Motion Planning for a Self-Reconfigurable Modular Robot

Eiichi Yoshida, Satoshi Murata, Akiya Kamimura,
Kohji Tomita, Haruhisa Kurokawa and Shigeru Kokaji
Mechanical Engineering Laboratory,
1-2 Namiki, Tsukuba-shi, Ibaraki 305-8564 Japan
eiichi@mel.go.jp

Abstract: This paper addresses motion planning of a homogeneous modular robotic system. The modules have self-reconfiguration capability so that a group of the modules can construct a robotic structure generating dynamic motion. Motion planning for self-reconfiguration is not straightforward because the modular structure allows many combinatorial configurations, and also the proposed module has only two degrees of freedom. We will show a motion planning method for a particular class of multi-module structure, based on global planning and local motion scheme selection. The fundamental module motion will also be demonstrated through hardware experiments.

1. Introduction

Reconfigurable robotic systems have been attracting more interest, as their feasibility has been examined through hardware and software experiments in recent years [1]–[7]. This paper focuses on self-reconfigurable and homogeneous modular robotic systems that can adapt themselves to the external environment by changing their configuration. They can also repair themselves by using spare modules without external help owing to homogeneous modular structure. They have various potential applications, especially for structures or robots that should operate in extreme environments inaccessible to humans, for instance, in space or deep sea, or in nuclear plants.

The reconfigurable modular robotic system are classified into two types, lattice type [1, 2, 3] and thread type [5, 6, 7]. The former corresponds to a system where each module has several fixed connection directions, and a group of them can construct various static structures like jungle-gym. However, it is difficult for such a system to generate some dynamic robotic motions. On the other hand, the latter has snake-like shape that can generate various dynamic motions, nevertheless self-reconfiguration is difficult.

We have therefore developed a new type of modular robotic system that can realize both static structure and dynamic robotic motion. This has been realized by simplified design of a module and connecting mechanism. Fundamental reconfiguration motions are demonstrated in the experiment section of the paper.

This paper addresses some aspects of motion planning of a modular robotic system. There have been a number of studies on reconfiguration method for lattice-type modular robots. We have developed a series of distributed self-reconfiguration methods for two-dimensional and three-dimensional homogeneous modular robots [8, 9].

These methods enabled homogeneous modular robots to self-assemble and self-repair in a distributed manner based on local inter-module interactions. In contrast, most of other methods are based on centralized planning. Kotay et. al [10] developed robotic modules called "Molecules," and described a global motion synthesis method for a class of module group to move in arbitrary directions. Ünsal et. al [11] reported two-level motion planners for a bipartite module "I-Cube" composed of cubes and links, based on heuristic graph search between module configurations. These methods are dedicated to modules that have sufficient degrees of freedom to move to desired neighboring lattice position. Since our module has incomplete spatial symmetry, the formerly developed motion planning method cannot be directly applied.

We propose a two-layered motion planning method, a global and local planners to transfer a class of module cluster, which is classified into a centralized method. The former part of the planner provides the *flow* of the cluster, which corresponds to a global movement. The latter generates local coordinated motions called *motion schemes* in consideration with incomplete spatial symmetry of modules. The planned motion have not been fully implemented yet in the hardware, however we will discuss issues which arise when the planned motions are executed.

2. Hardware Design

The developed module consists of two semi-cylindrical parts connected by a link (Fig. 1). Servomotors are embedded in the link so that each of parts can rotate by 180°. Figure 2 shows a hardware prototype of the module. Each module has six connecting surfaces (three for each part) that can attach and detach other modules by using magnets and shape memory alloy (SMA) actuator [12]. The connecting surfaces have also electrodes for power supply and serial communication. All the connected modules can be supplied power from one module connecting to the power source. This eliminates the tether entanglement that becomes significant in three-dimensional configuration.

