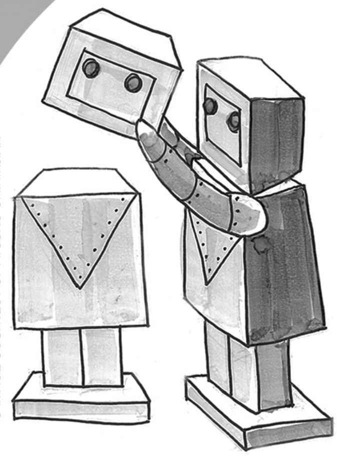
Robotic Self-Replication

A Descriptive Framework and a Physical Demonstration from Low-Complexity Parts

BY KIJU LEE AND GREGORY S. CHIRIKJIAN



Toward Robotic System Autonomy

We introduce a descriptive framework for robotic replicating systems that extends von Neumann's classical model of self-reproducing automata. A new physical prototype is developed to examine principles and uncover hardware limitations of kinematic reproduction at a low-complexity level in man-made physical systems. The initial functional robot assembles six subsystems located in a partially structured environment to form an exact functional replica of itself. Self-replication of the sort demonstrated here can be achieved by designing a robotic system composed of multiple simple parts/subsystems rather than a relatively few complex robotic modules. Therefore, this prototype hints at the feasibility of the concept of fully autonomous man-made machines that can construct functional copies of themselves from many very basic components. As a simple measure of system complexity, the number of active elements in each subsystem and interconnections between subsystems are counted. In addition, we present a measure of the degree of replication that includes the complexity distribution and the relative complexity of the total system to individual parts.

Background and Inspiration

Reproduction is considered to be one of the distinctive features of living organisms. It occurs at various length and time scales such as cell division, DNA replication with associated enzymes and proteins, and virus replication within a host cell. If the replication process is controlled and manipulated solely by the system to be reproduced, it is called a self-replicating system (for example, cell division). It is important to note that self-replication always takes place in a certain environment that can be filled with resources such as energy and parts and may also provide passive substrates, catalysts, or tools that the replicating system may use and then return to the environment when the replication is completed. In contrast, if the replication process requires external elements that actively control and manipulate the resources, then it is simply a replicating system (for example, a virus requires the transcriptional and translational machinery of a host cell to reproduce its genetic material and proteins). If all members of a species are deemed to be in an equivalence class, then cloning and sexual reproduction can roughly be viewed as macroscale replication processes that produce equivalent replicas as well.

Digital Object Identifier 10.1109/M-RA.2007.908962

An Overview of Related Works

Designing and building a robotic system (or a machine) capable of replication or self-replication has been a dream of many researchers in the robotics field (as well as the science fiction literature) since the first theoretical work on machine self-reproduction was introduced by John von Neumann more than 50 years ago. In a lecture in 1948, von Neumann suggested a model of an automaton that is sufficiently complex to reproduce itself [1], [2]. According to this model, a self-reproducing automaton consists of four components:

- A, an automatic factory (constructor) that collects raw materials and processes them into outputs using a specified written instruction.
- B, a copier that takes an instruction and copies it.
- C, a controller that controls A and B, actuating them alternatively.
- D, a list of written instructions.

Therefore, in von Neumann's model, the automaton $(A+B+C+D)$ produces another $(A+B+C+D)$, such that

$$[(A+B+C+D)] \longrightarrow [(A+B+C+D), (A+B+C+D)]. \quad (1)$$

Exactly what is meant by raw materials and written instructions is not quantified in that theory, nor is the role of passive elements in the environment that may be used and returned to the environment after each cycle of the process. In fact, a

1070-9932/07/$25.00©2007 IEEE

variety of issues related to physical self-replication are not addressed, including the role of energy and the production of waste. According to [3], the von Neumann's model can be applied to cell reproduction, with A represented by ribosomes, B represented by the enzymes RNA and DNA polymerase, C represented by repressor and derepressor molecules and associated expression-control machinery in the cell, and D represented by DNA.

The concept of kinematic self-reproduction has been applied in many research areas such as cellular automata, nanotechnology, macromolecular chemistry, and computer simulations [4]. In the 1950s and 1960s, Penrose presented the first implementation of a passive self-replicating machine [5], [6]. He showed that simple units or bricks having certain properties could build a self-reproducing machine under external agitation. This work opened the possibility of physical application of von Neumann's ideas on machine self-reproduction even though it could not take the further step toward more complicated machines such as robotic systems that actively and autonomously reproduce.

In the 1980s, NASA became interested in self-replicating robots as a potential means for space development and exploration [7]. This has been revived recently with the long-term goal of self-replicating factories on the moon [8], [9]. More recent research includes self-assembly and self-reconfiguration of modular robots and self-repairable robots. Algorithms for self-assembly using modular robots [10] and self-reconfigurable modular robots [11]–[13] are some of the works representing this area. Self-replicating modular cubes capable of self-assembly and reconfiguration with human intervention during the process are presented in [14]. In [15]–[17], algorithms and locomotion for self-reconfigurable modules were described and demonstrated in simulations.

Recently, our lab has designed several prototypes to develop and demonstrate the concept of robotic self-replication. The first generations, including [18]–[21], progressed from prefabricated subsystems remotely controlled by a human to more advanced demonstrations of semiautonomous self-replication. The second-generation robots are fully autonomous and microprocessor-based self-replicating robots [22], [23]. These robots were able to replicate automatically without human intervention during the process. In every system, a properly designed environment holds important information about subsystem locations and catalyzes the self-replication process. In addition, a self-replicating, electromechanical circuit was presented in [24]. The system presented in this article is one of our third-generation models that are fully autonomous without computer control. The role of the microprocessor in previous prototypes is now replaced by a number of simple circuits that are distributed into subsystems keeping the same level of robot function, but with much lower component complexity.

Principles of Robotic Replication

In general, no system can self-replicate without certain environmental conditions and resources. Therefore, von Neumann's model in (1) may be incomplete when applied to physical self-replicating systems. Therefore, we start by revising von Neumann's model by adding external resources (α) and leftover (or waste) materials after self-replication (β), such that

$$[(\mathrm{A}+\mathrm{B}+\mathrm{C}+\mathrm{D}),\alpha]\longrightarrow[(\mathrm{A}+\mathrm{B}+\mathrm{C}+\mathrm{D}), (\mathrm{A}+\mathrm{B}+\mathrm{C}+\mathrm{D}),\beta]. \quad (2)$$

Note that this modification still says nothing about the role of the environment, and the descriptions of α and β are very coarse relative to the descriptions of A–D.

We introduce a new descriptive framework for reproducing systems that applies to both the self-replicating and replicating cases. Recall that self-replicating systems are a subset of replicating systems and satisfy certain properties associated with the word self.

On the basis of (2) and broadening von Neumann's early concept on self-replicating automata, we introduce a general model of physical replication. A simple architecture would contain three sets of components:

- M, a multiset of available parts to be used for building replicas by an initial system (In mathematics, a multiset differs from a set in that each element can have multiplicity. For example, $\{1, 1, 1, 2, 2, 3\}$ is a multiset).
- R, a multiset of an initial functional system to be reproduced. $\mathrm{R}\neq\emptyset$ to be a replicating system.
- E, a multiset of environmental structures and/or elements involved in the replication process (but not replicated).

