Design and Implementation of UBot: A Modular Self-Reconfigurable Robot

Yanhe Zhu, Jie Zhao, Xindan Cui, Xiaolu Wang, Shufeng Tang and Xueyuan Zhang

State Key Laboratory of Robotics and System
Harbin Institute of Technology
Harbin, Heilongjiang Province, China
yhzhu@hit.edu.cn

Jingchun Yin

Dipartimento di Automatica e Informatica
Politecnico di Torino
Corso Duca degli Abruzzi, 24 - 10129 Torino, Italy
jingchun.yin@polito.it

***Abstract* –The design and implementation of a novel modular Self-Reconfigurable Robot (SRR) called UBot is reviewed in this paper. Firstly, the philosophy of hardware design is presented. The module is designed with criteria such as cubic-shape, homogeneity, and strong connections to fulfill the requirements of complex three-dimensional reconfiguration and locomotion. Each robotic module has two degrees of freedom and four connecting surfaces with hook-type connecting mechanism. A group of modules can transform between different configurations by changing their local connections, achieve complicated modes of motion and accomplish a large variety of tasks. Secondly, a 3D dynamics simulator for UBot SRR is developed, where robot locomotion and transfiguration simulation could be done. A worm-like robot evolution is performed with results of a variety of high-performance locomotion patterns. Finally, Experiments are performed about autonomous docking, multi-mode locomotion and self-reconfiguration. The validity of docking method, CPG-network control and reconfiguration planning method is verified through locomotion and transformation tests of configurations such as snake-type, quadruped walking-type, omni-directional cross-type and loop-type.**

Index Terms - modular Self-Reconfigurable Robot, docking, multi-mode locomotion, self-reconfiguration.

I. INTRODUCTION

Self-Reconfigurable Robot (SRR) is made up of mechatronic modules that can dock, detach and communicate with adjacent modules autonomously. SRR can transform between functional configurations to adapt to changing environment without external help, e.g., a reconfigurable robot could transform into a snake to get through narrow places during a rescue operation, into a hexapod to carry a load or it may split into many smaller robots to perform tasks in parallel. Compared with the conventional robot, self-reconfigurable robot owns advantages of adaptability, self-repair, robustness. It can extend the application filed from manufacturing industry to aerospace, exploration, nucleus, medical treatment, search and rescue, polar scientific research, military detection, anti-terrorism, etc.

Many successful prototypes have been developed since Fukuda developed the first self-reconfigurable robot named CEBOT [1] in 1988. M-TRAN modular robot system [2-5] is made up of two U-shaped blocks connected by a link rod. Each block has one DOF by connecting with the link rod. But the two degrees of freedom are in the same direction, therefore, the module can only rotate in one plane. SuperBot [6-8] module innovated from M-TRAN modules. Each SuperBot module has three joints, which is more flexible for locomotion, but the modules have to be connected to each other manually. Other typical system were researched, such as Polybot [9-11], CONRO [12], Micro-Unit [13] and ATRON [14-16] etc.

One requirement for SRR is the feasibility to different locomotion modes for a wide range of terrain types, since a robot may travel through terrains that may not be fully characterized ahead of time such as transportation, assembly, inspection and exploration. Another requirement is versatility in self-reconfiguration to realize transformation between different configurations to adapt to the changes in the environment and in the task, which is the key distinguishing feature of SRR. The UBot [17] module is constructed in cubic shape with two universal rotational joints and reliable active/passive connecting mechanism, suitable for locomotion and self-reconfiguration.

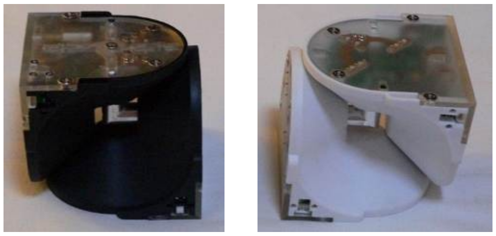
This paper is organized as follows. Section 2 introduces the UBot hardware system. Section 3 describes the 3D dynamics simulator for UBot SRR and some examples of locomotion simulation and evolution. Section 4 presents a set of experiments about autonomous docking, multi-mode locomotion and self-configuration. Conclusions and the future work are given in section 5.

