Distributed Self-reconfiguration Control of an M-TRAN System

Haruhisa Kurokawa, Kohji Tomita, Akiya Kamimura, Satoshi Murata*, Shigeru Kokaji
National Institute of Advanced Industrial Science and Technology (AIST)
* Tokyo Institute of Technology

A third prototype of a self-reconfigurable modular robot, called M-TRAN III, has been developed. Our current research objective is experimental verification of autonomous and distributed self-reconfiguration using more than 20 modules. Key issues are performance and reliability of each module hardware, a design of self-reconfiguration procedures, design of a distributed controller, and so on.

Hardware The size and weight (65 × 65 × 130 mm, 420 g) of the new prototype is similar to the former prototypes[1)]. Its connection mechanism is much faster (about 5 s) and less power consuming (1 W for 5 s) than the former. Each module has a 32 bit computer mutually connected by CAN (Controller Area Network) bus. Besides electrical connection for the CAN bus, there is a one bit connection, by which a module can detect whether it has neighbors. Each module has a 3D acceleration sensor to measure the gravitational direction, and 10 infrared transmitter and receiver pairs for proximity sensing. A lithium polymer battery in each module enables operation more than 30 minutes.

Self-reconfiguration procedure and simulators We have been developing various procedures of self-reconfiguration based on the idea of regular structure[2)]. They are suitable for parallel operation such that a group of neighboring modules works cooperatively by local communication and several groups work simultaneously. In order to design and verify a self-reconfiguration procedure, we have developed various programs, called M-TRAN simulators and tools. The former simulators are only useful to verify globally synchronous operation. The latest simulator contains a code of the distributed controller which is source code compatible to the onboard controller, and emulates all the parallel operations of modules. Dynamics simulation is not used but duration of each motor or connector motion can be made random in order to verify a program for synchronization.

Onboard controller The onboard controller is designed so that both a centralized single-master control and a fully decentralized and asynchronous control are easily implemented. The core of the controller is a virtual CPU with a communication channel. The virtual CPU interprets and executes an incoming command either from a program memory, from another module's CPU via a communication channel, or from the host PC. According to the command, usual operations of a general CPU as well as joint angle control or connection control are made. By this mechanism, two important functions, remote control of another module and shared memory, are realized. They are useful not only for a single master control but also for a decentralized control.

Experiments Homogeneity of the system is sought in programming, i.e., all the modules start with the same program. It is made that difference in modules' behaviors after startup are due to the difference of their local connective relations. After manual design of program and verification by the simulator, the same program is loaded to all the module and executed. Fig. 1 is an example of such decentralized and homogeneous control of self-reconfiguration. Two modules (a, b in Fig. 1) on the edge ascend, move on the plane made of other modules, and reach the other edge. Then other two (c, d) follows. The program for each module is 394 command lines in mnemonic and 963 bytes.

There still remain problems. Some trials of experiments failed either by a communication failure or by misalignment caused by gravitational force. The latter problem is not simple, as alignment accuracy depends on mechanical rigidity, joint control performance, and sequences of reconfiguration. However, we believe most of the problems will be solved by improvement of software and sequence design.

Local communication We are currently developing a program for neighbor to neighbor communication by the IR devices on all the connection surfaces. It is very slow (about 150 bps) compared with the global communication but may be adequate because hardware motion is not fast and simultaneous communication in parallel is possible. By using this communication, unique ID to distinguish each module can be removed from the controller to make the system fully homogeneous and scalable.

Future work Autonomous self-reconfiguration is the next objective. By using proximity sensing, a structure such as in Fig. 1 decides its motion and either makes a planar configuration or move to one of the four directions. Distributed decision making must be developed for that.

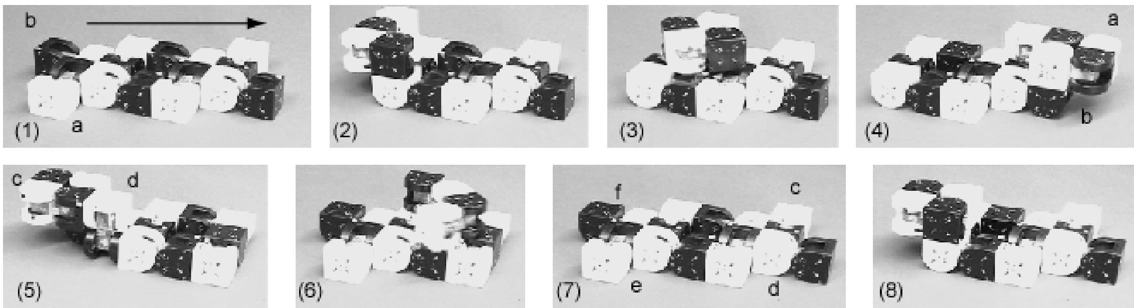

Fig. 1 Experiment of distributed self-reconfiguration by 8 modules

One cycle motion from (1) to (7) took about 8 minutes.

Reference

[1] Murata S, Yoshida E, Kamimura A, Kurokawa H, Tomita K, Kokaji S (2002) M-TRAN: Self-Reconfigurable Modular Robotic System, IEEE/ASME Trans. Mech. 7-4:431-441.

[2] Ostergaard EH, Tomita K, Kurokawa H (2004) Distributed Metamorphosis of Regular M-TRAN Structures, Distributed Autonomous Robotic Systems 6 (to appear).