2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

An Autonomous Robot that Duplicates Itself from Low-complexity Components

Kiju Lee and Gregory S. Chirikjian

***Abstract*— This paper presents an autonomous self-replicating robot consisting of four low-complexity modules. The entire system is composed of a parent robot, four unassembled modules provided as resources, and an environment where the self-replication takes place. The parent robot grows itself by attaching the resource modules onto itself until it doubles its physical size, and then splits in the middle thereby returning the parent to its original state and producing a daughter robot. We call these processes *expansion* and *separation*, respectively. The environment plays a passive role as a catalyst that helps generating a spiral trajectory for the parent robot and does not hold any information about the resource modules. To assess the physical changes made by self-replication, structural and informational complexities associated with the robotic system and the self-replication process are quantified and compared to previous prototypes.**

I. Introduction

Many aspects of robotics research are inspired by biological systems with regard to mechanical design, locomotion, and control algorithms. Among a number of unique capabilities, replication or self-replication[1] is one of the most defining features of life that can be observed from single-celled organisms to higher-level creatures. Replication in a biological system takes place at cellular- and subcellular-levels, such as cell-division, DNA replication and transcription, RNA self-replication, and virus replication within a host cell. Living organisms are composed of one or more cells, and each of these cells is multi-functional, handling a tremendous amount of information [2]. The process that generates two cells (the "daughters") from a single original (the "parent") is called the cell cycle. During this division process, environmental conditions and external elements are essential and critical. These external factors may include passive tools used by the cell and returned to the environment, such as catalysts, nutrients for a cell to grow, and environmental conditions such as temperature, humidity and pH.

For several decades, robotics researchers have been trying to mimic this special capability of biological systems in control algorithms and hardware design as an attempt to achieve some level of self-sustainability in robotic systems. This paper presents an autonomous self-replicating robot consisting of four heterogeneous modules as shown in Fig. 1. Our approaches to realizing robotic self-replication are *modularity* and *low-complexity*. The entire system is composed of a *parent* robot, a set of *resource* modules, and an *environment*:

K. Lee is with the Department of Mechanical and Aerospace Engineering, Case Western Reserve University, Cleveland, OH 44106, USA. e-mail: kiju.lee@case.edu

G. S. Chirikjian is with the Department of Mechanical Engineering, Johns Hopkins University, Baltimore, MD 21218, USA. e-mail: gregc@jhu.edu

[1]The distinction between replication and self-replication depends on whether there exists an external agent that actively assists or controls the replication process or not [1].

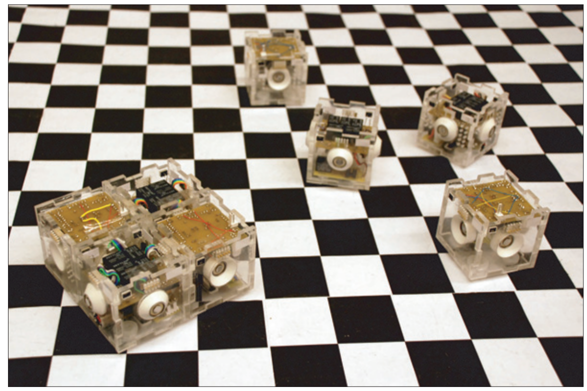

Fig. 1. A parent robot and unassembled modules are placed in a checkerboard environment.

- **Parent Robot**, $\{R\}$: A fully functional robot that duplicates itself given resource modules in a specified environment.
- **Resource Modules**, $\{M_1^A, M_2^A, M_1^B, M_2^B\}$: A set of resource modules provided to R for self-replication. These modules are identical to those composing R.
- **Environment**, $\{E\}$: A confined environment, a 1067×1067 [mm^2] checkerboard, that passively assists R for self-replication.

Each of the resource modules represents a different type of resource or nutrient needed by the parent robot to replicate. The parent robot physically grows by attaching resource modules onto itself until it doubles in its physical size (Expansion). It then separates in the middle resulting in two robots: the original parent and a replica called the *daughter* robot (Separation). Unlike the environmental structures used for the robots presented in [3] and [4], the checkerboard environment does not hold any information about the resource modules. This environment simply plays a passive role as a catalyst that promotes the duplication process by generating a 'circular-like' trajectory of the parent while it expands. In addition, the parent robot is the only *active* system during the replication process and the unassembled resource modules are viewed as incomplete parts. To assess and examine our robot in terms of the structural properties and performance, two quantitative measures, the degree of self-replication and the configurational entropy changes defined in [3], are computed and the results are compared to previous prototypes. We first begin by reviewing related works on self-replicating systems.

