Cross-Ball: A new Morphogenetic Self-Reconfigurable Modular Robot

Yan Meng, Yuyang Zhang, Abhay Sampath, Yaochu Jin, Bernhard Sendhoff

2011

Preprint:

This is an accepted article published in IEEE International Conference on Robotics and Automation (ICRA). The final authenticated version is available online at: https://doi.org/[DOI not available]

2011 IEEE International Conference on Robotics and Automation
Shanghai International Conference Center
May 9-13, 2011, Shanghai, China

Cross-Ball: A New Morphogenetic Self-Reconfigurable Modular Robot

Yan Meng, Yuyang Zhang, Abhay Sampath, Yaochu Jin, and Bernhard Sendhoff

Abstract— **We aim to develop a new self-reconfigurable modular robot, Cross-Ball, so that we can apply bio-inspired morphogenesis mechanisms to modular robots to adapt to dynamic environments automatically. To this end, the mechanical design of modular robots has to be flexible and robust enough for various complex configurations. The major contributions of the design of this Cross-Ball robots include: 1) it provides several flexible 3D reconfiguration capabilities, such as rotating, parallel, and diagonal movements; (2) a flexible and robust hardware platform for modular robots using more complex self-reconfiguration algorithms; and (3) the mobility of each individual module. Furthermore, a skeleton-based approach is proposed for the motion control of the modules, where the module movements can be conducted in groups to improve the system reconfiguration efficiency. Some simulation results have demonstrated the feasibility of the proposed module design and the corresponding controller by reconfiguring the robots to various complex configurations.**

I. INTRODUCTION

MODULAR self-reconfigurable robots (MSR) refer to the robots consisting of a large number of identical modules that can reconfigure their connections to form a variety of structures and configurations to suit the tasks in hand. Compared with conventional fixed morphology robotic systems, MSRs are more flexible and robust to adapt to various environments and tasks in hand. Meanwhile, the cost and performance of MSRs may have to be compromised for each specific task. The major applications of the MSRs include those tasks where it is difficult or impossible for human to access, or the tasks which are too tedious for human to do but require higher flexibility and robustness, such as space exploration, search and rescue, etc.

Generally, the mechanical design of MSRs can be categorized into two basic types: chain-based and lattice-based. A chain-based MSR consists of modules which are connected in serial chains to form line, tree and loop structures. The advantages of the chain-based modular robot include its ability to traverse rough terrain, as well as its ability to fit into small spaces. However it is hard for a chain-based MSR to build arbitrary complex 3D patterns. Some existing chain-based MSRs include Polypod [17] and its successor Polybot [18], CONRO [15], RBR [19], and CKBot [14].

Y. Meng, Y. Zhang, and A. Sampath are with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ07030, USA. (Phone: 201-216-5496; e-mail: {yan.meng, yzhang14, asampath}@stevens.edu).

Y. Jin is with the Department of Computing, University of Surrey, Guildford, Surrey, GU2 7XH, UK. (e-mail: yaochu.jin@surrey.ac.uk)

B. Sendhoff is with Honda Research Institute Europe GmbH, 63073 Offenbach/Main, Germany. (e-mail: bernhard.sendhoff@honda-ri.de).

For lattice-based modular robots, each module only occupies a discrete position in a grid lattice. The lattice model can build more complex robot configurations compared to the chain-based modular robots. Meanwhile, the control and motion-planning for lattice-based MSRs are more complex. Some lattice-based MSRs were proposed in [2] [10] [20] [21]. Recently, several MSRs were developed to combine the advantages of these two types in one system, such as ATRON [6], M-TRAN III [7], SUPERBOT [13], and Cross-Cube [8][9].

