Measures of Reliability and Task Complexity for Self-Replicating Robotic Systems

Kiju Lee and Gregory S. Chirikjian

***Abstract*— A self-replicating robot is one that has the ability to actively assemble basic components or subsystems to form an exact functional copy of itself. Measures of reliability and task complexity for modular self-replicating systems are introduced and applied here to a physical prototype. We define a reliability ratio based on the probability that a robot successfully replicates itself. In order to quantify the task complexity of the replication process, the amount of Sanderson's parts entropy that is reduced during the replication process is computed. A fully automated self-replicating robotic system composed of six subsystems with distributed electronics is presented to demonstrate the concept of robotic self-replication. The reliability ratio and task complexity are computed for this prototype.**

I. Introduction

John von Neumann [1] developed the first theoretical framework for self-replicating machines in the early 1950s. His concepts of self-replicating systems have been applied in many research areas such as cellular automata, nanotechnology, macromolecular chemistry and computer simulations [2]. The first mechanical implementation was presented by Penrose [3],[4]. He showed that simple units or blocks with certain mechanical properties could build a self-reproducing machine (referred to as the Penrose block replicator [5] throughout this paper), and demonstrated the assembly of passive elements under external vibration.

A few decades later, NASA became interested in self-replicating robots as a potential means for space development and exploration [5], with the long-term goal of constructing self-replicating factories on the moon [6]∼[8]. More recent research includes self-assembly, self reconfiguration and self-repair of modular robots. Algorithms for self-assembly using modular robots were described by Murata [9]. Tomita ([10], [11]) and Yim [12] presented physical prototypes of self-reconfigurable modular robotic systems. A centralized control algorithm was introduced and simulated [13], featuring a filter that checks for any isomorphism between the given state and the known states and then finds an appropriate mapping. In addition, self-replicating modular cubes with the self-replication capability were introduced [14]. A distributed algorithm for self-replicating modular robots and a reinforcement learning approach to learning self-reconfigurable modular robots were developed [16], [17].

Recently, our lab has built several prototypes in order to develop and demonstrate the concept of robotic self-replication and uncover hardware limitations. Robotic self-replication is viewed as a process in which an initial functional robot assembles several man-made subsystems and builds a functional replica. The first generation of our lab's prototypes are called semi-autonomous self-replicating robots [18], [19]. These prototypes are remote-controlled and contain a microprocessor-based controller in each system. We then developed a fully autonomous self-replicating robotic system in a structured environment [20]. In a subsequent study [21], barcode labels were added to the track design, which enable the robot to distinguish between subsystems by reading the barcode on each location where a subsystem is placed. Each of the systems has LEGO$^{\mathrm{TM}}$ RCX controller in one of its subsystems. The distinction between a semiautonomous and a fully autonomous system is made by whether human intervention is needed during the self-replication processes or not.

Although several prototypes have been developed, there was no proper measure to evaluate robotic systems in terms of the reliability and complexity of the self-replication process. A metric of self-replicability was presented by Adams and Lipson [15], but it was not applicable to physical systems. In this paper, we introduce a reliability ratio based on the probability that a system successfully replicates itself, and a measure of task complexity to be performed by self-replicating modular robots. A prototype is also presented to demonstrate the concept and to apply the developed measures. The physical prototype is composed of six subsystems that are simple electromechanical parts. In order to keep the subsystem complexity low, the necessary circuitry is divided into the number of subsystems, instead of devising a microprocessor-based controller. The system has no computer or microprocessor, and demonstrates a higher degree of self-replication (in the sense that the number of subsystems to assemble is larger) than our previous prototypes with microprocessor-based controllers [20], [21].

