M-TRAN II: Metamorphosis from a Four-Legged Walker to a Caterpillar

Haruhisa Kurokawa, Akiya Kamimura,
Eiichi Yoshida, Kohji Tomita, and Shigeru Kokaji
National Institute of Advanced Industrial Science and
Technology (AIST)
1-2-1 Namiki, Tsukuba, Ibaraki, 305-8564 Japan
{kurokawa-h, kamimura.a, e.yoshida, k.tomita,
s.kokaji}@aist.go.jp

Satoshi Murata
Interdiciplinary Graduate School of Science and
Engineering, Tokyo Institute of Technology
4259 Nagatsuta-cho, Midori-ku, Yokohama, 226-8502 Japan
murata@dis.titech.ac.jp

Abstract – We have been developing a self-reconfigurable modular robotic system (M-TRAN) which can make various 3-D configurations and motions. In the second prototype (M-TRAN II), various improvements are integrated in order to realize complicated reconfigurations and versatile whole body motions. Those are a reliable connection/detachment mechanism, on-board multi-computers, high speed inter-module communication system, low power consumption, precise motor control, etc. Programing environments are also integrated to design self-reconfiguration processes, to verify motions in dynamics simulation, and to realize distributed control on the hardware. Hardware design, developed software and experiments are presented in this paper.

I. INTRODUCTION

A self-reconfigurable modular system is a mechanical realization of an autonomous distributed system [1-5]. It is expected to be highly adaptive to unpredictable environment by its metamorphic capability and reliable by self-repair capability due to its multi-module structure. Such advantages are useful to search and rescue operations or exploration on planets.

We have been developing "Modular Transformer (M-TRAN)" as a general-purpose modular system [2]. It is designed to realize three-dimensional self-reconfiguration and versatile motions such as locomotion. By experiments with the first model, we verified its ability of both above, which included metamorphosis from a crawler to a quadruped walking robot [2].

In order to realize more complicated self-reconfiguration and more versatile whole body motions, we have developed the second prototype (M-TRAN II) aiming at more reliable operation. The design objectives were 1) stand-alone operation (tethers restrict free reconfiguration and motions), 2) firm connection and reliable detachment mechanism, 3) distributed control by enhanced information processing and inter-module communication. For these purposes, improvements has been made such as downsizing, optimization of connection/ detachment mechanism, low power consumption for battery-driven operation, precise motor control, and upgraded on-board computers. By those improvements, hardware experiments of self-reconfiguration were successfully carried out.

An algorithm to obtain reconfiguration and motions of modular system for a given task is still a difficult problem [1,2,6-10]. We have developed several software such as an interface program supporting a human operator to design a reconfiguration process [11], an automatic planner of reconfiguration processes [12] and pattern optimizer for coordinated locomotive motions [13]. There are also works by other researchers for the M-TRAN which uses a cellular automata method [14] and a Petri net method [15]. Considering those researches, we are integrating the program environment by which such algorithms can be installed and tested by the hardware.

This paper describes the hardware design and control architecture of M-TRAN II including detailed design of the connection mechanism, multi-CPU and an inter-module communication system in Section 2. In Section 3, integrated software for multi-module operations is described, and experimental results are shown in Section 4. Section 5 summarizes future works.

II. HARDWARE DESIGN OF M-TRAN

The essential point of M-TRAN system is in its simple mechanical design. Its small DOF and easy connection/ detachment enable it to use as a platform of a modular robot [2]. The second prototype (M-TRAN II) has been developed aiming at more reliable operation for self-reconfiguration, powerful actuation for coordinated whole body motion, and enhanced information processing ability for the distributed control.

A. Concept and Key Technology of Module

The module is composed of three components, two semi-cylindrical blocks and a link (Fig. 1). Each semi-cylindrical block can change its angle from -90 to 90 degrees independently. Two modules can connect with each other by contacting at their connection surfaces. Various 3-D structures can be constructed by using many modules. The shape of the 3-D structure can be changed by a sequence of primitive processes of a few modules, e.g., releasing some connections, changing angles to move semi-cylindrical blocks or modules, and reconnecting at different places.

In designing the M-TRAN II hardware, we aimed to reduce mechanical restrictions in order to apply possible reconfiguration processes given by algorithmic researches. For this purpose, connection mechanism must be firm enough and also easy to release. Connection should be also fast enough and reliable without precise alignment.