Some self-reconfiguration algorithms have been proposed for MSRs. For example, Hou and Shen [4] provided a reconfiguration approach for SUPERBOT with a thorough analysis of computational complexity of optimal reconfiguration planning. However, this approach can only be applied to chain-based configurations. Murata and Kurokawa [11] developed a self-reconfiguration approach for a special class of periodic lattice structures on M-TRAN modular robots using some predefined sequences of local self-reconfiguration. But it is not a fully autonomous reconfiguration process, and some predefined reconfiguration plans are needed for other complex configurations. Brandt and Christensen [1] proposed a 2D meta-module for the ATRON modular robot. ATRON modules can cooperatively work in groups called meta-modules. As a result, motion constrains are significantly simplified. However, this approach can only handle 2D motions and there is no further autonomous controller proposed for the final target configuration building process. Stoy[16] proposed a self-reconfiguration approach, where the desired pattern is grown from an initial seed module and recruitment gradients. This approach is based on a hypothetical ideal lattice-based MSR which has less mechanical constrains than any existing hardware platforms.

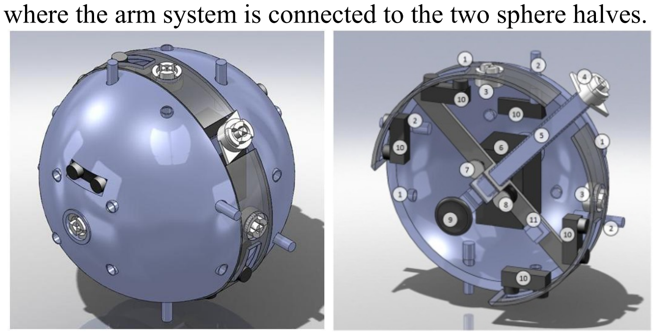
Most available MSR systems have to predefine the target configurations. Therefore, self-reconfiguration of MSRs to arbitrary complex configurations to adapt to dynamic environments is still a challenging problem remained to be solved. The two major issues of MSRs include: (1) the mechanical constraints of hardware designs of MSRs; and (2) more efficient and robust self-reconfiguration approaches to automatically change the target configuration based on the sensor information from the current environment.

In this paper, we aim to address the first issue by proposing a new lattice-based MSR, called Cross-Ball (the second issue will be addressed in a separate paper due to the page limit). The major features of Cross-Ball include: (1) several flexible reconfiguration capabilities, such as rotating, parallel and diagonal movements so that various 3D configurations can be built up; (2) a flexible and robust hardware platform for MSRs using more complex self-reconfiguration algorithms, such as the morphogenetic control algorithm we developed for modular robots introduced in [8] and [9]; and (3) the mobility of each individual module to simplify the configuration process under certain scenarios and potential applications to swarm robots.

II. The Design of the Cross-Ball Modular Robot

A. The Overall Design

The proposed Cross-Ball module, as shown in Fig. 1(a), is a sphere with 3-inchdiameter. The module is in a ball shape to allow individual mobility and to be spatially efficient during self-reconfiguration. It consists of three main components: a rotary arm system and two halves of a sphere, where the arm system is connected to the two sphere halves.



(a) (b)

Fig. 1. (a) The Cross-Ball module. The grey part is the rotary arm system with the main arm and two clasps. There are also two clasps on the sides of the module. (b)The detailed cross section view of the Cross-Ball module.

Fig. 1(b) shows the detailed cross section view of the Cross-Ball module. The components in Fig. 1(b)are listed as followings: (1) A threaded port for attachment of screw; (2) Screw of stationary attachment: contains a motor that can extend to connect to adjacent modules; (3) Side arm: contains electric magnet clasps located on other modules; (4) Clasp on the main arm equipped with a pancake motor to rotate; (5) Main arm: can rotate the clasp end, extend and retract; side arms are 45 degrees away from the main arm; (6) Center motor: provides connection and rotation between hemispherical shells and rotary arm system; (7) Stationary pinion: can spin freely, located there only for supports; (8) Pinion attached to motor: this pinion is controlled by the motor, and is used to extend/retract the arm; (9) Wheel: allows each module to move individually, thus delivering swarming capability; (10) Infrared sensor for distance detection, in total 6 sensors are equipped in 6 orthogonal directions per module; (11) Accelerometer: this sensor helps determine the orientation of the module relative to the force of gravity.