On the basis of the listed components, a replication process can be written as

$$(\mathrm{R},\mathrm{M},\mathrm{E})\longrightarrow(\mathrm{R}',\mathrm{M}',\mathrm{E}'), \quad (3)$$

where $|R|<|R'|$ and $|\mathrm{R}|$ indicates the number of initial systems. Replication processes are highly organized and timed events. If the replication process of the initial system is denoted by a time-dependent function Φ, then robotic replication can be written as $(\mathrm{R}',\mathrm{M}',\mathrm{E}')=\Phi(\mathrm{R},\mathrm{M},\mathrm{E},t)$, where t is the time required for replication. Note that here M takes the place of α in (2), and the environment is explicitly included. If there are waste products (β), they are viewed as a part of the modified environment, E'.

In (1), $(\mathrm{A}+\mathrm{B}+\mathrm{C}+\mathrm{D})$ can be seen as R in our model, and the environment in which the system functions (including catalysts and passive tools used by R) can be viewed as E, even though this part is ignored in von Neumann's model. If R itself requires an external constructor, i.e., all or part of the assembly machinery is external to R, then R is only a replicating system. Depending on whether R actively controls the replication (or self-replication) process or not, it is categorized as either an active or a passive system. In our view, replicating or self-replicating robotic systems must be active. Table 1 shows some examples for each category. Some of those examples are explained in detail in the following section.

To be a replicating system, R should not be the empty set. If $\mathrm{R}=\emptyset$ and $\mathrm{R}'\neq\emptyset$, then Φ here represents a self-assembly (or assembly) function rather than a replication process. Self-assembly is an interesting topic on its own, but is not addressed here.

Passive Replication

We define a passive replication process as one in which an object or system is reproduced by an external constructor while the object/system itself does not have active functionality during the replication process. As extreme examples, a document being replicated by a photocopier, or a three-dimensional (3-D) part being reproduced using a combination of laser scanner with a rapid prototyping machine, can be viewed as passive replication processes. We can also find such systems in nature. For example, DNA replication is performed by the action of DNA polymerase enzymes and associated proteins adding a deoxyribonucleotide triphosphate to the $3'$-hydroxyl group of a growing complementary DNA strand [25]. During this process, DNA plays the role of written instructions, D, but not as the constructor, A.

Active Replication

We call an actuated system able to replicate itself with the aid of an external constructor an active replicating system (ARS). For example, in a science-fiction-like scenario in which a team of robots hijacks a car factory and reprograms the manufacturing lines to produce copies of the robots rather than cars, in our view, these robots would constitute an ARS. In this case, the robots have enough functionality to control and manipulate the factory to produce what they need, but it is the factory that performs part of the constructor role. Viruses are another good example of ARSs [26]. A single virus can induce the replication of hundreds of copies by injecting its genetic material into a host cell. Within our framework, one can define R={virus}, M={nutrients available in a host cell}, and E = {a host cell including all transcriptional and translational machinery}. The whole machinery required for virus replication is provided by a host cell following the instructions of viral generic material, which is injected into the cell by the action of the virus.

Among existing robotic systems, the modular cubes (Lipson's group [14]) demonstrate a robotic replication process with some external machinery. The initial functional system composed of four identical modules is represented by $(\mathrm{N}+\mathrm{N}+\mathrm{N}+\mathrm{N})$. We define $R=\{(\mathrm{N}+\mathrm{N}+\mathrm{N}+\mathrm{N})\}$, $M=\{N,N,N,N\}$, and E = {feeding locations with human intervention}. The replication for this system is given by $\Phi(\mathrm{R},\mathrm{M},\mathrm{E},t)=(\mathrm{R}',\mathrm{M}',\mathrm{E})$, where $\mathrm{R}'=\{(\mathrm{N}+\mathrm{N}+\mathrm{N}+\mathrm{N}),(\mathrm{N}+\mathrm{N}+\mathrm{N}+\mathrm{N})\}$, $\mathrm{M}'=\emptyset$, $\mathrm{E}'=\mathrm{E}$, and $t\simeq 5$ min. We view this system as a replicating system, not self-replicating, because the resources (single modules) are fed by human intervention during the process, which uses an active external agent (i.e., the human).

Table 1. Categorization of replicating systems and some examples.

	Passive	Active
Replication	DNA + associated proteins [25]; photo + photocopier; 3-D object + rapid prototyping machine	Virus + host cell [26]; molecubes [14]; RCX robot [21]; robot hijacking a car factory
Self-replication	Penrose block replicators in [5], [6]; RNA world [25]	Cell division [25]; fully autonomous RCX robots [22], [23]

Passive Self-Replication

If a system itself does not have active functionality, but there exists a replication process associated with that system without any external constructor, we consider such a process as a passive self-replicating process. Random sources that may exist in nature, such as random noise and Brownian motion are not considered as an external constructor. RNA is "uniquely able to both serve as a template for and to catalyze its own replication" [25]. Although RNA holds genetic information, it is not an active system, i.e., it can be seen as written instructions containing a copier.

Penrose block replicators can be viewed as passive self-replicating systems (PSRS). The first example of the Penrose system consists of two simple blocks, B_1 and B_2, with certain mechanical properties [5]. When a system $(\mathrm{B}_1+\mathrm{B}_2)$ [or $(\mathrm{B}_2+\mathrm{B}_1)$] is provided in a box containing unassembled parts of B_1 and B_2, $(\mathrm{B}_1+\mathrm{B}_2)$ reproduce more copies of itself from individual parts with nothing but random agitation to the container box. This system can be simply represented by the replicating system model, with $\mathrm{R}=\{(\mathrm{B}_1+\mathrm{B}_2)\}$, $\mathrm{M}=\{\mathrm{B}_1,\mathrm{B}_1,\mathrm{B}_1,\mathrm{B}_2,\mathrm{B}_2,\mathrm{B}_2,\mathrm{B}_2\}$, and E = {a container box with random agitation}. By a self-replication function, Φ, $\mathrm{R}'=\{(\mathrm{B}_1+\mathrm{B}_2),(\mathrm{B}_1+\mathrm{B}_2),(\mathrm{B}_1+\mathrm{B}_2)\}$, $M'=\{\mathrm{B}_1,\mathrm{B}_2,\mathrm{B}_2\}$ and $\mathrm{E}'=\mathrm{E}$. In [6], Penrose presented a simple mechanical system consisting of two blocks with more complicated structure than B_1 or B_2, but the self-replication process is similarly driven with external agitation.

Active Self-Replication

An active self replicating system (ASRS) has sufficient functionality to replicate itself, possibly with the aid of some passive environmental structures. For these systems, some written instructions can be considered to be a part of the environment for reuse by both the original and replica robots, and so strictly speaking, von Neumann's architecture need not be followed. Cell division is an example of active self-replication in which the environment does not include instructions. Here Φ represents the self-replication function, with R = {cell}, M = {nutrients provided into a cell}, and E = {environmental structures around a cell}. After self-replication, R′ = {cell, cell}, M′ = M − {nutrients used for cell division}, and E′ = E + {some waste materials produced during the process}.

