Proceedings of the 2007 IEEE
International Symposium on Assembly and Manufacturing
Ann Arbor, Michigan, USA, July 22-25, 2007

A Memoryless Robot that Assembles Seven Subsystems to Copy Itself

Andrew Liu, Matt Sterling, Diana Kim, Andrew Pierpont, Aaron Schlothauer,
Matt Moses, Kiju Lee, Greg Chirikjian

***Abstract*— This paper presents a robot that can assemble exact functional replicas of itself from seven more basic parts/subsystems. The robot follows lines on the floor using light sensors and a simple control circuit without any onboard memory. It performs a self-replication task comparable in difficulty to those of previous self-replicating robots, but with a greatly simplified control system and reduced overall system complexity. Three methods are presented that quantify aspects of the complexity of the robot and the pattern of lines it follows. The complexity measures provide a way to compare existing self-replicating robot systems and to evaluate new designs. Robotic self-replication is an aspect of automated assembly that has not been studied extensively in hardware, and this work (which was the outcome of a project in a Mechatronics course at JHU) is one step in a larger effort to quantify and demonstrate various aspects of this research area.**

I. INTRODUCTION

In principle, a self-replicating machine given suitable inputs (energy, raw materials, etc.) can rapidly grow from a small "seed" device into a much larger system. This potential for rapid growth from small initial investment has led self-replicating manufacturing systems to be proposed for a range of applications, including nanotechnology [3], desktop manufacturing [1] [8], and space-based industrial installations [2] [4].

The formal study of artificial self-replicating systems was initiated by von Neumann [16] more than 50 years ago. Since then, most work in this area has focused on computer simulations of non-physical systems such as cellular automata [12], although recent physics-based simulations include [13] and [14]. The earliest physical self-replicating machines were presented in [7] and [11]. In [11] mechanical components were confined to a box and randomly agitated. The components would remain unassembled until a "seed" mechanism, composed of two assembled components, was added to the box. The seed would "catalyze" the assembly of the free components into duplicates of the seed via manipulation of locking mechanisms on each component. The system in [7] worked in a similar way, but used electric carts operating on an oval track instead of passive mechanical components confined in a box.

Most recent work in machine self-replication has dealt with the assembly of pre-fabricated components [5] [6] [10] [15] [18], although some efforts are beginning to address the problem of fabricating components from raw materials [1] [8]. The physical implementation in [5] is very similar to the simulation in [13], and it works with externally agitated components supplied in random intial configurations. The other implementations [6] [10] [15] [18] are internally-powered, but rely on components supplied (more or less) in specific, fixed configurations. The pre-fabricated components used in current self-replicating robots play the same role as the modules used in modular reconfigurable robotics [9] [17]. Self-replicating robots are designed specifically for replication, whereas modular robotic systems are typically designed for reconfiguration, and replication is not a requirement.

Our lab has built several prototype machines that exhibit some degree of self-replication, including specialized mechanisms [6] and systems built around line-following mobile robots [15] [10]. In [15] a track guides a microprocessor-controlled robot to various locations where it either grasps an unassembled component or attaches a component to the replica robot. The initial placement of components is critical for success, and there is only one arrangement that will work. In subsequent work, we have improved this system by allowing less structured environments. In [10], for example, the robot could determine the type of component it was about to pick up, and could therefore function correctly given a variety of initial arrangements. The robot presented in this paper represents an improvement over [15] because it accomplishes essentially the same task as [15] but with a greatly simplified control system. The microprocessor controller is by far the most complex part of these robots, so simplifying (or eliminating) it provides a significant reduction in overall system complexity. The motivation for reducing complexity (while maintaining the same system functionality) is that ultimately the components may be so simple that the system will be able to replicate given very basic input materials.

