The International Journal of Robotics Research

http://ijr.sagepub.com

Robotic Self-replication in Structured Environments: Physical Demonstrations and Complexity Measures

Kiju Lee, Matt Moses and Gregory S. Chirikjian

The International Journal of Robotics Research 2008; 27; 387
DOI: 10.1177/0278364907084982

The online version of this article can be found at:
http://ijr.sagepub.com/cgi/content/abstract/27/3-4/387

Published by:
SAGE Publications
http://www.sagepublications.com

On behalf of:

Multimedia Archives

Additional services and information for *The International Journal of Robotics Research* can be found at:

Email Alerts: http://ijr.sagepub.com/cgi/alerts

Subscriptions: http://ijr.sagepub.com/subscriptions

Reprints: http://www.sagepub.com/journalsReprints.nav

Permissions: http://www.sagepub.com/journalsPermissions.nav

Downloaded from http://ijr.sagepub.com by Kiju Lee on July 1, 2008
© 2008 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

Kiju Lee
Matt Moses
Gregory S. Chirikjian
Department of Mechanical Engineering
Johns Hopkins University
Baltimore, MD 21218, USA
{kiju, matt.moses, gregc}@jhu.edu

Robotic Self-replication in Structured Environments: Physical Demonstrations and Complexity Measures

Abstract

In this paper we define a complexity ratio that measures the degree to which a robot is self-replicating based on the number and complexity of subsystems that it can assemble to form a functional replica. We also quantify how structured the environment must be in order for a robot to function. This calculation uses Sanderson's concept of parts entropy. Together, the complexity measure of the robot and environmental entropy provide quantitative benchmarks to assess the state of the art in the subfield of self-replicating robotic systems, and provide goals for the design of future systems. We demonstrate these principles with three prototype systems that show different degrees of robotic self-replication. The first robot is controlled by a microprocessor and consists of five subsystems. The second has no microprocessor and is implemented as a finite-state machine consisting of discrete logic chips that are distributed over five subsystems. The third design consists of six subsystems and is able to handle greater environmental entropy. These systems demonstrate the desired progression towards self-replicating robots consisting of greater numbers of subsystems, each of lower complexity, and which are able to function in environments with increasing levels of disorder.

KEY WORDS—robotic self-replication, self-assembly, modular robots, distributed robotic systems

1. Introduction

Robotics research is often influenced by observing and mimicking the attributes of biological systems. Self-replication at the cellular level is one of the most fundamental features of all living creatures. A long-term goal of our research group is to develop robots that self-replicate in unstructured environments from the most basic building blocks. Progression towards the realization of this goal can be measured by assessing the complexity of the input parts (relative to the complexity of the whole robot) and quantifying the amount of structure in the environment required for the robot to function. For example, a trivial self-replication process might consist of a robot that nudges two complex prefabricated halves of the replica to move towards each other. If the halves of the replica are constrained to move on a physical track, the process is even more trivial. In contrast, a less-trivial (more desirable) process would involve assembling many simple components randomly situated in an open space.

This paper describes three self-replicating robots (SRRs) developed in our lab. These systems demonstrate a progression from "more trivial" to "less trivial" in that the newer systems perform the same (or more complex) tasks as the older ones, but do so with much simpler components. Later in the paper we develop quantitative measures to assess "how self-replicating" a robot is based on the ratio of complexity of the input parts to the complexity of the robot, and based on the amount of configurational entropy the robot reduces when making a replica. However, first we review the relevant literature.

1.1. Related Work

The first theoretical work on self-reproducing machines was done by John von Neumann in the 1950s, and this work is summarized in von Neumann and Burks (1962). His concepts on self-replicating systems have been applied in many research areas such as cellular automata, nanotechnology, macromolecular chemistry and computer simulations (Sipper 1998; Freitas and Merkle 2004). The earliest physical replicating machines were presented by Jacobson (1958) and Penrose (1959). In

The International Journal of Robotics Research
Vol. 27, No. 3–4, March/April 2008, pp. 387–401
DOI: 10.1177/0278364907084982
©SAGE Publications 2008 Los Angeles, London, New Delhi and Singapore
Figures 1–6, 8–14, 16 appear in color online: http://ijr.sagepub.com

Downloaded from http://ijr.sagepub.com by Kiju Lee on July 1, 2008
© 2008 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

the work of Penrose (1959), simple units self-reproduced via the assembly of passive mechanical elements under external vibration. The system of Jacobson (1958) used electric carts operating on an oval track instead of mechanical components confined in a box. In the 1980s, NASA performed a feasibility study on SRRs as a potential means for space development and exploration (Freitas and Gilbereath 1982). Other studies have been performed more recently with the long-term goal of self-replicating factories on the Moon (Chirikjian et al. 2002) and large-scale industrial installations on Earth (Lackner and Wendt 1995; Chadeev 2000; Friedman 2002).

Recent experimental work related to robotic replication falls loosely into four categories: directed replication via module assembly (Suthakorn et al. 2003a,b; Hastings et al. 2004; Park et al. 2004; Zykov et al. 2005), directed replication via fabrication (Lipson 2005; Bowyer 2006), self-reconfigurable modular robots (Murata et al. 1994; Yim et al. 2001; Shen et al. 2003; Yoshida et al. 2003) and self-assembly of randomly agitated modules (White et al. 2004; Griffith et al. 2005; Napp et al. 2006; Zykov and Lipson 2006). In directed replication via module assembly, an initial robot composed of modules assembles a duplicate from unconnected modules by executing some directed sequence of movements. Modules can be homogeneous or heterogeneous, but they need to be presented to the initial robot in some specific, fixed configuration. The robot can be a specialized mechanism (Hastings et al. 2004), a manipulator arm (Zykov and Lipson 2006) or, like the robots presented in this paper, a mobile manipulator (Suthakorn et al. 2003a,b; Park et al. 2004).

Investigations on directed replication via fabrication focus on how to produce components from raw materials (Bowyer 2006; Lipson 2005). At this stage of development these systems resemble low-cost rapid-prototyping tools. We envision that at some point these fabrication systems may be combined with modular-robotic assembly systems into “complete” machine self-replicators that can produce modules from raw materials and assemble them into complex devices.

The modules used in current SRRs play the same role as those used in modular reconfigurable robotics (Murata et al. 1994; Yoshida et al. 2003; Shen et al. 2003). Robots labeled as “self-replicating” do many of the same things as robots labeled “self-reconfigurable” and *vice versa*. For purposes of classification, we use the term “replicating” if a system is designed specifically for replication and “reconfigurable” if it is designed primarily for reconfiguration.

In self-assembly from random agitation, the systems typically use homogeneous electromechanical modules in a low-friction environment (either on an air-table for two-dimensional systems or suspended in a liquid medium for three-dimensional systems). Modules are not self-mobile, but instead are kept in constant motion by an external work source, such as oscillating fans. The formation of assemblies is induced by controlling the affinity each module has for making and breaking connections on each of its faces, through the use of mechanical latches (Friedman 2002), actuated permanent magnets (Napp et al. 2006), electromagnets (White et al. 2004) or fluidic valves (Zykov and Lipson 2006).

Depending on the task at hand, each approach listed has certain advantages. Self-assembly from random agitation is inconvenient for macro-scale modules because of the need to reduce friction and provide an agitating force, however this method would work well for systems composed of small (micrometer to millimeter) sized modules. Self-reconfigurable modular robots are suited to tasks requiring versatility and fault tolerance, but require a source of modules. Directed self-replication provides an efficient way to create modules and assemblies, and in our view replication via module assembly and via fabrication are really two sides of the same problem.

Our lab has built several prototypes in order to develop and demonstrate the concept of robotic replication. Some of these were self-replicating while others required actions from external agents. As the first step, remote-controlled and semi-autonomous replicating robots were presented by Chirikjian and Suthakorn (2002) and Suthakorn et al. (2003b). These systems require external control (by human) for replication. As the second step, autonomous self-replicating systems were developed (Suthakorn et al. 2003a; Park et al. 2004). The system described by Suthakorn et al. (2003a) is able to retrieve subsystems and assemble them in a structured environment. The structured environment includes tracks on a flat surface and metal foil to trigger a gripper on the robot. The robot trajectory is determined by line tracking. Park et al. (2004) added bar code labels to the track design, which enables the robot to distinguish between subsystems by reading bar codes on each location where a subsystem is placed. The robots in these projects were microprocessor-based systems consisting of several prefabricated subsystems. A self-replicating, electromechanical circuit was presented by Hastings et al. (2004). The circuit uses an electromechanical device as a substrate in order to construct functional copies of itself.

In this paper we present three SRRs. These systems represent improvements over our older work because they function in less structured environments and use simpler components. In Section 2, we discuss quantitative measures of complexity and configuration entropy that are of use in analyzing the self-replication process. In Section 3, we apply these measures to a microprocessor-based robot, SRR I (Park et al. 2004), a self-replicating robotic system with distributed electronics, SRR II (Eno et al. 2007), and SRR III (Lee and Chirikjian 2007), an advanced self-replicating system with distributed electronics.

2. Complexity Measures for Self-replicating Systems

We all have intuitive notions about “how self-replicating” a proposed SRR is. A robot that builds a duplicate from many

Downloaded from http://ijr.sagepub.com by Kiju Lee on July 1, 2008
© 2008 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

simple parts seems "more self-replicating" (or "more powerful") than one that builds a duplicate from a few complex modules. Similarly, a robot requiring simple landmarks or none at all is better than a robot that requires complex landmarks or tracks to constrain its motion; and a robot that can assemble modules with random initial configurations is better than a robot that requires modules in a carefully arranged initial configuration. We introduce a simple measure of complexity that can be applied to a modular SRR to describe the intuitive notion of "many simple parts" versus "few complex parts". We also discuss the use of information entropy for quantifying the tolerable "randomness" in the initial configuration of modules. Whereas our focus in this paper is on self-replicating systems, the complexity measures introduced here can also be applied to self-assembling and other systems.