Each module is equipped with a PIC microprocessor that drives servomotors and SMA actuators. In the current development, all the modules are controlled from a host PC that provides motion commands through serial communication lines. The size of one semi-cylindrical part is 6cm cube and a module weighs approximately 400g.
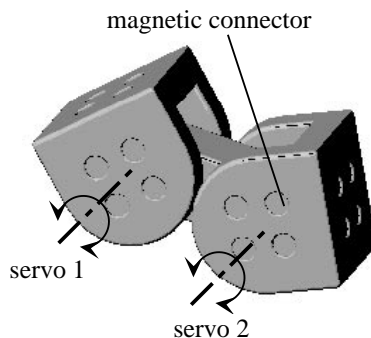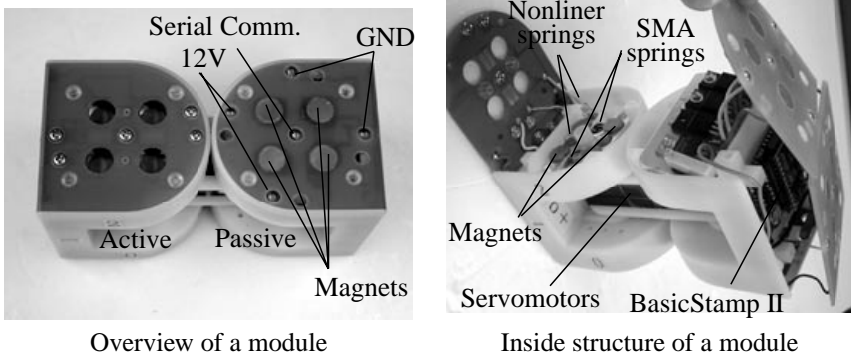

Figure 1. A robotic module.

Overview of a module   Inside structure of a module

Figure 2. A hardware module.

3. Reconfiguration Motion Planning

We have shown the proposed module can generate various shapes such as legged walking robot or slime-like locomotion machine [12]. However, the motion planning from one configuration to another is not straightforward because the proposed modules' mobility strongly depends on its local configuration. Therefore, we deal with the motion planning to a particular class of module cluster as a basic case. We take a two-layered approach, namely global planner and local motion scheme selector. The global cluster movement, called *flow*, is planned in the former part as a set of paths of modules from initial and goal position. The latter outputs a sequence of *atomic motions* for local reconfiguration, called *motion schemes* based on a rule database to realize the generated paths. The *atomic motions* is the simplest module motion by one or two modules which will be described in section 3.1. The incomplete spatial symmetry is considered in the motion scheme selector by means of a rule database of feasible local coordinated motion associated with corresponding local configuration.

In the following, we suppose that only one motion scheme is allowed during the cluster flow for simplicity. Another assumption is that one module can lift only one other module in the planning, which comes from the limited torque capacity of the hardware.

3.1. Atomic Motion

There are mainly three types of atomic motion, *pivot mode*, *forward-roll mode* and *mode conversion*. Figures 3 and 4 show two different atomic motions on a plane, forward-roll and pivot, whose orientation of rotational axes are in different direction. Mode conversion is a two-module motion to convert from one mode to the other, where a helper module is required as illustrated in Fig. 5.

To construct three-dimensional structures, a cluster of module should include at least one module in forward-roll mode and pivot mode.

3.2. Cluster Flow and Global Planner

As a basic case of motion planning, we consider a class of module cluster mainly composed of two layer of pivot mode modules and a couple of forward-roll mode modules (Fig. 6). As this cluster includes both modes of atomic motion, it can change there configuration in three dimensions. The connectivity condition of the whole cluster is
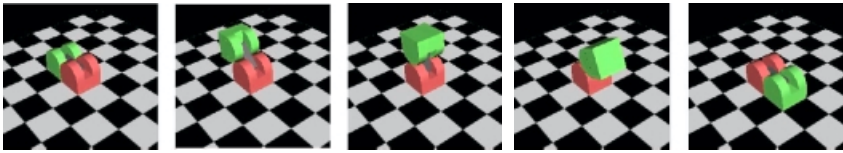
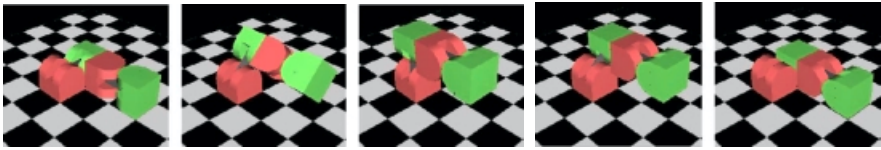Figure 3. Forward-roll mode.

Figure 4. Pivot mode.

Figure 5. Mode conversion from pivot to forward-roll.

satisfied by placing the modules so that the direction of pivot mode (i.e. the direction of links between semi-cylindrical parts) is different in each layer.