II. Measures for Self-Replicating Systems

A simple architecture for robotic self-replication can be represented with three sets of physical components:

$$\mathrm{S} = (\mathrm{R}, \mathrm{M}, \mathrm{E})$$

where R is a multiset[1] of initial functional robots, M is a multiset of subsystems (resources) to be used to form a replica and E is a multiset of environmental structures. A

Kiju Lee is with the Department of Mechanical Engineering at Johns Hopkins University, Baltimore, MD 21218, USA. kiju@jhu.edu

Gregory S. Chirikjian is with the Department of Mechanical Engineering, Johns Hopkins University, Baltimore, MD 21218, USA, gregc@jhu.edu

[1]In mathematics, a multiset differs from a set in that each member can have multiplicity. For example, {a, a, b, b, b, c} is a multiset.

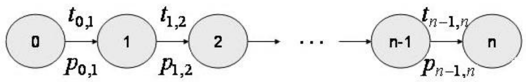

Fig. 1. Example1:self-replication process with n subsystems represented by a directed graph.

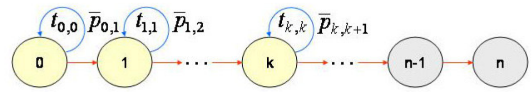

Fig. 2. Example2: if the robot fails to collect any of the first k subsystems, then it repeats the procedure until it completes the process successfully

robotic replication function, Φ, is defined such that $\Phi : \mathrm{S} \rightarrow \mathrm{S}'$ where $\mathrm{S}' = (\mathrm{R}', \mathrm{M}', \mathrm{E}')$ satisfying $|\mathrm{R}| < |\mathrm{R}'|$ (where $|\cdot|$ denotes the size of the set). An initial robot, or collection of robots, has sufficient functionality to build a copy of itself from provided resources. If there exists an external constructor that *actively* participates in the self-replication process, then R is not viewed as a self-replicating system even if R takes action in the production of a replica of R.

The structured environment may hold important information for a functional robot and play the role of the 'written instructions' in the von Neumann's model. In this case, the functionality of the robot may be reduced by the amount of information embedded in the environmental structures. If an environment is just an open space, i.e. there is no environmental structures built by human (or a constructor), then we consider $\mathrm{E} = \emptyset$, the empty set. The set of environmental structures, E, can be categorized as follows: (1) a completely structured environment when $\mathrm{E} = \mathrm{E}' \neq \emptyset$ and E is a strictly ordered set (i.e. no environmental structure can be either replaced or permuted), (2) a partially structured environment when $\mathrm{E} \neq \mathrm{E}'$, or when E is a partially ordered or unordered set (i.e. either a robot actively changes its environment during self-replication, or environmental structures have some flexibility on their locations without affecting to self-replication process) and (3) an unstructured environment when $\mathrm{E} = \mathrm{E}' = \emptyset$.

A. Reliability Ratio

In many cases, robotic self-replication or self-assembly is made by a step-by-step procedure. This procedure often include some repeated tasks. For example if a robotic system replicates itself by collecting n parts one by one, then the robot may repeat the similar process n times. For those kinds of systems, we can estimate the probability of the system to replicate successfully from given individual probabilities for distinctive tasks. The probability of each distinctive task is obtained by experiments. Fig. 1 shows an example of a simple self-replication process. The initial robot assembles n parts in a certain order. The node i represents the completion of assembling the i^{th} part, and the weighted edge shows the time required from node $i-1$ to node i, which is denoted by $t_{i-1,i}$. The probability that the robot assembles the i^{th} part successfully given that the first $i-1$ parts are already assembled is denoted by $p_{i-1,i}$.

We define *the reliability ratio* as the probability that a robot can assemble every subsystems correctly to build a replica. In the case of Fig. 1, if each step is statistically independent from the others, the reliability ratio is computed as

$$K_s = \prod_{i=1}^{n} p_{i-1,i} \tag{1}$$

where $0 < K_s \leq 1$. If $K_s = 0$, then the system is not able to self-replicate. One can design a robotic system to repeat the same task when it fails. In this case, in general, the reliability ratio of a system can be increased given more time. Fig. 2 shows an example that a robot is allowed to repeat failed procedures for the first k steps. We note that $\bar{p}_{i-1,i} = 1 - p_{i-1,i}$ and $t_{i,i}$ is the time for the robot to return to the previous stage. If the robot fails to collect the j^{th} subsystem for $j \leq k$, $p_{j-1,j}$ goes to 1 as t goes to ∞. Given unlimited time and energy, the reliability ratio for the system in Fig. 2 is

$$K_s|_\infty = \prod_{i=k+1}^{n} p_{i-1,i}. \tag{2}$$

For given t, such that $t_s \leq t \leq \infty$, K_s is bounded as

$$K_s|_{t_s} \leq K_s|_t \leq K_s|_\infty$$

where $t_s = \sum_{i=1}^{n} t_{i-1,i}$. The equalities hold when a robot is not allowed to repeat the failed process or a robot never fails.