Related Works

In a 1948 lecture, von Neumann presented a model of an automaton that has a sufficient complexity to reproduce itself [5], [6] (See [1] and reference therein). His concepts of kinematic self-replication have been applied in many research areas such as cellular automata, nanotechnology, macromolecular chemistry and computer simulations [6], [7]. However, since the earliest physical demonstration of the concept was performed by Penrose and Jacobson [8], [9], [10], research on kinematic (or physical) self-replicating systems has been lagging behind due to the limitations imposed by physical elements, such as geometry, force and energy.

As the first physical model, Penrose considered 2D components aligned in a 1D array with 1D translational and rotational motions [8]. The system contains two different types of blocks where each type of block can connect to the other. When these blocks are placed on a vibration table with an initially assembled 'seed' consisting of two blocks of each type, the unassembled blocks make assemblies identical to the seed. Penrose presented a more complicated mechanical system in [9], but the self-replication process was similar to the one presented in [8]. Jacobson presented several models of self-replicating machines based on cars and railroads. One of these models was actually constructed and presented in [10]. Although the systems by Penrose and Jacobson were no more than compositions of simple blocks, these works have been a pathway towards the concept of robotic self-replication.

Research on self-replicating/assembling systems has been being revived since NASA became interested in the concept for purposes of space exploration in the 1980s [7]. Recent experimental work includes directed replication via module assembly [3], [11], [4], directed replication via fabrication [12], [13], undirected self-replication [14] and self-assembly of randomly agitated modules [15], [16], [17]. Directed self-replication refers to a process that an initial functional system executes a sequence of steps, either preprogrammed or by reading information embedded in the environment, in order to create a functional replica. On the other hand, undirected replication is a process in which an initial system duplicates itself without following exact procedures. Also, it often inclues some 'random' or 'unknown' elements in its behaviors during the replication process. For example, the robot presented in [14] moves in Brownian motions and pushes unassembled parts randomly positioned in a bounded area to create replicas. In this case, the robot's motion is random as well as the initial locations of parts are unknown to the robot.

Self-replication, self-assembly and self-reconfiguration are highly related to each other in terms of their modularity. The concept on reconfigurable robots and its first physical implementation were presented by Fukuda in 1988 [18].

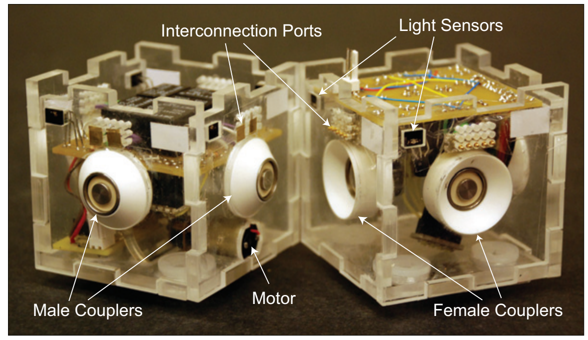

Fig. 2. M^A- and M^B-type modules. Each contact surface is equipped with an EM installed in male/female couplers, a light sensor and interconnection ports.

The distinctive function of a self-reconfigurable robot is to change its physical configuration to increase its adaptability to the environment while a self-replicating robot may not necessarily have this capability. The recent study and implementations include M-TRAN [19], PolyBot [20], CONRO and FIMER [21]. The modules used for self-reconfigurable systems must have mobility by themselves while those for self-replication or self-assembly may not.

II. System Description

The entire system consists of three elements: a parent robot consisting of four heterogenous modules, another set of these modules provided as resources, and a checkerboard environment where the parent robot duplicates itself. The parent robot is made of four cube-like modules connected to each other with electromagnets (EMs) installed in female and male couplers. The environment, where the resource modules are initially provided, is a black-and-white checkerboard with a size of about 1067×1067 [mm^2] containing 441 square grids (Fig. 1). The size of each grid is 51×51 [mm^2].

