Evolved and Designed Self-Reproducing Modular Robotics

Victor Zykov, *Student Member, IEEE*, Efstathios Mytilinaios, Mark Desnoyer, and Hod Lipson, *Member, IEEE*

***Abstract*—Long-term physical survivability of most robotic systems today is achieved through durable hardware. In contrast, most biological systems are not made of robust materials; long-term sustainability and evolutionary adaptation in nature are provided through processes of self-repair and, ultimately, self-reproduction. Here we demonstrate a large space of possible robots capable of autonomous self-reproduction. These robots are composed of actuated modules equipped with electromagnets to selectively control the morphology of the robotic assembly. We show a variety of 2-D and 3-D machines from 3 to 2*n* modules, and two physical implementations that each achieves two generations of reproduction. We show both automatically generated and manually designed morphologies.**

***Index Terms*—Evolutionary computation, modular robotics, self-repair, self-replication, self-reproduction.**

I. Introduction

SELF-REPRODUCTION is a process whereby a physical system is capable of producing another autonomous, functional system. Such system is called a self-replicator if the resulting system is an exact replica of the original [1]. The copy, by definition, also needs to be capable of self-reproduction. The concept of artificial self-replication had attracted attention over several decades [2], [3]. It facilitates understanding of one of the basic tenets of biological life, and represents a potentially important paradigm for machine sustainability and adaptation. Long-term sustainability and adaptation of robotic systems are achieved today mostly through durable hardware and adaptive controllers. In contrast, most biological systems are not made of robust materials; long-term sustainability and evolutionary adaptation in nature are provided instead through processes of self-repair and, ultimately, self-reproduction.

Self-reproduction differs from automatic manufacturing [4] or self-assembly [5], [6], where the resulting system is not necessarily capable of making, catalyzing, or in some way inducing more copies of itself. At small scales, such as at the molecular level, self-replication can occur through *stochastic* self-assembly processes catalyzed by the original self-replicating entity. At large scales, such as large multicellular organisms and robots, self-replication through ambient stochastic processes is energetically implausible, and *deterministic* self-reproduction is more plausible. However, macroscale stochastic self-replication powered by external artificial agitation has been recently demonstrated [7], [8].

Physical self-reproduction is potentially of great practical relevance, as it is the ultimate form of self-repair [9], [10]. The practical potential of physical self-reproduction was recognized in the early 1980s as a possible method for remote lunar colonization [11], but was abandoned due to many unresolved technical difficulties. More recently, interest is being revived in the area of kinematic self-replication [12] as a paradigm for both long-term self-sustaining and self-repairing robotic ecologies for space [13] and hazardous applications, and as a paradigm for micro- and nanoscale manufacturing [14]. Machine self-reproduction can enable new performance approaches. If one machine is insufficient to perform a task, given an appropriate supply of materials and parts, it can replicate the necessary amount of times, and the resulting group of machines may cope with the assignment in collaboration [15]. An artist's rendition of a possible future self-sustaining modular robotic system is shown in Fig. 1. However, physical machines capable of self-reproduction have been scarcely examined.

Artificial physical self-replication at macroscale was first demonstrated by Penrose [16] using stochastic tumbling blocks with special geometries and latching mechanisms. Deterministic self-reproduction of robotic systems was first demonstrated by Jacobson [17], who constructed a self-replicating system from locomotive toy cars operating on a fixed rail plan. A similar approach was taken by Chirikjian [18], who demonstrated a 2-D LEGO® robot composed of four modules that was able to assemble four other modules into a new identical robot by following tracks drawn on the ground.

Jacobson's and Chirikjian's work demonstrated physical, deterministic self-reproducing machines. In doing so, they raised several important questions: Are such simple reproduction processes comparable to what we consider as "true" self-reproduction in nature, and can that design space scale to more complex

Manuscript received March 15, 2006; revised September 5, 2006. This paper was recommended for publication by Associate Editor S. Ma and Editor L. Parker upon evaluation of the reviewers' comments. This work was supported in part by the NASA Program for Research in Intelligent Systems under Grant NNA04CL10A. This paper was presented in part at the 9th International Conference on the Simulation and Synthesis of Living Systems (ALIFE9), Boston, MA, September 2004.

V. Zykov is with the Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853 USA (e-mail: viktor.zykov@cornell.edu).

E. Mytilinaios was with the Department of Computing and Information Science, Cornell University, Ithaca, NY 14853 USA. He is now with Ingenio Inc., San Francisco, CA 94111 USA (e-mail: em249@cornell.edu).

M. Desnoyer is with the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14853 USA (e-mail: md246@cornell.edu).

H. Lipson is with the Schools of Mechanical and Aerospace Engineering and Computing and Information Science, Cornell University, Ithaca, NY 14853 USA (e-mail: hod.lipson@cornell.edu).

This paper has supplementary multimedia material available at http://ieeexplore.ieee.org, provided by the author. This material includes three video files containing footage of two generations of self-replication of various Molecube™ robots. These files are 12.3, 12.3, and 1.9 MB in size.

Color versions of Figs. 1–9 and 11–15 are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2007.894685

1552-3098/$25.00 © 2007 IEEE

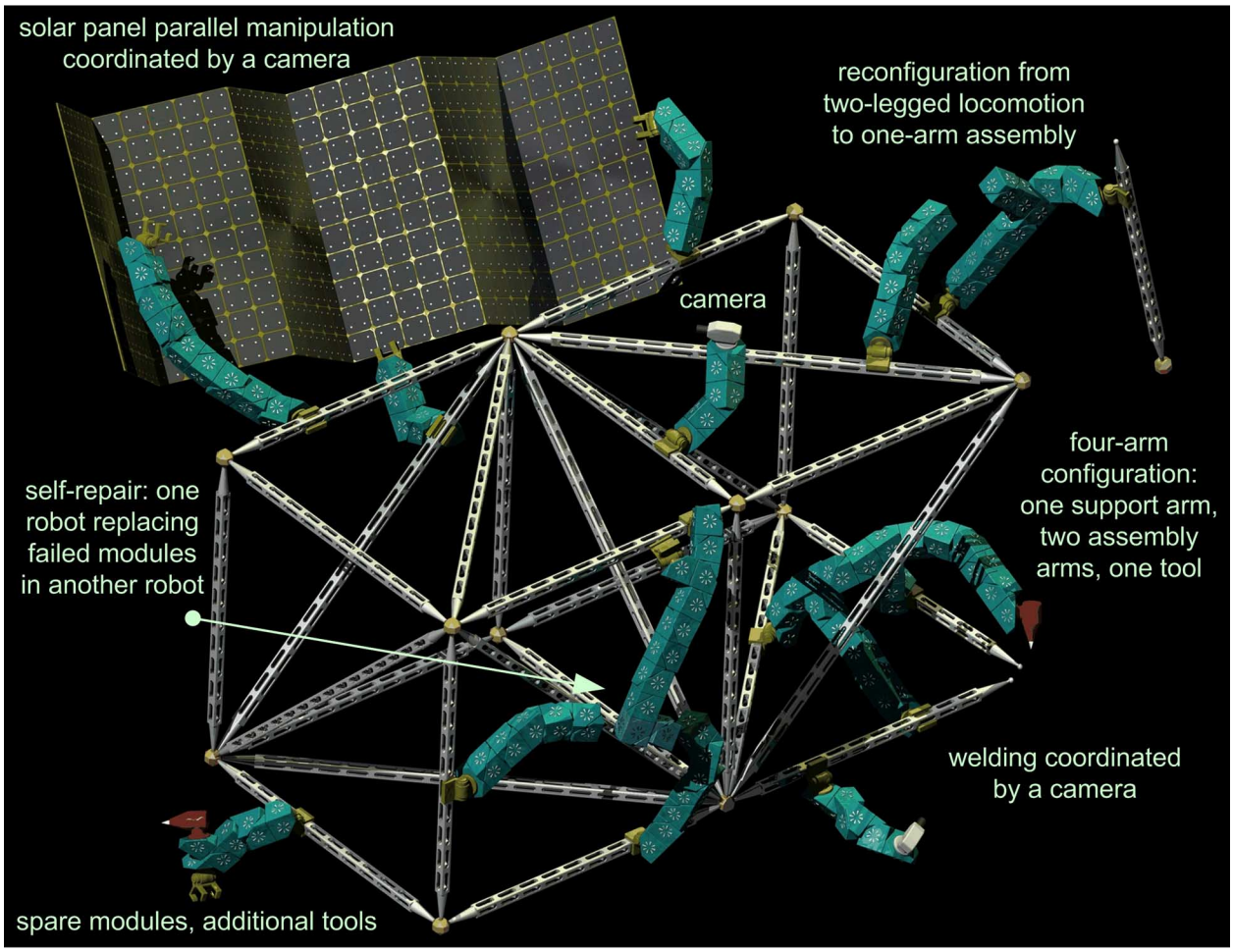


Fig. 1. Artist's rendition of a space application of the modular robotic system, involving self-repairing activity, coordinated manipulation, and reconfiguration.

reproducing systems? Moreover, are the ground tracks themselves to be considered part of the machine, and therefore also need to be self-replicated for it to qualify as full self-replication? To illustrate this point, consider a robot composed of two components capable of reproducing by actively assembling these two components together. Self-replication has indeed occurred, but in a way nowhere as impressive as a machine able to reproduce itself from raw materials or as biological life that is able to reproduce from amino acids; nor does it possess the same scaling potential into more complex systems. Such questions regarding "what counts as self-replication" recur each time a claim is made in this field. We therefore suggest a different metric that captures the subtleties of different forms of self-reproduction.