The reliability ratio provides an estimate (or at least upper and lower bounds) on the success ratio of a self-replication process for a given amount of time by testing individual steps rather than performing many experimental trials of the whole procedure. Once we get the success ratio of each step from partial experiments, we can compute lower and upper bounds on K_s. If the time-efficiency is the main consideration, then a higher value of $K_s|_{t_s}$ for a minimum t_s would be desirable. In that case, the optimal trajectory of the robot during the self-replication process and high sensing accuracy are important factors to achieve a high time-efficiency. If the reliability is the main consideration rather than time, then by designing the process to have closed loops at each step, i.e. the system will return to one of the previous steps when it fails to go to the next step, the system would obtain $K_s|_\infty \rightarrow 1$.[2]

In biology, self-replication processes are much more complicated than that in robotic systems and often include parallel procedures and random factors. Also in robotic systems, if a robot replicates by a random process (e.g. a self-replication made by an external agitation of a box containing subsystems), then we cannot determine the reliability of the

[2] In fact, the robot cannot run for $t = \infty$. Therefore the maximum running time will depend on the lifetime of the robot, and $K_s|_\infty$ may be viewed as the maximum reliability ratio of the system for its lifetime.

system with the presented method. Otherwise, for a modular self-replicating system in which a self-replication process is viewed as a step-by-step procedure, and the success ratio of each step is measurable by experiments, the reliability ratio can be computed for the total self-replication process over time.

B. Task Complexity

The number of tasks (or amount of work) done by a robotic system during a self-replication process varies according to each system. From an information theoretic point of view, robotic self-replication can be viewed as a process reducing uncertainty on unassembled parts by collecting and assembling them to make a functional copy of an initial robot. All machinery associated with control and manipulation during the process is solely made by an initial robot. Then, how can we quantify the difficulty of tasks done by a robot for self-replication? We compute the entropy reduction made by self-replication using Sanderson's parts entropy concept [22]. The parts entropy for subsystems before self-replication and that after self-replication are calculated and compared to get difference between two results. This difference tells the amount of uncertainty reduced by the self-replication process, and we call this as *the task complexity* of the system.

Entropy is a useful statistical measure that can be addressed by a modular system or a self-replicating (or self-assembly) system. Given a discrete space consisting of points $\mathbf{x}_1, \cdots, \mathbf{x}_n$, and a discrete probability distribution $f_i = f(\mathbf{x}_i)$, such that $\sum_i f_i = 1$, the corresponding discrete entropy is defined as

$$H_x = -\sum_{\mathbf{x}_i} f(\mathbf{x}_i) \log f(\mathbf{x}_i). \tag{3}$$

A property of discrete entropy is that $H_x \geq 0$. For a uniform probability density distribution, i.e. $f(\mathbf{x}_i) = \frac{1}{n}$ for $i = 1, \cdots, n$, Eq. (3) is simply computed as

$$\widehat{H}_x = \log n.$$

According to the parts entropy concept for robotic assembly planning described in [22], for an object on the 2-dimensional plane, its positional and orientational uncertainty can be described as a joint probability distribution $f(x, y, \theta)$. In the case when x, y, and θ are statistically independent, $f(x, y, \theta) = f(x)f(y)f(\theta)$, and the part entropy for the i^{th} subsystem is given by $H_i = H_i(x) + H_i(y) + H_i(\theta)$. For n objects, the total entropy can be computed as

