Development of a self-reconfigurable modular robotic system

Kohji Tomita, Satoshi Murata, Eiichi Yoshida, Haruhisa Kurokawa, Akiya Kamimura and Shigeru Kokaji
Mechanical Engineering Laboratory
Agency of Industrial Science and Technology
Ministry of International Trade and Industry
Tsukuba, 305-8564 JAPAN

ABSTRACT

We present a novel type of modular robotic system that has a capability of both autonomous shape reconfiguration and robotic motion generation. We have developed hardware modules and examined basic mechanical functions and reconfiguration. A simulation system of the modular robotic system has also been developed to design the reconfiguration sequence and motion for a cluster of the modules.

Keywords: Self-reconfigurable robot, distributed autonomous system, modular robot

1. INTRODUCTION

Homogeneous reconfigurable robotic systems, which are composed of one kind of autonomous modules, have considerable advantages compared with conventional robotic systems. Owing to its self-reconfigurability, it can change its shape without external help, and can perform several functionalities intrinsic to the shape. Moreover, such systems have potential ability of self-repair, which is hardly realized by conventional systems.

Recently, several homogeneous reconfigurable systems have been studied. They are classified into two categories: lattice type and linear type. The lattice systems (e.g., \cite{1-4}) are built by the same modules with some spatial symmetry, such as honeycomb or jungle gym. This type is suitable to construct various static lattice configurations, but difficult to generate group motion. The linear type systems (e.g., \cite{5-7}) are basically serial link robots made of many joint modules. Although this type is suitable for generating robotic motion, it is difficult to reconfigure because of its mechanical redundancy.

This paper presents a novel modular robotic system that has advantages of both types; capability of reconfiguration and generation of robotic motion \cite{8-10}. The system is composed of modules of one kind, which consists of two semicylindrical boxes and a link between them. Though each module has limited degree of freedom, i.e., rotation of the link and connection/disconnection mechanism, a cluster of the modules have abilities of various functions. For instance, a cluster of modules in a packed box configuration can metamorphose into a legged robot, and then, it can walk with the legs.

In this paper, we show module design, reconfiguration process by simulation, and design of prototype modules.

2. MODULE DESIGN

2.1. Geometrical Design

The robotic module consists of two semicylindrical boxes and a link between them (Fig. 1). Owing to its semicylindrical shape, each box can rotate by 180 degrees. Two servomotors are embedded in the link to control the rotational angles. The output axis of the servomotor is directly connected to the box. Three surfaces of each semicylindrical box are used to connect with other modules. Bus electrodes for electricity supply and communication channel are embedded on each connecting surface.

Correspondence: Email: tomita@mel.go.jp; WWW: http://www.mel.go.jp; Telephone: +81-298-61-7126; Fax: +81-298-61-7091
2.2. Motion Modes

Though each module has limited degrees of freedom, a cluster of the modules has ability of self-reconfiguration. We illustrate three basic methods of the reconfiguration: two translation modes and one conversion method between them. For easier understanding, we assume here that translation is carried out on a plane tiled by the same modules. The forward roll mode (Fig. 2(a)) is a linear translation on a line, where the module cannot change the direction in this mode. The pivot translation mode (Fig. 2(b)) enables the modules to traverse the plane to any direction. The third method converts these modes (Fig. 2(b)). The left module in the figure acts as a converter, and the right module is converted from the pivot translation mode to the forward roll mode.

![Magnetic connector](image1)

**Figure 1.** Geometrical design of robotic module.

![Translation by a forward roll](image2)

(a) Translation by a forward roll.

![Translation by pivot motion](image3)

(b) Translation by pivot motion.

![Mode conversion](image4)

(c) Mode conversion.

**Figure 2.** Basic reconfiguration methods.

3. MOTION SIMULATION

3.1. Simulator

We have built an interactive simulation system of the modular system. This allows us to design a sequence of reconfiguration, and to give motion on a module cluster.
The simulator is developed on the basis of OpenGL. It reads two files which describe an initial configuration and a motion sequence of the modules. The initial configuration of a module is decided by its position, orientation and two rotational angles of semicylindrical boxes.

The motion sequence is a list of “motion steps” that specifies the identifiers (IDs) of moving modules and their motion description using the rotational angles of the semicylindrical boxes. Figure 3 shows an example of a step motion sequence. We assume that the “Part A” of Module 1 is fixed to the its underlying modules (not drawn in this figure). In this example, Module 1 carries Module 2 by rotating the “part A” by 90 degrees clockwise as the configuration in Fig. 3(a) is changed into that in Fig. 3(b). The description corresponding to this motion step is given as a series of “name=value” pairs as follows:

ID=1 ; the ID of module in motion.
fix=A ; the fixed part during the motion.
rot=(90, 0) ; the rotation angle of each part A and B.
with=1 ; the number of the carried modules.
ID=2 ; the ID of the carried module.

where comments are added after “;”. The simulator processes these motion steps in the file at each “simulation time step” and updates the positions and orientations of modules. Concurrent motions of multiple modules can be simulated by giving several motion steps at one simulation time steps, as long as no collisions between modules occur.

![Figure 3. Example of step motion.](image)

Main functions of the simulator are:

- Graphical display of the modules. Motion of the modules is animated.
- We can edit the motion sequence (adding/deleting steps and changing motion of a step). The new sequence can be stored as a motion sequence file.
- Displaying internal states of the modules, such as the position, direction, and angles of the links.
- GUI facilities: zooming in and out, changing the view point, and visible/invisible objects such as floors and walls.

Figure 4 shows the state after opening the edit window.