II. DESCRIPTION OF EXPERIMENT

This section describes a self-replicating machine system consisting of a mobile robot, unassembled robot components, and a track. Unassembled components are placed at certain locations on the track. As the robot follows the track, it picks up unassembled components and attaches them to a replica assembly in the center of the track. When the robot passes over certain locations, it changes the pattern of lines on the track, causing the robot to be redirected the next time it visits that location. After the final component is placed on the replica, the original robot halts activity and the replica automatically begins moving along the track. As a demonstration that the replica is fully functional, the original robot may be dissassembled and it's components placed on

A. Liu, M. Sterling, D. Kim, A. Pierpont, A. Schlothauer, M. Moses, and K. Lee are students in the Johns Hopkins Whiting School of Engineering
G. Chirikjian is a Professor and Chair of the Department of Mechanical Engineering at Johns Hopkins University `gregc@jhu.edu`

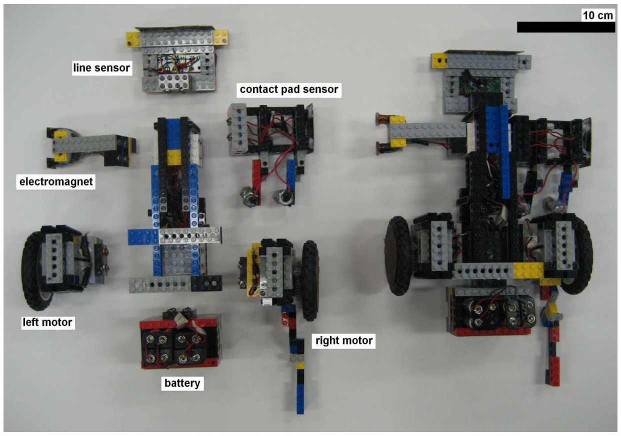

Fig. 1. Each robot is made of seven distinct components. Magnets provide mechanical and electrical connection between components.

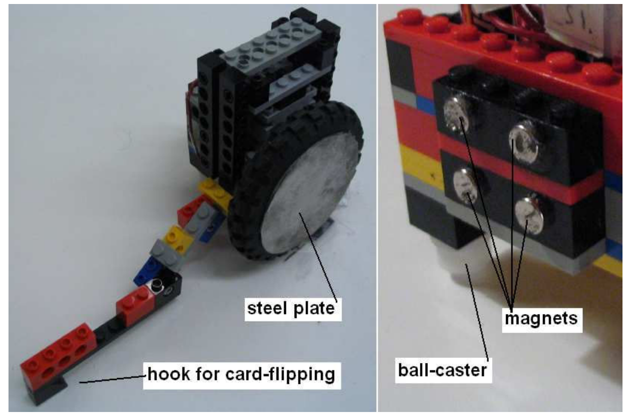

Fig. 2. Detail of the right drive motor and the magnets on the battery component. The arrangement and polarity of magnets is the same for all components.

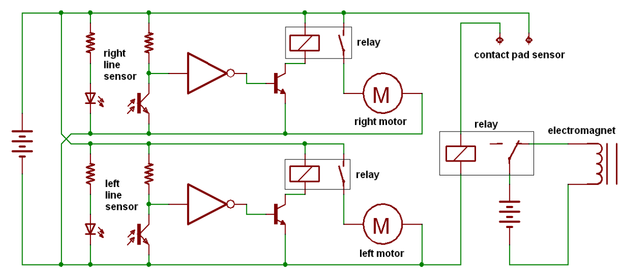

Fig. 3. Control circuit for robot.

the track. The replica will then assemble these components into a functional robot.

A. Robot

Each robot contains seven separate components: a central chassis, two drive motors, an electromagnet, a battery, a line sensor, and a contact pad sensor (see Fig. 1). Component-to-component interconnection is provided by an arrangement of four nickel-plated rare earth magnets (see Fig. 2). The magnets are strong enough to provide an acceptable mechanical connection between components. Wires are soldered to the surfaces of the magnets inside each component, so that when magnets of separate components make contact an electrical connection is established between components. It was found that a small amount of iron filings on the magnets improved the quality of the electrical connection. With the exception of the central chassis (which is not grasped or moved during replication) and the electromagnet (which is made of steel plate), the other components have a vertically oriented steel plate so they can be grasped with the electromagnet and moved by the robot.

The robot has three sensors (right line sensor, left line sensor, contact pad sensor) and three actuators (right motor, left motor, electromagnet). Each line sensor is an IR emitter/detector pair that rides about 1cm over the surface of the track. The sensor outputs a 5V level when it is over a white area of the track and 0V when over a black track mark. The contact pad sensor is a mechanism that uses a simple suspension linkage to slide two electrical contacts over the surface of the track. The presence of a conductive metal pad on the track closes the circuit between the two contacts on the sensor.