II. HARDWARE

A. UBot Basic Module Design

1) Mechanical System Design: As shown in Fig. 1, the UBot modules are classified as active module and passive module by the function of connecting mechanism. Active module is composed of one twin-rotation mechanism and four active connecting mechanisms. Passive module is composed of one twin-rotation mechanism and four passive connecting mechanisms. The size of active and passive module is both 80mm×80mm×80mm, the mass of active module is 350g, and the mass of passive module is 280g.

The twin- rotation mechanism is the central part of UBot module to which the active or passive connecting mechanism is attached. The twin-rotation mechanism determines the basic shape of the module and it is the main motion mechanism of the UBot for locomotion, self-reconfiguration and operation. As shown in Fig. 2, the right angle shaft and the two L-shaped parts connecting to the tips of the shaft form two rotational joints which perpendicularly intersect with each other in the geometric center of the module.



a) Active module b) Passive module

Fig. 1 UBot modules

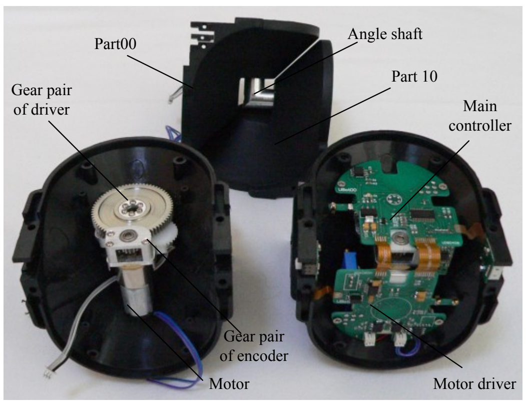
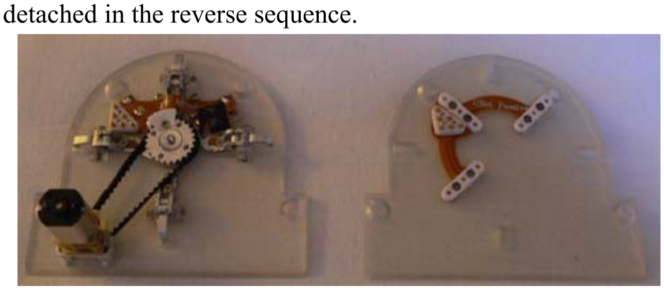


Fig. 2 Basic shape of UBot module

On each connecting surface of the active module, four holes are symmetrically distributed with every hole equipped with one hook-type connector driven by the same motor. While for each passive module, only four empty holes without connectors are symmetrically distributed in the same way on each connecting surface. The active and the passive modules could be connected between any connecting surfaces in four possible orientations. The connecting mechanism is shown in Fig. 3. Fig. 4 shows the process of connection. To realize connection process, the sliders move outward driven by the motor, and drive hooks rotate around fixed-shaft via moving-shaft until two surfaces are fastened. The connection is detached in the reverse sequence.



a) Active connecting mechanism b) Passive connecting board

Fig. 3 Connecting mechanism

2) Electrical System Design: The UBot SRR system consists of UBot modules, relay unit and PC. Each module communicates with the relay unit by wireless communication. PC sends and/or receives commands to/from UBot module through the relay unit. The modules share the ID and orientation information through serial ports communication. To realize reconfiguration and locomotion, the control system should be able to control the motor, the connecting mechanism, the sensors, the orientation recognition system, the power management system and the communication between neighbouring modules. The control schematic diagram of UBot active module is shown in Fig. 5.