$$H = \sum_{i=1}^{n} H_i. \tag{4}$$

For a self-replicating system in a structured environment, each subsystem is located properly in a structured environment with some tolerance. If the pose of the i^{th} subsystem, denoted as $g_i = (x_i, y_i, \theta_i)$, is bounded by $x_i \in [0, a_i]$, $y_i \in [0, b_i]$ and $\theta_i \in [0, c_i]$, the tolerance is given by

$$\delta g_i = (a_i, b_i, c_i).$$

Assuming that the variables are statistically independent, the part entropy of this subsystem in a structured environment is

$$H_i = -\sum_{j=1}^{\alpha_i} f_j(x_i) \log_2 f_j(x_i) - \sum_{k=1}^{\beta_i} f_k(y_i) \log_2 f_k(y_i) - \sum_{l=1}^{\gamma_i} f_l(\theta_i) \log_2 f_l(\theta_i)$$

where, for some positional accuracy ϵ_p and rotational accuracy ϵ_r,

$$\alpha_i = \frac{a_i}{\epsilon_p}; \quad \beta_i = \frac{b_i}{\epsilon_p}; \quad \gamma_i = \frac{c_i}{\epsilon_r}.$$

The objects may each have different tolerance values. Thus, the tolerance for each object and the part entropy should be computed individually. The total parts entropy of n objects in a structured environment is $H = \sum_{i=1}^{n} H_i$. In general, if we do not allow overlapping of the objects, then the total parts entropy will be slightly smaller. Neglecting overlap makes the entropy computation much simpler, so we make the assumption that the reduction in entropy resulting from preventing overlap is negligible when the number of parts is small compared to the size of the environment.

Now we compute the parts entropy when all the subsystems are assembled. If the i^{th} subsystem has some tolerance when it is assembled, $\delta g'_i = (\delta x'_i, \delta y'_i, \delta \theta'_i)$, such that

$$\delta x'_i < \delta x_i; \quad \delta y'_i < \delta y_i; \quad \delta \theta'_i < \delta \theta_i$$

for all $i = 1, \cdots, n$, then the part entropy of the i^{th} object is computed by

$$H'_i = -\sum_{j=1}^{\alpha'_i} f'_j(x'_i) \log_2 f'_j(x'_i) - \sum_{k=1}^{\beta'_i} f'_k(y'_i) \log_2 f'_k(y'_i) - \sum_{l=1}^{\gamma'_i} f'_l(\theta'_i) \log_2 f'_l(\theta'_i)$$

where

$$\alpha'_i = \frac{\delta x'_i}{\epsilon_p}; \quad \beta'_i = \frac{\delta y'_i}{\epsilon_p}; \quad \gamma'_i = \frac{\delta \theta'_i}{\epsilon_r}.$$

If there is no overlapping allowed, the total parts entropy of the system after assembly process is given by $H' = \sum_{i=1}^{n} H'_i$.

We define the *task complexity* of a robotic system as the entropy reduction by the assembly process, such as

$$\Delta H = H - H'. \tag{5}$$

If a system is capable of self-replication in an unstructured environment, we compute the parts entropy for the subsystems randomly placed in a bounded area and compare it to the parts entropy when the subsystems are assembled. The entropy reduction by the structured environment is smaller and therefore the entropy reduction by the assembly process should be bigger in a less-structured environment than that in a more-structured environment.

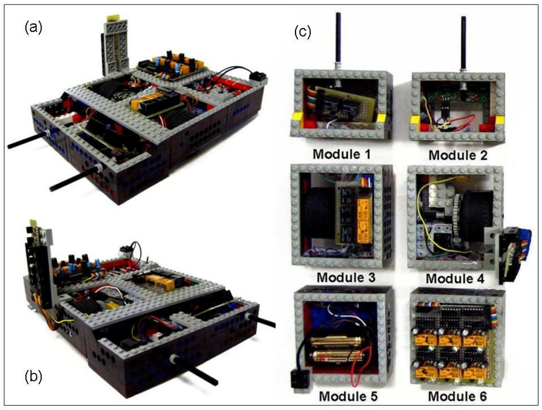

Fig. 3. A self-replicating robot: front-side views of the prototype ((a),(b)) and (c) six subsystems unassembled.