Prototypes presented in [22] and [23] can be considered as ASRSs. Both prototypes work in a highly structured environment, i.e., the robots follow instructions embedded in properly structured environments. Although those structures hold important information about parts (elements of M) locations and robot trajectory, they do not actively control or manipulate parts during the process. All necessary machinery is made solely by an initial robotic system itself. A new prototype to be presented in the next section is also viewed as an ASRS.

Self-Replicating Robotic System

In our prototype, the initial functional robot consists of six man-made subsystems that are relatively simple electromechanical parts. Another set of subsystems are provided and placed in a structured environment to be used to build a replica (Figure 1). Following our definitions, the system can be represented by $\mathrm{R} = \{(M_1 + \cdots + M_6)\}$, $\mathrm{M} = \{M_1, \ldots, M_6\}$, and $\mathrm{E} = \{\text{tracks, bar codes, contact codes, wall, metal line}\}$. By self-replication function, $\Phi(\mathrm{R}, \mathrm{M}, \mathrm{E}, \mathrm{t} \simeq 13 \text{ min})$, the system becomes $R' = \{(M_1 + \cdots + M_6), (M_1 + \cdots + M_6)\}$, $M' = \emptyset$, and $\mathrm{E}' = \mathrm{E}$. There is no external machinery other than R itself in this procedure.

Experimental Overview

The robot has six states (Figure 2) and three distinctive behaviors for each state: forward line-tracking (line-tracking mode), reversing (reversing mode), and turning left while the timer is on (left-turning mode). The behavior is decided by outputs from four different kinds of sensors: bar code readers reading bar codes; contact sensors detecting contact codes; a metal detector detecting the metal line at the station (where the replica is made); and a touch sensor detecting the wall. If the robot initially set to State 1, it starts moving along the outer track in line-tracking mode until it detects Bar code 1. When the robot finds Bar code 1, the system triggers the timer ON and the robot turns left for five seconds before going back to line-tracking mode. Left turning results the robot to find a cross way leading to the inner track.

Each subsystem is placed on one of the cross ways (connecting the outer track and the inner track, Figure 1) so that the robot can drag it to the inner track. A contact code is attached on each cross way, and it triggers the state machine to the next state. When the robot brings a collected subsystem to the station, the metal foil line on the surface triggers the robot from the line-tracking mode to the reversing mode. The robot reverses until the touch sensor hits the backside wall and it returns to the outer track in line-tracking mode. Since the robot now sets in State 2, it turns left to enter the cross way when it finds Bar code 2. The robot repeats this process until the replica is fully assembled. A time lapse sequence of the whole self-replication process is shown in Figure 3.

The Robot

The robot (Figure 4) consists of six subsystems: $M_1, \ldots, M_6$. Electrical and mechanical connections between subsystems are made through spring/metal contacts and rare-earth permanent magnets. Once all the subsystems are assembled correctly, the electrical circuit of the total system is completed and the robot starts working immediately. This prevents any unexpected malfunction of the incomplete replica during the assembly process.

The robot has a passive end-effector consisting of two 50-mm rods. Once the robot finds a subsystem by reading a bar code, the subsystem is captured by this end-effector as it moves along the track. We note that each subsystem is placed in a location with some tolerance in its position and orientation. On the basis of experimental observations, the tolerances of the subsystems for which the robot will still function, $\delta g = (\delta x[\text{mm}], \delta y[\text{mm}], \delta\theta[\text{radian}])$, are given by

$$\delta g_1 = \delta g_2 = (55, 13, 0.30); \quad \delta g_3 = (40, 13, 0.20);$$
$$\delta g_4 = (10, 13, 0.10); \quad \delta g_5 = \delta g_6 = (45, 13, 0.20),$$

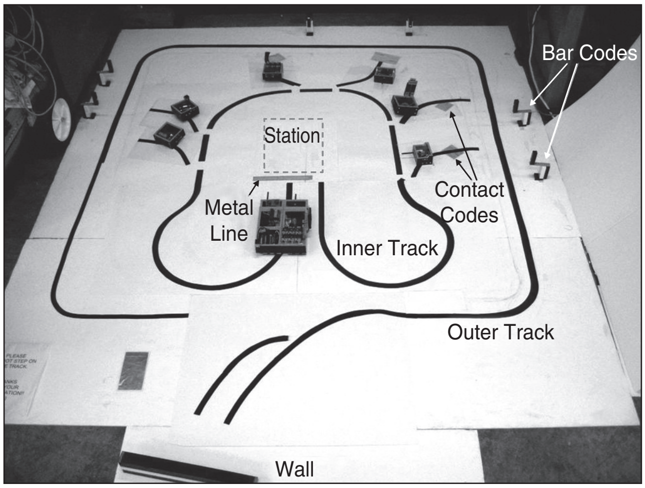


Figure 1. *Initial set-up of the robot and six subsystems of the replica in a partially structured environment.*

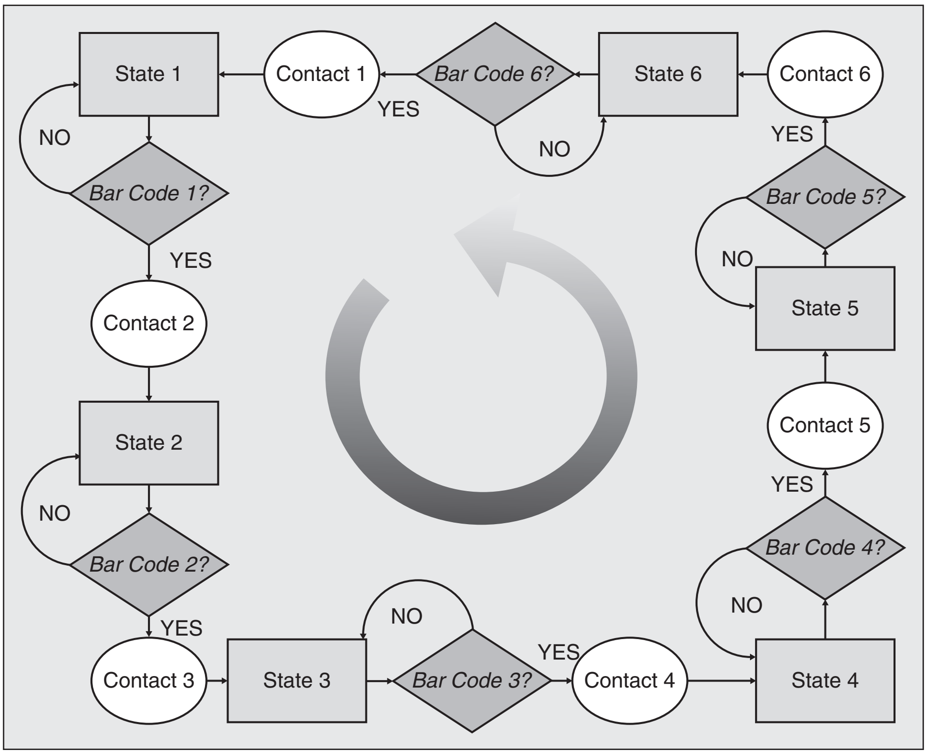


Figure 2. *State transitions of the robot according to the sensor inputs.*

where δg_i is the tolerance of M_i. We can observe that the subsystems with the same size and shape share the same tolerance value in their positions and orientations. To reduce friction while dragging a subsystem to the station, three ball casters are attached at the bottom of each subsystem.