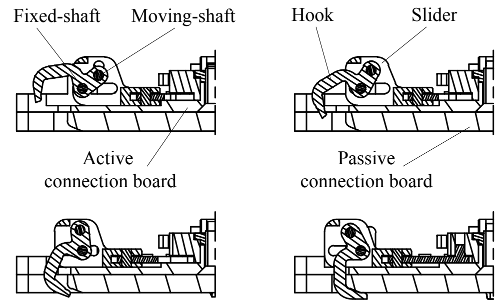


Fig. 4 The process of connection

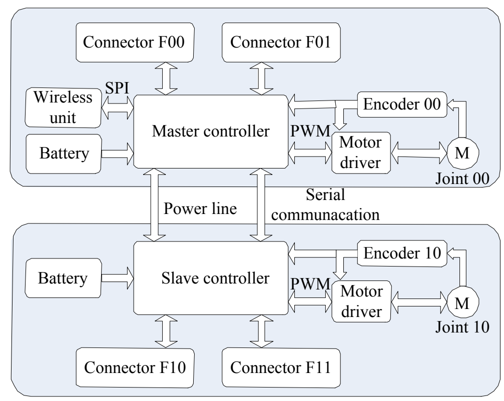


Fig. 5 The control schematic diagram of UBot active module

B. Sensory Module

We designed the sensory module. It has the same cubic structure and the same dimensions as the active/passive module. Inside the sensory module are four types of sensors: wireless vision sensor, accelerometer, infra-red range finder, and linear Hall sensor. The sensory module is shown as Fig. 6.

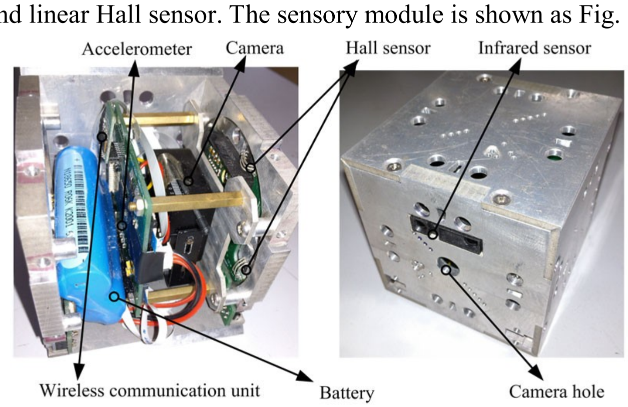


Fig. 6 The sensory module

III. SOFTWARE

A. 3D Dynamics Simulator for UBot

Generic dynamics simulator like Adams is not appropriate for SRR because of its various configurations; low-level programming ability makes some applications hard to execute like evolution calculation. Thus lots of SRR groups have developed their own robot simulator to simulate reconfiguration and coordinated dynamical features for various typical configurations. We developed a 3D dynamics simulator for UBot as shown in Fig. 7. It has the ability of simulating rigid body dynamics like joints, collisions, frictions et al. The simulator user interface is shown in the following picture, which has a gorgeous texture and gives us a strong virtual reality feeling.

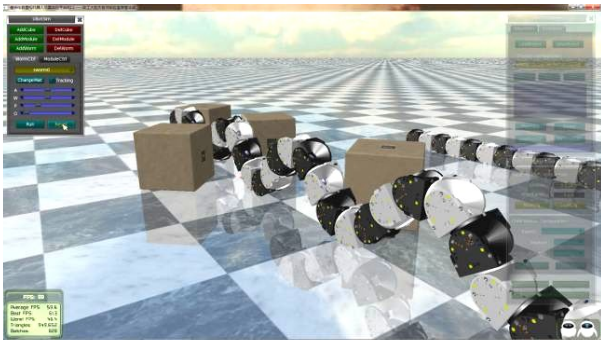


Fig. 7 3D dynamics simulator for UBot

This simulator is developed using PhysX as the physics calculation engine and OGRE as the graphic engine. To make it more useable, we add some additional functions to it:

1) To create and save new configurations by hand quickly,
2) To evolve any configuration's locomotion .
3) To save and load the evolution results in XML format.

The typical configurations constructed in the simulator is shown in Fig. 8.