A closer examination of self-reproduction shows that it is not a clear-cut property, as it may first appear. Self-reproducing systems may differ in dimensions such as quality, rate, and amount of self-reproduction. Self-reproduction inevitably requires material and energy supply from the environment, but the level of dependency on the environment may also differ across systems. For example, atomic patterns of a mineral crystal can replicate exactly, but only in a specific chemical solution. On the other hand, rabbits can reproduce in a much broader range of environmental conditions, but do so less precisely.

These various aspects can be collapsed into a single continuum, quantifiable based on the amount of information being replicated. For example, the amount of information needed to reproduce a two-component robot given these two components is less than the amount of information needed to reproduce a robot given many lower-level components. Similarly, the amount of information needed to reproduce an inaccurate copy of a machine is less than the amount of information necessary to produce a more exact duplicate. The amount of information needed to assemble a machine that is likely to independently and spontaneously appear in a domain is less than the amount of information needed to assemble a unique machine unlikely to appear by chance. The amount of information involved in the replication depends on the definition of the replicating system and the contribution of the environment to the replication process.

Based on these assumptions, we have formally proposed a domain-independent metric R of self-replicability [19]

$$\begin{aligned} R_T &= (S, E_s, E_0, \varepsilon, \tau_0, \tau_1) \\ &= \log \frac{\int_{t=\tau_0}^{\tau_1} P_\varepsilon(T(E_s,t), S)dt}{\int_{t=\tau_0}^{\tau_1} P_\varepsilon(T(E_0,t), S)dt}. \end{aligned} \tag{1}$$

Here, R_T is the relative replicability of system S over the period of time $t \in [\tau_0, \tau_1]$. This metric compares the probability of emergence P_ε of system S in environment E_s which at time

τ_0 already contained one instance of system S to the probability that this system would spontaneously appear in environment E_0 where system S was not present *a priori*. The presence of systems is compared with some finite tolerance ε. Details and examples of application of this metric are provided in depth elsewhere [19].

This metric not only avoids some of the difficulties of Von Neumann's "binary" definition [20], [21] (what counts as a nontrivial replication, and how to apply it to nonformal systems), but also provides a graded value that can serve as a basis for comparison and improvement of physical and artificial self-reproducing systems.

Following this formal definition, we have sought a physical substrate in which a variety of self-reproducing machines can be constructed and analyzed systematically. We also seek a substrate that is both physically plausible and faithfully simulatable, so that questions about physical self-replication can be studied computationally.

II. Physical Self-Replicating Machines

Modular self-reconfigurable robots are machines composed from robotic modules that can change their shape and topological connectivity, thus changing the overall morphology of the robotic structure [22]–[28]. These capabilities make them a particularly appealing substrate for self-replication. The space of machines we put forward is based on a single cubical building block, which we call a "Molecube" (see detailed description below). We believe that other known types of modular self-reconfigurable robots, such as M-TRAN [25], Polybot [29], ATRON [30], CONRO [31], and Superbot [32], may be composed into structures capable of kinematic self-reproduction; however, their physical reproductive functionality has not been investigated so far.

In choosing a conceptual design for the modular robots, we were looking for a modular substrate that could provide us with a variety of reconfiguration capabilities and, at the same time, be simple to construct. Following our proposed metric, self-replicability is increased when the machine is composed of simpler units and larger numbers of units, suggesting that simpler one-motor units are preferable to fewer two-motor units. To facilitate the precision of complex shape formation, we selected among symmetric module designs occupying unit volume in a 3-D lattice. To simplify the mechanical design, we only considered the homogeneous robot systems composed of robots with one rotational degree of freedom (DOF). These criteria limited our selection to the concepts summarized in Fig. 2 and compared in Table I. For Molecubes, we chose the design shown in Fig. 2(d). This design is the only one of the four that enables a single module to perform out-of-plane reconfiguration; also it allows the construction of dense structures.

A. Conceptual Design

Each Molecube, as shown in Fig. 3(a), is split into two parts along a plane that is perpendicular to its long diagonal (e.g., vector [1, 1, 1] in 3-D); the plane is shaded in Fig. 3(a). For reconfiguration along the lattice, one half of a cube can swivel relative to another half about the long axis in increments of 120°, each time cycling three faces of the cube. In this way, every module has one actuated DOF. Swiveling a cube with other cubes attached to it may cause the attached structure to reconfigure. For example, the cube shown in Fig. 3(b) has another cube attached to its left side. Swiveling the cube shown in solid lines causes the adjacent cube to cycle through three positions. In this way, the concept originally proposed in [33] is here physically implemented.

Each cube has three possible swivel states, and four possible orientations for the swivel axis. Fig. 3(c) shows a physical model of a set of five nonactuated Molecubes. Swiveling two of these

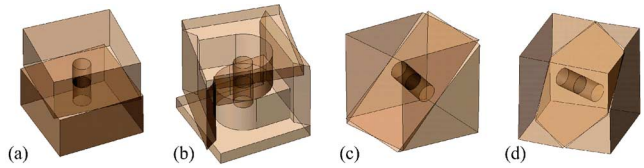


Fig. 2. Symmetric modular robot concepts, each occupying unit volume in 3-D grid.

TABLE I
Comparison of Modular Robot Concepts Shown in Fig. 2

Design type	(a)	(b)	(c)	(d)
No. of degrees of freedom	1	1	1	1
No. of connectable surfaces	2	4	4	6
No. of reachable lattice locations	1	3+1	2	3
Planar Manipulator	Yes	Yes	Yes	No

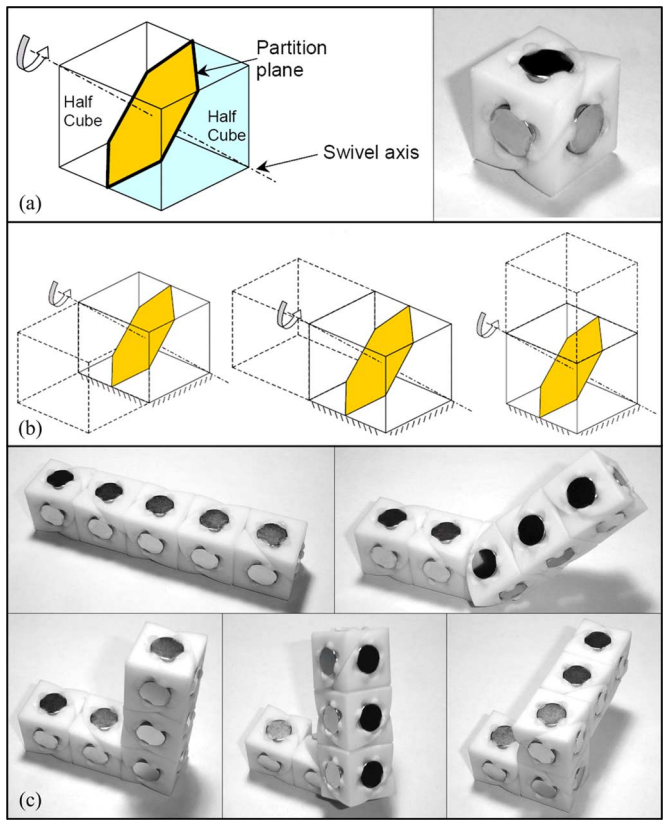


Fig. 3. Geometry of 3-D "Molecubes." (a) A cube is split into two halves along the (1,1,1) plane. (b) Swiveling the left half causes an adjacent block to cycle into new configurations. (c) A physical model of a 3-D Molecube: A sequence of swivels transforming a line to an L-shape to a 3-D structure.