B. Attachment Mechanisms

To self-reconfigure modules to various configurations, the attachment mechanism between modules is critical for the success of reconfigurations. To make it clear, we define the attachment procedure into stationary attachment and dynamic attachment. Stationary attachment means that modules will be connected to build up the target configuration unless new configuration is triggered. Dynamic attachment means that modules may need to be moved to different grid locations by using the dynamic arm for transient movements.

First we will discuss the stationary attachment. Cross-Ball has 6 symmetric stationary attachments on 6 orthogonal directions. Here, male-female design for stationary attachment is not suitable because the male-male and female-female conflictions may happen frequently due to the rotation and relocation of the self-reconfiguration motions. A stationary attachment part has 2 threaded ports with screws and 2 empty threaded ports, as shown in Fig. 2. The screws are equipped with a vibration motor to rotate, retract and extend. The threaded ports can accept screws from other modules. In addition, in cases where there is a screw-screw conflict, one screw can draw itself within the ports to accept the other screw. Therefore, modules can do a stationary connection after any relocation and rotations by 90 degrees. The stationary attachment doesn't consume any power while being connected.

For dynamic attachments, Cross-Ball is equipped with a rotary arm system and two independent clasps on two sides of module, by which self-reconfiguration motions can be executed. The clasps are equipped with electromagnets to easily attach to or repel from other modules because the poles of the electromagnet can be dynamically changed. When two clasps are attached, they are restricted to move and rotate together. The dynamic attachment needs to power the electromagnets while being connected.

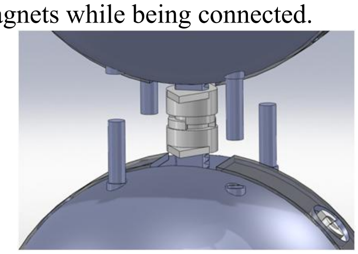
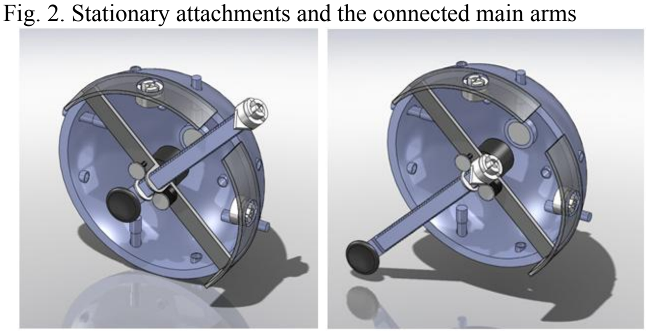


Fig. 2. Stationary attachments and the connected main arms



(a) (b)

Fig. 3. (a) A semi-sphere of a Cross-Ball with its arm completely extended. (b) A semi-sphere of a Cross-Ball with its wheel extended.

The rotary arm system consists of an arm rack with one main arm and two stationary clasps (side arms) located on either side of the main arm at an angle of 45 degrees. The main arm and side arms can rotate along the center of the module by rotating the arm rack. An electromagnets clasp is also equipped at the end of the main arm. The main arm can extend and retract as shown in Fig. 3, as well as rotate the clasp end. Therefore a module's main arm can connect to the main arm, side arms, and side clasps of another module.

A wheel is set up on the other end of the main arm. This wheel enables the module to move independently as an individual mobile robot so that Cross-Ball has more flexibility for complex configurations. This individual mobility also provides the potential of Cross-Ball to be used in a swarm robotic system. When the main arm extends as shown in Fig. 3(b), the wheel will touch the ground to fix the pose of the arm rack. Then the two hemisphere parts can rotate differentially to move the whole module forward and turn.

Therefore, 4 motors are needed in total. Two central motors for each semi-sphere respectively to rotate the arm rack (6 in Fig. 1(b)), and two motors equipped on the arm rack for extension/contraction and rotation, respectively. For the central motors, the torque is more important than the speed because the arm carriage needs to be powerful enough to lift other modules, but not necessarily rapidly. In addition, the central motor, the arm rotation motor and the arm extension motors need to be servo motors so that the rotations of the motors can be measured to obtain the localization accuracy of the module self-reconfiguration.