We define the cluster *flow* as *paths* of the *tail* modules towards the *head* (from Fig. 7a to 7d). A *path* denotes a rough routing of a module to move from the head position to the tail. The *global planner* is in charge of generating these paths of modules to realize the desired flow. While there are several ways of generating reconfiguration motion of this kind of cluster, we adopt a simple conveyer-like motion to realize the desired flow. The tail modules move toward the heads by using forward-roll and some coordinated atomic motions on the side of the cluster (Fig. 7b and 7c). They become new heads when they reach the other end of the cluster. The next tails will be sent to
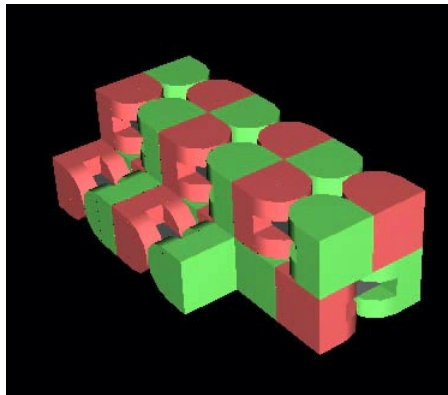

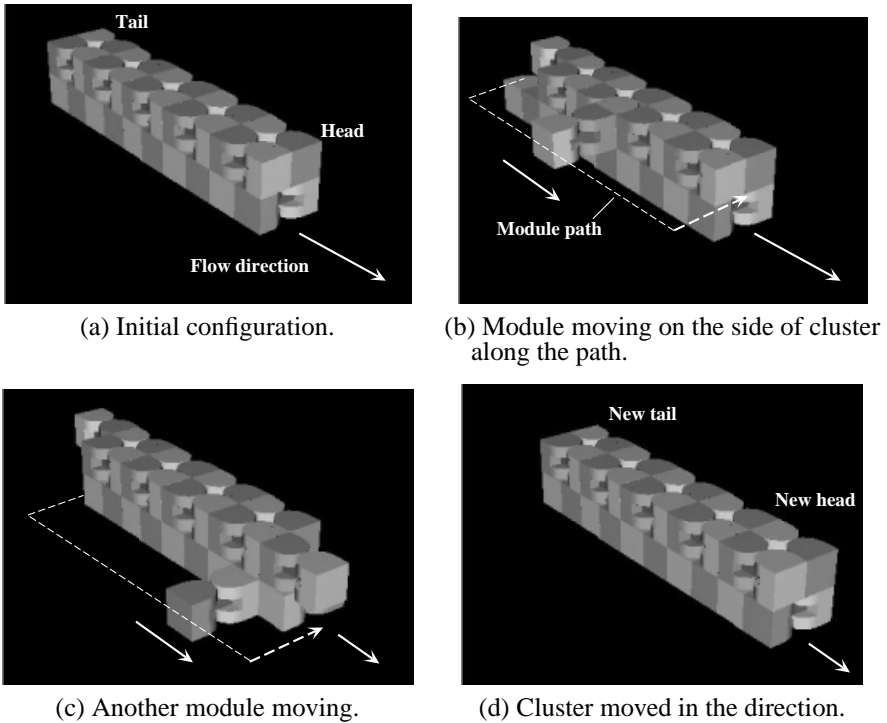Figure 6. A cluster composed of two layers of pivot mode modules with two converter modules.

(a) Initial configuration.

(b) Module moving on the side of cluster along the path.

(c) Another module moving.

(d) Cluster moved in the direction.

Figure 7. Cluster flow based on "conveyer-like" module transfer.

the heads, and so forth. After these global paths are generated, the detailed sequence of motions for local configuration, named *motion scheme*, will next be determined by the *motion scheme selector*.

3.3. Motion Scheme Selector

After the global planner outputs the module path, appropriate *motion schemes* should be selected to achieve the paths, considering connectivity condition and collision avoidance. The *motion scheme selector* does this job based on a database of rules for local coordinated motion. Each rule includes a motion scheme associated with an initial configuration of involved modules that is described as a graph tree (Fig. 8). If a rule matches the current configuration, the associated motion scheme is added to the candidates to be applied. Then the best motion scheme for desired path is selected to update the motion plan.

In order to implement the motion scheme selector, we extracted several fundamental motion schemes as follows.