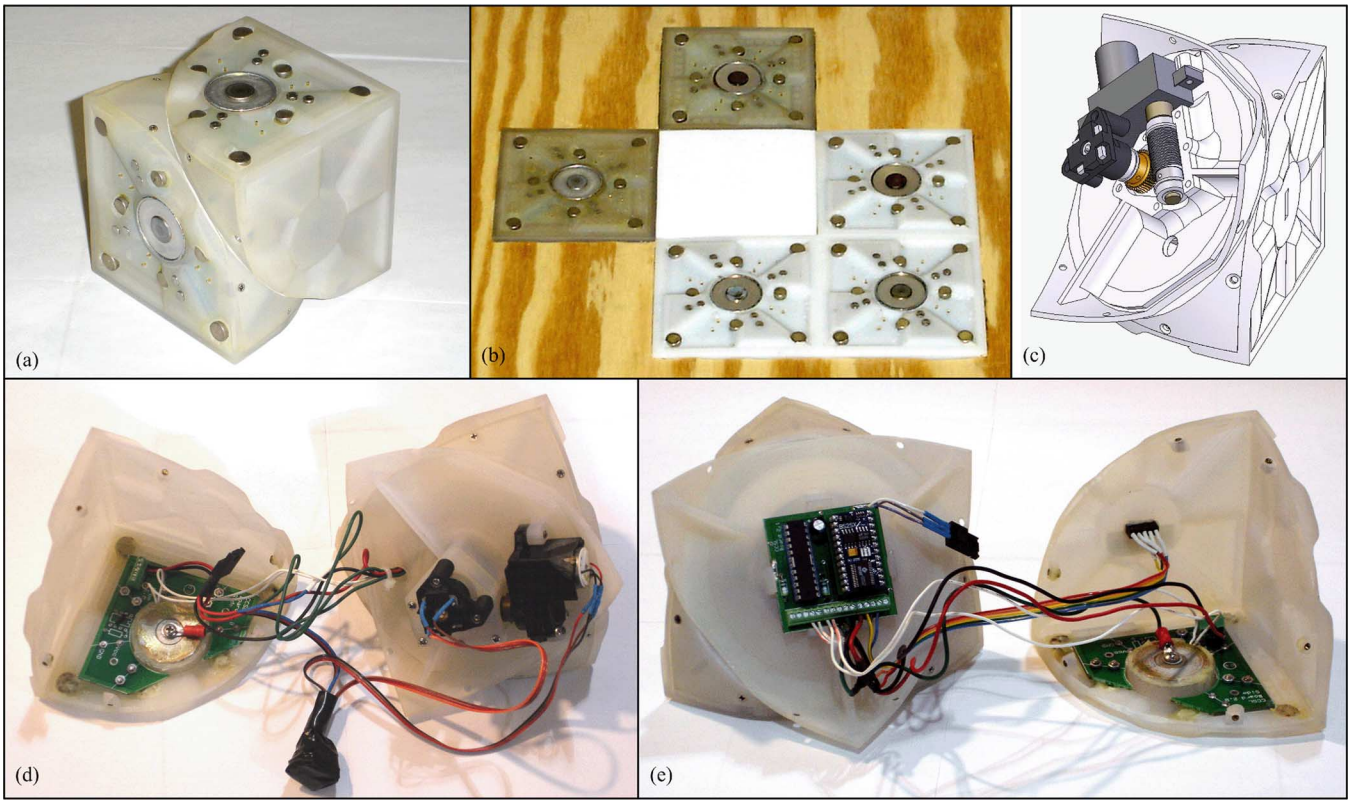


Fig. 4. Physical experimental setup. (a) Assembled Molecube consists of two halves separated along a plane perpendicular to the long diagonal. Each Molecube half is equipped with one connectable interface. (b) The base plates arrangement on the experimental platform. Each base plate has an embossed pattern mechanically compatible with any Molecube face, electrical terminals to facilitate power supply to the robot, and electromagnets for additional retaining force. (c) Dissected view of the worm gearbox half of the Molecube. (d) Molecube servo drive half accommodates the servo, worm gearbox, angle sensor (potentiometer), and voltage regulator for the servo. It also has one Molecube connector that consists of an electromagnet and spring-loaded electrical terminals. (e) Molecube microcontroller half is also equipped with one connector and accommodates the microcontroller and voltage regulator.

blocks by 120° in turn causes the entire structure to reconfigure into a Z-shape and then into an L-shape.

Each surface of a Molecube can be equipped with a connector. Connectors can be used to attach to adjacent neighbors. Geometric patterns embossed on connector surfaces assure that the assemblies are perfectly aligned when joined. Specific bonding and patterning mechanisms depend on the scale of the implementation, and can be, e.g., mechanical, magnetic, electrostatic, or hydrophilic/hydrophobic.

Module connectors can attract or repel each other, or be inert. If electromagnetic bonding is used, then electromagnets can switch between "north," "south," and "off" states. Each cube thus has at most $3^6 = 729$ possible states of electromagnetic activation if all six surfaces are equipped with connectors. Transitioning between states allows modules to pick up, hold, and drop other modules or groups of blocks, as well as grip and climb over other structures.

The machine is controlled using a sequence of swiveling and bond-state switching commands. It is possible to envision more elaborate controllers that incorporate sensing, branching, memory, and stochastic elements, as well as distributed control where cubes and groups of cubes execute programs locally.

The topologies of the structure may have loops and branches, and multiple blocks can swivel simultaneously. While swiveling, a structure goes through intermediate states. Due to collisions, some intermediate configurations may not be physically possible. Similarly, undesired bonding may occur if two attracting cube connectors temporarily become adjacent during reconfiguration. Other physical constraints may be placed on reconfiguration depending on robot environment, e.g., ground collisions avoidance, gravitational stability, actuation torque limits, and motion dynamics. However, as long as the target locations of modules lie at regular lattice locations, actuation sequences can be calculated rapidly and simulated without error accumulation.

B. Design of Physical Robotic Modules

The robot modules are $10\times10\times10$ cm in size and weigh 625 g each. The casing is made of two plastic shells, rapid-prototyped using stereolithography (SLA), each shaped to cover half a cube, as shown in Fig. 4. The module swiveling mechanism is driven by an internal servo motor Fig. 4(c) geared down with a worm gearbox with a ratio of 1:40. This allows rotation speed of 15°/s and torque of 1.41 Nm. Swivel angle precision of 1.7° was achieved through a feedback potentiometer attached to the Molecube axis.

The bonding mechanism must provide reliable connection with other modules while transferring electrical power, data signals, and mechanical torques between the units. In our demonstration, we chose to implement only two out of the six electromechanical connectors per cube, one in each half. Controllable

bonding of modules is implemented with a set of rare earth magnets capable of retaining up to two horizontally cantilevered units without energizing the electromagnet in the center of the interface plate, and three units using additional electromagnetic retaining force. An energizing electromagnet at the center of the plate allows disconnecting or weakening a bond, when facing another energized electromagnet with identical polarity, or temporarily strengthening a bond when facing another energized electromagnet with opposite polarity.

Every connection interface has 16 spring-loaded contacts arranged in two concentric rings, allowing transfer of power with eight-fold redundancy Fig. 4(a). Data is transmitted using the shells of electromagnets as the electrical terminals; power and data signals sharing common ground. Symmetric contact arrangement allows joining the connectors of separate modules at four different relative angular orientations with 90° increments. The cubes do not have autonomous power supplies; they are powered by a 12 V source available at each base plate [Fig. 4(b)]. Modules connected to the base plates propagate the power to all consecutive modules.

Each module's individual Parallax BS II microcontroller is preprogrammed with a complete collection of roles it might assume during the self-replication process. Role nomination during the process of replica construction depends on the stage of the process when the module is picked up. All modules are morphologically and functionally identical.

In this way, information is passed from one structure to the other in the form of role nomination. For example, when the replica construction is accomplished, the final command it receives from the parent structure before it finally detaches from it is to switch from executing the role of the child structure to executing the role of the parent structure.

C. Possible Modes of Self-Replication

There are a number of ways self-replication may occur in a Molecube space. Attempts to classify the types of self-replication have been made earlier [34]. Since any replication process requires an external material supply, we assume some lattice positions may act as *dispensers*, where new cubes reappear when removed from that location. A machine is considered replicated only when the new copy is *identical* to and *autonomous* from its parent.

- *Direct reproduction*: A machine reconfigures to pick cubes from a dispenser and place them in a new location, gradually building a copy from the ground up.
- *Multiparent reproduction*: Multiple machines are required to produce a single copy. One machine may place cubes, while the other reorients the constructed machine.
- *Self-assisted reproduction*: The machine being constructed reconfigures during the construction process to facilitate its own construction.
- *Multistage reproduction*: Intermediate constructions are required before the target machine can be made. The intermediate machine (or scaffold) is then discarded as waste, or can be reused to catalyze the production of additional machines.

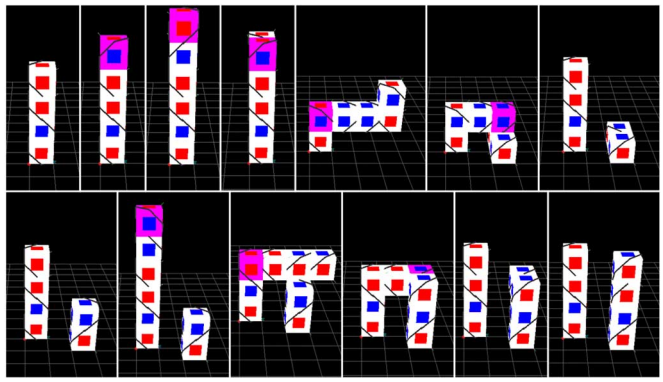


Fig. 5. Hand-designed self-assisted reproduction sequence.

Various combinations and extensions of these modes of operation are possible, and one may imagine an ecology of competing and cooperating machines.

D. Hand-Designed Replication Sequences

We initially explored this space manually, looking for possible self-replicating designs. Exploration of the 3-D space was carried out using a simulator able to simulate arbitrary 3-D Molecube structures and execute sequences of swiveling and bonding commands. The simulator accounts for realistic physical constraints such as collisions during transformation, loops, and locked structures, as well as incompatible bonding polarities and maximum torque loads due to gravity and moment arms.