- *Module 1:* M_1 (Figure 5) includes a relay circuit that receives inputs from the line tracker sensor in M_2 and gives output commands to the motor driving circuit in M_3 resulting in the line-tracking movements of the robot. The circuit consists of two relays and two transistors. The bases of the transistors connected to the line tracker sensor outputs through the interconnections between M_1 and M_2. This module also has a part of a passive end-effector and a metal detector (two metal nails facing down for detecting a metal line on the surface) to trigger the motor driving circuit to reverse directions of two motors.

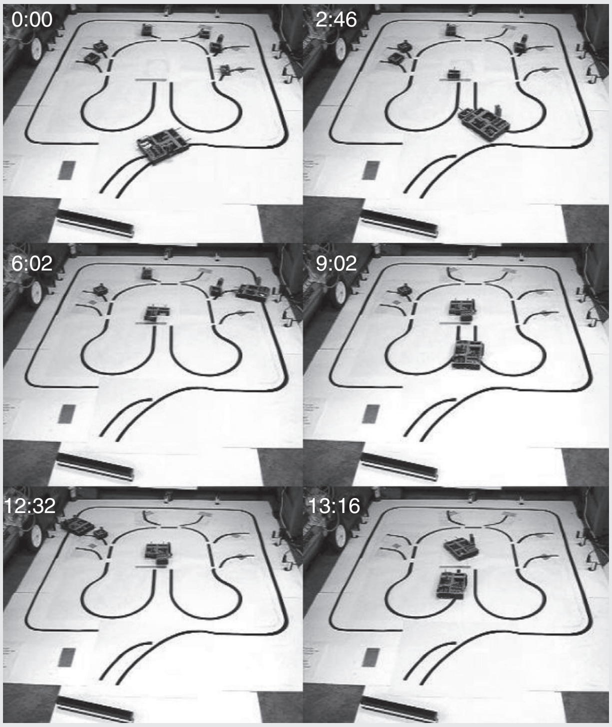


Figure 3. *The self-replication process at various times. The video clip is available at http://custer.me.jhu.edu/~kiju/srr1-lee.wmv.*

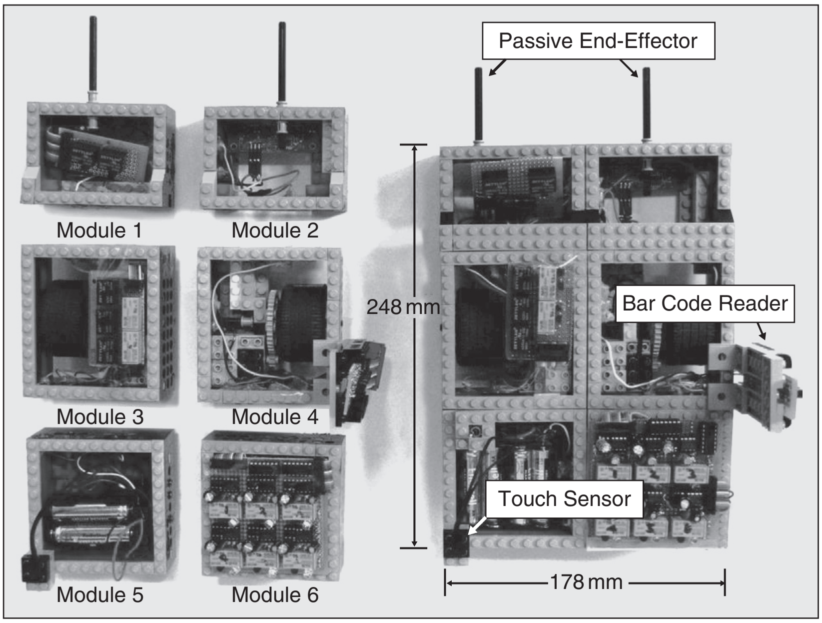


Figure 4. *The prototype consisting of six subsystems. (a) Before assembly. (b) After assembly.*

- *Module 2:* M_2 contains a line-tracker sensor as shown in Figure 6. The photo transistors in the line-tracker sensor detect reflected light from the surface and transfer the outputs to the relay circuit in M_1. We use a tracker sensor manufactured from Lynxmotion that has three pairs of infrared (IR) light emitting diode (LED) and photo diode. We only use two of three sensors (which are enough for stable line-tracking) among three pairs of LEDs and photo transistors in the line tracker sensor. The module has a rod in front that forms a passive end-effector with M_1 when they are assembled.
- *Module 3:* This module (Figure 7) is composed of the left motor and wheel with gears and a motor driving circuit that controls the left and right motors. The circuit is built with two latches, three transistors, and three relays. The circuit receives inputs from the relay circuit resulting line-tracking mode of the robot. When the metal detector in M_1 detects metal, the circuit triggers two motors to reverse directions, so that the robot moves backward blindly (without line-tracking).
- *Module 4:* M_4 has a bar code reader, the right motor, and wheel with gears (Figure 8). The bar code reader is mounted on the left side of M_4 so that the robot can read the bar codes placed along the outer track by moving in the counter-clockwise direction. The bar code reader is composed of six QRD1114 reflective object sensors that are IR LED and photo diode pairs. The bar code reader is able to read the bar code correctly within about 10 mm. The state machine in M_6 is triggered by the outputs from the bar code reader. The right motor in this module is controlled by the motor driving circuit in M_3 through the electrical connections.
- *Module 5:* Four 1.5-V AA batteries are used to power the system. The battery unit is built in M_5 (Figure 9), and all subsystems share the same power source through the electrical connections between subsystems. The circuit is designed to be closed only when all subsystems are assembled correctly to prevent any unexpected behavior of the incomplete replica during the self-replication process. After four to five experiments (each experiment takes about 780 s), the voltage dropped down to

approximately 3.7 V. This results in a reduction in the speed of the robot and an increase in sensor errors. As a solution for such problems, we can add a voltage regulator to provide a steady voltage input to the system. At the bottom of M_5, there are three contact sensors that can detect three contact codes attached at the surface.

- *Module 6:* M_6 in Figure 10 has the most complex circuitry among the six subsystems. It is a state machine consisting of six latches, 18 capacitors with different capacities, several resistors and transistors, and AND gates. The state machine has six states according to the robot's mission, such as in State 1 the robot is supposed to find M_1. The state machine receives outputs from bar code reader in M_4 and triggers the timer for five seconds. Once the robot enters to the cross way to collect a subsystem, the contact sensors in M_5 and M_6 recognize a contact code triggering the state machine to the next state.

The Partially Structured Environment

Structured environments can be divided into three categories: a completely structured environment, a partially structured environment, and an unstructured environment. We define a completely structured environment to be one in which every environmental structure is fixed at a precise location and no change or permutation is allowed in these locations. A partially structured environment is like a structured environment, but with structures whose locations can be permuted and perturbed by a small random error in pose. Thus, in a partially structured environment, the robot can still perform self-replication when the location of one or more environmental structure is switched with that of another structure. An unstructured environment is an environment without any structure resulting from initial human intervention.