1. rolling on a side of the cluster (Fig. 8)
2. conveying a module by right-angle on a plane (Fig. 9)
3. converting the rotational axis of a module (Fig. 10)

Figure 8 shows a rule corresponding to a simple motion scheme of the rolling on the side of the cluster. Figures 9 and 10 illustrate how module configuration changes
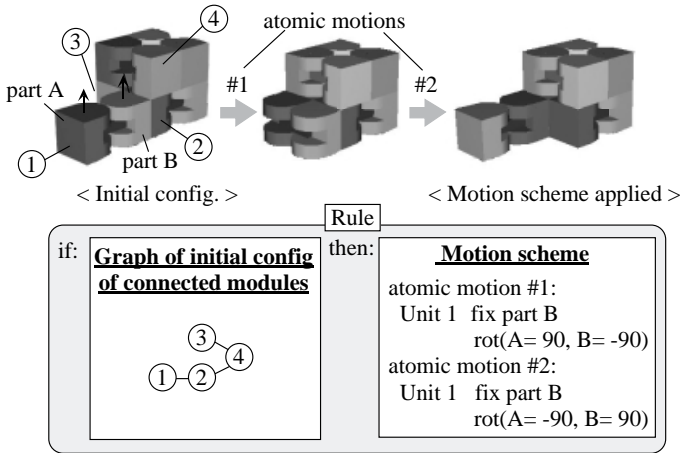
Figure 8. Example of a rule for a rolling motion scheme
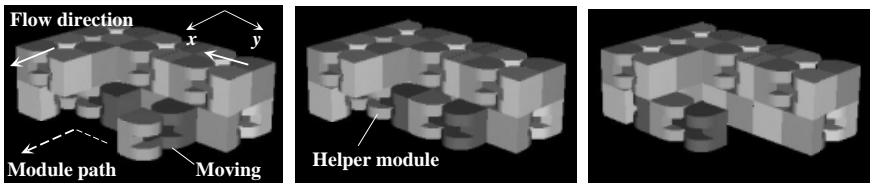

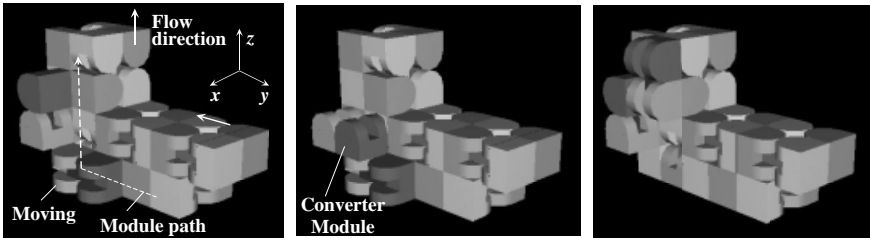Figure 9. Direction change of cluster on a plane.

Figure 10. Direction change to vertical direction on a plane.

in the latter two motion schemes. Supposing that the initial module cluster are located on $x - y$ plane, the direction change between $x - y$ axis is done by alternating the layers (Fig. 9). The converter modules are used when the desired cluster flow requires change of rotational axis of the module (Fig. 10). The number of converter modules can be augmented if necessary.

3.4. Planning Results

The motion planning method can generate simple three-dimensional path and is scalable, i.e. applicable to variable number of modules. If modules are equipped with some external sensors, the planning method will allow the module cluster to move around in unknown environments with bumps or walls, adapting its shape to the outside world.

Figure 11 show some snapshots taken from planned motion of a cluster of twenty-
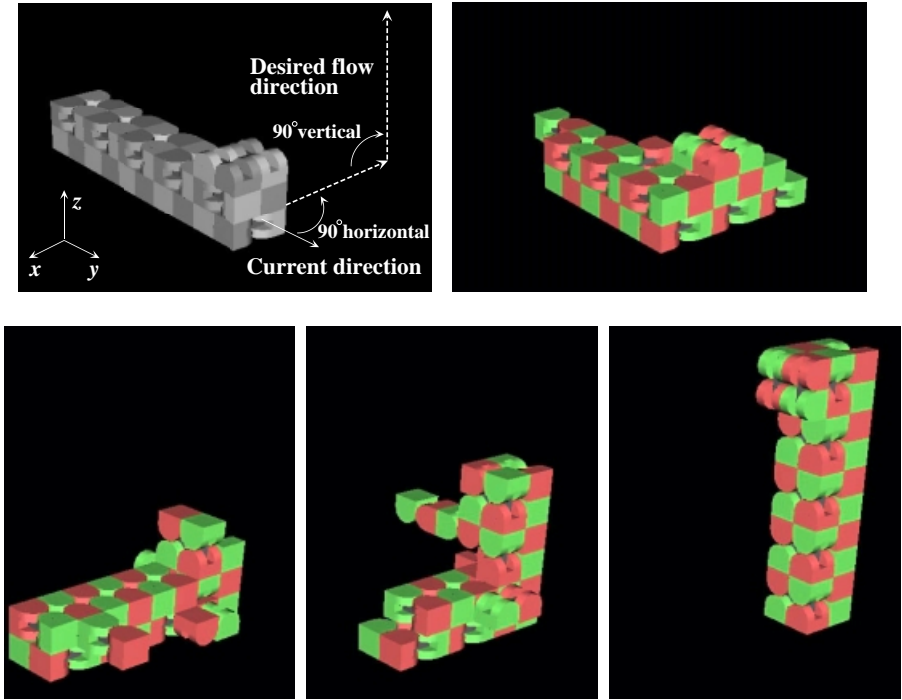
Figure 11. Simulated plan of motions in different flow directions from initial configuration on a plane.