TABLE I
COMPONENTS IN EACH SUBSYSTEM

Module	Components
M_1	Line tracking circuit/metal detector
M_2	Line tracker sensor
M_3	Motor driver circuit/left motor
M_4	Barcode reader/right motor
M_5	Power supply/3 contact sensors/touch sensor
M_6	State machine/3 contact sensors

We note that ϵ_p and ϵ_r must be carefully chosen to satisfy that

$$\epsilon_p \leq \min\{\delta x, \delta y\}; \quad \epsilon_r \leq \min\{\delta\theta\}$$

where $\min\{\delta x, \delta y\}$ and $\min\{\delta\theta\}$ are the smallest value of the positional tolerance and the rotational tolerance.

III. PHYSICAL PROTOTYPE

We present a robot prototype demonstrating robotic self-replication from several prefabricated subsystems. The robot (Fig.3) consists of six subsystems. Mechanical and electrical connections among the subsystems are made through permanent magnets and spring/metal contacts. Components in each subsystem are listed in Table I. The trajectory of the robot is determined by the track and 12 landmarks (6 barcodes and 6 metal contact codes). The self-replication process of the prototype over the lapse of time is shown in Fig. 6. The robotic self-replication of this prototype can be represented by R $= \{(M_1 + \cdots + M_6)\}$, M $= \{M_1, M_2, M_3, M_4, M_5, M_6\}$ and E $=$ {tracks, 6 barcodes, 6 contact codes, wall, metal line} with a self-replication function Φ: S $\rightarrow$S$'$, where R$'$ $= \{(M_1 + \cdots + M_6), (M_1 + \cdots + M_6)\}$, M$' = \emptyset$ and E$'$ = E. Thus we have $|\text{R}| = 1$ and $|\text{R}'| = 2$.

A. Reliability Ratio

The self-replication process is composed of six similar procedures (collecting six subsystems) and each procedure has two different steps, reading a barcode and detecting contact patches. Thus the total procedure can be represented by 12 independent steps as shown in Fig. 4. In order to estimate p_m and p_b, we performed 20 trials for each metal contact sensor or barcode reader. The robot failed to read or misread the metal contact code twice and failed to detect the barcode once among each 20 trials. The experiments are made in the same structured environment for reading a specific barcode or contact code. Based on experimental data, we have $p_b = 0.95$, the probability for the robot to read the barcode correctly, and $p_m = 0.9$, the probability for the robot to read the metal contact code correctly. We assume that the probability for the same kind of sensors to read the information embedded in the environment is the same.

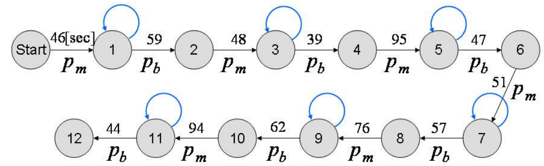

Fig. 4. Directed graph representing the self-replication process. The steps that the robot can repeat the process when it fails are indicated with blue edges. The time required for returning process is about 180 [sec].

If the robot fails in reading any barcode, it returns to the previous step after making a full round around the outer track. This processes are marked with blue directed loops in Fig. 4. The time required to return to the previous step is $t_r = 180$[sec]. If the robot replicates without missing any barcode, then the time required for this process is $t_s = 718$[sec]. Each step in the graph is independent each other, therefore, the *reliability ratio* of the system for given $t = t_s$ is computed as

$$K_s|_{t_s} = (p_m)^6 \cdot (p_b)^6 = 0.3907.$$

The reliability ratio in general increases over time. If the given time is $t = t_s + t_r$, the success ratio is increased because if the robot fails to read one of the barcodes, it is allowed to come back and try to read it once again. There the reliability ratio is computed as

$$K_s|_t = K_s|_{t_s} \left\{1 + \left\langle \begin{array}{c} 6 \\ 1 \end{array} \right\rangle \overline{p_b} \right\},$$

where $\left\langle \begin{array}{c} n \\ k \end{array} \right\rangle$ is the multiset coefficient given by

$$\left\langle \begin{array}{c} n \\ k \end{array} \right\rangle = \frac{(n+k-1)\cdots(n)}{k!}.$$