Our prototype works in a partially structured environment. The environmental structures include outer track,

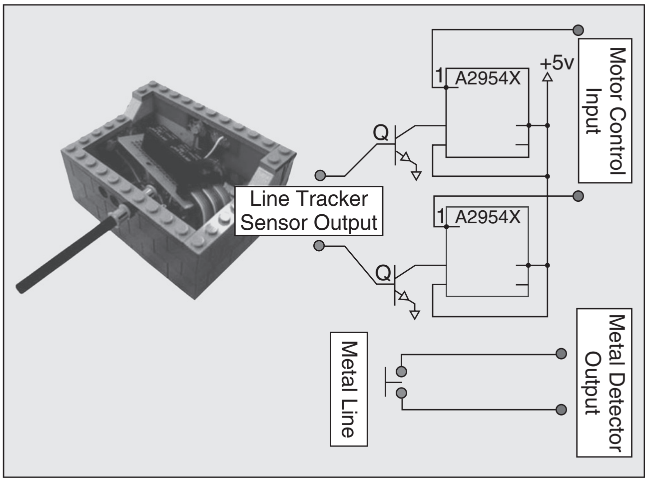


Figure 5. *Module 1 with the circuit diagram of the relay circuit and a metal detector, and a part of a passive end-effector (a bar). Blue dots indicate the input terminals and red dots denote the output terminals.*

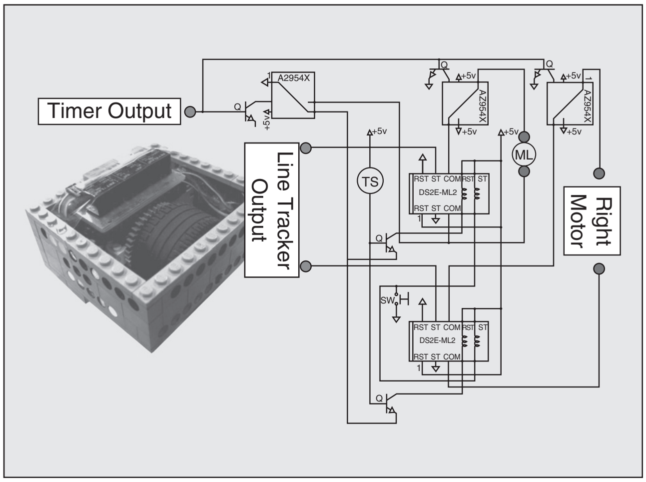


Figure 7. *Module 3 with the motor driving circuit diagram. ML is the left motor inside M_3 and TS is a touch sensor installed in M_5. Output ports described as Right motor are connected to motors in M_4.*

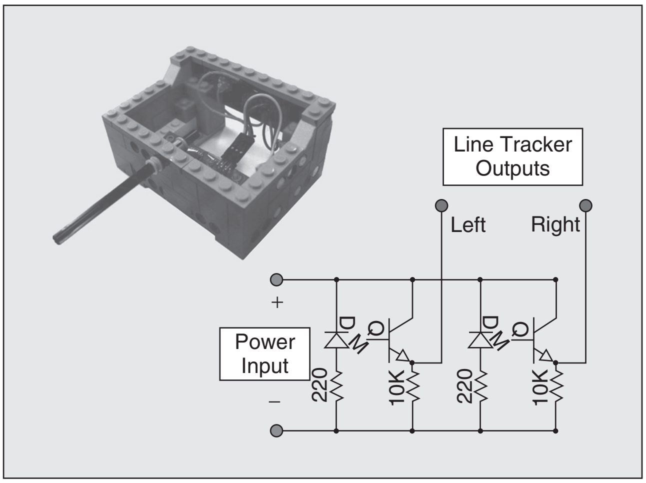


Figure 6. *Module 2 with the circuit diagram of the line tracker sensors. Two pairs of IR LED and photo diodes are used for line-tracking mechanism and a part of passive end-effector is installed in M_2.*

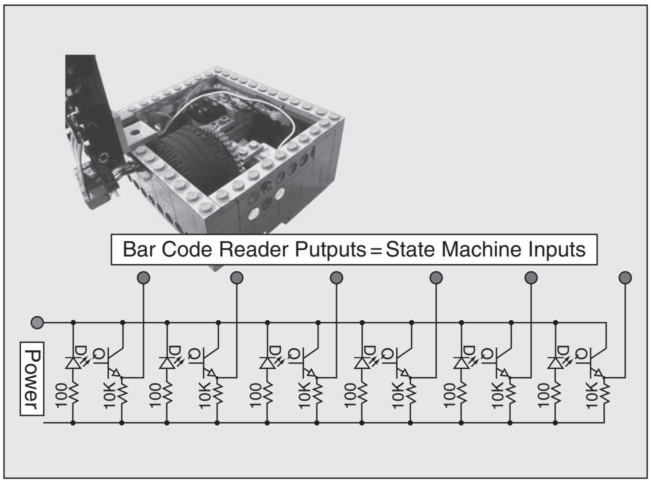


Figure 8. *Module 4 includes a right wheel, a motor, and a bar code reader. The circuit diagram of the bar code reader is shown. Every module shares the same power source from M_5.*

inner track, cross ways, six bar codes, six contact codes, backside wall, and metal line at the station (Figure 1). As shown in Figure 4, the robot composed of three layers of subsystems, and the replication process is made by stacking them in a certain order. The priority of assembly among the six subsystem is given by

$$(M_1, M_2) \succ (M_3, M_4) \succ (M_5, M_6),$$

where $\succ$ is defined as "assembled prior to." We note that there is no priority between two subsystems inside the parenthesis and the actual order is determined by the arrangement of environmental structures. There are six locations to place the subsystems along the outer track, and therefore there are 6! ways to arrange six bar codes around the outer track, each of them indicating a subsystem's ID. A contact code tells the robot which subsystem to collect next. For each arrangement of six bar codes, there are $2 \times 2 \times 2 = 8$ ways to arrange six contact codes to build the

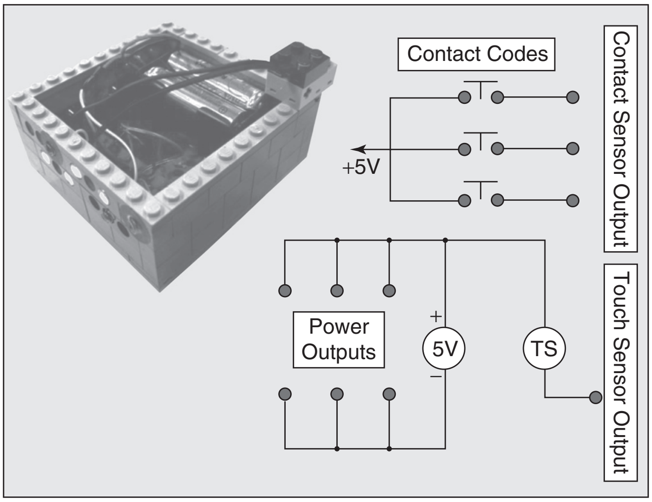


Figure 9. *Module 5 with the circuit diagram of 5-V power source and a LEGO touch sensor. M_5 contains four AA size batteries and three contact sensors installed at the bottom. TS is the touch sensor attached on the frame.*