Control of the robot is very simple (Fig. 3). The electromagnet is always powered unless the contact pad sensor detects a metal pad. The right motor drives forward when the right line sensor is over a white area and is motionless otherwise. The left motor drives forward when the left line sensor is over a white area and is motionless otherwise. The result of this arrangement is that the robot will follow the black line on the track until it comes to a black mark that covers both line sensors, where it will stop. When the robot moves past an unassembled component, the electromagnet will "grasp" the steel plate on the component and the component will move with the robot until the contact pad sensor is over a metal pad on the track. The presence of the metal pad causes the electromagnet to turn off, and the component will be dropped. If the component is dropped close enough to the central chassis (within about 0.5cm), attraction between the magnets on the central chassis and the component will cause the component to attach to the chassis. The underside of each component is supported by a number of plastic ball-casters (a plastic sphere that rotates freely within a housing). The ball-casters reduce friction between the component and track, and facilitate movement of the component from storage to replica robot.

Unlike our previous robots [10] [15] the controller for this robot is not a state machine - there is no memory of previous states. While memory is not necessarily required in general for self-replication, this system does contain a mechanism for storing information about state, but it is in the track rather than the robot.

B. Track

The dimensions of the track are about 2m by 2m. The backing is made of white poseterboard and the track is black tape (See Fig. 4 and Fig. 5). Contact pads are made of metal foil and secured to the track with transparent adhesive tape.

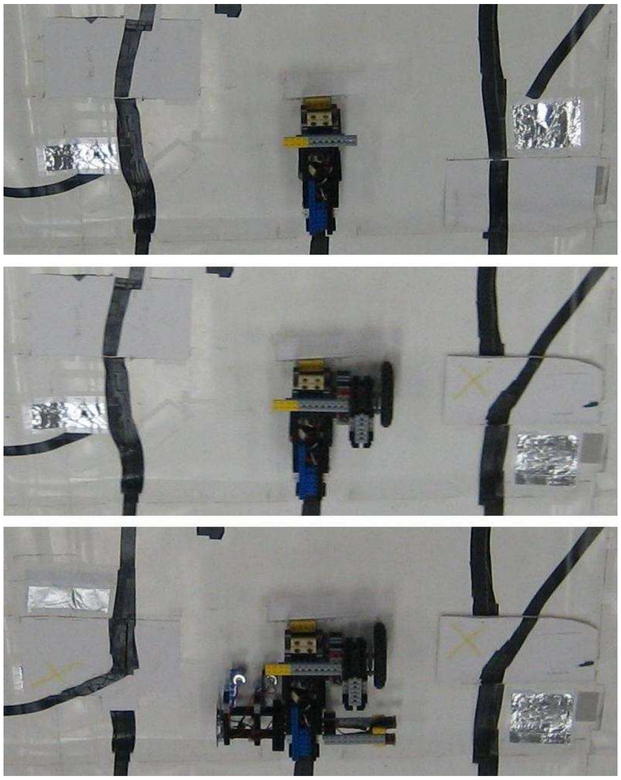

Fig. 4. Detail of the center portion of the track. The robot turns over flip-cards on each side of the central chassis as it drives past. Flipping a card changes location of the dropoff point and redirects the track line. Initial (top), intermediate, and final states of the cards are shown.

At each component storage location, there is a component-specific fixture that holds an unassembled component in the proper orientation so that a passing robot can easily grasp it with the electromagnet.

The track has six component storage locations and six "dropoff" locations. During replication, the central chassis remains in the center of the track and the other components are sequentially attached to it. Two components need to be placed on each side of the chassis, so the robot must make two approaches to each side. Because of the close spacing of the side components, the robot must use the same section of track to place both components. This is facilitated with a "flip-card" mechanism built into the track. The initial arrangement of the two flip-cards is shown in the top of Fig. 4. After the robot places the first component on one side of the chassis, a hook mechanism on the right motor (Fig. 2) flips over the card as it drives past. The card covers the contact pad used to place the first component, reveals the contact pad for the next component, and changes the direction of the track line so the robot will travel in a different direction after the next pass (middle image of Fig. 4). The same events take place when the robot attaches components to the other side of the chassis. The final state of the flip-cards is shown in the bottom image of Fig. 4.