The parent robot moves in spiral motions due to a light sensor installed at the bottom of the parent robot that controls the motor in M_2^A depending on the black and white patterns of the checkerboard. While making a spiral trajectory, the parent robot assembles the resource modules arbitrarily placed in the environment. Once it completes assembling all of the resource modules, it immediately turns off the EMs resulting in separation. This process returns the parent to its original state and produces a daughter robot. The system does not require a specific location or fixture for self-replication. We note that, in most existing systems, the replication process was performed by collecting modules one-by-one and assembling them in a designated area with certain fixtures where the replica is being assembled [3], [4]. Therefore, the robot must not only locate the resources, but also bring each of them to a specific location to assemble. This requires additional functionality of the robot for replication.

In the robotic system presented here, the parent robot is neither capable of distinguishing between individual modules nor assembling the modules in certain orientations. That is, the parent robot simply attaches the opposite type of module detected within the sensing distance. Hence, the duplicated

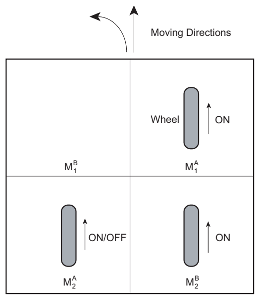

Fig. 3. The moving directions of the parent robot and the behaviors of three wheels: the motor in M_2^A is controlled by the light sensor installed at the bottom of the module reading the color patterns on the floor.

TABLE I

COMPONENTS IN FOUR MODULES AND THE EQUIVALENT NUMBER OF ACTIVE ELEMENTS REPRESENTING THE STRUCTURAL COMPLEXITY OF THE COMPONENT.

Components	M_1^A	M_2^A	M_1^B	M_2^B
Chassis(1)	1	1	1	1
Electromagnet(1)	4	4	4	4
Motor/Wheel(5)	1	1	0	1
Ball caster(1)	4	4	4	4
LED/Photo TR Pair(2)	4	5	4	4
Battery(1)	1	1	1	1
Relay(3)	5	6	5	5
Latch(4)	0	1	0	1
Transistor(1)	4	5	4	4
Total A.E.	42	52	37	46

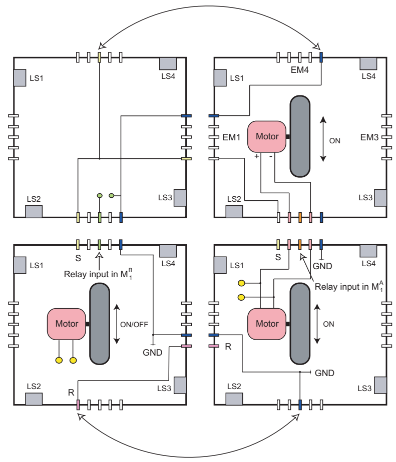

Fig. 4. Schematic of the parent robot composed of four modules. Two interconnection ports in each of the front or back side (indicated with arrows) are closed when four resource modules are fully assembled signaling the completion of the expansion process.

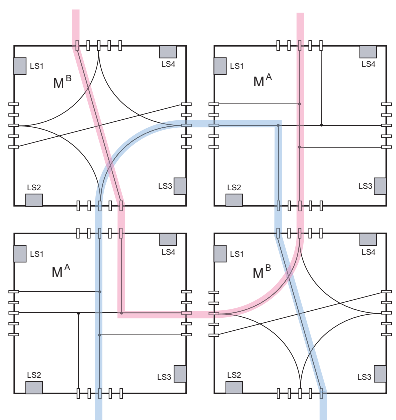

Fig. 5. Interconnections across four resource modules form an open loop which is connected to the parent robot as shown in Fig. 4.

robot may not be an 'exact' copy of the parent robot, but robots consisting of the same set of modules are viewed as being in an equivalent class.

A. Module Design

The modules are divided into two different types depending on the configurational geometry. The M^A-type module contains four electromagnets in male couplers and the M^B-type module has them in female couplers as shown in Fig. 2. With this geometric property and the limited sensing distance of the light sensor controlling the EM on each contact surface, each type of module can make a sustainable assembly only with the other type of modules. Each module is about $60 \times 60 \times 60$ [mm^3] in size and contains an integrated circuit, a Li-ion polymer battery, four ball casters on the bottom, and four contact surfaces. Each of these contact surfaces is equipped with an EM in a female or a male coupler, a light sensor, and interconnection ports (Fig. 2). The light sensor is a coupled infrared LED and photo transistor. Each EM is controlled by a light sensor installed on the same contact surface. When the light sensor detects another module within about 10 [mm], the EM is turned ON for assembly.