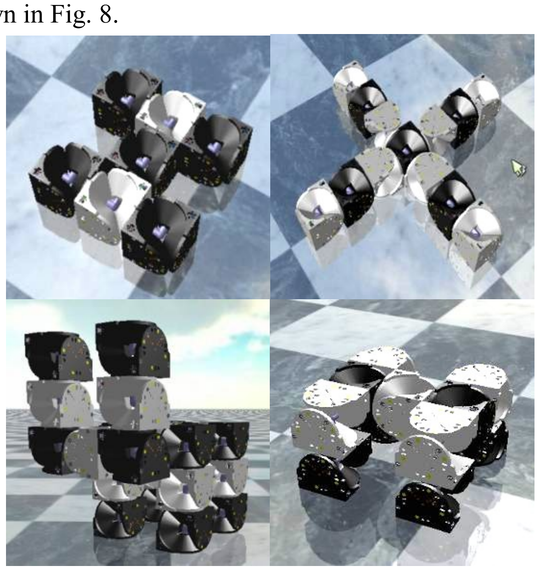


Fig. 8 Robots of various configurations composed of UBot modules

B. Locomotion Evolution

Finding appropriate gait for various configurations with different joint setups of SRR is really a hard task. To realize a useable quick gait planner or controller for any configuration, we can use the method in Evolutionary Robotics.

The procedures of UBot locomotion evolution is the same as normal robot gait evolution. The goal configuration robot is first loaded into our simulator, then each module's controller type and variation scale is set, and a certain number of robots with random controller parameters is initialized. A group of controller parameters in one robot is called a gene or an individual. After initialization, the performance of the population of robot is simulated during a certain period and the desired is recorded to calculate each individual's fitness value according to a certain evaluation function. The individuals' fitness values are used to refresh the population using selection, mutation, crossover operator. This evaluation and refresh procedure is repeated until the maximal iterations of generations or defined stop condition is satisfied.

After the evolution, we will get a list of acquired results with high fitness and can playback the result in the simulator to see if it meets our desire. We should also examine it in real robot hardware platform to prove the validity.

A locomotion evolution of worm configuration composed of 8-modules using Sinusoidal control signal is shown in Fig.9. Fig. 10 shows some locomotion patterns evolved by the simulator, they are worming locomotion, transportation locomotion, dynamic rolling locomotion and static rolling locomotion.

It is in evolution with a population of individual robot, each robot moves dynamically using parameters in corresponding gene and this dynamical locomotion is used for further fitness value calculation. We set the texture of each robot to simple ones in order to accelerate evolution speed.

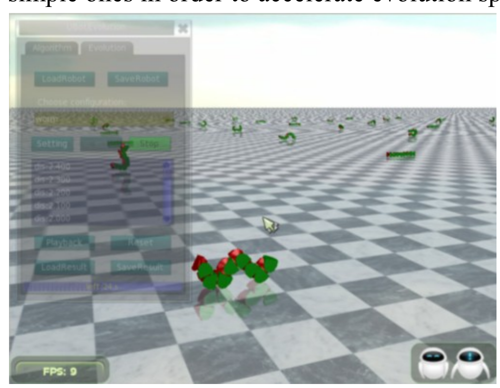
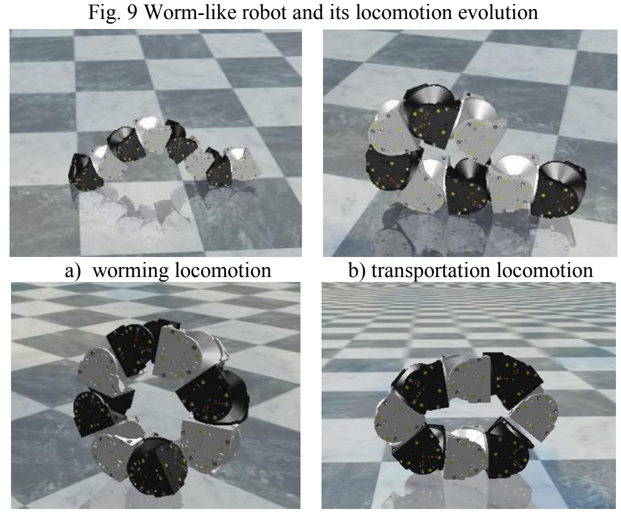


Fig. 9 Worm-like robot and its locomotion evolution



a) worming locomotion b) transportation locomotion

c) dynamic rolling locomotion d) static rolling locomotion

Fig. 10 Achieved multi-pattern locomotion through evolution.

