dock and undock with minimal planning. During our testing, we observed deflections beyond the capabilities of our passive self-alignment mechanism that required additional extension and contraction to address. We were able to actively correct by using additional motion in the plane of deflection. To enable fully autonomous motion, we have included multi-axis accelerometers and absolute encoders in each arm and are currently developing software to autonomously detect misalignment significant enough to require active correction and to respond appropriately.

Another issue with the hardware has been binding during latching and unlatching. We have observed that extending bound faces often causes further angular misalignment, increasing binding in a sort of snowball effect. We believe that this variety of failure can also be corrected by detecting the telltale change in angle resulting from binding and adjusting orthogonal faces to alleviate it.

Our goal for the next stage of development is to take advantage of these sensing capabilities to allow the module hardware to perform a variety of movement primitives completely autonomously. In parallel we intend to demonstrate several existing higher-level planning algorithms running on top of these movement primitives in simulation. We believe that demonstrating both autonomous low-level reconfiguration and interesting higher-level behaviors will make a strong case for developing ensembles of prismatic-cube-style modules as a platform for experimentation.

VII. ACKNOWLEDGEMENTS

This research was supported in part by the National Science Foundation under Grants ITR-0326054 and CNS-0428738. We would also like to thank Sam Tuchin for his assistance assembling ES flaps.

REFERENCES

[1] K. Stoy and R. Nagpal, “Self-reconfiguration using directed growth,” in *Distributed Autonomous Robotic Systems (DARS)*, 2004, pp. 1–10.

[2] R. Fitch and Z. Butler, “Million module march: Scalable locomotion for large self-reconfiguring robots,” *Intl Journal of Robotics Research*, vol. 27, no. 3–4, pp. 331–343, 2008.

[3] M. De Rosa, S. C. Goldstein, P. Lee, J. D. Campbell, and P. Pillai, “Scalable shape sculpting via hole motion: Motion planning in lattice-constrained module robots,” in *Intl Conf on Robotics and Automation (ICRA)*. IEEE, 2006.

[4] D. Dewey, S. S. Srinivasa, M. P. Ashley-Rollman, M. De Rosa, P. Pillai, T. C. Mowry, J. D. Campbell, and S. C. Goldstein, “Generalizing metamodules to simplify planning in modular robotic systems,” in *Intelligent Robots and Systems (IROS)*. IEEE, 2008.

[5] M. Yim, S. Wei-Min, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. S. Chirikjian, “Modular self-reconfigurable robot systems,” *Robotics and Automation*, vol. 14, no. 1, pp. 43–52, 2007.

[6] E. H. Ostergaard, K. Kassow, R. Beck, and H. H. Lund, “Design of the atron lattice-based self-reconfigurable robot,” *Autonomous Robots*, vol. 21, no. 2, pp. 165–183, 2006.

[7] S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, and S. Kokaji, “M-tran: self-reconfigurable modular robotic system,” *Transactions on Mechatronics*, vol. 7, no. 4, pp. 431–441, 2002.

[8] D. J. Christensen and K. Stoy, “Selecting a meta-module to shape-change the atron self-reconfigurable robot,” in *Intl Conf on Robotics and Automation (ICRA)*. IEEE, 2002, pp. 2532–2538.

[9] J. W. Suh, S. B. Homans, and M. Yim, “Telecubes: mechanical design of a module for self-reconfigurable robotics,” in *Intl Conf on Robotics and Automation (ICRA)*, vol. 4. IEEE, 2002, pp. 4095–101.

[10] M. E. Karagozler, J. D. Campbell, G. K. Fedder, S. C. Goldstein, M. P. Weller, and B. W. Yoon, “Electrostatic latching for inter-module adhesion, power transfer, and communication in modular robots,” in *Intelligent Robots and Systems (IROS)*. IEEE, 2007, pp. 2779–2786.

[11] M. Yim, D. G. Duff, and K. D. Roufas, “Polybot: a modular reconfigurable robot,” in *Intl Conf on Robotics and Automation (ICRA)*, vol. 1. IEEE, 2000, pp. 514–20.

[12] V. Zykov, E. Mytilinaios, M. Desnoyer, and H. Lipson, “Evolved and designed self-reproducing modular robotics,” *Trans on Robotics*, vol. 23, no. 2, pp. 308–319, 2007.

[13] B. Salemi, M. Moll, and W.-M. Shen, “SUPERBOT: A deployable, multi-functional, and modular self-reconfigurable robotic system,” in *Intelligent Robots and Systems (IROS)*. IEEE, 2006, pp. 3636–3641.

[14] D. Rus and M. Vona, “Crystalline robots: Self-reconfiguration with compressible unit modules,” *Autonomous Robots*, vol. 10, no. 1, pp. 107–124, 2001.

[15] S. Vassilvitskii, M. Yim, and J. Suh, “A complete, local and parallel reconfiguration algorithm for cube style modular robots,” in *Intl Conf on Robotics and Automation (ICRA)*, vol. 1. IEEE, 2002, pp. 117–122.

[16] Z. Butler, K. Kotay, D. Rus, and K. Tomita, “Generic decentralized control for a class of self-reconfigurable robots,” in *Intl Conf on Robotics and Automation (ICRA)*. IEEE, 2002, pp. 809–816.

[17] D. J. Christensen, E. H. Ostergaard, and H. H. Lund, “Meta-module control for the ATRON self-reconfigurable robotic system,” in *Intelligent Autonomous Systems (IAS)*, 2004, pp. 685–692.