A number of designs of self-reproducing machines (both structure and control) were found. Fig. 5 shows one of the simplest designs, containing four cubes. New cubes are dispensed from the top. The original machine accepts and manipulates these cubes to build its replica; the newly formed machine reconfigures during the reproduction to assist in its own construction.

Larger and more complex designs were found, as well as patterns for creating arbitrarily sized self-reproducing machines from smaller ones. For example, we have designed an algorithm that allows the construction of self-replicating machines of arbitrary length $2n$, where $(n > 1)$. Fig. 6 illustrates this concept using an example of a 12-module robot.

All such robots are similar in their structure: they all have the lower three blocks arranged in the same manner as shown in Fig. 6 (see inset in the top left corner), and all the rest of the blocks repeat the orientation of a block immediately underneath them. The process of replication starts with receiving two building blocks from the dispenser (which can be in any location reachable by the top of the original robot) and placing them into the position $(2n-1, -1)$ grid locations away from the base of the parent robot, as shown in the top row of Fig. 6. Next, the original tower detaches the two blocks that become the base of the daughter structure, reconfigure to receive two more blocks from the dispenser, attach the top new module to the top of the daughter structure and pass two more modules to it, and continue passing pairs of modules to the daughter structure until it reaches the length of the original robot.

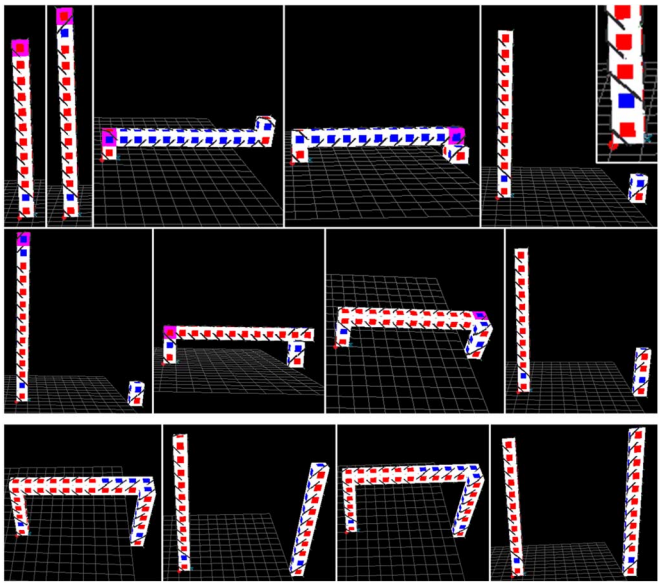


Fig. 6. $2n$-module self-assisted reproduction sequence (hand designed). A set of repetitive actions between the middle and bottom rows are omitted.

E. Experiments in Physical Self-Replication

Several different self-reproduction sequences are possible. We made a total of eight cubes, allowing us to demonstrate self-reproduction of two physical machines, one consisting of three, and the other of four, robotic modules. A brief summary of the project and the results of physical self-replication of a four-module robot were presented in [35]. The top three rows of Fig. 7 show a sequence of key frames from a self-reproduction process that spans about 2.5 min. The entire reproduction process ran continuously without human intervention, except for replenishing building blocks at two "feeding" locations. At the end of the process, the original four-module robot has created a second, identical, four-module robot. The copy is also capable of self-reproduction: the lower three rows of Fig. 7 show a second self-reproduction cycle where the second robot (just built) produces another replica. The four-module robot has four DOFs, according to the number of swiveling axes of its individual Molecubes. Fig. 8 shows three transient configurations of a four-module robot depicting its motion capabilities. Fig. 9 demonstrates two cycles of self-reproduction for a three-module robot.

During both simulated and physical experimentation with reconfiguring 3-D Molecubes along the lattice, we observed the geometric property of the system whereby the sum of all coordinates of any cube can only change with increments measuring an even number of lattice cells. For example, if cube A is manipulated by a cube B located at coordinates (0, 0, 0), then its own coordinates can change (depending on the orientation of the cube B and the original location of cube A) between values $(0,0,1),(0,1,0),(1,0,0),(0,0,-1),(0,-1,0)$, and $(-1,0,0)$. If we add up the axial coordinates of all these locations, we will obtain the following numbers, correspondingly: $1,1,1,-1,-1,-1$. Naturally, as a result of swiveling cube B around any of its possible axes orientations, the distance between cubes A and B, D_{AB} remains unchanged, and

$$\Delta x_1 + \Delta y_1 + \Delta z_1 = \Delta x_2 + \Delta y_2 + \Delta z_2,$$
$$\Delta x_1, \Delta y_1, \Delta z_1, \Delta x_2, \Delta y_2, \Delta z_2 \in Z. \quad (2)$$

Thus, any change in one of the global coordinates is either compensated by an equal and opposite change in some other coordinate, or two coordinates change their values for the same amount simultaneously.

Consequently, if we try to find a reconfiguration resulting in any odd change of a sum of global coordinates, it will require cube B to have fractional coordinates, which contradicts our initial assumption that we only consider reconfigurations along the lattice.

This geometric property of the system separates the lattice Molecube world into two subspaces that could be imagined as black-and-white 3-D checker board. The cubes that were originally placed into the white subspace will remain in the white space forever, and will not be able to cross into the black subspace as long as they reconfigure along the lattice. Conversely, cubes initially located in the black subspace will always remain there.

Such dichotomy naturally leads to an idea that a meta-module composed of two adjacent Molecubes would be a natural formation. For example, without such a meta-module, any Molecube-based manipulator using only one Molecube as an end-effector will not be able to reach half of the lattice space. However, if we use two adjacent Molecubes as an end-effector, one of them can be applied to reach any modules in white subspace, and the other, in black subspace.

Essentially, we have used this meta-module in manually designing self-replication sequences for 2 n-module machines shown in Fig. 6.

III. Comparing Replicability of Physical Systems

It would be interesting to formally apply our information-theoretical self-replicability metric [19] to the physical self-replicating Molecube systems and, for comparison, to a self-replicating system developed earlier by Suthakorn *et al.* [36]. Application of this metric would require careful definition of a closed environment and measurement of spontaneous connection tolerances, from which the likelihood of spontaneous self-assembly could be estimated and serve as the denominator of the self-replicability factor. These calculations require measurements that are currently unavailable.

Instead, we chose to estimate self-replicability of our system in a formal, simulated environment. We investigated self-replicability and spontaneous emergence of self-replicating structures in 2-D Molecube automata [37]. That work demonstrated that various self-replicating entities of different sizes and morphology can emerge in the 2-D Molecube world spontaneously. These results provided numeric estimates of relative self-replicability over time for a series of 2-D Molecube automata. The next step in investigating artificial self-replicability will be to simulate the 3-D Molecube world, accounting for the physical limitations of the real robots.