Powerful actuation is also necessary to realize efficient reconfiguration process and to make a whole body motion such as locomotion. Autonomous operation by battery is a key issue to make complicated reconfiguration because tethers for power supply and control signals are serious obstacles for reconfiguration motions.

To realize complicated reconfiguration process, information processing system and inter-module communication capability should be also enhanced, to support distributed control by multi-module system.

B. Mechanical and Electrical Design

Fig. 2 shows M-TRAN I & II and Figs 3 and 4 are photos of the appearance and the inner structure of the M-TRAN II module respectively. Table I shows the specifications of M-TRAN II.

When passive and active blocks touch with each other, they automatically connect by magnetic force. Each passive or active block has three surfaces for connection. The active one has a mechanism for releasing this connection (Section 2.3).

Inside the passive block, there are a power supply circuit, a main CPU board, and a battery. Power supply by a single battery is enough for driving two motors and for releasing one connection, and all the modules operate by their own batteries.

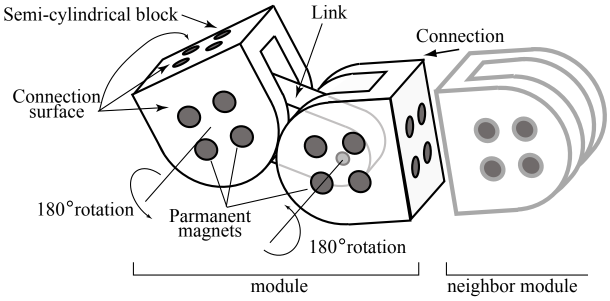


Fig 1. Schematic view of the M-TRAN module

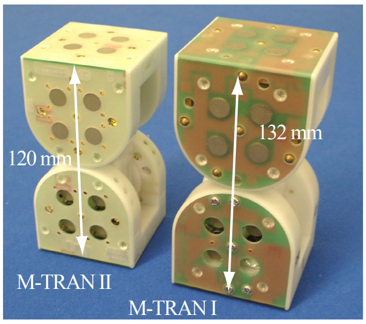


Fig 2. Comparison of M-TRAN I and II

The link part contains two geared motors to rotate passive and active blocks and potentiometers to measure those angles. A motor driver circuit including a microprocessor is also in this part, which realizes PID positioning control. The torque of the motor is enough to support and lift other two modules excluding the worst cases such as three modules arranged in straight line horizontally [2]. Magnetic connection force is enough to maintain the structure for such lifting motion.

Inside the active block, there are connection plates, each of which has several electrodes (Fig. 3). They are stored inside the casing to avoid short circuit with other module's

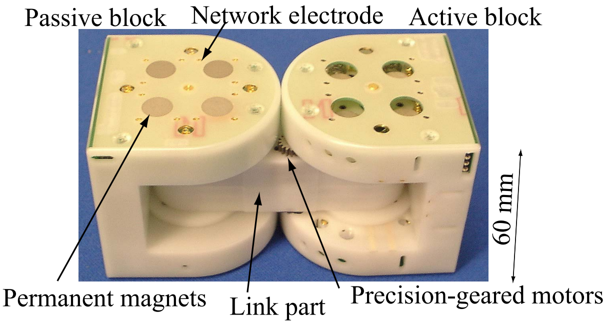


Fig 3. Appearance of the M-TRAN module

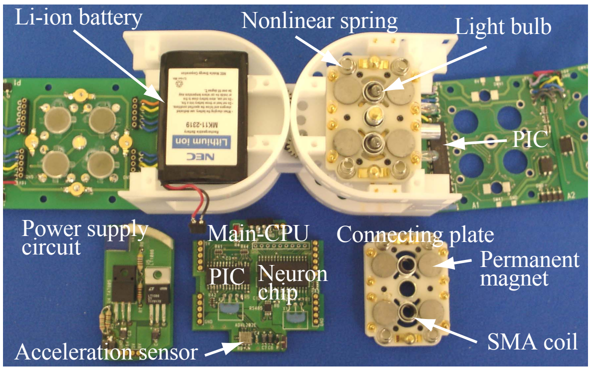


Fig 4. Inner structure of module

TABLE I
SPECIFICATIONS of M-TRAN II

Dimension	60x120x60mm
Weight	400 g (including battery)
CPU	Neuron chip (TMPN3120FE5M, Echelon Corporation) Three PICs (PIC16F873/877)
Global communication	LonWorks & RS-485 (39kbps)
Local communication	Asyncronous serial 4,800 bps
Torque of each axis	1.9 Nm (rating)
Connection force	80 N
Battery	Li-ion (3.8V, max 1.4 A, 700mAh)
Power consumption	0.4W (nominal) 4W (detachment)
Sensor	Acceleration sensor (3 axes)

electrodes, and come out and connect with others in other passive blocks when modules are connected (Fig. 5). Some of the electrodes are for communication and others are for sensing mechanical connection/detachment of modules.