As shown in Fig. 3, three DC motors are installed in M_1^A, M_2^A, and M_2^B, and every module has four ball casters installed at the bottom for balancing and reducing friction. M_2^A has an additional circuit containing another light sensor installed at the bottom of the module and a latch changing the direction of all three motors simultaneously. Interconnection ports create electrical connections throughout the assembled modules. The motors in M_1^A and M_2^B are always ON while the batteries are connected and all four modules are fully assembled; the motor in M_2^A is turned ON and OFF based on the light sensor output. This motor is turned ON when the light sensor detects white color and OFF while facing black color from the checkerboard environment. Therefore, the robot moves forward when all three motors run and turns left when the motor in M_2^A is OFF. This algorithm generates spiral motions. Table I lists all electrical and mechanical components installed in each module and the corresponding number of active elements.[2]

B. Interconnections and Mutations

The parent robot has an internal network through the interconnection ports. The total number of interconnections in the original parent robot is counted by the number of

[2]An 'active element' is defined by a moving mechanical part or a fundamental electronic component [3].

connected pairs of interconnection ports, given by $I_{total} = 14$. It also has two open ports on either the front or the back side of the robot indicated with arrows as shown in Fig. 4. Once the parent finishes the expansion process, that is, the four resource modules shown in Fig. 5 are attached to the parent robot shown in Fig. 4, two open ports in the original parent robot are closed. This triggers two latches resulting in changing the directions of all three motors in the original parent robot while turning off the EMs where four newly assembled modules are attached. This results in complete separation between the original parent and the newly assembled modules which become a new daughter robot. We note that the initial moving direction of the parent can be either forward or backward.

As shown in Fig. 5, a single open loop through interconnection ports can be made by assembling four resource modules. Since each resource module can be attached to the parent robot in any of four orientations, the loop made by assembly must remain the same under rotations, 0°, 90°, 180°, and 270°. In addition, each type of module must have the same internal connections. There are 4 possible ways for the parent robot to assemble two M^A-type modules and two M^B-type modules, and each of these resource modules can have one of four orientations when attached to the parent. Therefore, the daughter robot can have one of 1024 assembly configurations after successful replication. These 1024 robots with distinct configurations are viewed as in an equivalent class while all of them except of the exact replica of the parent can be viewed as *mutations*.

III. Experiments

Figures 6 and 7 show experimental snapshots with time lapse sequence. All system components, a parent robot and resource modules, are initially provided in the checkerboard environment. The parent robot follows a spiral trajectory in a counter-clock-wise direction in the environment by reading the color patterns on the board. It then grows itself by attaching the unassembled modules along the moving direction. This assembly occurs rather randomly as the robot moves towards resource modules. In addition, the poses of resource modules may change from the initial status as the parent robot pushes them without making successful assembly.

The environment was designed not to confine the robot's trajectory completely, but rather allow for the possibility that the parent robot may deviate from the boundary without replicating itself. In addition, the sizes of the environment and each grid were determined based on experimental observations in order to keep the robot within the boundary for a certain amount of time; therefore, its trajectory covers most of the area before running away from the environment. In 20 trials, the parent robot successfully replicated 9 times, corresponding to a 45% success ratio. Table II shows the experimental results. For each trial, the resource modules and the parent robot were placed with different poses. t_s is the time required for the parent robot to produce a daughter robot successfully, and t_o is the time until the parent robot runs away from the environment without replicating. The average time required for self-replication was about 247 seconds. As shown in the table, the experimental results vary significantly, which is mainly attributed to different initial poses and sensor delays. The experiments showed that the robot deviated from the environment more quickly when its initial pose was close to the boundary than the other cases. Since the robot moves in the counter-clock-wise direction, when the parent first assembles the left module on the front side, it rarely succeeded in duplication, because assembling a module on the right front side while moving in the counter-clock-wise direction is physically difficult.

TABLE II

The time required for self-replication (t_s) and the time the parent robot stays in the checkerboard before crossing the boundary without replication (t_o).