TABLE I
ACTIVE ELEMENTS PER COMPONENT

Element (value)	CC	LS	RM	LM	CP	EM	BA
support structure (1)	1	1	1	1	1	1	1
magnet (1)	24	4	4	4	4	4	4
motor (4)			1	1			
electromagnet (1)						1	
relay (3)	3						
transistor (1)	2	4					
resistor (1)		4					
IR emitter (1)		2					
IR detector (1)		2					
ball caster (1)		2	2	2	2	1	4
wheel (1)			1	1			
mechanical linkage (1)				2	2		
electrical contact (1)	24	4	4	4	6	4	4
battery pack (1)							2
Total	60	23	16	18	15	11	15

C. Replication Process

Fig. 5 shows four images in the replication process. In the top image, the replica components have been placed at their storage locations on the track, and the original robot begins on the section of track just beyond the central chassis of the replica. Components are added to the replica in the order: left motor, electromagnet, contact pad sensor, right motor, line sensor, battery. In the second image, the left motor and electromagnet are in place, the first flip-card is turned over, and the robot has just grasped the contact pad sensor. In the third image, the contact pad sensor and right motor are in place, the second flip-card is turned over, and the robot is headed to pick up the line sensor. In the final image, the original robot has placed the battery component and come to a halt at the end of the track. The replica robot automatically starts when the battery component is attached, and it has begun to move along the track. The replica robot can assemble another replica, provided that the original robot is removed, new components are placed at the proper storage locations, and the track is "reset" by turning the flip-cards to their initial state. The replication process takes about two minutes and 40 seconds.

III. COMPLEXITY MEASURES

The self-replicating system described in this paper is one of a series of similar demonstrations using line-following mobile robots. As mentioned in the Introduction, one of our goals is to simplify the system components as much as possible while maintaining the same or greater level of functionality. Compared to our previous experiments, the system described here has a much simpler controller, but a somewhat more complicated track (due to necessity of flip-cards and specialized component storage fixtures). In this Section, we present complexity measures that can be applied to a robot and a track. These measures help to provide a quantitative measure of how "complicated" a robot or track is. This provides a basis for comparing total complexity