two modules starting from a configuration on a plane. The cluster first changes its flow direction on the horizontal plane, then moves in vertical direction. To reduce the time required for reconfiguration, the simultaneous motions of several modules should be implemented by merging multiple module paths.

In the current implementation, the rule database does not cover all the situations that may occur. Thus the planner is not yet fully automated and human operator intervenes if the planner does not found applicable rules during planning.

4. Hardware Experiments

We are building hardware prototype of robotic modules. In this section, we will verify the motion capacity of hardware module and discuss some issues to be fixed in the future development.

Figure 12 and 13 show the experiment of forward-roll motion (Fig. 3) and mode conversion (Fig. 5). In these experiments, the connecting mechanism showed reliable performance; it has enough strength to hold the module against gravity and the smooth detachment is realized as well. We can also verify the module has sufficient torque to conduct certain two-module motions from Fig. 13. By combining these basic module motions, various motions are possible. Figure 14 is an example of combined motion using three modules.

Although we have verified the basic capacity of motion, the hardware still need

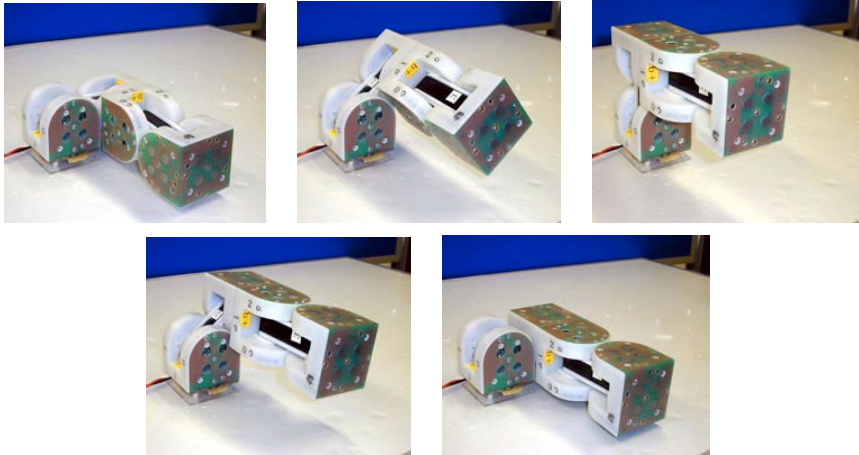Figure 12. Experiment of forward-roll motion.

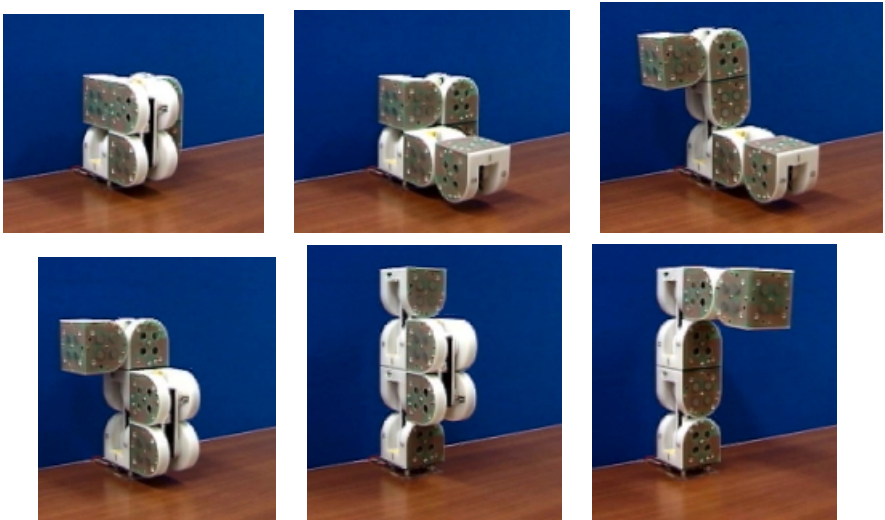
Figure 13. Experiment of mode conversion.