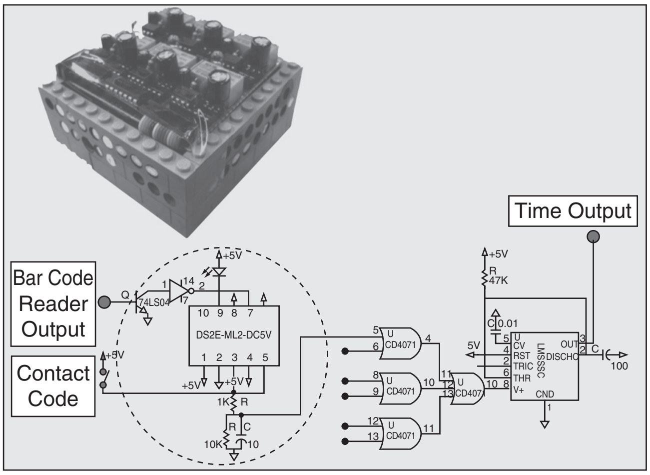


Figure 10. *Module 6 with the circuit diagram of the state machine. Three contact sensors (same as in M_5) are installed at the bottom, but omitted in the figure.*

same replica. Therefore, we have in total of $6! \times 8 = 5760$ possible permutations among the six bar codes and six contact codes. Since some of the environmental structures can be permuted and the system will still replicate successfully, this system can be seen as a simple, self-replicating robotic system in a partially structured environment.

The information about subsystem locations is embedded with bar codes and contact codes, and we call them landmarks of the system. The bar codes are located outside the tracks so that the robot can detect the bar code on the right side when it moves along the outer track. The bar code triggers the timer of the robot resulting in a left turn for five seconds. Bar codes are placed according to subsystem locations, i.e., if each subsystem's location is considered as a room, a bar code is a name card at the door. After entering to the cross way, the robot passes on a contact code attached on the surface while collecting a subsystem, resulting in a state transition of the robot. Those landmarks can be viewed as a list of written instructions. Thus the system follows instructions embedded in a properly designed environment.

Figure 11 shows a bar code made of LEGO blocks and a square piece of reflective paper. We first used a piece of white paper, but the reflective paper showed better results in sensing. A piece of reflective paper is located on one of six different heights indicating each location of six subsystems. The bar code is designed not to interfere with the robot trajectory but to be close enough to the bar code reader when the robot passes by. Six bar codes are located along the outer track.

A contact code is made of metal foil and transparency film with two holes as shown in Figure 12. Each contact code has

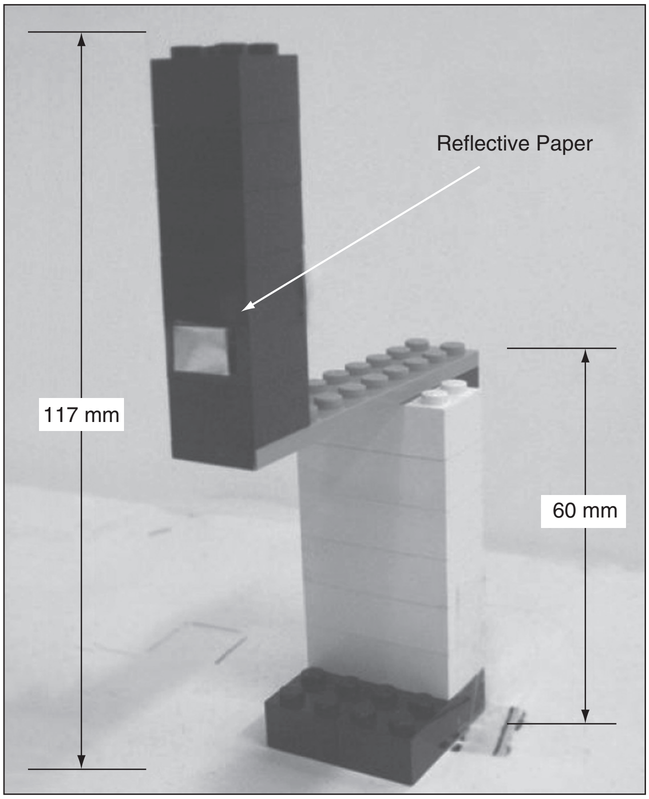


Figure 11. *Each bar code has a reflective metal piece at a different height indicating from Bar code 1 to Bar code 6.*

two holes in different locations, so that the contact sensor of the robot can read a different code. Contact codes are attached around six cross ways where subsystems are located. Therefore, the robot finds a subsystem and is triggered to the next state at the same time.

Degree of Replication

Although replicating/self-replicating and self-reconfigurable robotic systems have been developed by several researchers, there is currently no proper measure to evaluate the degree to which they replicate, i.e., a measure of how much order is created by the assembly process relative to the existing order in the unassembled parts. In this section, we first define a combined measure of system complexity distribution and relative complexity for ARS. The defined measure is applied to our prototype and some other replicating robotic systems.

Definitions

We define an active element to be a moving mechanical part or a fundamental electronic component. As a simple measure of subsystem complexity, we count the number of active elements in each subsystem. Some special parts, such as a chassis and fixed mechanical parts with special purpose (e.g., a passive end-effector in M_1 and M_2), are also viewed as active elements even if they are not moving mechanical parts. This measure can provide a reasonable estimate of the complexity of a part as long as the same criteria are applied through all systems being compared.

Each of the following are counted as a single active element: chassis, gear, wheel, shaft, switch, transistor, coil, capacitor, battery, etc. We did not count resistors as active elements, but a potentiometer is viewed as an active element. For example, the complexity of a relay consisting of a switch, a coil and an electromagnet is 1 (switch) + 1 (coil) + 1 (magnet) = 3, and the complexity of a brush motor is quantified as 1 (brush) + 1 (coil) + 1 (magnet) + 1 (shaft) = 4. In a decomposed robot, new mechanical and electrical connections between parts/subsystems are made when they are assembled. We call them interconnections, and each of them is counted as one active element since they represent the completion of otherwise passive components.

For a robotic system (recall that all robotic systems are active by our definition) consisting of n subsystems, we define the degree of replication for the system, D_s, as

$$D_s = \frac{C_{\min}}{C_{\max}} \cdot \frac{C_{\text{total}}}{C_{\text{ave}}} \cdot \frac{1}{C_{\text{ave}}}, \tag{4}$$

where

$$C_{\text{ave}} = \frac{1}{n}\sum_{i=1}^{n} C_i;$$

$$C_{\max} = \max\{C_1, \ldots, C_n\};$$

$$C_{\min} = \min\{C_1, \ldots, C_n\}; \text{ and}$$

$$C_{\text{total}} = \sum_{i=1}^{n} C_i + \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} C_{ij}.$$

We note that C_i is the number of active elements in M_i and C_{ij} is the number of interconnections between the ith and jth subsystems when they are assembled. $C_{ij} = 0$ if the ith and jth subsystems are not directly connected by the assembly process. In (4), the first term measures the complexity distribution throughout the subsystems, the second term measures the relative complexity of the total system to the individual subsystems, and the last term penalizes for complex subsystems. D_s is related to the system complexity measures defined in our conference papers [27], [28].