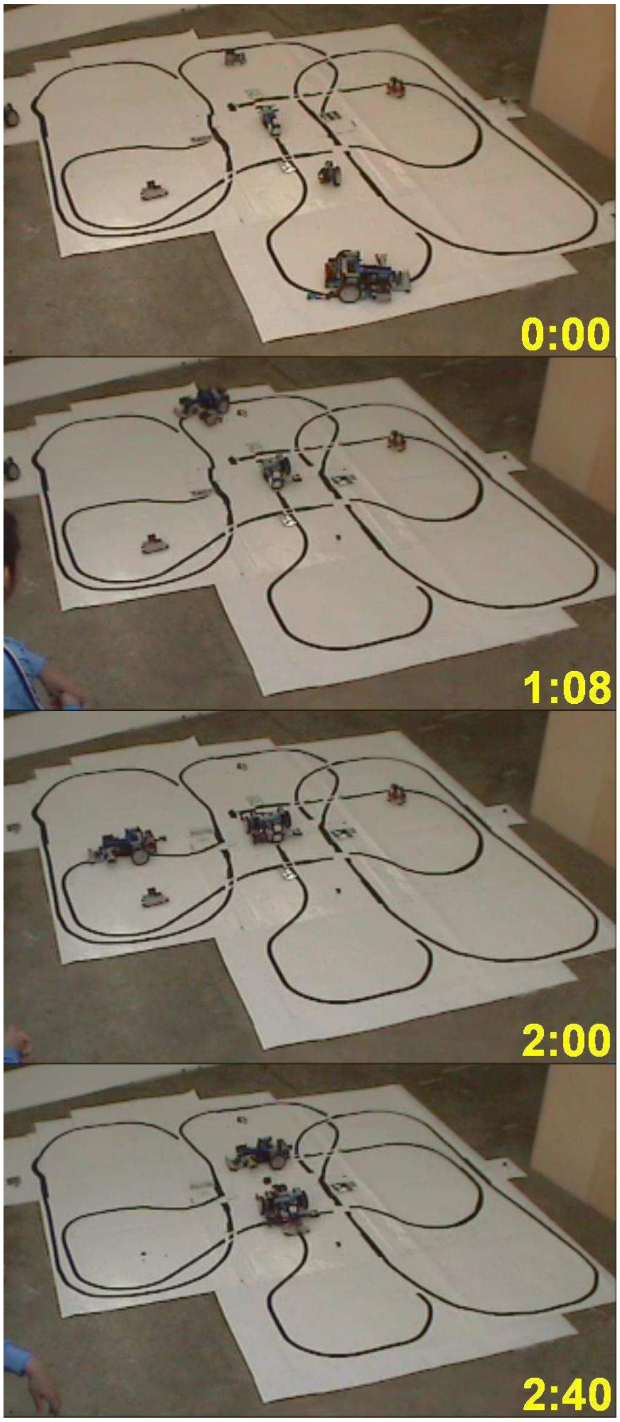

Fig. 5. Four images of the replication process. The labels indicate elapsed time in minutes:seconds.

between different systems, and may be of use in evaluating future system designs.

A. Active Elements

As a simple measure of robot complexity, we count the number of "active elements" for each component. What constitutes an active element is somewhat subjective and arbitrary, but this measure can provide a reasonable estimate for comparisons, as long as the same criteria are used across all systems being compared. In general we define an active element as a moving mechanical part or a fundamental electronic component. Each of the following are counted as a single active element: gear, shaft, magnet, electromagnet, switch, transistor, resistor, capacitor, electrical contact, mechanical linkage. As an example, we count a relay as 3 active elements (1 switch, 1 electromagnet, 1 mechanical linkage), and a motor as 4 (1 shaft, 1 magnet, 1 coil, 1 brush). Table I lists the active element count for each component. Table abbreviations are: CC central chassis, LS line sensor, RM right motor, LM left motor, CP contact pad sensor, EM electromagnet, BA battery.

For a robot consisting of n components, the set of component "complexities" is given by $\{C_1, \cdots, C_n\}$ where C_i denotes the active element count for the i^{th} component. We define the *active element distribution ratio*, r_a, as

$$r_a = \frac{C_{max}}{C_{min}} \cdot C_{avg} \tag{1}$$

where

$$C_{avg} = \frac{1}{n} C_{tot} = \frac{1}{n} \sum_{i=1}^{n} C_i,$$

$$C_{max} = max\{C_1, \cdots, C_n\},$$

$$C_{min} = min\{C_1, \cdots, C_n\}.$$

A low value of r_a indicates components have low average complexity and are of similar complexity. High values of r_a indicate high complexity and large variation in complexity between components. For a fixed C_{tot}, assuming each component has at least one active element and that a robot contains at least two components, upper and lower bounds for r_a are given as

$$r_{amin} = \frac{C_{tot}}{n},$$

$$r_{amax} = (C_{tot} - n + 1)\frac{C_{tot}}{n},$$

$$2 \leq n \leq C_{tot}.$$

The robot described above has a total active element count of $C_{tot} = 158$ and a distribution ratio of $r_a = 123.1$. This is plotted along with upper and lower bounds for r_a in Fig. 6. The r_a values for our previous robots [15] [10] are difficult to calculate because we do not have data for the active element count (number of transistors) in their microcontrollers. If we assume that the controller components in the previous robots have a C value $\approx 10^5$ while the other components have C

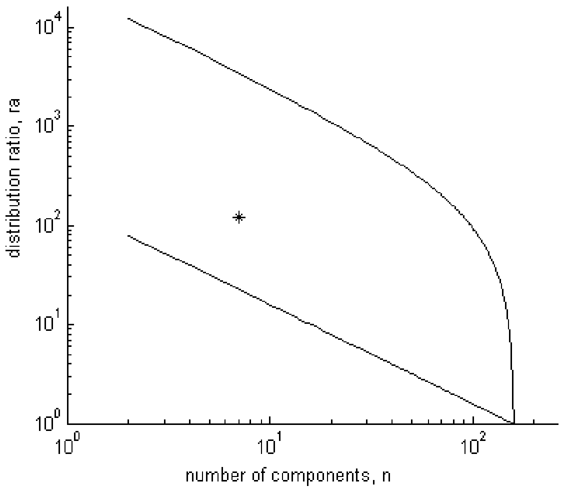

Fig. 6. Upper and lower bounds of r_a for a robot with a fixed number of active elements, $C_{tot} = 158$. The star indicates n and r_a for the robot described in this paper.

much less ($\approx 10^1$), it is easy to see that r_a for these robots is quite high.