C. Self-Reconfiguration Motions

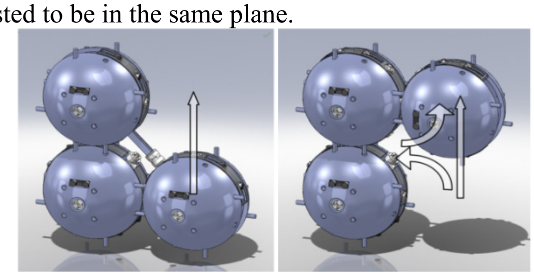
Using the rotary arm and side clasp, a Cross-Ball module is able to conduct three types of self-reconfiguration movements: rotating, parallel and diagonal movements. In the following part of the paper, we will call the main rotary arm as the "thread". The thread could have three poses: perpendicular to x, y and z axis, respectively.

1) *Rotating movement.* A module can connect its main arm to a side clasp, a side arm or the main arm of a neighboring module which touches its thread. Then the module disconnects all its stationary attachments and rotates the main arm clasp. In this way, the whole body will rotate because the other side of the main arm has been fixed by another module.

Rotating movement is very useful for some motions of a MSR such as rotating a module as a wheel when a MSR is in a vehicle configuration. Rotation can also adjust the thread pose of a module. By rotating 90 degrees, the pose of thread translates. We will show in next section that the thread configuration is very important for parallel and diagonal movements.

2) *Parallel movement.* For parallel movement, a module moves to one of its neighboring positions. In this movement, two more modules are involved in addition to the moving module, which are called supporting modules, as shown in Fig. 4. First, the main arm of the moving module connects to the main arm of the upper supporting module. Second, the moving module disconnects all stationary attachments. Third, the main arms of the two modules retract and the arm racks rotate to lift the moving module into the new position. Last, the stationary connections are setup between the two modules and the motion is finished. This motion can also be executed by the moving module and lower support module in a similar way. This motion can also be performed downwards and sideways as long as the following rule is satisfied.

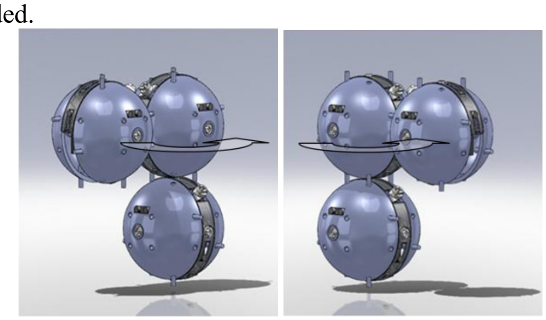
Parallel movement rules: There exist a pair of supporting modules, one is the neighbor of the moving module and the other is the neighbor of the destination grid. The thread of the moving module and one supporting module can be adjusted to be in the same plane.



(a) (b)

Fig. 4. Parallel movement. (a) Before the parallel movement; (b) After the parallel movement.

3) *Diagonal Movement*. Diagonal movement means a module moves into a neighboring grid in the diagonal direction, as shown in Fig. 5. By rotating the module on the upper right in Fig. 5(a), the upper left module is also rotated so the diagonal movement can be achieved. Therefore two supporting modules are needed. There is another way to achieve the diagonal movement similar to the parallel movement: one module lifts its neighbor using the main arm by 90 degrees. In this case only one supporting module is needed.



(a) (b)

Fig. 5. The diagonal movement. The target module is rotated by the arm of the bottom module. (a) Before the diagonal movement; (b) After the diagonal movement

Diagonal movement rules: diagonal movement can be executed when any of the following rules satisfies. 1) There exist a supporting module which is the neighbor of both the moving module and the destination grid. The thread of the supporting module can be adjusted in the same plane which contains the centers of the moving module, the supporting module and the target grid. 2) There exist two supporting modules. One is the neighbor of the moving module and the destination grid. The other is the neighbor of the first supporting module but its center is not on the plane containing the centers of the moving module, the supporting module and the target. 3) The common neighbor grid of the moving module and the destination grid are not occupied.

Generally speaking, the parallel movement and diagonal movement will allow the Cross-Ball to build various complex configurations with enough modules. Next we will introduce the reconfiguration approach for Cross-Ball.