2.1. Descriptive Framework

A descriptive framework for physical replicating systems was presented by Lee and Chirikjian (2007). In this framework, a robotic replicating system consists of three sets of components, M (a multiset of available parts for building replicas), R (a multiset describing the initial functional system to be reproduced by the replication process) and E (a multiset of environmental structures involved in the replication process as catalytic elements, but not replicated). The replication process can be represented by

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

such that obviously $|\mathrm{R}| < |\mathrm{R}'|$ where $|\cdot|$ denotes the number of elements. In order to be a self-replicating system, R should not be the empty set and all necessary machinery or manipulation related to the replicating process must be done by R. If E contains an agent that actively controls the replication process, then we say R replicates, but R does not *self*-replicate. According to the order of subsystem assembly, M can be either a strictly ordered, partially ordered or unordered set. For instance, if M is a set of identical modules, then it is an unordered set. An environment can be categorized into one of three classes: a completely structured environment, a partially structured environment or an unstructured environment. If E is a completely structured environment, then no change or modification is allowed in the structures. If some or all environmental structures can be moved or permuted without affecting the replication process (i.e. the robot can still successfully replicate when the structures are relocated), then E is called a partially structured environment. When $\mathrm{E} = \emptyset$, the empty set, then it is called an unstructured environment. As our prototypes are mobile robots moving in two-dimensional space, an unstructured environment is defined as a two-dimensional flat, bounded surface without any obstacles or landmarks in the environment. Depending on the physical details of a system, a

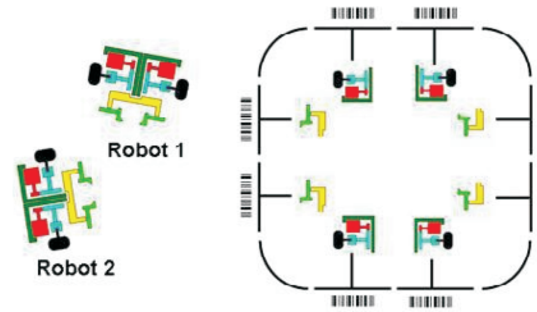

Fig. 1. (R, M, E), a system before the self-replication process

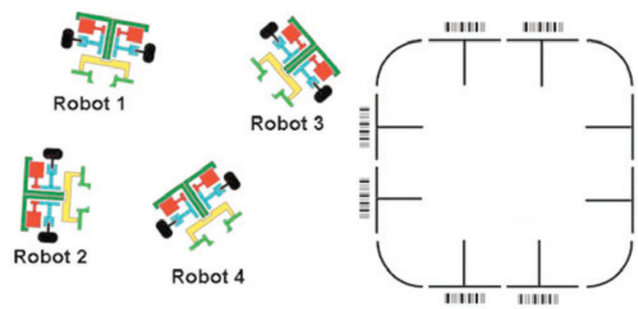

Fig. 2. $(\mathrm{R}', \mathrm{M}', \mathrm{E}')$ a system after the self-replication process.

corresponding unstructured environment can be defined differently. For example, for robots free to move in three dimensions (Zykov and Lipson 2006), a bounded three-dimensional volume may be viewed as an unstructured environment.

Figures 1 and 2 show an example of a self-replicating system with fixed environmental structures. The robot consists of four subsystems, a left gripper (g_l), a right gripper (g_r) a left wheel (w_l) and a right wheel (w_r). The structured environment includes eight bar codes ($b_1, \ldots, b_8$), eight T-shaped tracks ($t_1, \ldots, t_8$) and four curved tracks ($a_1, \ldots, a_4$). There are two initial functional robots and eight subsystems (two for each kind of subsystem) to be assembled. Then, the self-replication process can be described in our framework as

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

such that

$$\begin{aligned} \mathrm{R} &= \{r, r\} \rightarrow \mathrm{R}' = \{r, r, r, r\}; \\ \mathrm{M} &= \{g, g, w_l, w_l, w_r, w_r\} \rightarrow \mathrm{M}' = \emptyset; \\ \mathrm{E} &= \{b_1, \ldots, b_6, t_1, \ldots, t_8, a_1, \ldots, a_4\} \rightarrow \mathrm{E}' = \mathrm{E}. \end{aligned}$$

Here, there is no external source providing subsystems. For another cycle of replication, there must be an external supply of subsystems. In this example and all that follow, the power

Downloaded from http://ijr.sagepub.com by Kiju Lee on July 1, 2008
© 2008 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

supply is considered to be a replenishable resource. Otherwise, this too could be considered as a quantity that diminishes during the replication process. If the tracks and bar codes are replicated by the self-replication process, then R can be redefined including these structures in addition to the functional robots, while E becomes the empty set.

2.2. The Degree of Self-replication

As a simple measure of system complexity, we count the number of "active elements" for each subsystem and the number of "interconnections" between subsystems when they are assembled (Eno et al. 2007; Liu et al. 2007). 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: chassis, gear, shaft, switch, coil, transistor, capacitor, etc. Logic gates can count as more than one active element, depending on how many transistors they contain. For example, we count an AND gate as two active elements, OR as two, NOT as one and NOR as two. The complexity of a large logic circuit is measured as the equivalent number of transistors. A brushed motor is counted as four active elements (coil, rotor, brushes, magnet). When subsystems are assembled forming a functional robot, new electrical and mechanical connections are made. These are called interconnections. In addition to counting the active electrical/mechanical parts in each subsystem, each interconnection between subsystems is counted as one active element.

In order for a self-replicating robotic system to be more powerful, the complexity of the fully assembled system must be far greater than that of individual parts before assembly. Also, if most of the complexity is concentrated on one subsystem and the rest are relatively simple parts, then this is viewed as more trivial than a case where complexity is distributed evenly throughout all subsystems. If the ith of n modules has complexity C_i, we define the *degree of self-replication* as (Lee and Chirikjian 2007):

$$D_{\mathrm{s}} = \frac{C_{\min}}{C_{\max}} \cdot \frac{C_{\mathrm{total}}}{C_{\mathrm{ave}}} \cdot \frac{1}{C_{\mathrm{ave}}}, \tag{1}$$

where $C_{\mathrm{ave}} = 1/n \sum_{i=1}^{n} C_i$, and $C_{\max}$ and $C_{\min}$ are the maximum and minimum subsystem complexity among $(C_1, \ldots, C_n)$, respectively. When we count the number of interconnections made by assembly, then the total complexity is computed as

$$C_{\mathrm{total}} = \sum_{l=1}^{n} C_l + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} I_{ij},$$

where I_{ij} is the number of interconnections between the ith and the jth subsystems. If interconnections are not counted, $C_{\mathrm{total}} = n \cdot C_{\mathrm{ave}}$ and therefore in this case $D_{\mathrm{s}} = C_{\min}/C_{\max} \cdot n/C_{\mathrm{ave}}$. The first term on the right-hand side of (1) indicates 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. We claim that a more powerful self-replicating system has higher value of D_{s}.

If a system consists of n identical modules, D_{s} is given by

$$D_{\mathrm{s}} = \frac{n\tilde{C} + I}{\tilde{C}^2}, \tag{2}$$

where $\tilde{C}$ is the module complexity and I is the total number of interconnections among n modules. As another example, if the subsystem complexities are given by $C_1 = \cdots = C_m = Q$ and $C_{m+1} = \cdots = C_n = q$ for some $m < n$ where $Q > q$, then

$$D_{\mathrm{s}} = \frac{qn^2(mQ + (n-m)q + I)}{Q(mQ + (n-m)q)^2}. \tag{3}$$

If both systems share the same average complexity and the number of interconnections, such that $\tilde{C} = (mQ + (n-m)q)/n$, then (3) is always less than (2). In other words, among all systems composed of n subsystems with the same C_{ave} and C_{total}, a system with a uniform complexity distribution has the highest degree of self-replication. If the complexity is concentrated in one of the subsystems and each of the rest has only one active element, then D_{s} is a minimum for this system. This relation is written as

$$\frac{C_{\mathrm{total}}}{C_{\mathrm{ave}}^2 (C_{\mathrm{total}} - n + 1)} \leq D_{\mathrm{s}} \leq \frac{C_{\mathrm{total}}}{C_{\mathrm{ave}}^2}.$$

2.3. Entropy

Entropy is a useful statistical measure that can describe the sophistication of tasks to be addressed by a modular reconfigurable system or a replicating (including self-replicating) system. The concepts of discrete and continuous entropy from information theory are different from each other, and continuous information-theoretic entropy is not exactly the same as statistical mechanical/thermodynamic entropy. The discrete entropy depends on whether modules are labeled (distinguishable) or not, and is a measure of how complicated the space of all possible arrangements of parts is. In this paper, we only use discrete entropy. Given a discrete space consisting of points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and a discrete probability distribution $f_i = f(\mathbf{x}_i)$, such that $\sum_i f_i = 1$ and all $f_i \geq 0$, the corresponding discrete entropy is defined as

$$H_f = -\sum_{\mathbf{x}_i} f(\mathbf{x}_i) \log_2 f(\mathbf{x}_i). \tag{4}$$

Downloaded from http://ijr.sagepub.com by Kiju Lee on July 1, 2008
© 2008 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

A property of discrete entropy is that $H_f \geq 0$.

We adapt the *parts entropy* method (Sanderson 1984) to compute the entropy of objects located in a bounded area. For an object in a two-dimensional space, 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 of each other,

$$f(x, y, \theta) = f_x(x) f_y(y) f_\theta(\theta)$$

and the corresponding parts entropy is given by

For n parts, if we assume that there is no penalty for overlaps between parts, then the total entropy can be computed as

$$H = \sum_{i=1}^{n} H_i, \tag{5}$$

where H_i denotes the parts entropy of the ith part. When we do not allow overlapping among the objects, the total parts entropy will be slightly smaller. Neglecting overlaps makes the entropy computation much simpler, so we make the assumption that the reduction in entropy resulting from preventing overlap is negligible when the total footprint of all parts is very small compared with the area of the environment.