The active element distribution ratio provides a method of comparing the complexity of different robot systems made of prefabricated components. We claim that it is desirable to build self-replicating robots with low distribution ratio for two reasons. First, the replication process with a low r_a robot is "less trivial" because the complexity of the total system must arise more from the replication process than from an individual component. In other words, the replication process is more than simply adding "bells and whistles" to one component that contains most of the system complexity. Second, low r_a indicates simple components. As components for self-replicating robots become simpler, it becomes easier to address the question of how to build the components themselves.

B. Track Graph

We can think of the track as a program of instructions that the robot "runs". By quantifying the essential information contained in the track, we get a notion of the tradeoff between robot and track complexity. The track can be represented as a directed graph (Fig. 7). The vertices of the graph correspond to important locations on the track (storage sites, dropoff locations, etc) and the edges correspond to the track lines. The flip-cards are represented as special vertices that direct the robot to one vertex on its first visit and to a second vertex on the next visit. The replication process is modeled as a walk from the first vertex to the last vertex in the graph.

We consider two aspects of track complexity: geometrical placement of the vertices, and non-geometrical connectivity between vertices. We assume it is reasonable to neglect the information contained in the geometry of the track lines, because given vertex placement and connectivity it is computationally easy to connect the vertices with edges. In other words, the placement of the vertices and the order they are visited is important, but the route taken between them is not.

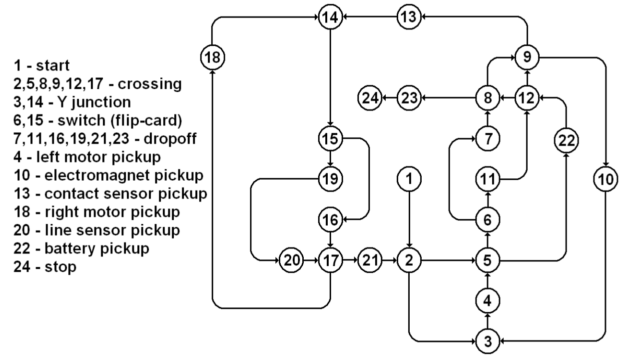

Fig. 7. The track can be represented as a directed graph with 31 edges and 24 vertices. Compare to the physical track in Fig. 5.

The parameters needed to specify a vertex are position, orientation, and the type of track site it represents. We define the *geometrical vertex information*, I_{gv}, as the sum of information needed to specify each individual vertex

$$I_{gv} = \sum_{j=1}^{v} \log_2 \left(\alpha \frac{2\pi xy}{\delta x_j \delta y_j \delta \theta_j} \right) \tag{2}$$

where v is the number of vertices, α is the number of types of vertices, x and y are the overall dimensions of the track, and δx_j, δy_j, $\delta\theta_j$ are the positional and angular tolerances for the j^{th} vertex. For the track described above, $v = 24$, $\alpha = 12$, and $x = y = 2m$. Using the same positional tolerances for each vertex of $\delta x = \delta y = 1mm$ and $\delta\theta = 5°$ results in a value of $I_{gv} = 761$ bits.

A directed graph can be represented by a list of ordered pairs, with each pair corresponding to an edge. For example, the track graph in Fig. 7 can be represented as

$$\begin{gathered}\{\{1,2\},\{2,3\},\{3,4\},\{4,5\},\{5,6\},\{6,7\},\{7,8\},\\ \{8,9\},\{9,10\},\{10,3\},\{6,11\},\{11,12\},\{12,9\},\{9,13\},\\ \{13,14\},\{14,15\},\{15,16\},\{16,17\},\{17,18\},\{18,14\},\\ \{15,19\},\{19,20\},\{20,17\},\{17,21\},\{21,2\},\{2,5\},\\ \{5,22\},\{22,12\},\{12,8\},\{8,23\},\{23,24\}\}.\end{gathered}$$

We define the *vertex connectivity information*, I_{vc}, as the information needed to specify the edge list of the track graph

$$I_{vc} = E \log_2 (2v) \tag{3}$$

where E is the number of edges. The track graph has 31 edges and 24 vertices, so $I_{vc} = 174$ bits.

I_{gv} and I_{vc} provide a method for comparing the complexity of different tracks. In future self-replicating robot systems, we would like to either eliminate the track entirely or design the track so that it may be replicated by the robot. In either case, reducing I_{gv} and I_{vc} brings us closer to the goal.