No.	t_s [sec]	t_o [sec]	No.	t_s [sec]	t_o [sec]
1	×	265	11	×	201
2	×	270	12	×	70
3	225	-	13	×	134
4	140	-	14	248	-
5	245	-	15	268	-
6	×	325	16	×	312
7	×	299	17	×	265
8	178	-	18	330	-
9	×	330	19	278	-
10	314	-	20	×	308

IV. Complexity Analysis

To assess the physical changes made by self-replication in the presented robotic system, the degree of self-replication and the configurational entropy changes presented in [3] are revisited and computed. The degree of self-replication, D_s, is a combined measure of the complexity ratio of the robot to individual modules, the complexity distribution over the individual modules, and the average complexity in each module defined by [3]

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

where C_{min} and C_{max} are the minimum and the maximum values among $\{C_1, \cdots, C_n\}$, $C_{total} = \sum_{i=1}^{n} C_i + I_{total}$, and $C_{ave} = 1/n \sum_{i=1}^{n} C_i$. For given numbers of active elements $(C_1^A, C_2^A, C_1^B, C_2^B) = (42, 52, 37, 46)$, the degree of self-replication is computed as

$$D_s = \frac{37}{52} \cdot \frac{191}{44.25^2} \simeq 6.57 \times 10^{-2}. \tag{2}$$

This result is higher than the values computed for three previous prototypes presented in [3]:

$$D_s^1 \ll 1.56 \times 10^{-6}; \quad D_s^2 \simeq 1.45 \times 10^{-2}; \quad D_s^3 \simeq 2.99 \times 10^{-2}.$$

In order to penalize an uneven complexity distribution among the modules more precisely, one can replace 'C_{min}/C_{max}' in (1) by a form of '$k/(1+\sigma_c)$' where σ_c is the standard deviation of the module complexities and k is a constant. We note that any of these two definitions can be used to quantify the complexity distribution as long as the same measure is applied to the systems being compared.

Configurational entropy changes are also computed based on the configurational entropy method [3], [22]. Since the environment does not hold any information about the resource modules, no uncertainty is reduced by structuring the environment. Therefore, the entropy reduction resulting from structuring the environment can be considered simply zero. The amount of uncertainty reduced by self-replication can be computed as the difference between the configurational entropy computed for four modules randomly placed in a bounded environment and the entropy remaining in these modules after being completely assembled. We assume that the effect of physical overlaps among these modules in entropy computation is trivial and neglected.

If each module can have any position and orientation within the boundary, $\mathscr{X}_i^u = [0,1067]$, $\mathscr{Y}_i^u = [0,1067]$ and $\mathscr{Z}_i^u = [0,2\pi]$ for all $i = 1,\cdots,4$. For $\varepsilon_p = 0.5$ [mm] and $\varepsilon_r = 0.01$ [radian], the numbers of possible positions and orientations are given by

$$\alpha_i^u = \frac{1067}{\varepsilon_p} = 2134; \ \beta_i^u = \frac{1067}{\varepsilon_p} = 2134; \ \gamma_i^u = \frac{6.28}{\varepsilon_r} = 628.$$

The corresponding configurational entropy for a module randomly placed in the environment is computed as

$$\widehat{H}_i^u = \log_2 \alpha_i^u + \log_2 \beta_i^u + \log_2 \gamma_i^u \simeq 31.41.$$

Since all modules are about the same size, when assuming that each module is placed independently from the others, the total configurational entropy is computed as $\widehat{H}^u = \sum_{i=1}^4 \widehat{H}_i^u \simeq 125.65$.

Due to the geometric design, the modules have fairly small tolerances in the positions and orientations when they are assembled to form a functional robot. These tolerances are estimated as $\delta g_i^a = (0.5, 0.5, 0.03)$, and the corresponding entropy for each module is

$$\widehat{H}_i^a = \log_2 1 + \log_2 1 + \log_2 3 \simeq 1.59.$$

The total entropy for four assembled modules is given by $\widehat{H}^a = \sum_{i=1}^4 \widehat{H}_i^a \simeq 6.34$. Based on the definition in [3], the configurational entropy reduced by self-replication is then computed by

$$\Delta H_R = \widehat{H}^u - \widehat{H}^a = 119.31. \quad (3)$$

This value is higher than the results presented in [3]:

$$\Delta H_R^1 \simeq 30.80; \quad \Delta H_R^2 \simeq 54.84; \quad \Delta H_R^3 \simeq 90.97.$$

The same measures can be applied to other self-replicating or self-assembling systems to provide a useful insight about the overall system changes in terms of structural complexity and information associated with the process.

V. Discussion and Conclusion

This paper presented a new robotic system that demonstrated self-replication from low-complexity components by extending our continuing effort on self-replicating robots without computer control. The parent robot composed of four heterogeneous modules duplicated by growing itself and separating in the middle resulting in two robots. The checkerboard environment functions as a passive catalyst that keeps the robot within the boundary for a certain amount of time while covering the most of the area to locate and assemble the resource modules. This environment does not contain any information about the parent robot or the resources. In addition, the structural complexity of each resource module was even further simplified and the amount of configurational entropy reduced by self-replication was increased compared to the previous prototypes presented in [3].