In a robotic self-replication or self-assembly process, the total entropy reduction is made through two steps: (i) a human structuring the environment and (ii) the assembly process performed by the robot. A structured environment can play an important role, such as storing information about subsystem locations and guiding the robot's trajectory. Positional and rotational uncertainty in subsystem locations can be reduced by a properly designed environment. The entropy reduction resulting from structuring the environment can be computed by comparing the parts entropy of the system, when we assume that the subsystems are randomly located in an unstructured environment, to that when the subsystems are placed in a structured environment within some tolerance values. If the parts entropy in an unstructured environment is denoted by H^{u} and the parts entropy in a structured environment is denoted by H^{s}, the entropy reduction by a structured environment is defined by

$$\Delta H_{\mathrm{E}} = H^{\mathrm{u}} - H^{\mathrm{s}}. \tag{6}$$

To compute the entropy reduction by the robot, we compare the parts entropy of the system in a structured environment before and after the assembly process. The difference between the two is the amount of entropy reduction by the assembly process performed by the robot. When discrete entropy is used, the parts entropy after the assembly process will be close to zero. Positional and orientational uncertainty of components in the assembled robot, due to component tolerances, can result in a small non-zero entropy for the assembled robot. Let H^{a} be the parts entropy when all subsystems are assembled. Then the entropy reduction made by an assembly process of the functional robot is defined by

$$\Delta H_{\mathrm{R}} = H^{\mathrm{s}} - H^{\mathrm{a}}. \tag{7}$$

In the case where the assembly process is partially made by an external agent, i.e. the system can replicate, but does not self-replicate, ΔH_{R} is *reduced* by the amount of entropy change resulting from the external manipulations on the subsystems. This amount is *added* to ΔH_{E} because manipulations by an external agent are counted as an effect of the environmental structure. The quantity ΔH_{E} is one way to measure the complexity of the environment itself. We have investigated other measures of environment complexity based on graph theory in Liu et al. (2007).

For a self-replicating system consisting of n subsystems in a structured environment, H^{u}, H^{s} and H^{a} can be computed as follows. We assume that an unstructured environment is a flat, bounded area with no structures or landmarks inside the boundary. Let the pose of the ith subsystem in the environment be parameterized by three coordinates, $g_i^{\mathrm{u}} = (x, y, \theta)$. Each coordinate is bounded on an interval and discretized with uniform spacing such that $x \in \{x_j\}$ ($j = 1, \ldots, \alpha_i^{\mathrm{u}}$) on $[0, a]$, $y \in \{y_k\}$ ($k = 1, \ldots, \beta_i^{\mathrm{u}}$) on $[0, b]$, and $\theta \in \{\theta_m\}$ ($m = 1, \ldots, \gamma_i^{\mathrm{u}}$) on $[0, 2\pi]$. For convenience we write this as $x \in \mathcal{X}_i^{\mathrm{u}}$, $y \in \mathcal{Y}_i^{\mathrm{u}}$ and $\theta \in \mathcal{Z}_i^{\mathrm{u}}$. The number of discrete values for each coordinate is given by

$$\alpha_i^{\mathrm{u}} = \frac{a}{\epsilon_{\mathrm{p}}}; \quad \beta_i^{\mathrm{u}} = \frac{b}{\epsilon_{\mathrm{p}}}; \quad \gamma_i^{\mathrm{u}} = \frac{2\pi}{\epsilon_{\mathrm{r}}},$$

where ϵ_{p} is the positional resolution and ϵ_{r} is the rotational resolution. The parts entropy of the ith subsystem in an unstructured environment is then computed as the joint entropy of the pose coordinates

$$H_i^{\mathrm{u}} = -\sum_{x \in \mathcal{X}_i^{\mathrm{u}}} \sum_{y \in \mathcal{Y}_i^{\mathrm{u}}} \sum_{\theta \in \mathcal{Z}_i^{\mathrm{u}}} f_i^{\mathrm{u}}(x, y, \theta) \log_2 f_i^{\mathrm{u}}(x, y, \theta),$$

where $f_i^{\mathrm{u}}(x, y, \theta)$ is the joint probability distribution of the ith subsystem having a given initial pose. We assume that all initial poses are equally likely, so f_i^{u} is uniform over all coordinates. In addition, we assume that x, y and θ are statistically independent, so that $f(x, y, \theta) = f_x(x) f_y(y) f_\theta(\theta)$. In this case $f_x(x) = 1/\alpha_i^{\mathrm{u}}$ for all $x \in \mathcal{X}_i^{\mathrm{u}}$ and $f_x(x) = 0$ everywhere else. Likewise, $f_y(y) = 1/\beta_i^{\mathrm{u}}$ and $f_\theta(\theta) = 1/\gamma_i^{\mathrm{u}}$ for all $y \in \mathcal{Y}_i^{\mathrm{u}}$ and $\theta \in \mathcal{Z}_i^{\mathrm{u}}$ and zero everywhere else. Given these assumptions, the parts entropy of the ith subsystem simplifies to

$$\widehat{H}_i^{\mathrm{u}} = \log_2 \alpha_i^{\mathrm{u}} + \log_2 \beta_i^{\mathrm{u}} + \log_2 \gamma_i^{\mathrm{u}}.$$

If we assume that the probability distributions across subsystems are statistically independent, the joint parts entropy of n subsystems in an unstructured environment, denoted by H^{u}, can be found by summing H_i^{u} for all $i = 1, \ldots, n$ as (5). As mentioned earlier, this is not an entirely realistic way to model the system because it allows poses that would result in collisions between subsystems in a physical system.

Downloaded from http://ijr.sagepub.com by Kiju Lee on July 1, 2008
© 2008 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

However, we feel it is reasonable to assume that the discrepancy due to allowing overlap is negligible when the environment is large compared with the area occupied by the subsystems.

We next consider the scenario in which there are some landmarks in the environment, that the robot knows the locations of all landmarks and the subsystems are carefully placed in certain locations relative to the landmarks within some tolerance. The effect of the landmarks is to constrain the available initial poses to some subset of those available in an unstructured environment. By reducing the number of available poses, the parts entropy is reduced. As before, each coordinate is bounded on an interval and discretized, only now the intervals represent a much smaller region surrounding the local vicinity of the landmark. We now have $x \in \{x_j\}$ $(j = 1, \ldots, \alpha_i^s)$ on $[a_1, a_2]$, $y \in \{y_k\}$ $(k = 1, \ldots, \beta_i^s)$ on $[b_1, b_2]$ and $\theta \in \{\theta_m\}$ $(m = 1, \ldots, \gamma_i^s)$ on $[\theta_1, \theta_2]$. Shorthand notation for this is $x \in \mathcal{X}_i^s$, $y \in \mathcal{Y}_i^s$ and $\theta \in \mathcal{Z}_i^s$. The reduced number of discrete values for each coordinate in the vicinity of the landmark is given by

$$\alpha_i^s = \frac{a_2 - a_1}{\epsilon_p}; \quad \beta_i^s = \frac{b_2 - b_1}{\epsilon_p}; \quad \gamma_i^s = \frac{\theta_2 - \theta_1}{\epsilon_r}.$$

We assume that within the pose tolerance of the landmark, every available pose is equally likely. Following the same reasoning as above, the structured parts entropy for the ith subsystem is given by

$$\widehat{H}_i^s = \log_2 \alpha_i^s + \log_2 \beta_i^s + \log_2 \gamma_i^s.$$

We now have the relations $(a_2 - a_1) < a$, $(b_2 - b_1) < b$ and $(\theta_2 - \theta_1) < 2\pi$. The subsystems may each have different tolerance values. Hence, $\mathcal{X}_i^s$, $\mathcal{Y}_i^s$, $\mathcal{Z}_i^s$ and the corresponding H_i^s should be obtained individually for each i. If parts are identical to each other, then they may have the same tolerances and parts entropy. We again assume that probability distributions across subsystems are independent, i.e. that there is no overlapping among the subsystems, and that the joint parts entropy of n objects in a structured environment, denoted by H^s, is obtained by $H^s = \sum_{i=1}^n H_i^s$. Note that in this case it is entirely reasonable to make this assumption because landmarks would not be intentionally located in a structured environment in such a way as to cause overlap between subsystems.

When subsystems are assembled, the available poses for each subsystem are reduced even further. In a perfect assembly process, each subsystem would be constrained to exactly one pose and the parts entropy would be zero. However, we allow multiple poses to account for small variations even when all subsystems are assembled. In this case the coordinates are bounded on intervals that represent poses within the allowable assembly tolerances. The available coordinates are $x \in \{x_j\}$ $(j = 1, \ldots, \alpha_i^a)$ on $[a_3, a_4]$, $y \in \{y_k\}$ $(k = 1, \ldots, \beta_i^a)$ on $[b_3, b_4]$ and $\theta \in \{\theta_m\}$ $(m = 1, \ldots, \gamma_i^a)$ on $[\theta_3, \theta_4]$. Shorthand notation is $x \in \mathcal{X}_i^a$, $y \in \mathcal{Y}_i^a$ and $\theta \in \mathcal{Z}_i^a$. The number of discrete values for the coordinates is given by

$$\alpha_i^a = \frac{a_4 - a_3}{\epsilon_p}; \quad \beta_i^a = \frac{b_4 - b_3}{\epsilon_p}; \quad \gamma_i^a = \frac{\theta_4 - \theta_3}{\epsilon_r}$$

and using the same simplifying assumptions as earlier, the parts entropy of the ith assembled module is

$$\widehat{H}_i^a = \log_2 \alpha_i^a + \log_2 \beta_i^a + \log_2 \gamma_i^a.$$

The same assumption is made about statistical independence of poses across subsystems and the total parts entropy of an assembled system is given by $H^a = \sum_{i=1}^n H_i^a$. Note that $(a_4 - a_3) < (a_2 - a_1)$, $(b_4 - b_3) < (b_2 - b_1)$ and $(\theta_4 - \theta_3) < (\theta_2 - \theta_1)$. The parts entropy for each object must be computed individually according to its tolerance. The same values of ϵ_p and ϵ_r must be used when computing unstructured, structured and assembled parts entropy. There is no clear-cut way to chose ϵ_p and ϵ_r, but it is important that they be less than the smallest values of the positional tolerance and the rotational tolerance. In addition, the same values must be used when comparing parts entropy values across different systems.