IV. Experiments

A. Autonomous Docking

The distinguishing property in self-reconfigurable robot is the ability of reconfiguration. To realize reconfiguration, accurate autonomous docking between modules is the essential process. Docking is implemented between chain module group with the sensory module as the head and the active module as the target module. We attach four symmetrically distributed yellow round tags on each of the lateral target module surfaces as the characteristic features for visual recognition. The target module is shown as Fig. 11(a).

We have proposed one novel autonomous docking approach which consists of two phases such as pre-positioning preparation and precise positioning:

1) In the phase of pre-positioning preparation, the sensory module obtains vision information, extracts the features and generates the motion adjustment parameters through computation in PC. Then the motion module group repeats the process of parameters adjustment to achieve pre-positioning.

2) After pre-positioning, linear Hall sensors are initialized to capture range information, and the motion adjustment parameters are calculated through precise docking algorithm.

We implement experiments for motion module groups to dock with a target module and two identical motion module groups to dock with a static motion module group with the target module set as the head. The results are shown in Fig. 11 and Fig. 12. These experiments prove effectively the validity of our docking algorithm for changeable configurations of UBot SRR system.

a) Target module b) Worm configuration docking

Fig. 11 The experiment of worm configuration

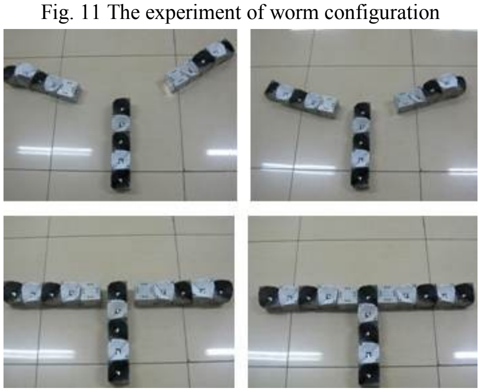


Fig. 12 The experiment of T configuration

B. Multi-Mode Locomotion

The UBot modules can switch between different motion configurations to satisfy different terrains. A feasible control method is necessary to support different motion configurations. CPG method as an effective motion control method has been used in an amphibious snake robot by Auke Jan Ijspeert at EPFL[18], we choose the CPG model by Matsuka. According to Matsuoka's papers [19-20], the CPG model we used for locomotion control of UBot robot is (1)-(3),

$$\begin{cases} T_r\dot{x}_{1i} + x_{1i} = -\omega_0 y_{2i} - \beta f_{1i} + x_e + F_{1i} + a\cdot\sum_j weight_{ij}x_{1j} \\ T_a\dot{f}_{1i} + f_{1i} = y_{1i} \\ y_{1i} = \max(0, x_{1i}), i = 0,\cdots, num-1 \end{cases} \tag{1}$$

$$\begin{cases} T_r\dot{x}_{2i} + x_{2i} = -\omega_0 y_{1i} - \beta f_{2i} + x_e + F_{2i} + a\cdot\sum_j weight_{ij}x_{2j} \\ T_a\dot{f}_{2i} + f_{2i} = y_{2i} \\ y_{2i} = \max(0, x_{2i}), i = 0,\cdots, num-1 \end{cases} \tag{2}$$

$$Output_i = -y_{1i} + y_{2i} \tag{3}$$

In these nonlinear equations above, (1) and (2) denote the two neurons inhibited each other in one CPG oscillator respectively; x_{1i} and x_{2i} are the membrane potential of the two neurons; f_{1i} and f_{2i} denote the fatigue state of the two neurons; T_r and T_a are time constants, y_{1i} and y_{2i} are the outputs of the two neurons; x_e is an external constant input for the neuron; β is a weight of inhibitory synaptic connection from the two neurons; *num* is the amount of oscillators in the network; $output_i$ is the output of the *i*-th CPG oscillator; F_{1i} and F_{2i} are the feedback of the environment the robot interact with. The matrix *weight* denotes the connection relationship among oscillators, element $weight_{ij}$ means the connection weight from the *j*-th neuron to the *i*-th neuron. $weight_{ij}$ has three states: 1, -1, 0, represent excitatory connection, inhibitive connection, no relation respectively.