C. Connection Mechanism

The connection mechanism of the module is based on IBMU [16] (Internally Balanced Magnetic Unit) and coil actuators made of shape memory alloy (SMA). The IBMU mechanism includes a connecting plate with permanent magnets and springs (Fig. 5). When two blocks, passive and active, are place near by, the connecting plate moves by the magnetic force and magnets of two blocks make contact (Fig. 5 from (a) to (b)). As the repulsive force of the springs is internally balanced inside the casing, firm connection is realized without reducing attractive force of magnets. When the SMA coils are heated and generate enough force to move the connection plate with the aid of the spring, connection force between two blocks become much smaller and the modules are separated by the modules' motion (Fig. 5 (c)).

In order to realize firm connection and reliable detachment, magnetic force should be large and releasing force (SMA + spring) should exceed the magnetic force over the operating range of distance *d* (Fig. 5). Fig. 6 shows the relationship between distance and force of all the components relating the connection mechanism.

We used an Neodymium magnet (ϕ7mm t 3mm) covered with an iron yoke for large attracting force. With full shaped yoke (Magnet A in Fig.6), magnetic force is largest when two magnets contact each other and it becomes very small when the magnets are apart. This force at *d*>>0 is too small to generate enough force to adjust misalignment of two modules in such a case as (a) in Fig. 5. Moreover, the steep slope of the characteristic curve (A in Fig. 6) near *d*=0 makes the design of the spring difficult. The yoke shape of Magnet B in Fig. 6 is designed for the real hardware. The slope of the curve B in Fig. 6 is not too steep and we can design the spring by which the releasing force (SMA+Spring in Fig. 6) exceeds the magnetic force for any distance *d* in the operating range. By this design, though the maximum connection force is about three times larger than that of M-TRAN I, such firm connection is still releasable by the SMA coils and the springs.

In M-TRAN I, electric current is applied directly through the SMA coils to heat them. In M-TRAN II, miniature light bulbs are used (Figs. 4 and 5) to reduce electric current. The power rate for detachment (4 W, 0.5 A by 8V) is small enough for battery operation, which is much less than that of M-TRAN I (36W). This method is reliable but the speed of detachment and reconnection is slow, requiring 1 and 3 minutes respectively.

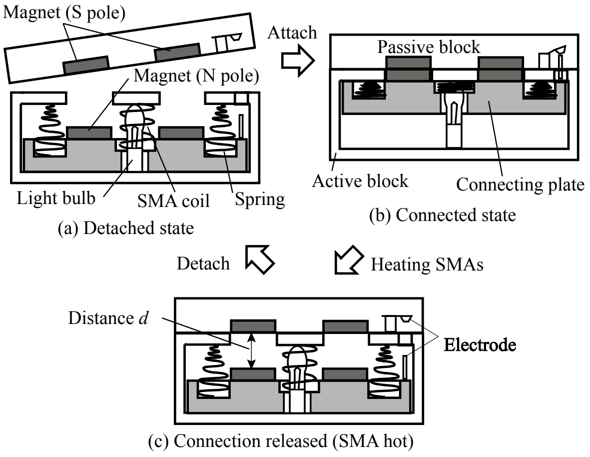


Fig 5. Connection mechanism.
(a) connection phase: misalignment is adjusted by magnetic force. (b) connected state: spring force is internally ballanced. (c) Connection released: Hot SMA coils and the springs are generating enough force to detatch the magnets.

D. Multi-CPU and Communication System

The control system of the M-TRAN II contains two layers of multi-CPU system (Fig. 7). The upper layer is a computer-network of the modules and the lower layer is a multi-microcontroller system inside each module.

The upper layer consists of the main-CPUs (Neuron chip, Echeron corp.) and a two wired network bus based on RS-485. The host PC is also in this network through an interface board with the same chip when the tethered attachment is connected to one module. Via this network, programs and data are transferred to each module from the host PC.