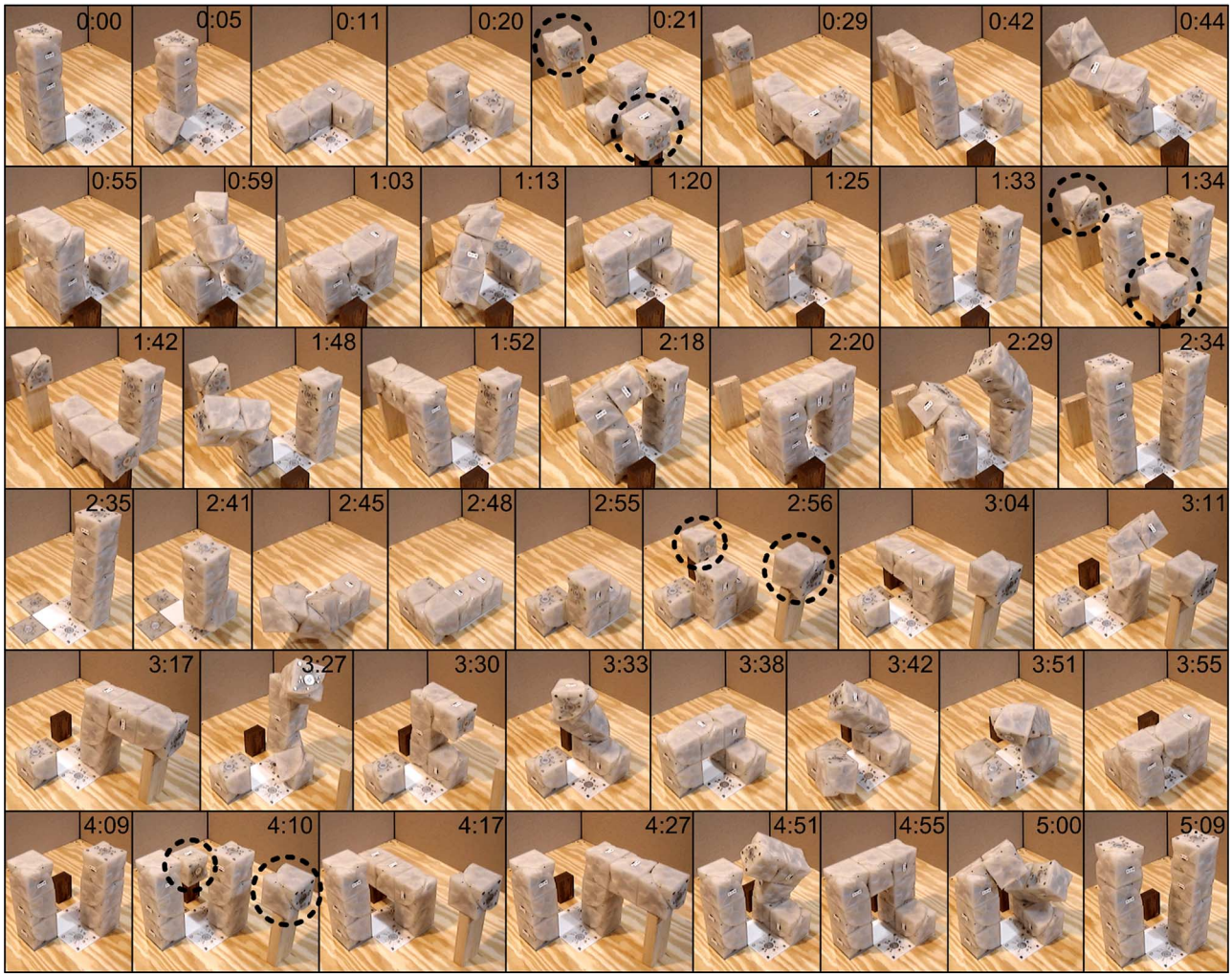


Fig. 7. Self-reproduction of a four-module robot. The top three rows show a sequence of frames from one self-reproduction process spanning about 2.5 min.

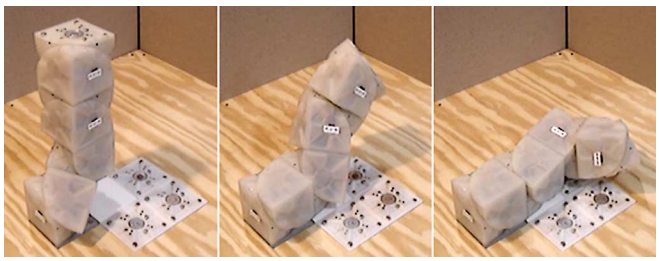


Fig. 8. Transient configurations of the four-module robot photographed at 3, 6, and 9 s after the beginning of the self-reproduction process (from left to right).

Despite the difficulty of precise self-replicability estimation of physical systems, qualitative comparison can still be performed. Both Molecubes and LEGO® robots depend on a reliable supply of modules in specific locations; however, while the LEGO® robot modules can be placed in their positions at any time before the experiment, Molecube robots also require that the building blocks are replenished at the appropriate times during the experiment. This dependency on a supply of modules in the right place at the right time reduces the self-replicability of the Molecube robots, and would be a key focus for improving the system in future designs. On the other hand, the self-replication program is external to the LEGO® robots and is encoded using tracks painted on the ground, while the Molecube robots contain all information necessary to create a replica internally.

Comparison among the three physical systems capable of physical self-replication can also be carried out using the design complexity as the basis. The summary of such a comparison is presented in Table II. This table reflects the facts that the Molecube structures pick up the spare modules from the dispenser locations repeatedly, and that the LEGO® robot only had to manipulate three modules, because the fourth module has been placed at the target location before the beginning of the experiment [36].

In all cases, the robots are preprogrammed (using either the software or the painted lines) to visit all of the locations in a specific sequence. This sequence is synchronized with grasping and releasing the spare modules and ensures the construction of a copy structure. All systems also require assembled subcomponents to be placed at specific locations.

The important difference between the systems is that the Molecube structures are built of identical modules. Their identical interfaces allow rearranging the units into many more permutations than if they had unique interfaces, thereby

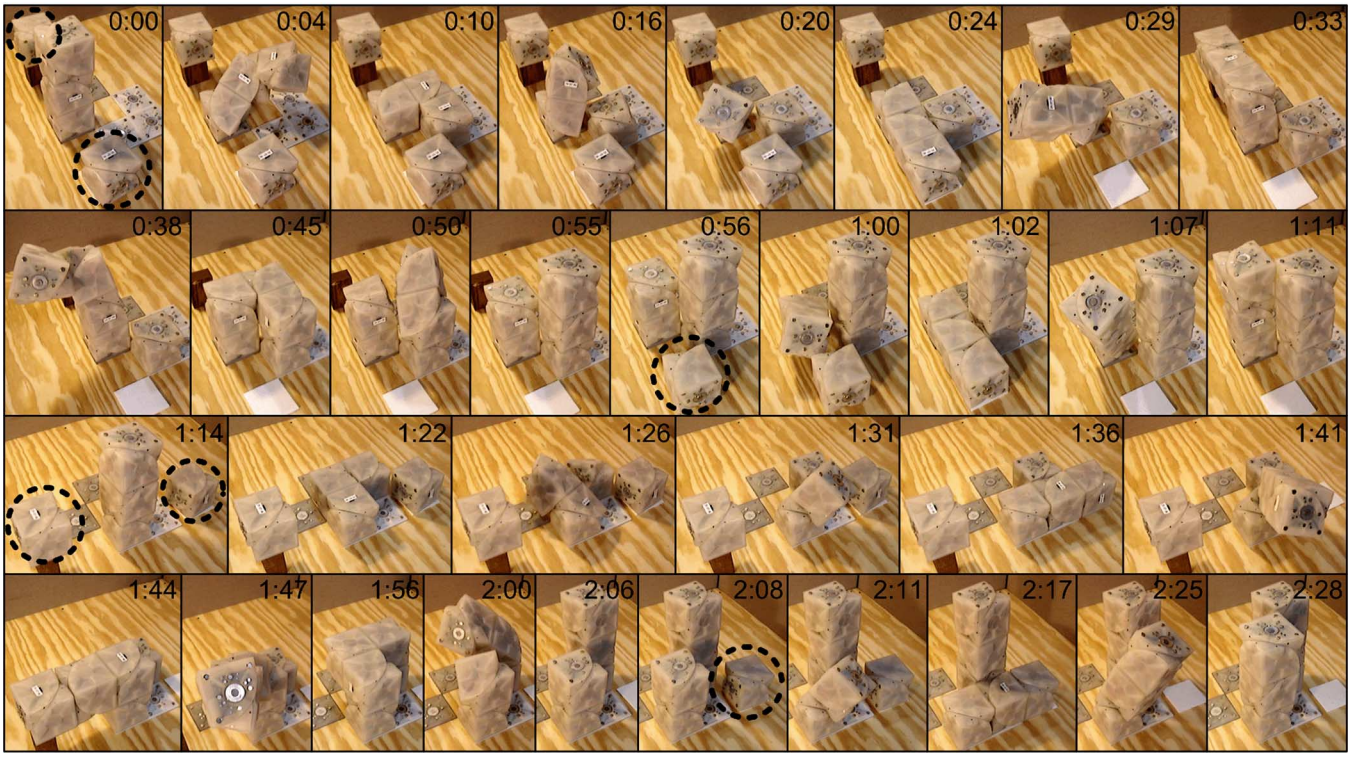


Fig. 9. Self-reproduction of a three-module robot. The top two rows show a sequence of frames from a self-reproduction process spanning about 1 min.

TABLE II
COMPARISON OF PHYSICAL SELF-REPLICATING MACHINES BY TASKS

Machine type	Four-module Molecube	Three-module Molecube	LEGO Mindstorms [31]
Number of locations to visit	6	5	6
- storage/pick-up locations	2	2	3
- target locations	4	3	3
Number of modules to operate	4	3	3
Replication program storage	Internal	Internal	External

spanning a much larger space of possible self-replicating morphologies.

There is no dependency on the order in which the modules are used to construct the four-module machine. The modules assume one of a full collection of their preprogrammed roles during the process of construction or self-replication depending on the stage of the process when they are picked up. Such module universality provides the system with additional reconfigurability and reliability. The concept of "roles" has been used earlier by Støy *et al.* in [38] with application to control of self-reconfigurable robots.

By using the same modules for constructing different self-replicating machines, we demonstrate another advantage of the Molecube substrate. If self-replication of a specific robot is for any reason not possible, the robot can regain this capability by reconfiguring into another self-replicating structure.

IV. EVOLVING SELF-REPLICATING MACHINES

We also attempted to *evolve* self-replicators, rather than design them manually. Successful results in evolving controllers for complex robotic tasks have been reported earlier by Østergaard and Lund in [39], where they presented a system that used competitive co-evolution to develop robot controllers for Khepera robot soccer.

Ideally, replicators would emerge spontaneously out of a primordial soup of cubes [37], where, as in nature, self-replication is an *implicit* reward for itself. However, here we experimented with direct artificial evolution of replicators where the measured amount of replication was used as an *explicit* fitness criterion.

Treating self-replication as a binary property would not provide any gradient for the evolutionary process to follow, and so would be unlikely to yield any viable solution in this vast space of machines which includes both structure and control. Instead, we divided the evolutionary process into two stages, and used the graded definition of self-replication to produce a gradient.