This method can modify the connection conditions of the neurons according to the connection modes of the modules, which can lead to different time-phased control signals, thus realizing variety in the motion modes.

1) Snake-type mode: A snake configuration consisted of nine UBot modules has been investigated. Since only one pitch joint rotate in a module in the snake-type motion, we use one CPG oscillator corresponding to one module. Through inhibiting the others in the sequence of the modules, the overall motion sequence of the configuration can be obtained. Fig. 13 shows the nine-module snake-type wiggling experiment controlled by CPG network. The locomotion is stable and smooth in the whole process, the robot moves 600mm in 12s, and the velocity of the robot is about 50mm /s.

2) Quadruped walking-type mode: Quadruped configuration is one of the basic limbed configurations. We construct the quadruped configuration with 16 UBot modules (three modules in each leg and four modules in the trunk).

Four CPG oscillators have been used corresponding to four legs' hip joints in the quadruped locomotion control. Quadruped motion experiment has been carried out as shown in Fig. 14, the robot walks with the diagonal legs lifting and landing simultaneously. The quadruped configuration walks forward about 500mm in 15s, the velocity is about 33.3mm/s.

Fig. 13 Snake configuration locomotion

Fig. 14 Quadruped configuration locomotion

3) Omni-directional cross-type mode: The cross-type locomotion is an omni-directional motion, which needs the least modules. The cross configuration consists of 5 modules (represented by 01~05); one module in center is connected with four adjacent modules. Through the cooperative motion of the five modules, the cross-configuration can go forward/backward and rotate, therefore, the robot can reach anywhere in the flat area. The experiment has verified the omni-cross-type motion, as shown in Fig. 15. The robot first moved along the line for 25s as shown in Fig. 15(a-c), and then turned around its own center for 90° couterclockwise as shown in Fig. 15(d-f). The robot has moved three modules' length in the first 25s, and rotated 90° in the next 15s.

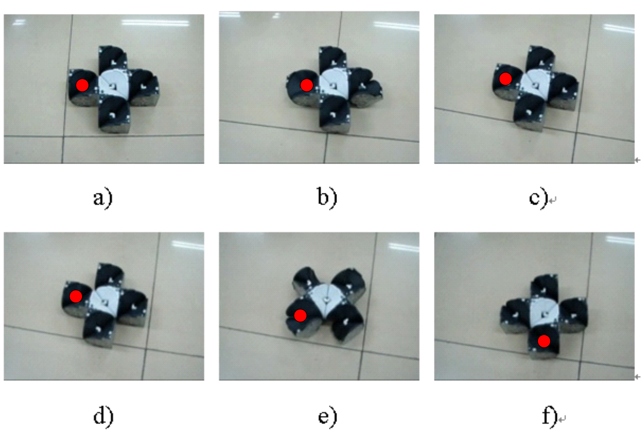


Fig. 15 Locomotion experiment of omni-cross-type configuration

4) Loop-type mode: The loop-type mode rolls like a tread, as a wheeled style of motion. This configuration is well suited to move on straight, flat terrain and even climb or roll down some small sloped terrain. Loop configuration is composed of several couples of modules (one active module and one passive module). Through changing the loop's shape, the robot can roll on the floor. To accelerate the rolling velocity, we could make the center of gravity stand in front of the support range of the configuration in the rolling process called dynamic rolling gait. We choose the loop shape of the rolling as an American football, as shown in Fig. 16. Fig. 17 shows the experiment result of the dynamic rolling with 8 modules. The configuration has rolled 1200mm in 6s.