IV. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

A new line-following self-replicating robot system was presented. The control system for this robot was significantly simpler than earlier demonstrations, while the robot maintained the same level of functionality. The simple control system necessitated a somewhat more complicated track. The track used special flip-cards to redirect the path of the robot as it moved along the track lines. Three measures of complexity were presented for quantifying aspects of complexity of both the robot and the track. These measures may be of use in comparing overall system complexity of different self-replicating robot systems. Additionally, the measures provide a quantitative method for evaluating new system designs as we work toward simplifying the constituent components of new self-replicating robots, and extending their overall capability.

B. Future Works

Some improvements for self-replicating robots include further reducing component complexities, eliminating the track or incorporating it into the replication process, allowing less structured environments, and demonstrating useful capabilities of robots in addition to replication, such as cooperation between multiple robots, self-repair, and assembly of larger structures.

V. ACKNOWLEDGMENTS

This work was supported by the Johns Hopkins University Department of Mechanical Engineering.

REFERENCES

[1] Bowyer, A., "The Self-replicating Rapid Prototyper - Manufacturing for the Masses", Invited Keynote Address, Proc. 7th National Conference on Rapid Design, Prototyping & Manufacturing, Centre for Rapid Design and Manufacture, High Wycombe, June 2006. (in press)

[2] Chirikjian, G. S., Zhou, Y., and Suthakorn, J., "Self-Replicating Robots for Lunar Development", IEEE/ASME Transactions on Mechatronics, Vol 7, No 4, Dec 2002, p462-472.

[3] Drexler, K. E., "Nanosystems", 1992, John Wiley and Sons, Inc.

[4] Freitas, R. A., Merkle, R. C. "Kinematic Self-Replicating Machines", Landes Bioscience, Georgetown Texas, 2004.

[5] Griffith, S., Goldwater, D., Jacobson, J. M., "Self-replication from random parts", Nature, Vol 437, 29 September 2005, p636.

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

[7] Jacobson, H., "On models of reproduction", American Scientist, Vol. 46, 1958, p 255-284.

[8] Lipson, H., "Homemade: The future of Functional Rapid Prototyping", IEEE Spectrum, May 2005, pp. 24-31

[9] Murata, S., Kurokawa, H., and Kokaji, S., "Self-assembling machine," *Proceedings of the IEEE International Conference on Robotics and Automation*, San Diego, CA, 1994, pp.441-448.

[10] Park, W., Algright, D., Eddleston, C., Won, W. K., Lee, K., and Chirikjian, G. S., "Robotic self-repair in a semi-structured environment," *Proceedings of Robosphere2004*, NASA Ames, November 2004.

[11] Penrose, L. S.,"Self-reproducing machines," *Scientific American*, Vol.200, No.6, 1959, pp.105-114.

[12] Sipper, M., "Fifty years of research on self-replication: An overview," *Artificial Life*, 4(3), 1998, pp.237-257.

[13] Smith, A., Turney, P., and Ewaschuk, R., "JohnnyVon: Self-replicating automata in continuous two-dimensional space" NRC Technical Report ERB-1099. National Research Council Canada, 2002.

[14] Stevens, W. M., "NODES: An Environment for Simulating Kinematic Self-Replicating Machines" Proc. of the Ninth International Conference on the Simulation and Synthesis of Living Systems (ALIFE9) 39-44, 2004.

[15] Suthakorn, J., Cushing, A. B., and Chirikjian, G. S.,"An autonomous self-replicating robotic system," *Proceedings of 2003 IEEE/ASME International Conference on Advanced Intelligent MEchatronics*, 2003.

[16] Von Neumann, J., Burks, A. W., "Theory of Self-Reproducing Automata," *University of Illinois Press*, 1962.

[17] Yoshida, E., Murata, S., Kamimura, A., Tomita, K., Kurokawa, H., Kokaji, S., "A Self-Reconfigurable Modular Robot : Reconfiguration Planning and Experiments", International Journal of Robotics Research, Vol. 21, No. 10, pp.903-916, 2003

[18] Zykov, V., Mytilinaios, E., Adams, B., Lipson, H., "Self-reproducing machines", Nature Vol. 435 No. 7038, 2005, pp. 163-164