For $t = t_s + n \cdot t_r$,

$$K_s|_t = K_s|_{t_s} \sum_{i=0}^{n} \left\langle \begin{array}{c} 6 \\ i \end{array} \right\rangle \overline{p_b}^i$$

and for given $t = \infty$, we can simply calculate the reliability ratio as

$$K_s|_\infty = (p_m)^6 \cdot 1^6 = 0.5314.$$

Fig. 5 shows a graph of K_s vs. t. K_s increases for given more time t, and converges to $K_s|_\infty$ for $t \geq 1258$[sec].

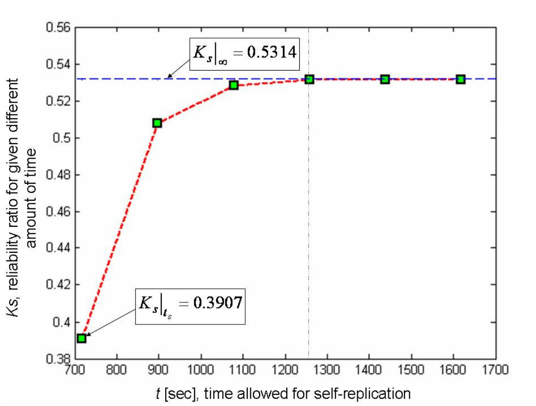

Fig. 5. The green square boxes indicate $K_s|_t$ for t=718:180:2338[sec]. The red line (connecting the values of K_s) converges to the maximum reliability ratio, $K_s|_\infty$, for a given time $t \geq$1258[sec].

B. Task Complexity

Now we compute the task complexity of the system. Subsystems are not identical in size and shape, and therefore the tolerance for each subsystem is obtained from experimental observation. The tolerance of each subsystem in the structured environment is given by

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

where $\delta g_i = (x_i[\mathrm{mm}]^3, \mathrm{y_i}[\mathrm{mm}], \theta_i[\mathrm{radian}])$. Each subsystem is assumed to have a uniform probability distribution within the tolerance. In addition to possible positions and orientations within the tolerance, there are 6! possible permutations to place subsystems in the environment. As shown in the first picture in Fig. 6, six subsystems are initially located in six spots in the environment. Therefore, the entropy resulting from the permutations must be included in the parts entropy.

For $\epsilon_p = 0.5$ and $\epsilon_\theta = 0.01$, the total parts entropy of the system in a structured environment is computed as

$$\widehat{H} = \sum_{i=1}^{6} \widehat{H}_i + \widehat{H}^{perm} \simeq 102.97.$$

The assembled subsystems have a fairly small tolerance of $\delta g_i' = (1, 1, 0.01)$ for all $i = 1, \cdots, 6$, then the parts entropy for an assembled subsystem is $\widehat{H}_i' = \log_2 2 + \log_2 2 + \log_2 1 = 2$ for a uniform distribution. The total parts entropy for the assembled subsystems is $\widehat{H}' = 2 \times 6 = 12$, and therefore *the task complexity* of the prototype, is given by

$$\Delta H = \widehat{H} - \widehat{H}' \simeq 90.97.$$

C. Discussion

The reliability ratio can be applied to any modular self-replicating or assembly system if the self-replication process can be represented by a directed graph and the probability of each step can be computed or given by experiments. If the self-replication procedure is not composition of independent events, then the computation for the reliability ratio would be much more complicated than the prototype presented.

We computed the task complexity under the assumption that the probability density functions (pdfs) are uniform within the tolerance and there is no overlapping among subsystems. In the prototype, initial locations for six subsystems are far from each other, so the assumption of no overlapping is true in this case. In general, however, pdfs may not be uniform and some entropy reduction may occur by prohibiting overlapping among subsystems.