We apply this analysis to actual robotic systems as follows. First the bounds of the unstructured environment (a, b) are determined. Usually these are just the dimensions of a bounding box containing the track. Next, the landmark and assembly tolerances are measured or estimated empirically from the physical system. The positional and orientational tolerances of the ith subsystem in a structured environment are defined as

$$\delta g_i^s = (\delta x_i^s, \delta y_i^s, \delta \theta_i^s)$$

and the tolerances of the jth subsystem, when it is in an assembled robot, are defined as

$$\delta g_j^a = (\delta x_j^a, \delta y_j^a, \delta \theta_j^a),$$

where $\delta x = (x_2 - x_1)$, $\delta y = (y_2 - y_1)$ and $\delta\theta = (\theta_2 - \theta_1)$ for $x \in [x_1, x_2]$, $y \in [y_1, y_2]$ and $\theta \in [\theta_1, \theta_2]$. Finally, resolutions ϵ_p and ϵ_r are chosen to be less than the minimum tolerance value.

If an initial functional robot is able to assemble all subsystems of the replica in an unstructured environment, then $\Delta H_E = 0$ and the parts entropy change induced by the robot will be determined by $\Delta H_R = H^u - H^a$. By building some landmarks or structures in the environment, the amount of entropy change associated with assembly performed by the robot is reduced by the amount of entropy change resulting from structuring the environment. We recall that a self-replicating system is capable of replication without the aid of deterministic external manipulation. A structured environment can hold information about part locations or passively guide the robot trajectory, but does not actively control or manipulate the subsystems. If an external agent partially or fully controls the replication process, we call the system “replicating” but not

Downloaded from http://ijr.sagepub.com by Kiju Lee on July 1, 2008
© 2008 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

"self-replicating". For the same level of replication, a system including an external agent will have a higher value of ΔH_E and a smaller value of ΔH_R than a system capable of self-replication.

The presented measures are applied to three prototypes and some other existing systems, and the results are described in the following section.

3. Prototypes

We present three physical prototypes in this section. A microprocessor-based self-replicating robot (SRR I) and two self-replicating systems with distributed electronics (SRR II and SRR III) are described. SRR I and SRR III are capable of distinguishing between subsystems by reading the information embedded in the structured environments. Each robotic system consists of five or six heterogeneous prefabricated subsystems. These prototypes are fully autonomous and capable of replication without human intervention during the assembly process. Therefore, they are viewed as self-replicating systems. The degree of self-replication based on the number of active elements and interconnections is computed for each prototype. In addition, the entropy reductions due to a structured environment and robot assembly process, ΔH_E and ΔH_R, are computed as well. For computational simplicity, we assume that the probability distribution on part position and orientation is uniform within the allowed tolerances, and that variables defining poses (x, y, θ) are independent of each other.

3.1. Microprocessor-based SRR: SRR I

The replication process of SRR I (Park et al. 2004) can be described in our framework as

$$(\mathrm{R}, \mathrm{M}, \mathrm{E}) \longrightarrow (\mathrm{R}', \emptyset, \mathrm{E}) \tag{8}$$

such that

$$\begin{aligned} \mathrm{R} &= \{r\} \rightarrow \mathrm{R}' = \{r, r\}; \\ \mathrm{M} &= \{M_1, M_2, M_3, M_4, M_5\} \rightarrow \mathrm{M}' = \emptyset; \\ \mathrm{E} &= \{E_{\text{movable}}, E_{\text{fixed}}\} \rightarrow \mathrm{E}' = \mathrm{E}. \end{aligned}$$

As shown in Figure 3, the base (with Lego™ RCX controller), M_5, must be initially placed at the center of the track and the other four subsystems can be placed on one of four sub-tracks. The assembly process is made by collecting four subsystems and attaching them to the base module in a strict order of $M_1 \rightarrow M_2 \rightarrow M_3 \rightarrow M_4$. The structures that can be relocated according to the subsystem locations are denoted by E_{movable} including bar codes on the track, and the structures fixed in the environment are denoted by E_{fixed} containing the sub-tracks and the main track.

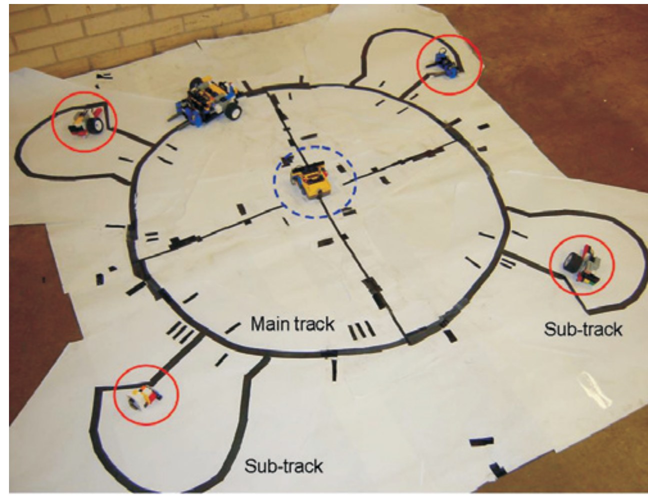

Fig. 3. SRR I: initial setup of the system, $M_1, \ldots, M_4$ and M_5 (fixed at the center of the track).

The main difference of SRR I from our earlier fully autonomous prototype (Suthakorn et al. 2003a), referred as "SRR-03", is that SRR I is able to read bar codes on the track, indicating each subsystem's location, while SRR-03 simply follows the pre-defined trajectory. The initial functional robot in SRR I can start out anywhere on the main track. Each subsystem is labeled by a bar code on the main track. When the robot finds this bar code, it takes the alternate path to the sub-track, which leads to a subsystem. It proceeds to pick it up and follow that path until it converges onto the main track again. After it gets back to the main track, it starts to search for a matching bar code to find the location where the subsystem is to be released. When it reaches the specified bar code, it takes the path into the center of the track and delivers the subsystem to the base. Afterwards, it turns around and follows the path back onto the main track. This bar code matching algorithm is then performed for the other subsystems.

The robot consists of five subsystems: a gripper, left wheel with a motor, right wheel with a motor, a balancing tail and the base with a Lego™ RCX controller as shown in Figure 4. Mechanical and electrical connections between modules are made through permanent magnets and spring-loaded metal contacts. The RCX controller contains a microprocessor and 512 bytes of RAM. We do not know the exact number of transistors equivalent to the RCX, but we estimate a lower bound to the complexity based on the size of the RAM as $512 \times 8 = 4,096$ active elements. Let α be the number of active elements in the RCX controller, then $\alpha \gg 4{,}000$. As shown in Table 1, the number of active elements of M_5 is far greater than that of $M_1, \ldots, M_4$. The number of interconnections between modules is 18, and so the degree of self replication is given by

Downloaded from http://ijr.sagepub.com by Kiju Lee on July 1, 2008
© 2008 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

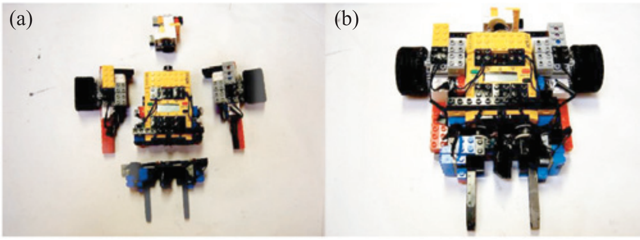

Fig. 4. SRR I: (a) five subsystems disassembled; (b) completely assembled robot.

Table 1. Five subsystems of SRR I.

Module	Components	Complexity
M_1	Right motor/wheel	12
M_2	Gripper	9
M_3	Left motor/wheel	12
M_4	Balancing tail	1
M_5	RCX Controller	$\alpha \gg 4{,}000$

$$D_{\mathrm{s}} \ll \frac{1}{4{,}000} \cdot \frac{4{,}052}{807^2} \simeq 1.56 \times 10^{-6}. \tag{9}$$

We now compute the parts entropy for this system. The size of the environment is defined by the size of minimum rectangle which covers the whole environmental structure. Then, we have $\mathcal{X}_i^{\mathrm{u}} = [0,\ 1{,}200]$, $\mathcal{Y}_i^{\mathrm{u}} = [0,\ 1{,}200]$ and $\mathcal{Z}_i^{\mathrm{u}} = [0,\ 2\pi]$ for all $i = 1, \ldots, 5$. The translational displacement is measured in millimeters and the rotational displacement is measured in radians. For $\epsilon_{\mathrm{p}} = 0.5$ and $\epsilon_\theta = 0.01$, if all variables are independent of each other and have uniform distributions, the parts entropy of the ith subsystem in an unstructured environment is given by

$$\widehat{H}_i^{\mathrm{u}} = \log_2 \alpha_i^{\mathrm{u}} + \log_2 \beta_i^{\mathrm{u}} + \log_2 \gamma_i^{\mathrm{u}} \simeq 31.75,$$

where

$$\alpha_i^{\mathrm{u}} = 2400; \quad \beta_i^{\mathrm{u}} = 2{,}400; \quad \gamma_i^{\mathrm{u}} \simeq 628.$$

If every subsystem has the same parts entropy, then the total parts entropy is given by

$$\widehat{H}^{\mathrm{u}} = \sum_{i=1}^{5} \widehat{H}_i^{\mathrm{u}} \simeq 158.75.$$

In SRR I, the system is tolerant to permutations of the four movable track segments, because each segment is identified by an attached bar code. There are 4! ways to locate $M_1, \ldots, M_4$ in four sub-tracks with some tolerance and M_5 must be placed at the center of the track with fairly small tolerance. In placing subsystems, each of them has some tolerance in its pose given by

$$\begin{aligned}
\delta g_1^{\mathrm{s}} &= \delta g_3^{\mathrm{s}} = (5, 5, 0.12);\\
\delta g_2^{\mathrm{s}} &= (10, 5, 0.17);\\
\delta g_4^{\mathrm{s}} &= (8, 5, 0.13);\\
\delta g_5^{\mathrm{s}} &= (1, 1, 0.02).
\end{aligned}$$