To design a robotic system capable of replication from low-level parts, the subsystem complexity should remain low relative to the total system complexity (i.e., high relative complexity). In addition, we view a system consisting of a few complex parts as more trivial than one composed of many low-complexity parts. A higher value of D_s indicates a more truly replicating system in terms of its complexity distribution and relative complexity. Although the concept of D_s may not generalize to all robotic systems, it is useful as *a* measure to evaluate robotic replicating systems.

Degree of Replication of Existing Prototypes

We first compute the degree of replication for our prototype. The number of active elements in each subsystem and the number of interconnections between adjacent subsystems are counted. As shown in Table 2, the subsystem complexities are given by

$$(C_1, \ldots, C_6) = (11, 16, 30, 24, 12, 91),$$

where C_i is the system complexity of M_i. Also the number of interconnections between modules counted is 49 as shown in Table 3. Therefore, the total and average system complexity is given by

$$C_{\text{total}} = 184 + 49 = 233; \quad C_{\text{ave}} = \frac{184}{6} \simeq 30.67.$$

The degree of replication, D_s, is computed as

$$D_s = \frac{11}{91} \cdot \frac{233}{(30.67)^2} \simeq 2.99 \times 10^{-2}.$$

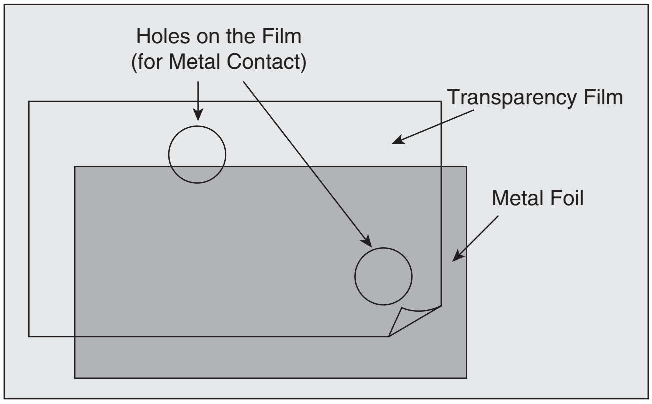


Figure 12. *A contact code is made by a thin metal foil covered by transparency film with two holes. Each contact code has holes in different locations in the transparency film.*

We calculate D_s for one of our previous prototypes [23] with a LEGO RCX controller in M_5. The RCX controller contains a microprocessor and 512 B of RAM. We do not know the exact number of transistors in the RCX, so we estimate a lower bound on the complexity based on the size of the RAM as $512 \times 8 = 4096$ active elements. The subsystem complexity is given by

$$(C_1, \ldots, C_5) = (12, 9, 12, 1, c),$$

where $c \gg 4000$, and the number of interconnections is 18. The system complexity ratio is computed as

$$D_s \ll \frac{1}{4000} \cdot \frac{4052}{807^2} \simeq 1.56\times10^{-6}.$$

The value of D_s for this system is much smaller than that of the new prototype presented in this article. A higher value of D_s indicates that a higher-order system is created by the assembly process relative to the initially provided subsystems.

Table 2. Number of active elements in each subsystem.

Components	M_1	M_2	M_3	M_4	M_5	M_6
Chassis (1)	1	1	1	1	1	1
End-effector (1)	1	1	0	0	0	0
Motor (4)	0	0	1	1	0	0
Wheel (1)	0	0	1	1	0	0
Line tracker (12)	0	1	0	0	0	0
IR emitter (1)	0	0	0	6	0	0
IR detector (1)	0	0	0	6	0	0
Transistor (1)	2	2	5	6	0	6
Capacitor (1)	0	0	0	0	0	12
Relay (3)	2	0	3	0	0	0
Latch (5)	0	0	2	0	0	6
OR gate (2)	0	0	0	0	0	4
Timer (28)	0	0	0	0	0	1
Touch sensor (1)	0	0	0	0	1	0
Metal detector (1)	1	0	0	0	0	0
Contact sensor (1)	0	0	0	0	3	3
Battery (1)	0	0	0	0	4	0
Ball casters (1)	0	0	0	0	3	3
Total	11	16	30	24	12	91

Table 3. Number of interconnections among six subsystems.

	M_1	M_2	M_3	M_4	M_5	M_6
M_1	0	7	9	0	0	0
M_2	7	0	0	4	0	0
M_3	9	0	0	6	5	0
M_4	0	4	6	0	0	11
M_5	0	0	5	0	0	7
M_6	0	0	0	11	7	0

For a system composed of n identical modules, we have

$$D_s = \frac{n(p+q)}{p^2}, \tag{5}$$

where p is the module complexity (the number of active elements in one module) and q is the normalized number of interconnections, such that $q = Q/n$, where Q is the total number of interconnections. In most existing systems with identical modules, each module contains a microprocessor, an actuator, etc., i.e., p is at least 10^6. The system described in [14] consists of four modules, such that $n = 4$ and $q = (19 \times 3)/n = 14.25$, has the degree of replication, $D_s \leqslant 4 \times 10^{-6}$.

As another example, if a system consisting of n subsystems with most of the system complexity concentrated in one of the subsystems, such that $C_1 = C_2 = \cdots = C_{n-1} = p$ and $C_n = P$ with $P \gg p$, then D_s is computed as

$$D_s = \frac{pn^2}{P} \cdot \frac{P + n(p+q) - p}{[P + p(n-1)]^2},$$

which is always smaller than the value of D_s in (5).

Conclusion and Future Work

Revising and extending von Neumann's early model of self-replicating automata, we presented a simple framework for robotic replication consisting of three sets of components. In addition, we characterized a system as being in one of four categories: a PSRS, an ARS, a passive self-replicating system, or an ASRS. The prototype presented in this article was viewed as an active self-replicating system in that there is no external control or machinery other than that by the initial functional robot. The replication process is composed of assembling six subsystems in a partially structured environment, which holds important information about the subsystems. A measure of the degree of replication, D_s, as defined and applied to the prototype and other robotic replicating systems. Although the prototype presented in this article showed the highest value of D_s among the systems being considered, it is still possible to have a higher D_s by reducing the subsystem complexity and increasing the number of parts, or having a uniform complexity distribution throughout subsystems.

Biological replication is a tremendously complicated process that we are attempting to mimic in engineering systems. The current trend in robotic self-replication is based on modular systems. To achieve sufficient functionality of the robot and for a replication process to be meaningful, the number of modules should be large and the module complexity should remain as low as possible. The same degree of replication achieved in biological systems may not be achievable by robotic systems. The main difference between robotic systems and living organisms is that robots are invented and used to satisfy human needs by performing specific tasks. In contrast, biological organisms have no innate purpose other than survival and reproduction, and the machinery to achieve these goals has been perfected over billions of years. One of the goals we want to achieve in robotic replication is minimizing human intervention and maximizing the functionality of the robot itself to perform given tasks while also being able to reproduce.

Keywords

Self-reproduction, robotic replication, self-replicating robot, modular robot, distributed system.