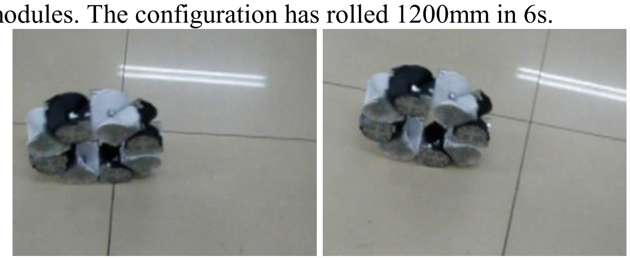


Fig. 17 Dynamic rolling experiment of loop configuration

C. *Reconfiguration*

Fixed particular configuration of the robot cannot adapt to all possible environments, and during performing the complex and varying tasks, it calls for the transformation from one configuration to another.For the reconfiguration between the fixed configurations, a transformation planning method that formulate the degree of the configuration similarity as the search-driven function has been proposed.

First of all, there should be a criterion to judge whether two configurations can be transformed. The criterion we proposed is:

$$\left.\begin{aligned}\min(na_{IC}, np_{IC}) \geq \min(na_{GC}, np_{GC})\\ \max(na_{IC}, np_{IC}) \geq \max(na_{GC}, np_{GC})\end{aligned}\right\} \tag{4}$$

Where IC and GC represent the initial configuration and goal configuration respectively, na and np represent the quantity of active modules and passive modules in corresponding configuration. This criterion can remove the search task in which configuration couples can't be deformed to each other.

Then we use the configuration similarity as the search-driven function to find appropriate deformation plans which is proposed in [21-22]:

$$\sigma_s(IC, GC) = \frac{\sum \min(C_R(IC), C_R(GC))}{\max(|IC|, |GC|)} \tag{5}$$

Where $\sigma_s(IC, GC)$ denotes the similarity degree between IC and GC, $|IC|, |GC|$ are the edge amount in IC and GC respectively. $C_R(IC), C_R(GC)$ are the quantities of R-connected states in IC and GC respectively(R represents any kind of 64 connection ways between the two UBot modules). This method can find a deformation path which needs few steps within a reasonable time.

We performed the self-reconfiguration experiment and the method was used in the transformation from three-limbed configuration to snake-type configuration on the real UBot system. Fig. 18 shows the self-reconfigurable process. There are four connecting actions and four detaching actions executed in the reconfiguration process.

Fig. 18 Reconfiguration experiment of UBot system

V. CONCLUSION AND FUTURE WORK

The hardware and software platform for UBot SRR are introduced and the experimental verification are performed for basic applications of the UBot system: autonomous docking, multimode locomotion and self-reconfiguration.

There are still several important issues remaining in the future UBot research. First, the hardware system will be innovated. We are developing the next generation of UBot module, which has better capability in communication and locomotion. Second, we plan to add some sensors in the modules for the system to perceive external environment information in real time and make decisions about which motion modes to choose and whether to transform.

ACKNOWLEDGMENT

This work was supported by National Natural Science Foundation of China(60705027 and 61273316) and Fundamental Research Funds for the Central Universities (HIT.NSRIF.2009021). The authors express gratitude for the financial support.

REFERENCES

[1] T. Fukuda and S. Nakagawa, "Dynamically reconfigurable robotic system," *IEEE International Conference on Robotics and Automation,* vol. 3, pp. 1581-1586, 1988.

[2] E. Yoshida, S. Murata, A. Kamimura, et al, "Self-reconfigurable modular robots-hardware and software development in AIST," *IEEE International Conference on Robotics, Intelligent Systems and Signal Processing, China*, pp. 339-346, 2003.

[3] H. Kurokawa, K. Tomita, A. Kamimura, et al, "Self-reconfigurable modular robot M-TRAN: distributed control and communication," *International Conference on Robot Communication and Coordination, Athens, Greece*, *no. 21,* 2007.