The part entropy for each subsystem can be computed as

$$\begin{aligned}
\widehat{H}_1^{\mathrm{s}} &= \widehat{H}_3^{\mathrm{s}} = \log_2 10 + \log_2 10 + \log_2 12 \simeq 10.23;\\
\widehat{H}_2^{\mathrm{s}} &= \log_2 20 + \log_2 10 + \log_2 17 \simeq 11.73;\\
\widehat{H}_4^{\mathrm{s}} &= \log_2 16 + \log_2 10 + \log_2 13 \simeq 11.02;\\
\widehat{H}_5^{\mathrm{s}} &= \log_2 2 + \log_2 2 + \log_2 2 = 3.
\end{aligned}$$

The structured environment includes four sub-tracks to locate four subsystems and, as shown in Figure 3, these areas do not overlap each other. Therefore, there are 4! ways to locate the subsystems in the environment. The entropy resulting from this permutations must be counted in addition to the entropy from the positional and rotational tolerances. If we define $\widehat{H}^{\mathrm{perm}}$ as the entropy from the possible permutations and assume that the probability distribution among permutations is uniform, then

$$\widehat{H}^{\mathrm{perm}} = -\sum_{k=1}^{4!} \frac{1}{4!} \log_2 \frac{1}{4!} = \log_2 4! \simeq 4.59.$$

The total parts entropy can be computed as

$$\widehat{H}^{\mathrm{s}} = \sum_{i=1}^{5} \widehat{H}_i^{\mathrm{s}} + \widehat{H}^{\mathrm{perm}} \simeq 50.80.$$

Therefore, the entropy reduction by the structured environment is given by

$$\Delta H_{\mathrm{E}} = \widehat{H}^{\mathrm{u}} - \widehat{H}^{\mathrm{s}} \simeq 107.95. \tag{10}$$

If the tolerance of the assembled subsystem is given by $\delta g_i^{\mathrm{a}} = (2, 2, 0.01)$ for all $i = 1, \ldots, 5$, the parts entropy is computed as

$$\widehat{H}_i^{\mathrm{a}} = \log_2 4 + \log_2 4 + \log_2 1 = 4$$

and the total parts entropy for the assembled system is

$$\widehat{H}^{\mathrm{a}} = 20.$$

Therefore, the entropy reduction by the robot assembly is given by

$$\Delta H_{\mathrm{R}} = \widehat{H}^{\mathrm{s}} - \widehat{H}^{\mathrm{a}} \simeq 30.80. \tag{11}$$

Downloaded from http://ijr.sagepub.com by Kiju Lee on July 1, 2008
© 2008 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

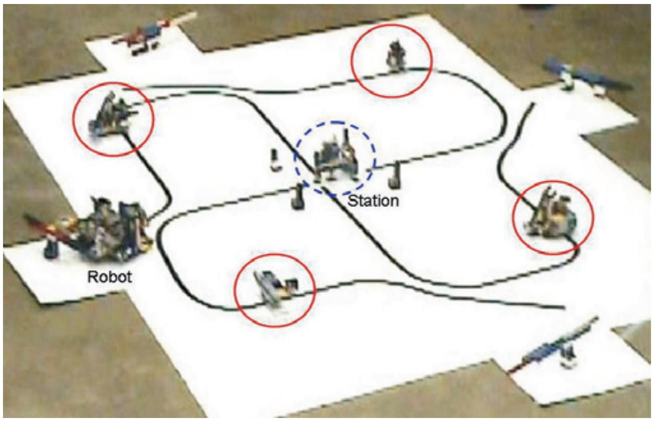

Fig. 5. SRR II: initial set-up in the environment, $M_1, \ldots, M_4$ and M_5 (at the center).

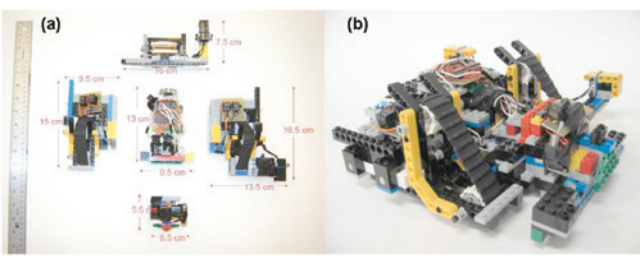

Fig. 6. SRR II: (a) five subsystems; (b) completely assembled robot.

3.2. SRR with Distributed Electronics: SRR II

SRR II (Eno et al. 2007) can be described in our framework as

$$(\mathrm{R}, \mathrm{M}, \mathrm{E}) \longrightarrow (\mathrm{R}', \emptyset, \mathrm{E}), \tag{12}$$

such that

$$\begin{aligned} \mathrm{R} &= \{r\} \rightarrow \mathrm{R}' = \{r, r\}; \\ \mathrm{M} &= \{M_1, M_2, M_3, M_4, M_5\} \rightarrow \mathrm{M}' = \emptyset; \\ \mathrm{E} &= \{E, E, E, E\} \rightarrow \mathrm{E}' = \mathrm{E}. \end{aligned}$$

The central subsystem M_5 must be fixed in the environment, as with SRR I; $M_1, \ldots, M_4$ can be placed anywhere along the length of one of four tracks, within a fairly small orientation tolerance. The environment consists of four identical subsets, $E = \{track, wall, pole\}$. The initial setup of the system and the robot and five subsystems are shown in Figures 5 and 6.

SRR II has *distributed electronics* instead of an integrated controller as used in SRR I and SRR-03. The trajectory of the robot is determined by the structured environment. When the robot picks up a subsystem, the track automatically leads the robot to place the subsystem in the correct location. Once the subsystem is attached to the central part, the robot reverses its direction and then goes back to the next track. The robot repeats this process until it assembles every subsystem. A flowchart of this logic is shown in Figure 7. The robot has two finite states in its behavior: moving forward along the line (mode 1) and moving backward blindly (mode 2). There are two events which trigger a change in state. A triggering of the front touch sensor causes the robot to transition from mode 1 to mode 2. A triggering of the rear touch sensor causes the robot to transition from mode 2 to mode 1. According to the orientation of M_5, $M_1, \ldots, M_4$ are placed in one of four tracks. They are placed with certain orientations anywhere along the track. Figure 8 shows a time-lapse of the self-replicating process.

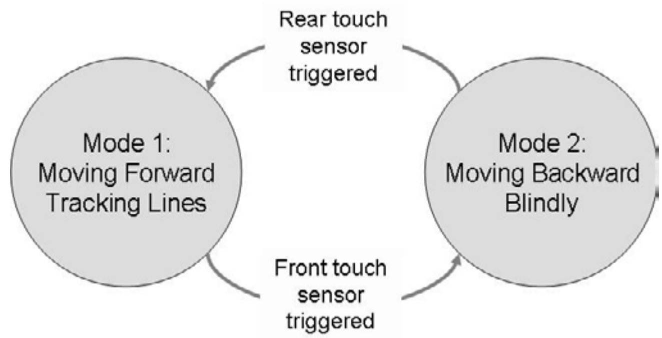

Fig. 7. SRR II: state transitions between two defined behaviors.

Table 2. Five subsystems of SRR II.

Module	Components	Complexity
M_1	Magnetic gripper/9V battery	3
M_2	Left motor/driving circuit	14
M_3	Right motor/driving circuit/touch sensor	15
M_4	5V battery/touch sensor	3
M_5	Main circuit/line tracker	65

Five subsystems and the number of active elements in each subsystem are described in Table 2. The number of interconnections is 26, and so the degree of self-replication of SRR II is

$$D_s = \frac{3}{65} \cdot \frac{126}{20^2} \simeq 1.45 \times 10^{-2}. \tag{13}$$

Although most active elements are still concentrated in M_5 rather than the other subsystems, D_s is much larger than in SRR I because the RCX controller is replaced by discrete circuit elements distributed into subsystems.

We now compute the entropy reductions for this system. The environment is bounded by $\mathcal{X}_i^u = [0,\ 1{,}016]$, $\mathcal{Y}_i^u = [0,\ 1{,}016]$ and $\mathcal{Z}_i^u = [0,\ 2\pi]$ for all $i = 1, \ldots, 5$. For $\epsilon_p = 0.5$

Downloaded from http://ijr.sagepub.com by Kiju Lee on July 1, 2008
© 2008 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

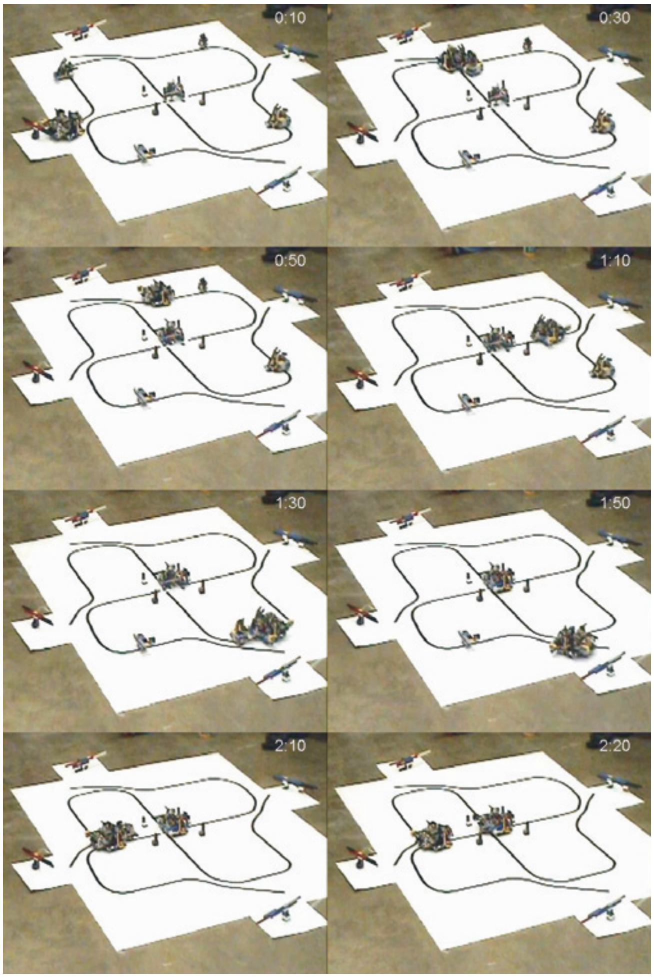

Fig. 8. SRR II: self-replication process in a structured environment.