References

[1] J. von Neumann and A. W. Burks, *Theory of Self-Reproducing Automata*. Urbana, IL: Univ. of Illinois Press, 1962.

[2] R. A. Freitas, Jr., and R. C. Merkle, *Kinematic Self-Replicating Machine*. Georgetown, TX: Landes Bioscience, 2004.

[3] A. J. Jones, "Self-replicating probes for galactic exploration," Lecture at Department of Computing Imperial College, London, Oct. 15, 2000.

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

[5] L. S. Penrose, "Mechanics of self-reproduction," *Ann. Hum. Genet.*, vol. 23, pp. 59–72, Nov. 1958.

[6] L. S. Penrose, "Self-reproducing machines," *Sci. Am.*, vol. 200, no. 6, pp. 105–114, 1959.

[7] R. A. Freitas, Jr., and W. P. Gilbereath, Eds., "Advanced automation for space missions," in *Proc. 1980 NASA/ASEE Summer Study*, Washington, DC, 1982.

[8] G. Friedman, "Self-replication technology for the space solar power mission," in *Joint NASA/NSF Workshop on Autonomous Construction and Manufacturing for Space Electrical Power Systems*, Apr. 2002.

[9] 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, Dec. 2002.

[10] S. Murata, H. Kurokawa, and S. Kokaji, "Self-assembling machine," in *Proc. IEEE ICRA*, San Diego, CA, 1994, pp. 441–448.

[11] K. Tomita, S. Murata, H. Kurakawa, E. Yoshida, and S. Kokaji, "Self-assembly and self-repair method for a distributed mechanical system," *IEEE Trans. Robot. Autom.*, vol. 15, no. 6, pp. 1035–1045, 1999.

[12] H. Kurokawa, K. Tomita, A. Kamimura, S. Murata, and S. Kokaji, "Distributed self-reconfiguration control of an M-TRAN system," in *Robitcs: Science and Systems Workshop on Self-reconfigurable Modular Robots*, Aug. 2006, .

[13] M. Yim, D. Duff, and K. Rufas, "PolyBot: A modular reconfigurable robot," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2000, vol. 1. pp. 514–520.

[14] V. Zykov, E. Mytilinaios, B. Adams, and H. Lipson, "Self-reproducing machines," *Nature*, vol. 435, no. 7038, pp. 163–164, 2005.

[15] Z. Bulter, S. Murata, and D. Rus, "Distributed replication algorithm for self-reconfiguring modular robots," in *Proc. DARS '02*, June 2002, pp. 25–27.

[16] P. Varshavskaya, L. P. Kaelbling, and D. Rus, "On scalability issues in reinforcement learning for self-reconfiguring modular robots," in *Robitcs: Science and Systems Workshop on Self-reconfigurable Modular Robots*, Aug. 2006.

[17] R. Fitch and Z. Butler, "Million module march: Scalable locomotion for large self-reconfiguring robots," in *Robitcs: Science and Systems Workshop on Self-reconfigurable Modular Robots*, Aug. 2006.

[18] G. S. Chirikjian and J. Suthakorn, "Toward self-replicating robots," in *Proc. ISER'02*, Italy, July 2002.

[19] J. Suthakorn, Y. Zhou, and G. S. Chirikjian, "Self-replicating robots for space utilization," in *Proc. Robosphere'02*, NASA Ames Research Center, CA, 2002.

[20] G. S. Chirikjian, Y. Zhou, and J. Suthakorn, "Self-replicating robots for lunar development," *ASME IEEE Trans. Mechatron.*, vol. 7, issue. 4, pp. 462–472, Dec. 2002.

[21] J. Suthakorn, Y. Kwan, and G. S. Chirikjian, "A semi-autonomous replicating robotic system," in *Proc. CIRA'02*, July 2003, pp. 776–781.

[22] J. Suthakorn, A. B. Cushing, and G. S. Chirikjian, "An autonomous self-replicating robotic system," in *Proc. 2003 IEEE/ASME Int. Conf. Advanced Intelligent Mechatron.*, 2003, pp. 137–142.

[23] W. Park, D. Algright, C. Eddleston, W. K. Won, K. Lee, and G. S. Chirikjian, "Robotic self-repair in a semi-structured environment," in *Proc. Robosphere'04*, NASA Ames, Nov. 2004.

[24] W. A. Hastings, M. Labarre, A. Viswanathan, S. Lee, D. Sparks, T. Tran, J. Nolin, R. Curry, M. David, S. Huang, J. Suthakorn, Y. Zhou, and G. S. Chirikjian, "A minimalist parts manipulation systems for a self-replicating electromechanical circuit," in *Proc. IMG'04*, July 2004, pp. 349–354.

[25] G. M. Cooper and R. E. Hausman, *The Cell: A Molecular Approach*, 3rd ed. Washington, DC: ASM Press, 2004.

[26] Department of Bacteriology, Univ. of Wisconsin-Madison [Online]. Available: http://www.bact.wisc.edu/themicrobialworld/Phage.html

[27] S. Eno, L. Mace, J. Liu, B. Benson, K. Raman, K. Lee, M. Moses, and G. S. Chirikjian, "Robotic self-replication in a structured environment without computer control," in *Proc. IEEE CIRA'07*, June 2007, pp. 327–332.

[28] A. Liu, M. Sterling, D. Kim, A. Pierpont, A. Schlothauer, M. Moses, K. Lee, and G. S. Chirikjian, "A self-replicating robot with simplified control," in *Proc. IEEE ISAM'07*, July 2007, pp. 264–269.

Kiju Lee is pursuing her Ph.D in mechanical engineering at the Johns Hopkins University, Baltimore, MD. She received her B.S.E. degree in electrical and electronics engineering from Chung-Ang University, Seoul, Korea, in 2002, and the M.S.E. degree in mechanical engineering from the Johns Hopkins University in 2006. She received a graduate research scholarship from the Korea Science and Engineering Foundation 2003–2005. Her research interests include self-replicating robotic systems, self-assembly, distributed systems, modular robots, and multibody dynamics. Her dissertation research focuses on robotic self-replication and the design of self-replicating robotic systems.

Gregory S. Chirikjian received the B.S.E. degree in engineering mechanics, the M.S.E. degree in mechanical engineering, and the B.A. degree in mathematics from the Johns Hopkins University, Baltimore, MD, in 1988, and the Ph.D. degree from the California Institute of Technology, Pasadena, in 1992. Since 1992, he has been on the faculty of the Department of Mechanical Engineering, Johns Hopkins University, where he is now professor and chair. His research interests include the kinematic analysis, motion planning, design, and implementation of hyperredundant, metamorphic, and binary manipulators. In recent years, he has expanded the scope of his research to include applications of group theory in a variety of engineering disciplines and the mechanics of biological macromolecules. He is a 1993 National Science Foundation Young Investigator, a 1994 Presidential Faculty Fellow, and a 1996 recipient of the ASME Pi Tau Sigma Gold Medal.

Address for Correspondence: Gregory S. Chirikjian, 223 Latrobe Hall, 3400 N. Charles Street, Baltimore, MD 21218 USA. Phone: +1 410 516 7127. Fax: +1 410 516 7254. E-mail: gregc@jhu.edu.