Stage One: Evolve morphologies of machines capable of reaching an area large enough to contain a detached copy of themselves. The percentage of coverage provides a gradient.

Stage Two: Evolve controllers to make a given morphology pick cubes from dispensers and place them at the correct positions. The number of dispensers needed provides a gradient (fewer is better).

The specific algorithms we used are detailed in Fig. 10. During the two stages of evolution, the MAINGA uses two distinct fitness evaluation procedures. The morphology fitness-evaluation procedure is $\mathrm{O}(n^3)$; the control evaluation procedure is $\mathrm{O}(n)$, where n is the number of modules.

Using the algorithms given in Fig. 10, we carried out the initial experiments on the 2-D version of the Molecube, as it is faster to simulate and spans a smaller search space. In two dimensions, each cube is a square; the square is split along its

```
function MAINGA returns array of genomes
    initialize first generation randomly
    for (i = 1 to maxGeneration)
        rank all genomes using one of the fitness functions (below)
        for (j = 1 to eliteCopies)
            copy best solutions to new generation
        while (j <= generationSize)
            perform dual point crossover
            perform mutations
    return lastGeneration

function MORPHOLOGYFITNESS (morphGenome) returns value
    for each state in states      // determine all reachable positions
        for each location adjacent only to end effectors
            reachableLocations.add(location)
    for each state in states      // find the best fit for the copy
        for each reachableLocation
            for each block in morphologyBlocks
                test if reachableLocations.contains(testLoc)
    return (bestFit / morphologySize)

function CONTROLFITNESS (controlGenome) returns value
    for (i=1 to genomeLength)  // execute command sequence
        switch (genome[i++] * 3)
        case < 1:  try swiveling a swivel block, continue if it fails
        case < 2:  attach a new block to an end-effector
        case < 3:  detach a copy block from its end-effector
    count the number of target locations covered goalCount
    count the number of extra copy blocks created extraBlockCount
    return g ( c, m, d, s )       // for details, see (1)
```

Fig. 10. Algorithms for evolving morphology and control of 2-D replicators.

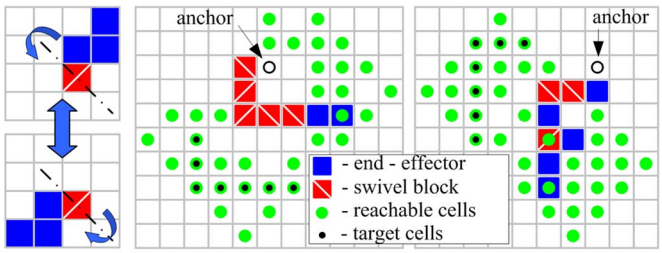


Fig. 11. 2-D self-replicating Molecubes. (a) Two halves split across the diagonal; swiveling the top half causes any adjacent blocks to cycle into new configurations. (b) Two evolved shapes (morphologies) that can cover (reach) an area that contains a detached copy of themselves.

diagonal. A swivel of the square causes two faces to switch, as shown in Fig. 11. Each square thus has two possible swivel states, two possible swivel axes, and $3^4 = 81$ possible bonding states. Since the swiveling motion causes the squares to go out of plane during transition, no intermediate collisions exist. Because of this reduced space size and simpler physics, a 2-D Molecube space is amenable to fast simulation.

In this particular experiment, we also required that each cube be either a swiveling block with permanent magnets or a nonswiveling block with switchable electromagnet (an "end-effector"), but not both. This restriction was placed both for practical consideration for physical implementation, and also to rule out the trivial solution of a single cube sitting at the dispenser location (this was, of course, one of the first "unintended" solutions to be found).

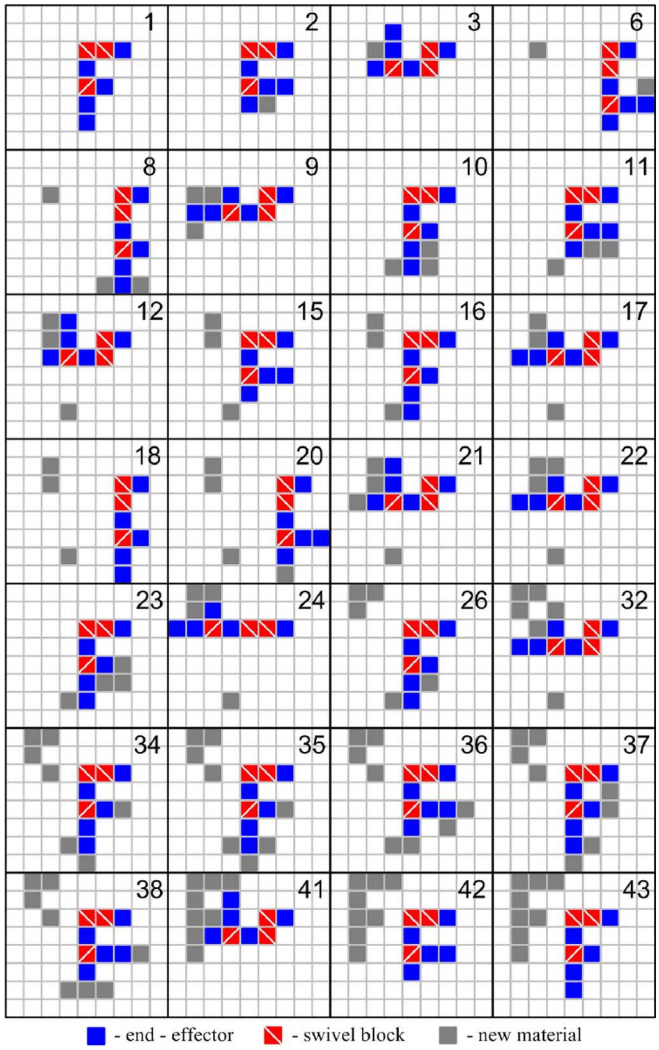


Fig. 12. Evolved replication sequence for the F-morphology. Step number is shown in the upper right corner, redundant steps and cycles are not shown.

The evolutionary process was carried out using a variable-length genome that specified the structure of the machine (a list of unit connections) and, in the second stage, a command sequence. Several morphologies were found, but only few were successful at the second phase, yielding a morphology and matching command sequence that would yield a detached, identical copy. Three of these 2-D machines are shown in Figs. 11(b) and 13(a), and their intermediate morphologies as they execute the command sequence are shown in Figs. 12, 13(b), and 14.

1) Evolutionary Search: The evolution of morphologies and controllers for self-replicating machines was done in the 2-D version of Molecube space, as it is faster to simulate and provides a smaller search space. At the initial morphology-search stage, the fitness function first exhaustively mapped out the area that the end-effectors covered, while pruning illegal configurations due to collisions and self-locking. This step can be done in polynomial time using convolution of reconfiguration steps. Once the coverage of the machine was obtained, then a copy of the machine was tried exhaustively to be fitted within that space in any of the four orientations. This step can be done in polynomial time. The maximum amount of the original structure that

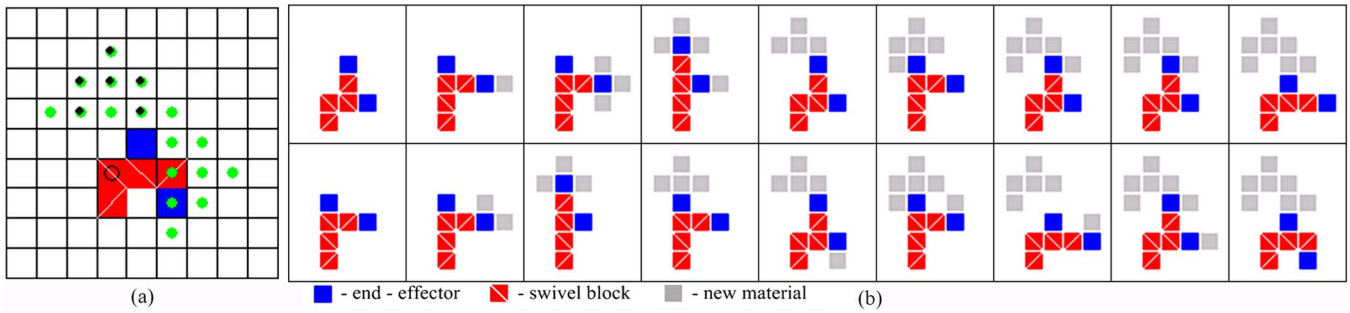


Fig. 13. Progress of 2-D self-replicating structure evolution. (a) The resulting evolved morphology that can reach an area that contains a detached copy of itself. (b) Evolved replication sequences for this resulting morphology.