and $\epsilon_\theta = 0.01$, the parts entropy for each subsystem in an unstructured environment is given by

$$\widehat{H}_i^{\mathrm{u}} = \log_2 \alpha_i^{\mathrm{u}} + \log_2 \beta_i^{\mathrm{u}} + \log_2 \gamma_i^{\mathrm{u}} \simeq 31.27,$$

where

$$\alpha_i^{\mathrm{u}} = 2{,}032; \quad \beta_i^{\mathrm{u}} = 2{,}032; \quad \gamma_i^{\mathrm{u}} \simeq 628,$$

and the total parts entropy is

$$\widehat{H}^{\mathrm{u}} = 31.27 \times 5 \simeq 156.35.$$

The gripper uses permanent magnets to "grasp" steel contact plates on each subsystem. Figure 9 shows the front and back view of the magnetic end-effector. As every subsystem uses the same kind of contact plate, we assume that tolerance to errors is the same for $M_1, \ldots, M_4$, and given by

$$\delta g_i^{\mathrm{s}} = (900, 5, 0.10).$$

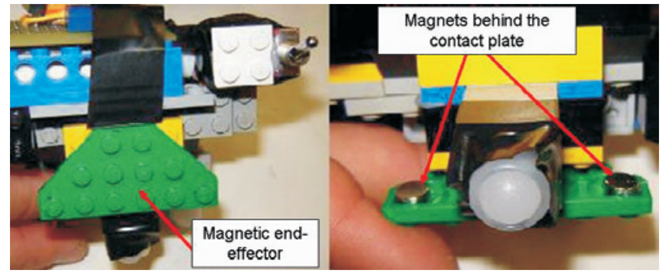

Fig. 9. SRR II: magnetic end-effector, M_1.

The first tolerance value is much larger for this system (900 compared with 10 for SRR I) because subsystems can be placed anywhere along the length of one of four tracks, as long as they do not interfere with the robot trajectory. The part entropy for each subsystem can be computed as

$$\widehat{H}_i^{\mathrm{s}} = \log_2 1{,}800 + \log_2 10 + \log_2 10 \simeq 17.46$$

for all $i = 1, \ldots, 4$. For M_5 with $\delta g_5^{\mathrm{s}} = (1, 1, 0.02)$, we have

$$\widehat{H}_5^{\mathrm{s}} = \log_2 2 + \log_2 2 + \log_2 2 = 3.$$

As four subsystems must be assembled in a strict order, there are 4! ways to arrange $M_1, \ldots, M_4$ in four tracks as discussed previously. Then, we have

$$\widehat{H}^{\mathrm{perm}} = -\sum_{k=1}^{4} \frac{1}{4} \log_2 \frac{1}{4} = \log_2 4 = 2.$$

Thus the total parts entropy in a structured environment is given by

$$\widehat{H}^{\mathrm{s}} = \sum_{i=1}^{5} \widehat{H}_i^{\mathrm{s}} + \widehat{H}^{\mathrm{perm}} \simeq 74.84.$$

The entropy reduction by the structured environment is computed as

$$\Delta H_{\mathrm{E}} = \widehat{H}^{\mathrm{u}} - \widehat{H}^{\mathrm{s}} \simeq 81.51. \tag{14}$$

The assembled subsystems have a small tolerance of $\delta g_i^{\mathrm{a}} = (2, 2, 0.01)$. The parts entropy for an assembled subsystem is

$$\widehat{H}_i^{\mathrm{a}} = \log_2 4 + \log_2 4 + \log_2 1 = 4$$

for all $i = 1, \ldots, 5$. The total parts entropy for the assembled subsystems is

$$\widehat{H}^{\mathrm{a}} = 20.$$

Therefore, the entropy reduction by the robotic assembly is given by

$$\Delta H_{\mathrm{R}} = \widehat{H}^{\mathrm{s}} - \widehat{H}^{\mathrm{a}} \simeq 54.84. \tag{15}$$

Downloaded from http://ijr.sagepub.com by Kiju Lee on July 1, 2008
© 2008 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

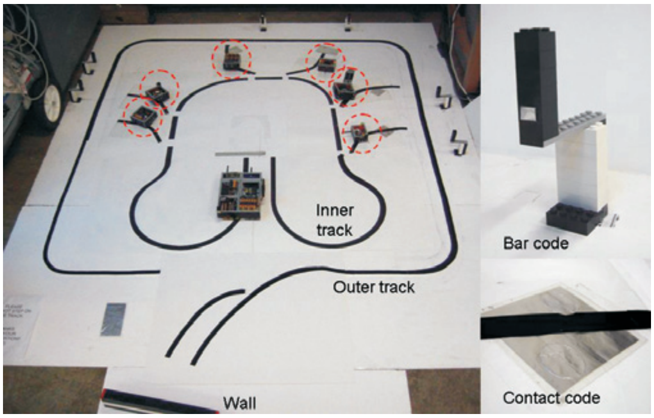

Fig. 10. SRR III: overview of the entire system and the environmental structures.

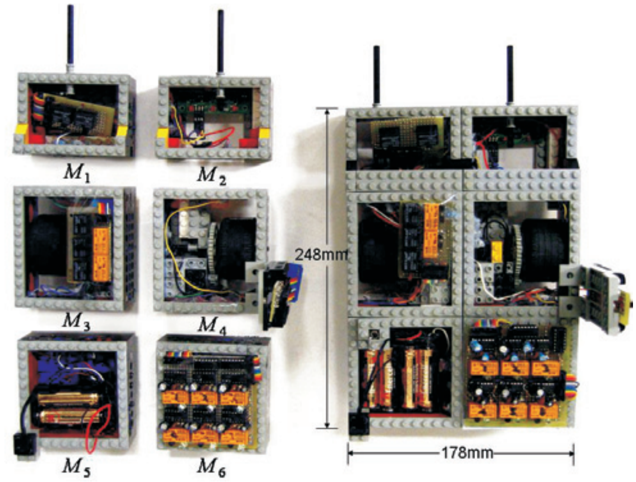

Fig. 11. SRR III: (a) six disassembled modules and (b) a completely assembled robot.

3.3. Advanced SRR with Distributed Electronics: SRR III

Self-replication by SRR III (Lee and Chirikjian 2007) can be represented as

$$(\mathrm{R}, \mathrm{M}, \mathrm{E}) \longrightarrow (\mathrm{R}', \emptyset, \mathrm{E}) \tag{16}$$

such that

$$\begin{aligned} \mathrm{R} &= \{r\} \rightarrow \mathrm{R}' = \{r, r\}; \\ \mathrm{M} &= \{M_1, M_2, M_3, M_4, M_5, M_6\} \rightarrow \mathrm{M}' = \emptyset; \\ \mathrm{E} &= \{E_{\text{movable}}, E_{\text{fixed}}\} \rightarrow \mathrm{E}' = \mathrm{E}. \end{aligned}$$

Here E_{movable} includes bar codes and contact patches on the track, and E_{fixed} includes the outer track and the inner track (Figure 10). The subsystems must be collected in a certain order to be assembled successfully as shown in Figure 11. From top to bottom, there are three layers of modules: the first layer with M_1 and M_2, the second layer with M_3 and M_4 and the third layer with M_5 and M_6. The first layer must be collected prior to the second or the third, and the second layer must be collected prior to the third.

SRR III is a combination and an extension of SRR I and SRR II. SRR III is able to distinguish between subsystems and assemble them automatically in a structured environment (much like SRR I). The robot consists of six distinctive subsystems with distributed electronics (much like SRR II). The electronic circuit is also divided into six sub-circuits, and each of them is mounted on one of six subsystems. Some improvements have been made in SRR III compared with the previous prototypes. These are: (i) an increased number of subsystems, (ii) no initially fixed hub, (iii) advanced modular design (easier to push and slide), (iv) extended functionality of the robot (more defined behaviors) and (v) more movable environmental structures.

Figure 10 shows an overview of the initial setup in the structured environment. The robot has a state machine which has six states according to the subsystem to be assembled. The robot follows the outer track until it finds a bar code indicating the correct subsystem and then it turns left to pick up the subsystem. During this process, the robot passes a contact metal patch that triggers the state machine to the next state. Next, the robot enters the inner track by making a rear left or right turn and follows the inner track until it detects the metal line at the center. The metal line triggers the robot to reverse the directions of the motors, resulting in the robot moving backward until it hits the wall. Since the robot grabs a subsystem and drags it with a passive fork, the subsystem is released when the robot reverses direction. The rear touch sensor reverses the direction of travel again when the robot hits the wall, and it returns back to the outer track by forward line-tracking movement to repeat the process. The distinctive behaviors of the robot and the mode transitions according to the sensor inputs are shown in Figure 13. Figure 12 is the entire self-replication process with a time-lapse sequence. The dimensions of the passive fork on modules M_1 and M_2 are carefully decided to allow increased position and orientation tolerance. As the robot manipulates subsystems by pushing and sliding operations, choosing the right fork spacing and length allows subsystems to self-align in the fork. In other words, a well-designed grasping mechanism can tolerate increased pose errors in subsystem placement.