As an example of other self-replicating systems, we consider the Penrose block replicator [3] composed of two simple blocks A and B with certain mechanical properties. The initial system (A+B) in a box containing A's and B's replicate copies of itself with random agitation applied to the box. We can draw a simple graph with two vertices named 'start' and 'end'. K_s can be obtained by experiments, such that for varying t,

$$K_s|_t = \frac{|\text{successful self-replication in time } t|}{|\text{total trials}|}.$$

We assume that the size of the box containing parts is given by $x \in [0, 1000]$. For $\epsilon = 0.5$ and for a uniform distribution, the parts entropy for two blocks is given by $H = 2\log_2 1000 \simeq 21.93$. If the parts entropy when two blocks are assembled is simply zero, then $\Delta H \simeq 21.93$.

One of our previous prototypes [21] is composed of five subsystems with an LEGO$^{\mathrm{TM}}$ RCX controller in one of subsystems. For the same ϵ_p and ϵ_r, the initial parts entropy before self-replication is about 50.80 and that after self-replication is 20. Therefore, the task complexity of the RCX system is computed as $\Delta H = 30.80$. Without any data obtained by experiments on this prototype, we cannot compute K_c for this system. Only from the provided information in [21], the process collecting one subsystem is made by (reading barcode indicating each subsystem), (reading barcode leading the robot to the center of the track), (reading barcode indicating the location to release the collected subsystem). The robot repeats this process until it assembles all four subsystems (except of one subsystem originally placed at the center where the replica is made). Therefore, if the probability of the robot to read a barcode is given by experiments, then we can estimate the overall success ratio (the reliability ratio) of the system over time.

IV. Conclusion

Measures of reliability and task complexity were presented and applied to a physical prototype consisting of six prefabricated subsystems. The prototype autonomously replicates a copy of itself by assembling six subsystems located in a partially structured environment. Although the environmental structures hold important information about subsystem locations and the robot's trajectory, all actions associated with the self-replication process are made solely by the initial robot. Much work remains in developing systems capable of self-replication in a less-structured or unstructured environment.

[3] [mm]=[millimeter]

Such systems would increase adaptability to the environment, as well as overall system robustness.

REFERENCES

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

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

[3] L.S. Penrose, "Mechanics of self-reproduction," *Ann. Human Genetics*, 23: 59-72, Nov. 1958

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

[5] R.A. Freitas, Jr., and W.P. Gilbereath, (Eds.) "Advanced automation for space missions," *Proceedings of the 1980 NASA/ASEE Summer Study, Chapter 5 (Conference Publication 2255)*, Washington DC, US government printing office, 1982.

[6] V.M. Chadeev,"Allowance for robot self-replication in industrial automation," *Automation and Remote Control*, vol.61, 2000, pp.1752-1757.

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

[8] Chirikjian, G.S., Zhou, Y., Suthakorn, J., "Self-Replicating Robots for Lunar Development," *IEEE/ASME Transactions on Mechatronics*, Vol. 7, No. 4, Dec. 2002., pp. 462-472.

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

[10] K. Tomita, S. Murata, H. Kurakawa, E. Yoshida and S. Kokaji, "Self-assembly and self-repair method for a distributed mechanical system," *IEEE Transaction on Robotics and Automation*, 15:1035-1045, 1999.

[11] H. Kurokawa, K. Tomita, A. Kamimura, S. Murata and S. Kokaji,"Distributed Self-reconfiguration Control of an M-TRAN System," *Robitcs: Science and Systems Workshop on Self-reconfigurable Modular Robots*, Philadelphia, August, 2006.

[12] M. Yim, D. Duff, and K. Rufas, "PolyBot: A modular reconfigurable robot," *IEEE Intl. conf. on Robotics and Automation*, vol.1, 2000, pp.514-520.

[13] M. Park, S. Chitta, M. Yim,"Isomorphic Gait Execution in Homogeneous Modular Robots," *Robitcs: Science and Systems Workshop on Self-reconfigurable Modular Robots*, Philadelphia, August, 2006.