References

[1] K. Lee and G. S. Chirikjian, "Robotic Self-Replication from low complexity parts," *IEEE Robotics and Automation Magazine*, 14(4): 34-43 (2007)

[2] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts and P. Walter, Molecular Biology of the Cell, Academic Internet Publishers (2002)

[3] K. Lee, M. Moses and G. S. Chirikjian, "Robotic Self-Replication in Structured Environments: Physical Demonstrations and Complexity Measures," *Int. Jour. of Robotics Research*, 27(3-4): 387-401 (2008)

[4] V. Zykov, S. Mytilinaios, M. Desnoyer, and H. Lipson, "Evolved and designed self- reproducing modular robotics," *IEEE Transactions on Robotics*, 23(2):308-319 (2007)

[5] J. von Neuman and A. W. Burks, *Theory of Self-reproducing Automata*, IL: Univ. of Illinois Press (1962)

[6] R. A. Freitas and R. C. Merkle, "Kinematic Self-replicating Machine," Georgetown, TX: Lands Bioscience (2004)

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

[8] L. S. Penrose, "Machines of self-reproduction," *Ann. Hum. Genet.*, 23:59-72 (1958)

[9] L. S. Penrose, "Self-reproducing machines," *Sci. Am.*, 200(6):105-114 (1959)

[10] H. Jacobson, "On Models of Reproduction," *American Scientist*, 46:255-284 (1958)

[11] J. Suthakorn, A. B. Cushing and G. S. Chirikjian, "An autonomous self-replicating robotic system," in *Proc. IEEE Intl. Conf. on Advanced Intelligent Mechatronics, AIM'03*, pp. 137-142 (2003)

[12] H. Lipson, "Homemade: The Future of Functional Rapid Prototyping," *IEEE Spectrum*, pp. 24-31 (May 2005)

[13] A. Bowyer, "The Self-replicating Rapid Prototyper-Manufacturing for the Masses," in *Proc. 7th National Conference on Rapid Design, Prototyping & Manufacturing*, Center for Rapid Design and Manufacture, High Wycombe (June 2006)

[14] G. Kaloutsakis and G. S. Chirikjian, "Self-replicating Robot in an Unstructured Environment," in *Proc. Romansy'08* (July 2008)

[15] P. J. White, K. Kopanski and H. Lipson, "Stochastic Self-reconfigurable Cellular Robotics," in *Proc. ICRA 2004*, 3(26): 2888-2893 (2004)

[16] S. Griffith, D. Goldwater and J. M. Jacobson, "Self-replication from Random Parts," Nature, 437:636 (2005)

[17] N. Napp, S. Burden and E. Klavins, "The Statistical Dynamics of Programmed Self-assembly," in *Proc. ICRA 2006*, pp: 1469-1476 (2006)

[18] T. Fukuda, S. Nakagawa, Y. Kawauchi and M. Buss, "Self Organizing Robots Based on Cell Structures - CKBOT," in *Proc. IEEE Intl. Workshop on Intelligent Robots*, pp. 145-150 (November 1988)

[19] E. Yoshida, S. Murata, A. Kamimura, K. Tomita, H. Kurokawa and S. Kokaji, "A Self-Reconfigurable Modular Robot: Reconfiguration Planning and Experiments," *International Journal of Robotics Research*, 21(10): 903-916 (2003)

[20] M. Yim, D. Duff and K. Rufas, "PolyBot: a Modular Reconfigurable Robot," in *Proc. IEEE ICRA02'*, pp. 514-520 (2000)

[21] W. Shen, P. Will and B. Khoshnevis, "Self-assembly in Space via Self-reconfigurable Robots," in *Proc. IEEE ICRA'03*, pp. 2516-2621 (2003)

[22] A. C. Sanderson, "Part Entropy Method for Robotic Assembly Design," in *Proc. International Conference on Robotics* (1984)