The communication protocol is the LON protocol that is implemented on the Neuron chip as firmware. This chip releases the user from troublesome communication process. All the modules in this network, including the host PC, are equivalent in their priority. In the experiments in Section 4, one modules with least ID number is selected as a master.

Inside each module, the main-CPU (Neuron chip) and three

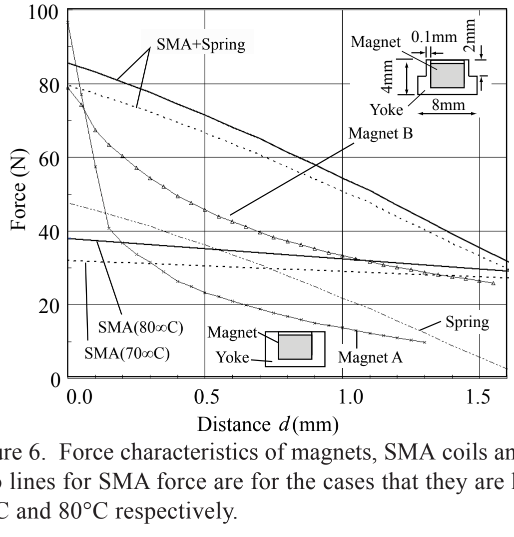


Figure 6. Force characteristics of magnets, SMA coils and spring. Two lines for SMA force are for the cases that they are heated in 70°C and 80°C respectively.

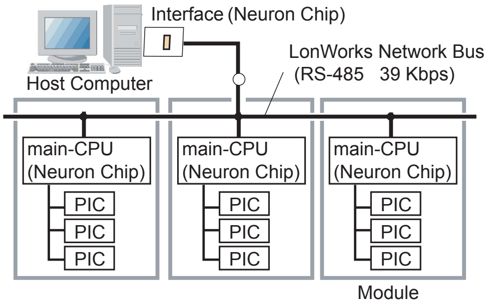


Fig 7. Multi-CPU and communication system

microcontrollers (PIC16F873 and F877) compose the lower layer. The main-CPU works as a master and sends commands to the others. The PIC in the link part controls two motors by its PID program and the PIC in the active block controls the light bulbs to release the connections. When rotational angles reach within a certain range or connection is completed or released, each PIC sends a completion signal to the master. The PIC in the passive block has a role to communicate locally with neighbor modules via surface electrodes, but currently it is not implemented. This controller is used only for receiving command from outside via the RF signal (Only one module has an RF receiver). Communication between those four controllers are made by two wire asynchronous serial lines running inside of the module.

III. SOFTWARE ENVIRONMENT

A modular robotic system like M-TRAN system can form various configurations such as a rigid structure and realize motions suitable for a variety of tasks, such as locomotion and manipulation. To realize such tasks, there are two types of research subjects. One is to find a sequence for reconfiguration from one shape to another, and the other is to find a cooperative robotic motion suitable for tasks.

We have developed several programs including; 1) a kinematic 3-D simulator to design 3-D configurations and reconfiguration sequences by GUI [11], 2) an automatic planner to obtain reconfiguration processes [12], 3) dynamics simulator to design coordinated motions for locomotion [13]. There also exist some research works for M-TRAN by other researchers [14,15]. We also need a control program on the real hardware which works in parallel and in distributed manner.

The above programs have been developed independently and their programming and data structure are different from each other. Therefore we need to convert the results obtained by each program for the hardware. We are trying to integrate them into a more general form.

By using integrated programming environment of the following three programs, we can design motions of modules, verify them by simulation and apply to the hardware.

A. Kinematics Program and Language

We have made a program which serves as an interface between human and hardware [11]. Its functions include 3-D display of modules' motion using Open GL library, GUI interface on the MS-Windows to design configurations and reconfiguration sequences, kinematic routines to check the connectivity of the whole structure and collision between modules, and statics analyzer to verify the stability of the whole structure under the gravity.