III. THE RECONFIGURATION APPROACH

A. A Morphogenetic Approach

Inspired by the embryonic development of multi-cellular organisms [22], a new emerging field in developmental robotics called morphogenetic robotics has been proposed in [5], which mainly focuses on modeling of neural and morphological development of robotic systems. Based on the multi-cellular mechanism of biological organisms, we have proposed morphogenetic approaches for self-reconfiguration of a modular robot in our previous work [8][9]. First, a two-layer approach is developed in [8], where layer 1 is a look-up-table to predefine the target configuration, and layer 2 is a gene regulatory network (GRN) based controller to move the modules to construct the target configuration. The major limitation of this work is that the target configuration has to be predefined. To address this issue, based on a mechano-chemical model for cell morphogenesis [12], a new approach is later proposed in [9]. In [9], a mechano-chemical model is proposed for layer 1, which is responsible for autonomous generation of chemical patterns in a changing environment. A GRN-based controller is developed for layer 2 to physically realize the target configuration. Furthermore, to optimize the configuration design of modular robots, the covariance matrix adaptation evolution strategy (CMA-ES) [3] is employed to evolve the configuration parameters of the mechano-chemical model. Both morphogenetic models are designed for a Cross-Cube modular robot, which is the preliminary version of the Cross-Ball. Compared to the Cross-Cube, the hardware design of Cross-Ball is further simplified. Meanwhile, Cross-Ball has more motion constraints. Therefore, we need to design a layer 3 controller which is dedicated to the motion control of Cross-Ball. Due to the page limit, in this paper, we mainly focus on the module motion control of the Cross-Ball. The high-level morphogenetic approaches for the Cross-Ball will be discussed in a separate paper.

B. Layer 3 Module Motion Controller

Based on previous discussion, thread configuration for moving and supporting modules are the key to implement rotating, parallel and diagonal movements to achieve complex configurations. In this manner, the module movements usually require the assistance of other modules. Therefore a major challenge for layer 3 controller is to schedule the movements of all the related modules to ensure the success of all modules' movements. Here, we propose a skeleton-based approach to solve this issue.

First, we define module A as a skeleton module if there are 3 adjacent modules B, C and D. B and C are immediate neighbors of A, and D is the immediate neighbor of B and C. With the help of B, C and D, skeleton module A can freely adjust its own thread and perfectly serve as the supporting module for the movements of all other non-skeleton modules (B, C, and D), as shown in Fig. 6. Therefore a module can move to any grid as long as it keeps connecting its thread to skeleton modules. Therefore the basic idea of layer 3 controller is to maximize the scale of the skeleton modules, and utilize the skeleton modules to support the movements of non-skeleton modules.

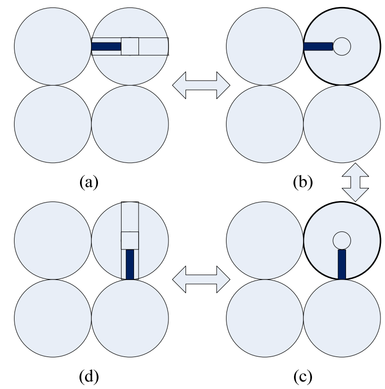


Fig. 6.The skeleton-based approach freely adjusts the thread pose. The upper right module (a) connects its main arm to the module on the left, disconnects all stationary attachments, and rotates the main arm clasp to change the thread pose; (b) reconnect the stationary attachments to finish the thread pose adjustment; (c) connects its main arm to the module below, disconnects stationary attachments, and rotates the main arm clasp to change the thread pose; and (d) reconnect stationary attachments to the finish thread adjustment. All the above processes are reversible.

Layer 3 motion controller of the Cross-Ball works in a decentralized manner, where the flowchart of layer 3 is shown in Fig. 7. Taking movement decisions from layer 2 of the reconfiguration approach, layer 3 controller first needs to judge if the movement is reasonable. The reasonability of the movement can be judged by the following rules.