The six subsystems and their number of active elements are listed in Table 3. The electrical and mechanical connections among six modules are made through the spring/metal mechanism and permanent magnets as shown in Figure 14. The number of interconnections among subsystems is 49 and the degree of self-replication of SRR III is

$$D_{\text{s}} = \frac{11}{91} \cdot \frac{233}{(30.67)^2} = 2.99 \times 10^{-2}. \tag{17}$$

Downloaded from http://ijr.sagepub.com by Kiju Lee on July 1, 2008
© 2008 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

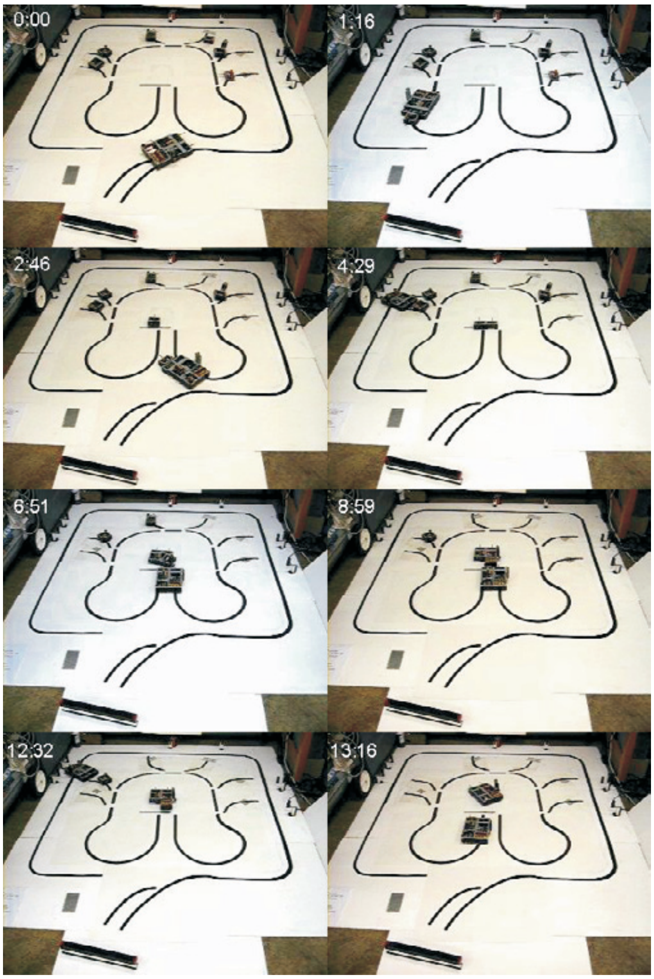

Fig. 12. SRR III: self-replication process with time-lapse sequence.

Table 3. Six subsystems of SRR III.

Module	Components	Complexity
M_1	Line tracking circuit	11
M_2	Line tracker sensor	16
M_3	Motor driver circuit/left motor	30
M_4	Bar-code reader/right motor	24
M_5	Power supply/3 contact sensors	12
M_6	State machine/3 contact sensors	91

We claim that SRR III demonstrates a higher degree of self-replication than SRR I or SRR II, in that it has a better complexity distribution and a higher relative complexity.

The environment is bounded by $\mathcal{X}_i^{\mathrm{u}} = [0,\ 2{,}032]$, $\mathcal{Y}_i^{\mathrm{u}} = [0,\ 2{,}286]$ and $\mathcal{Z}_i^{\mathrm{u}} = [0,\ 2\pi]$ for all $i = 1, \ldots, 6$. Using the same positional and rotational accuracy as before, the parts

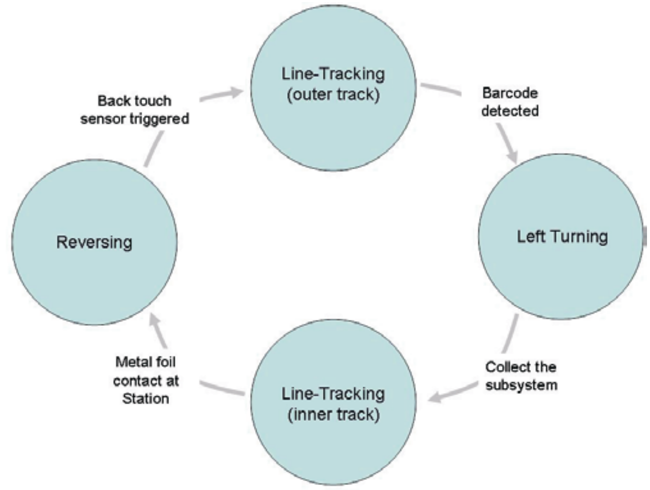

Fig. 13. SRR III: defined behaviors of the robot, the robot repeats this process until it replicates all subsystems.

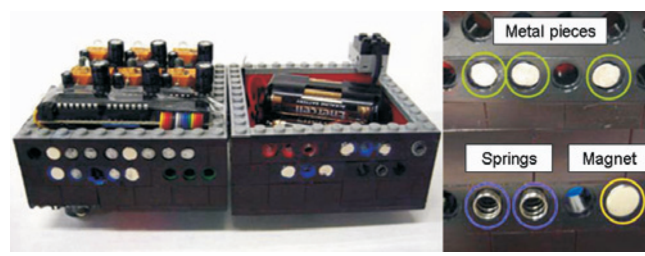

Fig. 14. SRR III: three metal pieces (top), two springs (bottom left) and a permanent magnet (bottom right).

entropy of the system in an unstructured environment is given by

$$\widehat{H}_i^{\mathrm{u}} = \log_2 \alpha_i^{\mathrm{u}} + \log_2 \beta_i^{\mathrm{u}} + \log_2 \gamma_i^{\mathrm{u}} \simeq 33.44,$$

where

$$\alpha_i^{\mathrm{u}} = 4064; \quad \beta_i^{\mathrm{u}} = 4572; \quad \gamma_i^{\mathrm{u}} \simeq 628.$$

Then, the total parts entropy is computed as

$$\widehat{H}^{\mathrm{u}} = \sum_{i=1}^{6} \simeq 200.65.$$

The tolerance of each subsystem location in the structured environment is given by

$$\begin{aligned} \delta g_1^{\mathrm{s}} &= \delta g_2^{\mathrm{s}} = (55, 13, 0.30); \\ \delta g_3^{\mathrm{s}} &= (40, 13, 0.20); \\ \delta g_4^{\mathrm{s}} &= (40, 13, 0.10); \\ \delta g_5^{\mathrm{s}} &= \delta g_6^{\mathrm{s}} = (45, 13, 0.20). \end{aligned}$$

Downloaded from http://ijr.sagepub.com by Kiju Lee on July 1, 2008
© 2008 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

The corresponding parts entropy is calculated as

$$\begin{aligned}
\widehat{H}_1^{\mathrm{s}} &= \widehat{H}_2^{\mathrm{s}} = \log_2 110 + \log_2 26 + \log_2 30 \simeq 16.39;\\
\widehat{H}_3^{\mathrm{s}} &= \log_2 80 + \log_2 26 + \log_2 20 \simeq 15.34;\\
\widehat{H}_4^{\mathrm{s}} &= \log_2 80 + \log_2 26 + \log_2 10 \simeq 14.34;\\
\widehat{H}_5^{\mathrm{s}} &= \widehat{H}_6^{\mathrm{s}} = \log_2 90 + \log_2 26 + \log_2 20 \simeq 15.51.
\end{aligned}$$

There are six locations to place six subsystems and, therefore, 6! possible ways to locate them,

$$\widehat{H}^{\mathrm{perm}} = \log_2 6! \simeq 9.49$$

and, therefore, the total parts entropy of the system in a structured environment is computed by

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

The entropy reduction by the structured environment is

$$\Delta H_{\mathrm{E}} = \widehat{H}^{\mathrm{u}} - \widehat{H}^{\mathrm{s}} \simeq 97.68. \tag{18}$$

The assembled subsystems have a small tolerance of $\delta g_i^{\mathrm{a}} = (1, 1, 0.01)$ for all $i = 1, \ldots, 6$. The parts entropy for an assembled subsystem is

$$\widehat{H}_i^{\mathrm{a}} = \log_2 2 + \log_2 2 + \log_2 1 = 2.$$

The total parts entropy for the assembled subsystems is calculated as

$$\widehat{H}^{\mathrm{a}} = 12$$

and, then, the entropy reduction by the robotic assembly is given by

$$\Delta H_{\mathrm{R}} = \widehat{H}^{\mathrm{s}} - \widehat{H}^{\mathrm{a}} \simeq 90.97. \tag{19}$$

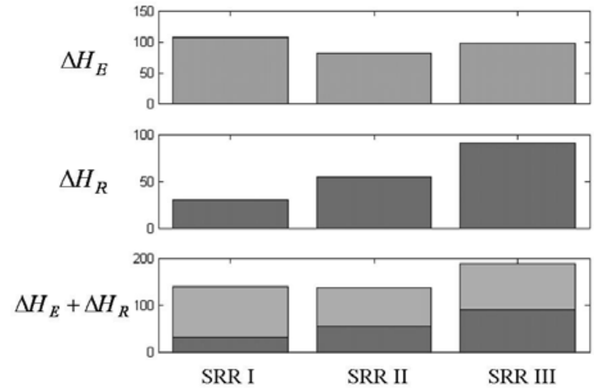

Fig. 15. Entropy reduction by structured environment, ΔH_{E}, and by the assembly process, ΔH_{R}, and the total entropy reduction, $\Delta H_{\mathrm{E}} + \Delta H_{\mathrm{R}}$, of three prototypes.

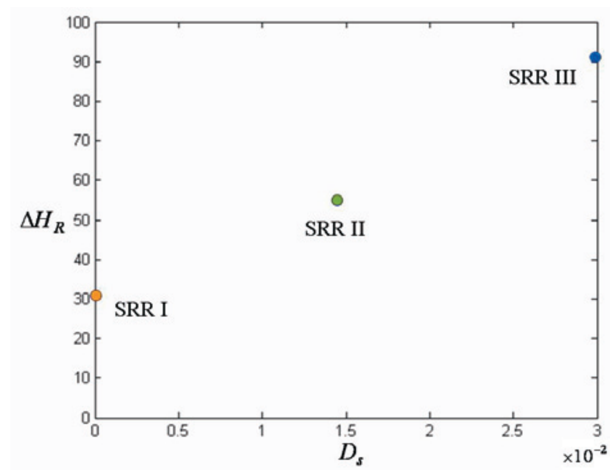

Fig. 16. D_{s} versus ΔH_{R}: SRR I, SRR II and SRR III.

3.4. Discussion on Prototypes

Figure 15 shows the amount of entropy reduced by the structured environment and by the robot assembly process for each prototype presented in Section 3. SRR I and SRR III show slightly higher ΔH_{E} than SRR II. SRR I has the smallest entropy reduction and SRR III has the largest entropy reduction resulting from the assembly process. To compare the tasks performed purely by the robots, one should observe ΔH_{R} for the systems being compared. A larger value of ΔH_{R} indicates that more sophisticated/complex tasks is performed by the robot.