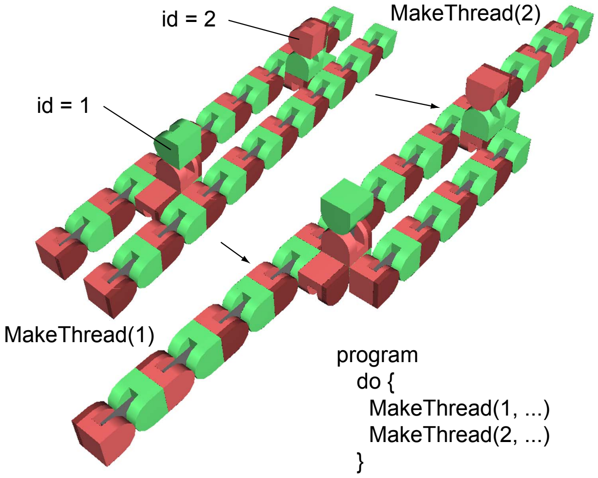
Improvement to this software is a subroutine call similar to usual programing languages, which is processed concurrently. Fig. 8 shows an example of concurrent programming. By defining a subroutine (MakeThread()) as sequences of modules' motions and calling them with different parameters, the same motion sequences are applied to different places at the same time.

Currently, this programming language supports simple structure such as if...then...else and while-loop, but any extension can be feasible common to general or concurrent programming languages. For instance, a state machine type algorithm or a cellular automata algorithm such as [14] can be also treated.

A difficulty of this concurrent programming is the arbitration in such a case that two subroutines issue commands to drive the motors of the same module at the same time. As this problem is based on the difficulty of reconfiguration planning [2], there seems to be no best solution. Currently such arbitration is made manually.

B. Dynamics Simulation

We have developed a dynamics simulator for our modular robot in order to realize whole body locomotion by a coordinated periodic motion [13]. The simulator is based on Vortex (Critical Math Lab) and most of the dynamical characteristic of the hardware module is implemented.



```
program
  do {
    MakeThread(1, ...)
    MakeThread(2, ...)
  }
```

Fig 8. Example of subroutine

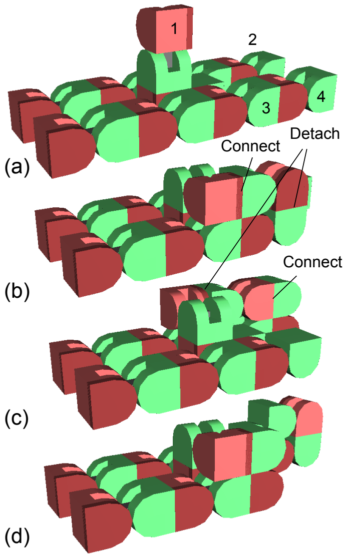
By using a modified simulator, we can verify a reconfiguration sequence made by the kinematics program in 3.1 before an experiment by the hardware.

C. Hardware Control Program

The M-TRAN II hardware is controlled in distributed manner. Though the network allows a decentralized operation, communication speed is currently critical. Therefore, all the modules are coordinated by a single master in order to reduce messages on the network.

A whole body motion such as walking and a caterpillar-like motion is realized as that each module drives motors following a table step-by-step by a global clock. For the reconfiguration process, clock synchronization is not appropriate, because all modules' motions must be completed before next motions start. Therefore the reconfiguration process is managed by a command/event table stored in each module which describes a motion and/or a condition for synchronization such as completion of connection/detachment as well as positioning.

The kinematics program in Section 3.1 generates this table. Fig. 9 shows a part of the original sequence of self-reconfiguration ((a) to (d)), the program and the command/event table for the reconfiguration experiment in the next section (Fig. 10 (b) to (d) via (c)). Global synchronization is managed by the master's polling.



```
//(a) -> (b)
con(3, 3, 1)
{
  mov(1,1, -90,0)
  mov(3,0, 0,90)
  mov(4,0, -90,0)
}

con(4, 3, 1)
con(3, 3, 0)
```

```
//(b) -> (c)
{
  mov(1,1, 90,0)
  mov(4,1, 0,0)
}
mov(2,0, -90,90)

//(c) -> (d)
con(4, 3 , 0)
con(2, 3, 1)
mov(1,1, -90,0)
{
  mov(2,0, 0,90)
  mov(4,0, 90,0)
  mov(3,0, -90,90)
}
```

(e) Reconfiguration program

[id:0]	[id:2]	[id:3]	[id:4]	
CC 2	NOP	CC 2	NOP	-(a)
MV 0,-90	NOP	MV 0,90	MV -90,0	-(b)
NOP	NOP	NOP	CC 2	
NOP	NOP	SMA 2	NOP	
NOP	NOP	CD 2	NOP	
MV 0,90	NOP	NOP	MV 0,0	-(c)
NOP	MV -90,90	NOP	NOP	
NOP	CC 2	NOP	NOP	
NOP	NOP	NOP	SMA 2	
NOP	NOP	NOP	CD 2	
MV 0,-90	NOP	NOP	NOP	
NOP	MV 0,90	MV -90,90	MV 90,0	-(d)
NOP	NOP	CC 2	NOP	

(f) Command/Event table

Fig 9. Reconfiguration and program.
In (e), mov( ) and con( ) are functions for angle control and connection control. Meanings of commands in (f) are; MV: change angles and check completion of positioning, CC: check connection, CD: check detachment, SMA: switch on, and NOP: no operation. In (f), operation of each row is managed by each module so that each column is syncronized.