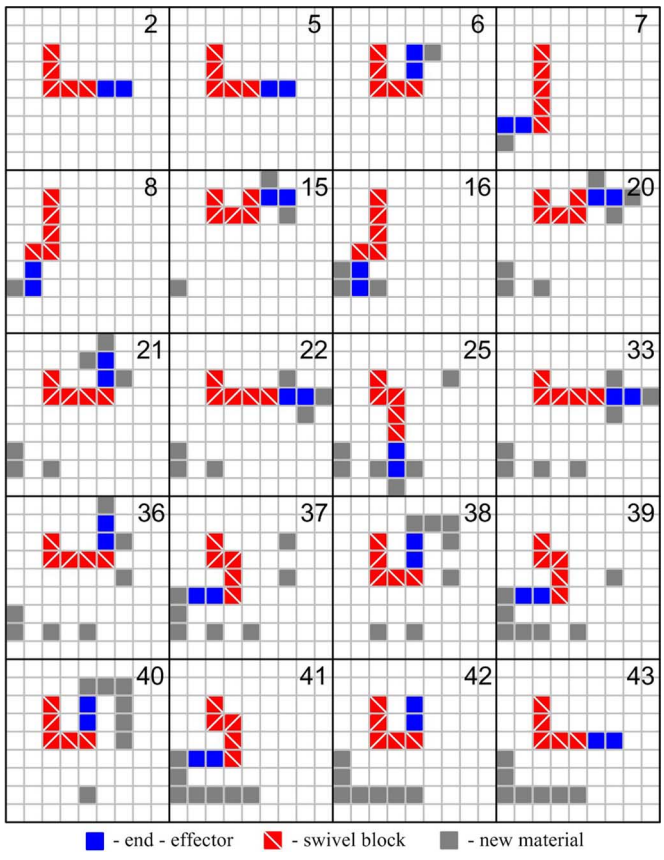


Fig. 14. Evolved replication sequence for the L-morphology. Step number is shown in the upper right corner, redundant steps and cycles are not shown.

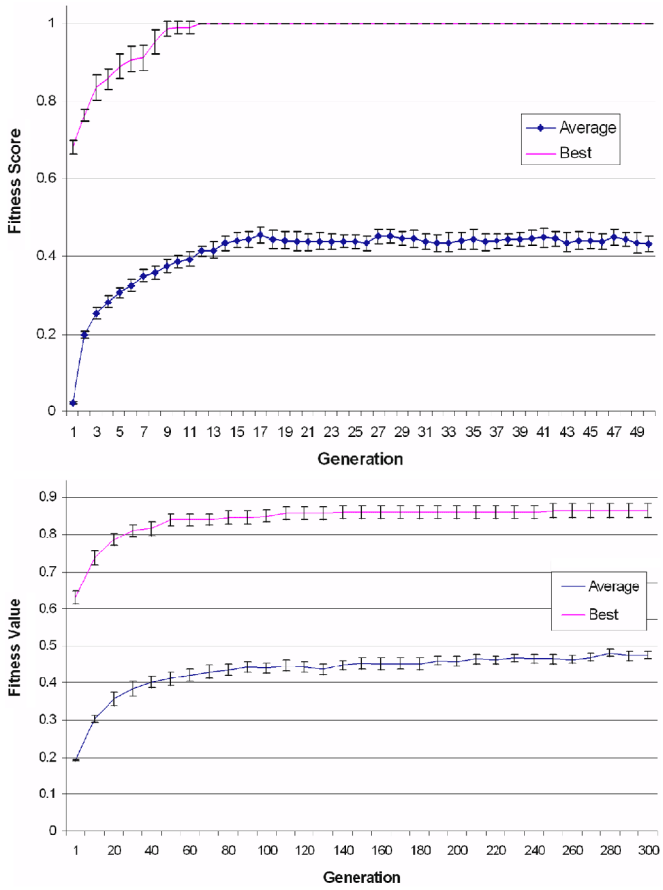


Fig. 15. (top) Evolutionary progress for morphology stage. Fitness versus generations for the run that produced the structure shown in Fig. 13. (bottom) Evolutionary progress for the controller stage for the structure shown in Fig. 13.

would fit in the mapped area provided the fitness for the first stage.

Morphologies were represented as a series of code pairs. Starting with a cursor at the origin, the first code moves the cursor in one of the four cardinal directions, while the second code defines the type of block to try to place. If there is already a block at that position, the new block is ignored and the cursor continues to move as defined by the next element in the array. Morphology strings may be of variable length, but were limited to fewer than 20 units. Variation was achieved through crossover (90%) and three types of mutation: change (0.1%), addition (6%), and removal (0.05%). The population contained 400 individuals and underwent generational fitness-proportionate selection. The top graph on Fig. 15 shows the progress of the morphology evolution resulting in the successful replicator presented in Fig. 13(a).

At the second controller-search stage, the fitness function evaluated a control sequence for the given morphology by executing that sequence and measuring the percentage of the potential duplicate that was covered. Controllers were represented as a series of code triplets, describing a set of commands to be executed in sequence. Each triplet first described a command ("Swivel," "Attach," or "Detach"), and a block number. For the "Attach" command, the third parameter also specified which of the four sides a new block should attach to. The attach

operation also implicitly defined where dispensers are expected to exist, a factor that influences fitness.

Control strings may be of variable size, but were limited to fewer than 300 commands. Variation was achieved through crossover (90%) and three types of mutation: change (0.2%), addition (12%), removal (0.1%). Population contained 1000 individuals and underwent generational fitness-proportionate selection for 300 generations. The bottom graph in Fig. 15 shows the progress at that stage for the structure presented in Fig. 13(a).

The final fitness function used was

$$g(\text{cmds}) = 0.8p + 0.2\frac{c - s + 1}{c} - 0.03e \qquad (3)$$

where $p =$ weighted fraction of the goal location covered; $c =$ number of blocks inside the model; $s =$ number of dispensers assumed; $e =$ number of excess blocks beyond those needed.

Of 20 morphologies found by the evolutionary process in the first stage, only three were successful at the second stage, yielding a functional, physically plausible self-replicating machine. The morphologies and sequences of reconfigurations associated with these two machines are shown in Figs. 11(b), 12, 13, and 14.

V. Conclusion

Our purpose in this investigation is to identify a rich substrate in which many self-reproducing machines can be found and physically realized in a systematic fashion. The results presented here demonstrate a robotic substrate composed of simple modular units, in which both simple and complex machines can construct and be constructed by identical machines in the same substrate. A number of designs, both hand-crafted and evolved, and both virtual and physical, were shown. Although our machines are relatively simple, it is easy to foresee that with more modules and more connectors per module available, larger and more complex self-reproducing machines could be constructed. However, such larger machines are likely to introduce new power, force, and additional physical constraints that would need to be addressed.

Looking at earlier work in self-replication [4], [12], [17], [18], [36], one may have concluded that machine self-reproduction involves sophisticated mechanisms and contraptions, and that this complexity will increase as we seek to self-reproduce machines with more parts. Our results seem to suggest the contrary. Increasing self-reproduction will involve machines with more units that are each simpler and more similar to each other. This ratio, of the number of lowest-level (“atomic”’) units to the number of unit types, is remarkably high for biological organisms that reproduce using roughly 10^{20} amino acids from a repertoire of only 20 types [40]. The more units involved, and the simpler and more similar the units are, the more information is being reproduced by the system itself, as compared with information pre-existing in the parts and environment. We have, therefore, tried to make our modular units all identical and simple, as compared with previous machines. We conjecture that future self-reproducing machines will be increasingly modular and simple, and self-reconfiguring robotics is the likely substrate for this type of machinery to develop.

There are several future challenges. First, the modules demonstrated here were few and large, making scaling to large numbers difficult and costly. Microscale designs based on other bonding methods and actuation mechanisms may prove to be a more economical and versatile alternative to continue this investigation, potentially allowing fine-grained reproduction of machines with thousands of parts covering a much broader space of morphologies. A second challenge is computational, addressing the planning of algorithms involved in carrying out robust self-replication.

Self-reproduction is one of the remarkable feats of biological systems which has remained largely outside the scope of capabilities of traditional engineered systems. As we strive for more robust robotic systems for applications where human maintenance is prohibitive (Fig. 1), self-repair and self-replication may be a viable design paradigm.

References

[1] R. A. Freitas and R. C. Merkle, *Kinematic Self-Replicating Machines*. Georgetown, TX: Landes Biosci, 2004.

[2] M. Sipper, “Fifty years of research on self-replication: An overview,” *Artif. Life*, vol. 4, no. 3, pp. 237–257, 1998.

[3] M. Sipper and J. A. Reggia, “Go forth and replicate,” *Sci. Amer.*, vol. 285/265, no. 2, pp. 35–43, 2001.

[4] H. Lipson and J. B. Pollack, “Automatic design and manufacture of artificial lifeforms,” *Nature*, vol. 406, pp. 974–978, 2000.

[5] R. J. Jackman, S. T. Brittain, A. Adams, M. G. Prentiss, and G. M. Whitesides, “Design and fabrication of topologically complex, three-dimensional microstructures,” *Science*, vol. 280, pp. 2089–2091, 1998.

[6] E. Winfree, F. Liu, L. A. Wenzler, and N. C. Seeman, “Design and self-assembly of two-dimensional DNA crystals,” *Nature*, vol. 394, pp. 539–544, 1998.

[7] J. Breivik, “Self-organization of template-replicating polymers and the spontaneous rise of genetic information,” *Entropy*, vol. 3, no. 4, pp. 273–279, 2001.

[8] S. Griffith, D. Goldwater, and J. M. Jacobson, “Self-replication from random parts,” *Nature*, vol. 437, no. 7059, p. 636, 2005.