3.2. Motion Programming

Basic motions were shown in section 2.2. We design motion for a cluster of the modules by combination of these basic motion modes. Let us consider a linear cluster composed of forward roll mode modules. We need at least two lines so that the cluster is not divided while it is reconfiguring. Figure 5 shows a sequence of the cluster to move rightward. (Only one line is shown in the figure.) At first, it makes a foothold for the leftmost two modules (Fig.
Then the modules rotate (d), connect with the foothold (e), and stand on it (f). By repeating the process the cluster can move rightward. Clusters of this type are suited for up-and-down paths (Fig. 6), but its mobility is restricted in one direction.

On the other hand, a cluster of pivot translation mode modules are suited for horizontal motion. Again we need two layers to maintain the connectivity. Figure 7(a) shows a two layered cluster. Top view and the first layer is shown in Fig. 7(b) and (c), respectively. Directions of the modules in these layers differ in order that the cluster is not divided. Horizontal motion is achieved by similar method to the case of forward roll modules. Vertical motion is not possible using only this mode modules as contrasted with the former one.

In order to maintain self-reconfigurability, clusters of modules must include both forward roll mode and pivot translation mode. Figure 8 is two examples of such clusters. In Fig. 8(a), the main cluster is in forward roll mode
**Figure 6.** Up-and-down motion by a forward roll cluster.

![Figure 6](image)

**Figure 7.** A cluster of pivot translation mode modules.

(a) ![Image](image) (b) ![Image](image) (c) ![Image](image)

and two side modules are in pivot translation mode, and (b) is the opposite. Each cluster can reconfigure into the other.

Next we consider a more complicated configuration change (Fig. 9). A box cluster of 22 modules reconfigures into a legged robot of 12 modules in about 200 steps, and then the robot begins a walking motion. Note that we do not assume that the floor is tiled by the same modules. In the initial state, the cluster is composed of a main box cluster of 20 modules in the pivot translation mode and two modules in forward roll mode at the side of the cluster. Each segment of the walking robot requires mode conversion for its leg module. Here, two side modules work as a

![Figure 8](image)

**Figure 8.** Clusters of rolling mode and pivot mode.
mode converter. We design a special sequence that rearrange the initial regular lattice structure into the articulated leg structure. We must omit the detailed process because of limited space.

![Building a legged robot and its walking motion.](image)

**Figure 9.** Building a legged robot and its walking motion.

4. HARDWARE MODULE

4.1. Hardware Implementation

4.1.1. Connection mechanism

The modules connect each other at three surfaces of each semicylindrical box. The connection mechanism using rare-earth metal permanent magnets attaches the adjacent boxes firmly. In order to reduce the force to detach the magnets, we adopt the idea of the internally-balanced magnetic unit\(^{(13)}\) using non-linear springs. Springs are designed to have slightly lower force than the magnets when they are compressed. Detachment is thus possible with relatively small additional force without lessening connection force. SMA (shape memory alloy) coil springs are used to reverse the balance between the magnets and the springs.

4.1.2. Control circuit

We adopt a PIC processor Basic Stamp II. The PIC generates control signal for servomotors and PWM output for SMAs.

In this first prototype, we adopt a centralized control method. A host PC controls the modules through a serial bus. The PC sends operation command consists of 5 bytes including module ID, SMA control, and reference angles of servo motors. PIC on each module decodes the commands and generates necessary control signals for actuators.

4.1.3. Bus connection

Each module has six connection surfaces, three passive and three active (Fig. 10). When the modules are connected, electricity supply and the serial communication channel must be established at the same time. For this purpose, we need at least five contact points on each surface.
4.2. Hardware Specification

We developed two modules as the first prototype based on the above design (Fig. 11). The specification is as follows:

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>66 mm (size of semi-cylindrical box)</td>
</tr>
<tr>
<td>Weight</td>
<td>400 g</td>
</tr>
<tr>
<td>Servomotor</td>
<td>Hitec 6.6 kg-cm</td>
</tr>
<tr>
<td>Processor</td>
<td>Parallax, Basic Stamp II, 20 MHz</td>
</tr>
<tr>
<td>Power supply</td>
<td>DC 12 V</td>
</tr>
<tr>
<td>Connection strength (to attach)</td>
<td>3.6 Kg</td>
</tr>
<tr>
<td>Reduced connection strength (to detach)</td>
<td>0.3 Kg</td>
</tr>
<tr>
<td>Communication Rate</td>
<td>9.600 bps</td>
</tr>
</tbody>
</table>

Figure 10. Connection surfaces.

Figure 11. Prototype modules.

Using a few modules, we have conducted experiments, and confirmed basic functions such as power supply, data communication, and simple reconfiguration.

5. CONCLUSIONS

We presented a novel modular robotic system that has capability of both reconfiguration and motion generation. We have developed hardware modules, and confirmed basic functions. A simulator of the system is developed, and we showed that several complicated process are planned using the simulator.
We have much future work. We are to conduct basic experiments of more hardware modules to show the feasibility of the system. Ten modules are currently being developed. More decentralized control will be employed in the next prototype. In the next model, six independent serial I/O is necessary on each surface for module-to-module communication. In terms of the simulator, we need to include physical constraint such as geometrical interference and static/dynamic analysis of the system.

A next important target is autonomous reconfiguration. The complicated motion sequences in section 3.2 were hand-coded using the simulator. We begin with simple motion sequences for a fixed simple configuration. For instance, if we consider a small cluster of forward roll modules, we can automatically generate the reconfiguration process of getting over several walls. However, reconfiguration into an arbitrary shape is quite difficult, because of its enormous search space. Functions adapting to the environment will be achieved on a hierarchy of such kind of basic functionality.

REFERENCES