IV. EXPERIMENTS

We have developed 20 modules of M-TRAN II. Experiments of whole body locomotion using four to nine modules were successfully carried out [13]. For those experiments, connection of modules were not changed.

Fig. 10 shows an experiment of reconfiguration by ten modules, which includes metamorphosis from a four-legged walker to a single thread caterpillar (see video). The four-legged walker (Fig. 10 (a)) moves using a coordinated motion pattern of eight modules which is obtained by the automatic motion generator using a GA process and a CPG network model [13]. Its locomotion speed is about 20 cm/sec.

After walking, it changes to an H-shape by spreading all the legs (Fig. 10 (b)). With this shape lying on the floor, it can move by caterpillar-like motion of double thread. Then it starts self-reconfiguration (Fig. 10 (c)) of 18 steps to a single thread structure. As releasing each connection requires about a minute, this process requires about 8 minutes. The single thread structure (Fig. 10 (d)) moves using a caterpillar-like motion at a speed of 6 cm/sec.

Fig. 11 shows other two experiments. In Fig 11 (a), the 4 legged walker is almost the same as Fig. 10 (a). This walker can perform a crawler motion by connecting two pairs of legs. The next reconfiguration (Fig. 11 (b)) starts from an axisymmetric walker and ends up with a straight line. All the modules forms a single thread in this case, while one or two modules are left aside in the other two experiments.

All the processes in the above experiments were autonomously performed by the real hardware without human intervention. By those experiments, performances of the M-TRAN II, especially its ability of self-reconfiguration, and its supporting software were verified (see http://unit.aist.go.jp/is/dsysd/mtran/English/index.html for several experiments.)

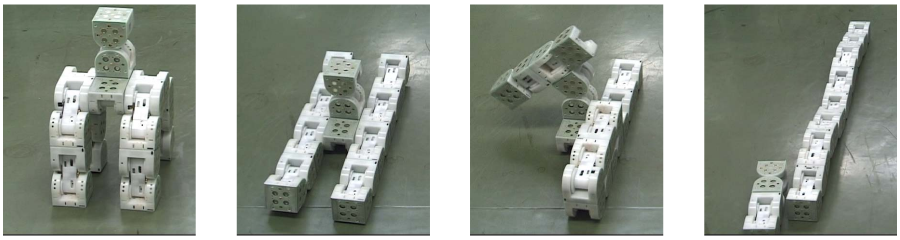
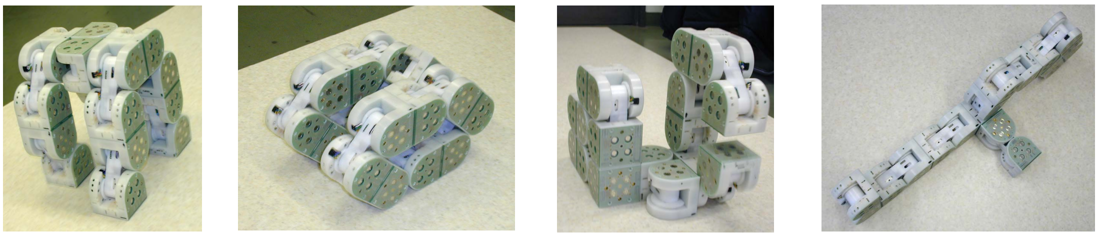
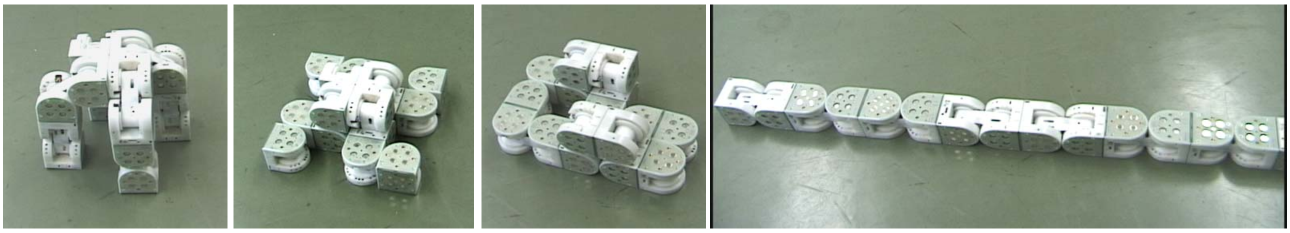