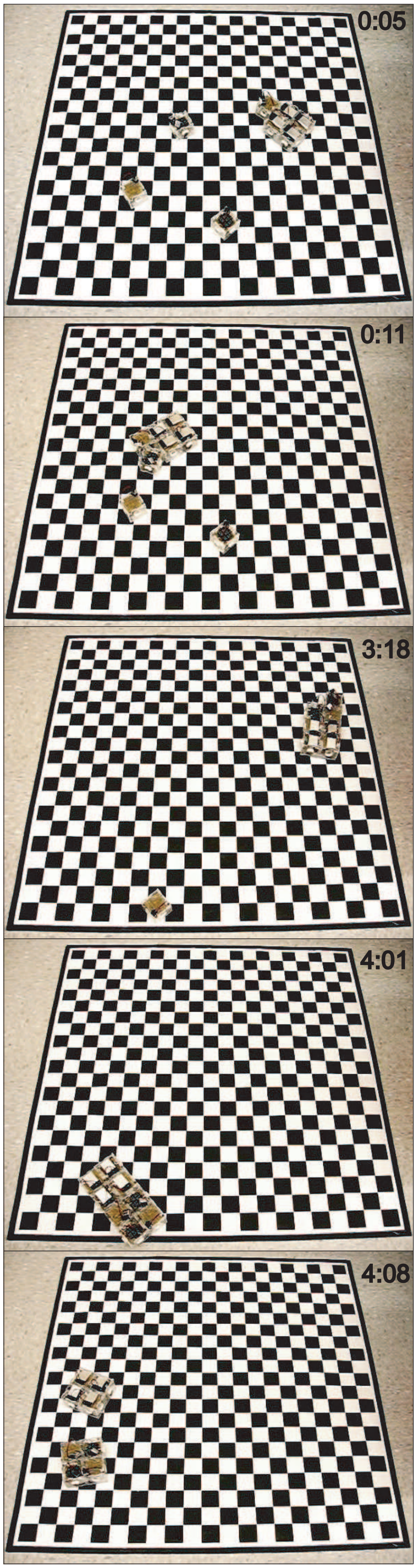

Fig. 6. Experimental snapshots 1: Initial poses of four resource modules and the parent robot were arbitrarily selected as shown in the pictures.

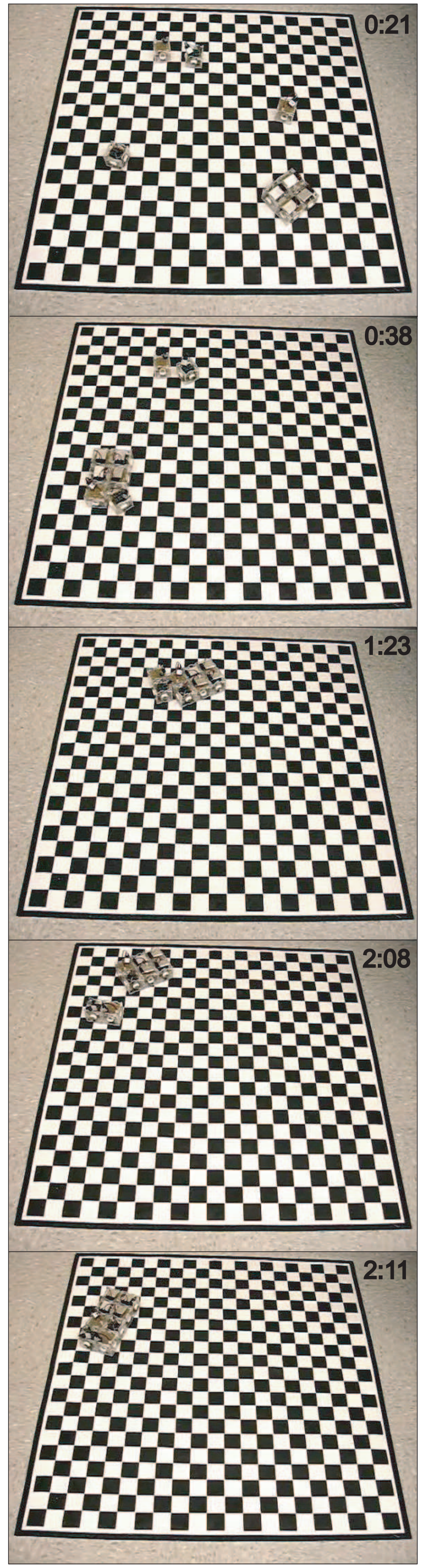

Fig. 7. Experimental snapshots 2: Initial poses of four resource modules and the parent robot were arbitrarily selected as shown in the pictures.