1) Skeleton modules will not move ifthere is non-skeleton modules at the current moment.
2) If there is no non-skeleton modules at the current moment, only the skeleton modules with the least number of neighbors will move. (The first two rules try to keep the scale of skeleton modules).
3) For non-skeleton modules and modules which are about to turn from a skeleton module to a non-skeleton module, by following the priority from high to low they should: a) connect the thread to a skeleton module;b) connect its thread to the thread of a non-skeleton module; or c) connectits thread to a non-skeleton module. These rules will enable the further movement capability of non-skeleton modules. They also affect how the moving modules choose the supporting modules.

If the movement satisfies all the rules, the moving module adjusts its own thread and the related supporting modules to implement the movement. By introducing the skeleton modules and allowing modules to work in groups (skeleton group and non-skeleton group), a module can easily decide whether to move, and how to choose and move with the supporting modules. In other words, the skeleton-based layer 3 motion controller can significantly reduce the searching complexity on the module movements plan. From the system level point of view, by introducing layer 3 controller, both layer 1 and layer 2 in the morphogenetic approaches(such as those developed in [8,9]) can be well integrated with the customized hardware design of Cross-Ball and its corresponding locomotion capabilities.

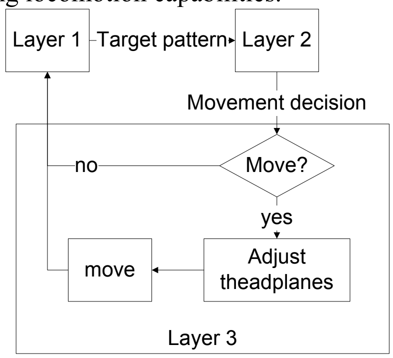


Fig. 7. The flow chart of the layer 3 motion controller.

IV. Experimental Results

Due to its unique and flexible mechanical design and the corresponding motion controller, the Cross-Ball modular robot is able to configure itself into various complex configurations, as shown in Fig. 8. Fig. 8 (a) shows one example of a snake-like modular robot constructed by the Cross-Ball module, where each module is connected to its neighbor(s) in a thread-touching-thread manner in order to allow easy movements.

Fig. 8(b) shows a vehicle-like modular robot, where the wheels are formed under the main body of the configuration. The advantage to this vehicle structure is that the robot can move faster in an open space, and can carry payloads on the top of the vehicle as needed. Due to the unique design of the mobility of the Cross-Ball, the wheel modules beneath the main body of the robot can be treated as omni-directional wheels. They can connect their main arms to vehicle chassis and rotate the main arms to adjust wheels' directions. By rotating hemisphere parts of wheel modules, the vehicle can move forward. Cross-Ball modules can also build other complex configurations, such as a caterpillar-like robot as shown in Fig. 8(c), and a hexapod-like robot as shown in Fig. 8(d).

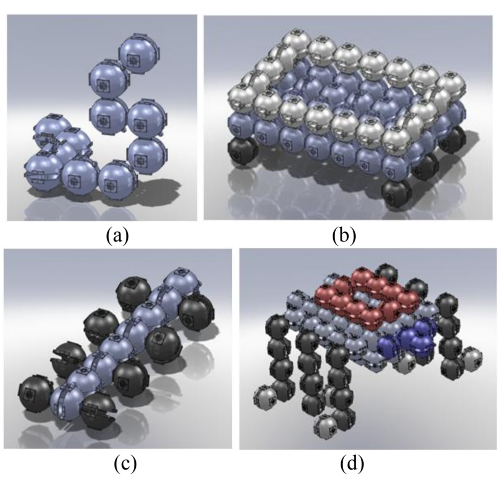


Fig. 8. Some configuration examples that the Cross-Ball modular robots can build up. (a)a snake-like robot; (b) a vehicle-like robot; (c) a caterpillar-like robot; (d) a hexapod-like robot.

To demonstrate the feasibility and efficiency of the layer 3 controller, a simulator is developed to simulate the behaviors and interactions of the Cross-Ball with a physical 3D world using C++ and the PhysX engine from nVidia. The reconfiguration process is automatically generated by layer 1 and layer 2 controllers. Several sets of snapshots of the experimental results are shown in Fig. 9, Fig. 10, and Fig. 11, respectively. Fig. 9 demonstrates how the Cross-Ball modular robot transforms into a snake-like configuration. The self-reconfiguration procedure using layer 3 controller to build a vehicle-like configuration is shown in Fig. 10. Finally, Fig. 11 shows the self-reconfiguration procedure of a complex legged robot configuration, where the target configuration has four legs, a cargo space on the top, two working arms in the front, and some modules in the back to keep the balance and also serve as the backup modules.