(a) Four-legged walker (b) H-shape (c) Reconfiguration (d) Caterpillar

Fig 10. Experiment of locomotion and self-reconfiguration



(a) Additional crawler motion



(b) An axisymmetric walker to a line

Fig 11. Experiments of self-reconfiguration

V. CONCLUSION AND FUTURE WORKS

We have developed M-TRAN II to realize complicated self-reconfiguration and coordinated whole body motions. Several improvements are achieved such as firm and reliable connection/detachment mechanism, high performance computation and inter-module communication, low energy consumption for autonomous and tetherless operation, powerful actuation, etc. We also improved and integrated programming environments for this modular system.

As future works, we will improve the computation of each module and execute various experiments using all the 20 modules. Connection mechanism is still a critical part and to be improved. For adaptive operation, sensor is necessary and we are planning both to install sensors to the current modules and to develop sensor modules compatible to the M-TRAN II.

VI. REFERENCES

[1] S. Murata, et al., "*Self-repairing mechanical systems*," *Autonomous Robots 10*, pp. 7-21, 2001.

[2] S. Murata, et al., "*M-TRAN: Self-Reconfigurable Modular Robotic System*," *IEEE/ASME Trans. Mech*. Vol. 7, No. 4, pp. 431-441, 2002.

[3] M. Yim, et al., "Connecting and Disconnecting for Chain Self-Reconfiguration With PolyBot," *IEEE/ASME Trans. Mech*. Vol. 7, No. 4, pp. 442-451, 2002.

[4] A. Castano, et al. "*The Conro Modules for Reconfigurable Robot,*" *IEEE/ASME Trans. Mech*. Vol. 7, No. 4, pp. 403-409, 2002.

[5] C.Unsal, et al. *"I(CES)-cubes; a modular self-reconfigurable bipartite robotic system*," *Proc. SPIE, Sensor Fusion and Decentralized Control in Robotic Systems II*, 258-269, 1999.

[6] K. Tomita, et al.: "*Self-assembly and self-repair method for distributed mechanical system*," *IEEE Trans. on Robotics and Automation*, 15-6, 1035-1045, 1999.

[7] Z. Butler, et al., "Distributed Control for Unit-Compressible Robots: Goal-Recognition, Locomotion, and Splitting," *IEEE/ASME Trans. Mech*. Vol. 7, No. 4, pp. 418-430, 2002.

[8] M. Yim, et al.: "*Distributed Control for 3D Metamorphosis," Autonomous Robots*, Vol.10, No.1, 41-56, 2001.

[9] K. Kota, et al., "*Motion synthesis for the self-reconfigurable molecule*," *Proc. IROS 1998*, 843-851, 1998.

[10] C. Ünsal, et al.: "*A modular self-reconfigurable bipartite robotic system: implementation and motion planning*," *Autonomous Robots*, 10-1, 23-40, 2001.

[11] H. Kurokawa, et al.: "*Motion simulation of a modular robotic system*," *Proc. 2000 IEEE Intl. Conf. on Ind. Elec., Con.and Instr. (IECON-2000)*, pp. 2473 - 2478, 2000.

[12] E. Yoshida, et al.: "A Motion Planning Method for a Self-Reconfigurable Modular Robot," *Proc. IROS 2001*, 590-597, 2001.

[13] A. Kamimura, et al., "*Automatic Locomotion Pattern Generation for Modular Robots," Proc. ICRA* 2003, to appear.

[14] Z. Butler, et al., "Generic Decentralized Control for a Class of Self-Reconfigurable Robots", *Proc. ICRA* 2002, pp. 802-816, 2002

[15] H. H. Jund, et al., "*Distributed Control in Self-Reconfigurable Robots," Proc. ICES2003*, 2003.

[16] S. Hirose, et al.: "*Internally-Balanced Magnet Unit*," *Advanced Robotics*, 1-3, 225-242, 1986.