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

[15] Adams, B., Lipson, H., "A universal framework for self-replication", European Conference on Artificial Life, ECAL03, September 2003, Dortmund Germany, pp. 1-9, 2003.

[16] Z. Bulter, S. Murata and D. Rus, "Distributed replication algorithm for self-reconfiguring modular robots," *Proceedings of 6th International Symposium on Distributed Autanomous Robotic Systems (DARS '02)*, Fukuda, Japan, June 2002, pp.25-27.

[17] P. Varshavskaya, L.P. Kaelbling and D. Rus, "On Scalability Issues in Reinforcement Learning for Self-Reconfiguring Modular Robots," *Robitcs: Science and Systems Workshop on Self-reconfigurable Modular Robots*, Philadelphia, August, 2006.

[18] G.S. Chirikjian and J. Suthakorn,"Toward self-replicating robots," *Proceedings of the Eight International Symposium on Experimental Robotics(ISER)*, Italy, July 2002.

[19] J. Suthakorn, Y. Kwan and G.S. Chirikjian, "A semi-autonomous replicating robotic system," *Proceedings of 2003 IEEE International Conference on Intelligent Robots and Applicatins (CIRA)*, Kobe, Japan, July 2003,pp.776-781.

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

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

[22] A.C. Sanderson, "Part Entropy Mothod for Robotic Assembly Design," *Proceedings of International Conference on Robotics, 1984.*

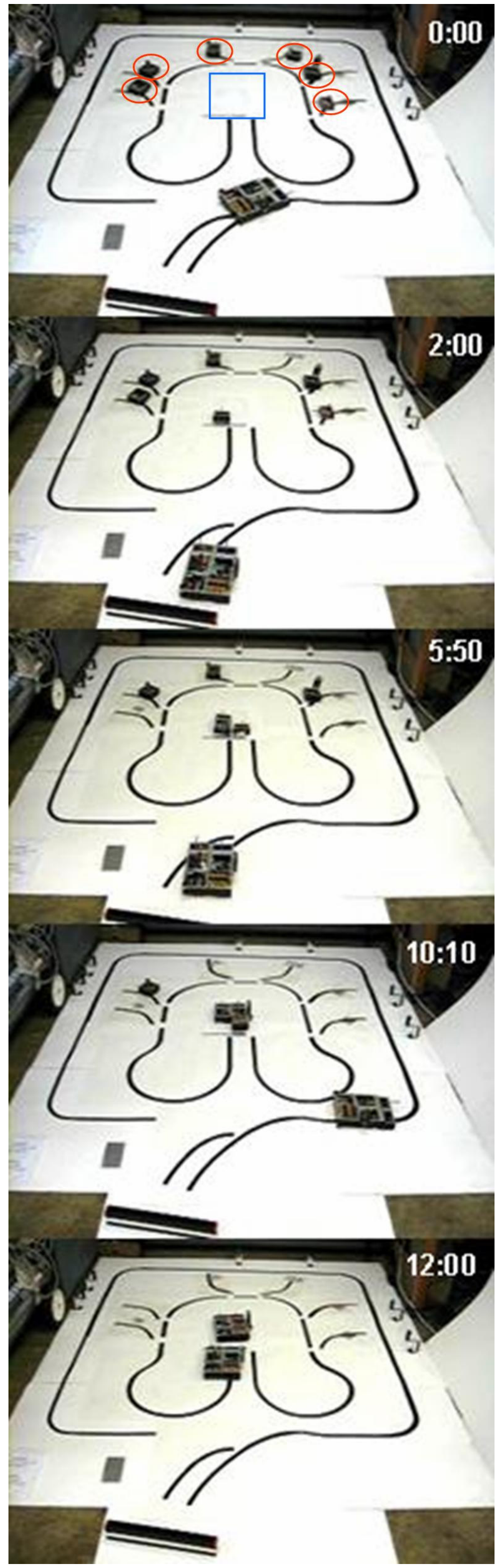

Fig. 6. Self-replication process: the red circles indicate initial locations of six subsystems and the blue rectangle is the station where the replica will be assembled.