[9] S. Murata, E. Yoshida, H. Kurokawa, K. Tomita, and S. Kokaji, “Self-repairing mechanical systems,” *Auton. Robots*, vol. 10, no. 1, pp. 7–21, 2001.

[10] R. Fitch, D. Rus, and M. Vona, “A basis for self-repair robots using self-reconguring crystal modules,” in *Proc. Intell. Auton. Syst. 6*, 2000, pp. 903–910.

[11] S. Colombano, “ROBOSPHERE is a NASA project to explore the possibility of long term or continuous robotic presence on planetary surfaces and in space,” 2002 [Online]. Available: http://robosphere.arc.nasa.gov/

[12] M. Moses, “A physical prototype of a self-replicating universal constructor,” M.S. thesis, Dept. Mech. Eng., Univ. New Mexico, Albuquerque, 2001.

[13] R. A. Freitas and W. Zachary, “A self-replicating, growing lunar factory,” in *Proc. 5th Princeton/AIAA Conf. Space Manuf.*, May 1981, pp. 109–119.

[14] R. A. Freitas, “Manufacturing systems for molecular nanotechnology: Kinematic self-replicating machines,” 2004, in preparation.

[15] Z. Butler, S. Murata, and D. Rus, “Distributed replication algorithms for self-reconfiguring modular robots,” *Distrib. Auton. Robot. Syst. 5*, pp. 37–48, 2002.

[16] L. S. Penrose, “Self-reproducing machines,” *Sci. Amer.*, vol. 200, no. 6, pp. 105–114, 1959.

[17] H. Jacobson, “On models of reproduction,” *Amer. Sci.*, vol. 46, pp. 255–284, 1958.

[18] G. S. Chirikjian, Y. Zhou, and J. Suthakorn, “Self-replicating robots for lunar development,” *IEEE/ASME Trans. Mechatron.*, vol. 7, no. 4, pp. 462–472, Jul. 2002.

[19] B. Adams and H. Lipson, “A universal framework for self-replication,” in *Proc. Eur. Conf. Artif. Life*, Dortmund, Germany, Sep. 2003, pp. 1–9.

[20] J. Von Neumann, "Von Neumann's self-reproducing automata," in *Essays on Cellular Automata*, A. W. Burks, Ed. Urbana, IL: Univ. Illinois Press, 1970, pp. 4–65.
[21] B. McMullin, "John von Neumann and the evolutionary growth of complexity: Looking backward, looking forward," *Artif. Life 6*, pp. 347–361, 2000.
[22] K. Støy and R. Nagpal, "Self-reconfiguration using directed growth," in *Proc. 7th Int. Symp. Distrib. Auton. Robot. Syst.*, Toulouse, France, Jun. 23–25, 2004, pp. 1–10.
[23] G. Chirikjian, "Kinematics of a metamorphic system," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1994, pp. 449–455.
[24] M. Yim, Y. Zhang, and D. Duff, "Modular robots," *IEEE Spectrum Mag.*, vol. 39, no. 2, pp. 30–34, Feb. 2002.
[25] S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, and S. Kokaji, "M-TRAN: Self-reconfigurable modular robotic system," *IEEE/ASME Trans. Mechatron.*, vol. 7, no. 4, pp. 431–441, Jul. 2002.
[26] D. Rus, A. Butler, K. Kotay, and M. Vona, "Self-reconfiguring robots," *Commun. ACM*, vol. 45, no. 3, pp. 39–45, 2002.
[27] W.-M. Shen and M. Yim, "Self-reconfigurable robots," *IEEE Trans. Mechatron.*, vol. 7, no. 4, pp. 401–402, Jul. 2002.
[28] A. Pamecha, I. Ebert-Uphoff, and G. S. Chirikjian, "Useful metrics for modular robot motion planning," *IEEE Trans. Robot. Autom.*, vol. 13, no. 4, pp. 531–545, Aug. 1997.
[29] M. Yim, K. Roufas, D. Duff, Y. Zhang, C. Eldershaw, and S. Homans, "Modular reconfigurable robots in space applications," in *Autonomous Robot Journal, Special Issue for Robots in Space*. New York: Springer Verlag, 2003.
[30] M. W. Jorgensen, E. H. Ostergaard, and H. H. Lund, "Modular ATRON: Modules for a self-reconfigurable robot," in *Proc. IEEE/RSJ Int. Conf. Robots d Syst.*, Sendai, Japan, Sep. 30–Oct. 2, 2004, pp. 2068–2073.
[31] A. Castano, W.-M. Shen, and P. Will, "CONRO: Towards deployable robots with inter-robots metamorphic capabilities," *Auton. Robots*, vol. 8, no. 3, pp. 309–324, Jun. 2000.
[32] B. Salemi, M. Moll, and W.-M. Shen, "SUPERBOT: A deployable, multi-functional, and modular self-reconfigurable robotic system," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Beijing, China, Oct. 2007.
[33] E. Mytilinaios, D. Marcus, M. Desnoyer, and H. Lipson, "Designed and evolved blueprints for physical self-replicating machines," in *Proc. 9th Int. Conf. Artif. Life*, 2004, pp. 15–20.
[34] G. S. Chirikjian and J. Suthakorn, "Toward self-replicating robots," in *Proc. 8th Int. Symp. Exp. Robot.*, Italy, 2002, pp. 392–401.
[35] V. Zykov, E. Mytilinaios, B. Adams, and H. Lipson, "Self-reproducing machines," *Nature*, vol. 435, no. 7038, pp. 163–164, May 11, 2005.
[36] J. Suthakorn, A. B. Cushing, and G. S. Chirikjian, "An autonomous self-replicating robotic system," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatron.*, 2003, pp. 137–142.
[37] G. Studer and H. Lipson, "Spontaneous emergence of self-replicating structures in Molecube automata," in *Proc. 10th Int. Conf. Artif. Life*, 2006, pp. 227–233.
[38] K. Støy, W.-M. Shen, and P. Will, "Using role based control to produce locomotion in chain-type self-reconfigurable robots," *IEEE Trans. Mechatron.*, vol. 7, no. 4, pp. 410–417, 2002.
[39] E. H. Østergaard and H. H. Lund, "Co-evolving complex robot behavior," in *Proc. 5th Int. Conf. Evolv. Syst.: From Biol. to Hardware*, Norway, Mar. 17–20, 2003, pp. 308–319.
[40] R. Duncan and E. H. McConkey, "How many proteins are there in a typical mammalian cell," *Clin. Chem.*, vol. 4, pt. 2, pp. 749–55, Apr. 28, 1982, Approximately 10^3 amino acids per protein, 10^3 polypeptides per mammalian cell, and 10^{14} cells in a human.



Victor Zykov (S'06) received the B.S. and M.Eng. degrees in electromechanical engineering from Ivanovo State Power Engineering University, Ivanovo, Russia, in 2000 and 2002, respectively. He is currently working toward the Ph.D. degree in mechanical engineering at Cornell University, Ithaca, NY.

While with Ivanovo State Power Engineering University, he was working on controller synthesis and optimization for multiphase induction motors. His current research interests belong to the area of damage-resilient robotic systems, and include autonomous methods for complete robot restoration, custom three-dimensional robot part restoration, unanticipated damage identification, and autonomous physical and functional recuperation from partial damage.



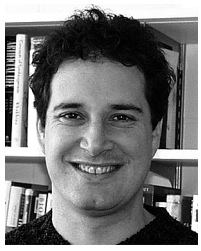
Efstathios Mytilinaios received the B.S. degree in mathematics and computer science from Brandeis University, Waltham, MA, in 2003, and the M. Eng. degree in computer science from Cornell University, Ithaca, NY, in 2004.

While with Brandeis, he was involved in construction of 3-D Genobots at the DEMO Lab and took part in the Research Experiences in Algebraic Combinatorics at Harvard (R.E.A.C.H.) program. After graduation from Cornell, he was with Microsoft Corporation, Redmond, WA, until 2006. Currently, he is with Ingenio Inc., San Francisco, CA. His current research interests include algebraic combinatorics, evolutionary robotics, machine learning, robotic self-replication, self-repair, and reconfiguration.



Mark P. Desnoyer is currently working towards the B.Eng. degree in electrical and computer engineering at Cornell University, Ithaca, NY.

He has worked summers as a Software Developer in the broadcast and robotic industries. Currently, he is with the Cornell University Astronomy Department, developing an automated image calibration system for NASA's Deep Impact mission. His research interests include automation, control systems, and physical self-replication.



Hod Lipson (M'98) received the B.Sc. and Ph.D. degrees in mechanical engineering from the Technion Israel Institute of Technology, Haifa, Israel, in 1989 and 1998, respectively.

Since 2001, he has been an Assistant Professor with the Mechanical and Aerospace Engineering and Computing and Information Science Schools, Cornell University, Ithaca, NY. Prior to this appointment, he was a Postdoctoral Researcher with Brandeis University's Computer Science Department, and a Lecturer with MIT's Mechanical Engineering Department, where he conducted research in design automation. He is interested biologically inspired approaches to robotics, as they bring new ideas to engineering and new engineering insights into biology.