SRR I and SRR II can be considered to demonstrate a similar level of self-replication in terms of complexity of the total replication process, because both self-replications are made by collecting four subsystems in a structured environment. As one can observe from the bottom graph in figure 15 SRR I and SRR II show similar values of $\Delta H_{\mathrm{E}} + \Delta H_{\mathrm{R}}$. However, the source of entropy reduction is different in the two systems. As SRR II can distinguish between subsystems by reading bar codes while SRR I cannot, one might say that SRR II performs a more complicated task than SRR I, and this is properly accounted for by our measure.

SRR I has a microprocessor in one of its five subsystems. Therefore, the complexity of the total system is mostly concentrated in one particular subsystem. By dividing electronics into several sub-circuits, SRR II and SRR III achieve higher degrees of self-replication (D_{s}). In addition, SRR III consists of six subsystems while SRR I and SRR II are both composed of five subsystems, and that fact leads SRR III to have the highest D_{s} among the three prototypes. Figure 16 plots ΔH_{R} versus D_{s} for the three prototypes.

Downloaded from http://ijr.sagepub.com by Kiju Lee on July 1, 2008
© 2008 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

The proposed measures are applicable to many modular robotic systems, including directed replication via module assembly (Zykov et al. 2005; Zykov and Lipson 2006), self-assembly of random modules (White et al. 2004; Napp et al. 2006) and replication from random modules (Penrose 1959; Griffith et al. 2005).

Systems that assemble structures from randomly agitated subsystems will naturally have a higher value of ΔH_R than those that require subsystems to be placed in fixed poses. The degree of self-replication D_s will typically be very low for systems that use microprocessors and higher for systems that use discrete circuit elements. The purpose of these measures is to compare the performance of systems within a category of task. We claim that for a modular robotic system designed for directed self-replication, high ΔH_R and high D_s are desirable. For other types of modular robots this may not apply. In addition, a comparison of ΔH_R and D_s across systems of different categories may not yield meaningful information. For example, it is not clear that values for a directed self-replication system compared with a self-assembly system will tell us anything useful, because what constitutes a task is different (in directed self-replication a functional replica must be assembled, whereas a self-assembly task might be simply to form a static structure from unassembled subsystems).

The system of Zykov et al. (2005) consists of four identical cubes in three-dimensional space. The robot assembles the modular cubes provided in one of two feeding locations. The cube is symmetric, but has only two surfaces equipped with connectors and, therefore, the initial configurations of the module provided at the feeding location can have one of two possible orientations, within a small tolerance to errors. The feeding locations are designed to be same as the cube connector surface, which means that the tolerance in placing a cube at the feeding location is the same as the tolerance of a cube in an assembled system. Therefore, there is no entropy reduction made by the assembly process. In this system, the two feeding locations can be viewed as a structured environment. Also, a human providing subsystems during the replication process is viewed as an external agent manipulating parts, and so in our terminology this system replicates, but does not self-replicate. Each modular cube has a micro-processor, actuators, etc., so their complexity can be considered roughly equal to that of the microprocessor-containing module in SRR I.

4. Conclusions

We have reviewed a descriptive framework inspired by von Neumann's early model of self-reproducing automata and presented two useful measures for robotic self-replicating or self-assembly systems. The system complexity, combining the complexity distribution among the subsystems with the relative complexity of the total assembled system compared with individual subsystems, has been defined as the degree of self-replication. In addition, the amount of task complexity by a functional robot was measured by computing the entropy reduction resulting from the robot assembly process. Sanderson's (1984) parts entropy method has been used to compute the parts entropy on locations of subsystems. Robotic replicating systems often include a structured environment in order to catalyze the replication process and reduce the task complexity. The amount of uncertainty reduced by structuring the environment was also computed by comparing two entropy values, one when subsystems are randomly placed in an unstructured environment and the other when the subsystems are placed in a structured environment. These measures were applied to three robots developed in our lab, although they could be applied to other types of modular robot. Without detailed knowledge of experimental data and specific design details, it is difficult to compute D_s, ΔH_E and ΔH_R for other robots. A larger value of D_s indicates a higher degree of self-replication, in the ratio of the total system complexity to the individual subsystem complexities is higher, and the complexity distribution of subsystems is closer to a uniform distribution. Also, a higher value of $\Delta H_E + \Delta H_R$ indicates a more complicated replication process, and a higher value of ΔH_R indicates a more complicated assembly task performed by a robot.

Acknowledgement

This work grew out of our investigations into mobile manipulation funded under NSF grant ITR 0205466. Prototypes SRR I and SRR II were designed and constructed by students in our Mechatronics course and reported in the conference proceedings cited in the text.

References

Bowyer, A. (2006). The self-replicating rapid prototyper—manufacturing for the masses. *Invited Keynote Address, Proceedings of the 7th National Conference on Rapid Design, Prototyping and Manufacturing*, Centre for Rapid Design and Manufacture, High Wycombe, June.

Chadeev, V. M. (2000). Allowance for robot self-replication in industrial automation. *Automation and Remote Control*, **61**: 1752–1757.

Chirikjian, G. S. and Suthakorn, J. (2002). Toward self-replicating robots. *Proceedings of the 8th International Symposium on Experimental Robotics (ISER)*, Italy, July.

Chirikjian, G. S., Zhou, Y. and Suthakorn, J. (2002). Self-replicating robots for Lunar development. *ASME and IEEE Transactions on Mechatronics*, **7**(4): 462–472.

Downloaded from http://ijr.sagepub.com by Kiju Lee on July 1, 2008
© 2008 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

Eno, S. et al. (2007). Robotic self-replication in a structured environment without computer control. *Proceedings of International Symposium on Computational Intelligence in Robotics and Automation*, Jacksonville, FL, June.

Freitas, R. A. and Gilbereath, W. P. (1982). Advanced automation for space missions. Proceedings of the 1980 NASA/ASEE Summer Study, Chapter 5: Replicating systems concepts: Self-replicating lunar factory and demonstration, NASA, Scientific and Technical Information Branch (Conference Publication 2255), Washington DC, US government printing office.

Freitas, R. A. and Merkle, R. C. (2004). *Kinematic Self-Replicating Machines*. Georgetown, TX, Landes Bioscience.

Friedman, G. (2002). Self-replication technology for the space Solar power mission. *Joint NASA/NSF Workshop on Autonomous Construction and Manufacturing for Space Electrical Power Systems*, April.

Griffith, S., Goldwater, D. and Jacobson, J. M. (2005) Self-replication from random parts. *Nature*, **437**: 636.

Hastings, W. A. et al. (2004). A minimalist parts manipulation systems for a self-replicating electromechanical circuit. *Proceedings of IMG'04*, Genova, Italy, July.

Jacobson, H. (1958). On models of reproduction. *American Scientist*, **46**: 255–284.

Lackner, K. S. and Wendt, C. H. (1995). Exponential growth of large self-reproducing machine systems. *Mathematical and Computer Modelling*, **21**: 55–81.

Lee, K. and Chirikjian, G. S. (2007). Robotic self-replication from low-complexity parts. *IEEE Robotics and Automation Magazine*, **14**(4).

Lipson, H. (2005). Homemade: the future of functional rapid prototyping. *IEEE Spectrum*, May, pp. 24–31.

Liu, A. et al. (2007). A memoryless robot that assembles seven subsystems to copy itself. *Proceedings of the 2007 IEEE International Symposium on Assembly and Manufacturing*, Ann Arbor, MI, July.

Murata, S., Kurokawa, H. and Kokaji, S. (1994). Self-assembling machine. *Proceedings of the IEEE International Conference on Robotics and Automation*, San Diego, CA.

Napp, N., Burden, S. and Klavins, E. (2006). The statistical dynamics of programmed self-assembly. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1469–1476.

Park, M., Chitta, S. and Yim, M. (2006). Isomorphic gait execution in homogeneous modular robots. *Proceedings of Robotics: Science and Systems Workshop on Self-reconfigurable Modular Robots*, Philadelphia, PA, August.

Park, W. et al. (2004). Robotic self-repair in a semi-structured environment. *Proceedings of Robosphere 2004*, NASA Ames, CA, November.

Penrose, L. S. (1959). Self-reproducing machines. *Scientific American*, **200**(6): 105–114.

Sanderson, A. C. (1984). Part entropy method for robotic assembly design. *Proceedings of International Conference on Robotics*.

Shen, W., Will, P. and Khoshnevis, B. (2003). Self-assembly in space via self-reconfigurable robots. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2516–2621.

Sipper, M. (1998). Fifty years of research on self-replication: an overview. *Artificial Life*, **4**(3): 237–257.

Suthakorn, J., Cushing, A. B. and Chirikjian, G. S. (2003a). An autonomous self-replicating robotic system. *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics*.

Suthakorn, J., Kwan, Y. and Chirikjian, G. S. (2003b). A semi-autonomous replicating robotic system. *Proceedings of IEEE International Conference on Intelligent Robots and Applications (CIRA)*, Kobe, Japan, July.

von Neumann, J. and Burks, A. W. (1962). *Theory of Self-Reproducing Automata*. Urbana, IL, University of Illinois Press.

White, P. J., Kopanski, K. and Lipson, H. (2004). Stochastic self-reconfigurable cellular robotics. *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 3, pp. 2888–2893.

Yim, M. et al. (2001). Distributed control for 3D metamorphosis. *Autonomous Robots*, **10**: 41–56.

Yoshida, E. et al. (2003). A self-reconfigurable modular robot: reconfiguration planning and experiments. *International Journal of Robotics Research*, **21**(10): 903–916.

Zykov, V. and Lipson, H. (2006). Fluidic stochastic modular robotics: revisiting the system design. *Proceedings of Robotics: Science and Systems Workshop on Self-reconfigurable Modular Robots*, Philadelphia, PA, August.

Zykov, V. et al. (2005). Self-reproducing machines. *Nature*, **435**(7038): 163–164.

Downloaded from http://ijr.sagepub.com by Kiju Lee on July 1, 2008
© 2008 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.