[4] H. Kurokawa, E. Yoshida, K. Tomita, et al, "Self-reconfigurable M-TRAN structures and walker generation," *Robotics and Autonomous Systems*, vol. 54, pp. 142-149, 2006.

[5] H. Kurokawa, K. Tomita, A. Kamimura et al, "Distributed self-reconfiguration of M-TRAN III modular robotic system," *International Journal of Robotics Research*, vol. 27, no. 3-4, pp. 373-385, March, 2008.

[6] B. Salemi, M. Moll, W.M. Shen, "SUPERBOT: A deployable, multi-functional, and modular self-reconfigurable robotic system," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, *Beijing,* pp. 3636-3641, 2006.

[7] W.M. Shen, M. Krivokon, H. Chiu, et al. "Multimode Locomotion via SuperBot robots," *IEEE International Conference on Robotics and Automation Orlando*, *Florida*, pp. 2552-2557, May 2006.

[8] W.M. Shen, M. Krivokon, H. Chiu, et al, "Multimode locomotion via SuperBot reconfigurable robots," *Autonomous Robot*, vol. 20, pp. 165-177, 2006.

[9] M. Yim, D.G. Duff, K.D. Roufas, "PolyBot: a modular reconfigurable robot," *IEEE International Conference on Robots and Automation, San Francisco, CA*, pp. 514-520, 2000.

[10] M. Yim, Y. Zhang, D. Duff, "Modular Robots," *IEEE Spectrum Magazine*, pp. 30-34, 2002.

[11] M. Yim, D.G. Duff, and K.D. Roufas, "Walk on the wild side: designers of the PolyBot robot system solve the challenges of locomotion by mimiching locomotion in the animal world," *IEEE Robotics & Automation Magazine*, pp. 49~53.

[12] W.M. Shen, B. Salemi, and P. Will, "Hormone- inspired adaptive communication and distributed control for CONRO self- reconfigurable robots," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 700-712, 2002.

[13] E. Yoshida, S. Kokaji, S. Murata, et al, "Micro self-reconfigurable robot using shape memory alloy," *Robotics and Mechatronics*, vol. 13, no. 2, pp. 212-219, 2001.

[14] M.W. Jorgensen, E.H. Ostergaard, and H.H. Lund. "Modular ATRON: modules for a self-reconfigurable robot," *IEEE/RSJ International Conference on Intelligent Robots and Systems* , vol. 2, pp. 2068-2073, 2004.

[15] D.J. Christensen, D. Brandt, U.P. Schultz, et al, "Neighbor detection and crosstalk elimination in self-reconfigurable robots," *International Conference on Robot Communication and Coordination*, pp. 1-8, 2007.

[16] D. Brandt, and D.J. Christensen, "A new meta-module for controlling large sheets of ATRON modules," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2375~2380, 2007.

[17] J. Zhao, X.D. Cui, Y.H. Zhu, and S.F. Tang, "A new self-reconfigurable modular robotic system UBot: Multi-mode locomotion and self-reconfiguration," *International Conference on Robotics and Automation*, *Shanghai*, pp. 1020-1025, 2011.

[18] A.J. Ijspeert, A. Crespi, D. Ryczko, J.M. Cabelguen, "From swimming to walking with a salamander robot driven by a spinal cord model," Science, vol. 315, pp. 1416-1419, 2007.

[19] Matsuoka K, "Sustained oscillations generated by mutually inhibiting neurons with adaptation," *Biological Cybernetics*, vol. 52, pp. 367-376,1985.

[20] Matsuoka K, "Mechanism of frequency and pattern control in the neural rhythm generators," *Biological Cybernetics*, vol. 56, pp. 345-353, 1987.

[21] X. Jiang, and H. Bunke, "Optimal quadratic-time isomorphism of ordered graphs," *Pattern Recognition*, vol. 32, no. 7, pp. 273–1283, 1999.

[22] J.W. Raymond, E.J. Gardiner, P. Willett, "RASCAL: calculation of graph similarity using maximum common edge subgraphs," *The Computer Journal*, vol. 45, no. 6, pp. 631–644, 2002.