Figure 14. An example of motion sequence for three modules.

to be improved in several aspects to carry out the planned motion in section 3 using the hardware modules. The future improvements includes:

Redesigning electrodes. During the pivot motion (Fig. 4), it happens that the electrodes of power supply, ground and serial line of two modules contact, which causes short circuit. In the next prototype model, the electrodes will be allowed to slide with connecting mechanism to prevent the unnecessary contact.

Improving motor torque. Large torque is required for two-module motion in severest conditions. The actuator mechanism will be improved to provide enough torque by increasing gear reduction ratio.

Equipping modules with some sensors. By detecting the external environment, the group of modules are enabled to adapt to the various situations.

5. Conclusions

This paper discussed motion planning of a self-reconfigurable modular robot. The module was designed to generate both static structure and dynamic robotic motions. This module can form various three-dimensional shapes in spite of simple module design. On the other hand, it has incomplete spatial symmetry due to its limited degree of freedom, which imposes a stronger constraint on motion planning compared to ordinary lattice-type modules. The motion planning should consider this constraint, and we adopt a planning method based on global planning and motion scheme selector. The former outputs the global path to realize overall cluster flow, and the latter selects feasible motion schemes as a combined atomic motions based on a rule database associating a local configuration with a motion scheme. The constraints against module motion is considered in the latter part of the planner. The future work concerning the motion planner includes building a general global path-finding algorithm applicable to wider classes of configuration and investigating more compact and robust motion rule description. We also verified the basic functions of hardware modules through experiments. Some issues on hardware design were clarified to carry out the planned motions. After these refinement, we intend to demonstrate the capacity of self-reconfiguration and dynamic robotic motion, by using more than ten modules.

References

[1] Murata S, Kurokawa H, Yoshida E, Tomita K, Kokaji S 1998 A 3-D Self-Reconfigurable Structure. *Proc. 1998 IEEE Int. Conf. on Robotics and Automation*, 432–439

[2] Kotay K, Rus D, Vona M, McGray C 1998 The Self-Reconfiguring Robotic Molecule. *Proc. 1998 IEEE Int. Conf. on Robotics and Automation*, 424–431.

[3] Ünsal C, Kılıççöte H, Khosla P K 1999 I(CDS)-cubes: a Modular Self-Reconfigurable Bipartite Robotic System. *Proc. SPIE, Sensor Fusion and Decentralized Control in Robotic Systems II*, 246–257.

[4] Yoshida E, Kokaji S, Murata S, Tomita K, Kurokawa H 2000 Miniaturization of Self-Reconfigurable Robotic System using Shape Memory Alloy. *J. of Robotics and Mechatronics*, **12**-2: 1579–1585.

[5] Hamlin G, Sanderson A 1998 *A Modular Approach to Reconfigurable Parallel Robotics*. Kluwer Academic Publishers, Boston.

[6] Yim M and Casal A 1999 Self-Reconfiguration Planning for a Class of Modular Robots. *Proc. SPIE, Sensor Fusion and Decentralized Control in Robotic Systems II*, 246–257.

[7] Castano A, Chokkalingam R, Will P 2000 Autonomous and Self-Sufficient CONRO Modules for Reconfigurable Robots. *Distributed Autonomous Robotics 4*, Parker L E, Bekey G, Barhen J eds., Springer, 155–164.

[8] Yoshida E, Murata S, Kurokawa H, Tomita K, Kokaji S 1999 A Distributed Method for Reconfiguration of 3-D homogeneous structure. *Advanced Robotics*, **13**-4:363–380.

[9] Tomita K, Murata S, Yoshida E, Kurokawa H, Kokaji S 1999 Self-assembly and Self-Repair Method for Distributed Mechanical System. *IEEE Trans. on Robotics and Automation*, **15**-6:1035–1045.

[10] Kotay K and Rus D 1998 Motion Synthesis for the Self-Reconfigurable Molecule. *Proc. 1998 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 843–851.

[11] Ünsal C, Kılıççöte H, Patton M E, Khosla P K 2000 Motion Planning for a Modular Self-Reconfiguring Robotic System. *Distributed Autonomous Robotics 4*, Parker L E, Bekey G, Barhen J eds., Springer, 165–175.

[12] Murata S, Yoshida E, Tomita K, et al. 2000 Hardware Design of Modular Robotic System. *Proc. 2000 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, CD-ROM, F-AIII-3-5.