Please be noted that in Fig. 9, Fig. 10, and Fig. 11, the white lines represent the threads. The dark blue objects are the skeleton modules and the light blue ones are non-skeleton modules. Skeleton modules serve as a supporting platform for non-skeleton modules to move, and the thread of non-skeleton module always tries to connect the skeleton modules or the threads of non-skeleton modules to facilitate further module motions.

These experimental results demonstrate the effectiveness of layer 3 controller for building various complex configurations. In all these experiments, layer 2 controllers have defined the module movements in advance for layer 3 controller. The fully autonomous self-reconfiguration of the Cross-Ball modular robot based on the sensor information to adapt to dynamic environment will be discussed in a separate paper.

V. Conclusion

In this paper we have shown a new mechanical design of the Cross-Ball modular robots and the corresponding layer 3 controller for the module movements. This new design can provide enough flexibility and robustness for the robots to reconfigure to various complex configurations as needed. Under this new design, each individual module can make three different movements: rotating, parallel, and diagonal movements, which are necessary for the Cross-Ball modular robots to build complex configurations. In addition, with the dedicated layer 3 for modular motion controller, layer 1 for pattern generation and layer 2 for pattern formation, the generic morphogenetic self-reconfiguration approaches proposed in [8] [9] can be easily integrated with the Cross-Ball modular robots.

However, one issue in layer 3 controller remains. Although it works in a decentralized manner, it still depends on some global information (a module may have to collect position information from all other modules). This may increase the communication and computational costs when the size of modules increases. We will investigate this issue in our future work. Other future work for this modular robot includes: (1) Build the prototype of real physical Cross-Ball modular robots. (2) Update and implement our high-level hierarchical morphogenetic controller on the prototype modules to form various configurations in real experiments.

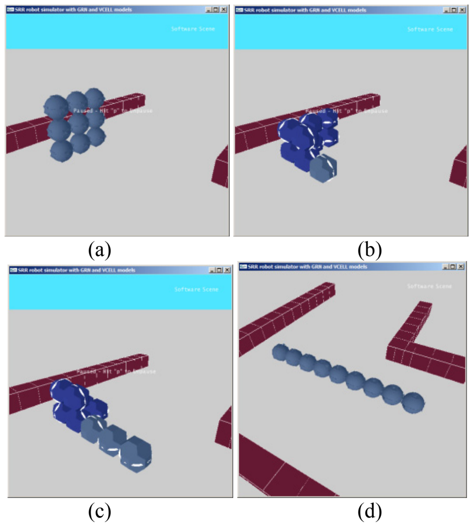


Fig. 9. Simulation ofthe Cross-Ball on building a straight-line configuration using layer 3 controller.

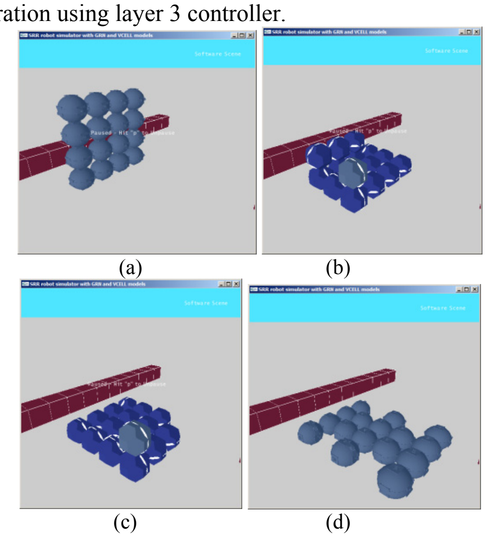


Fig. 10. Simulation ofthe Cross-Ball on building a vehicle-like configuration using layer 3 controller.

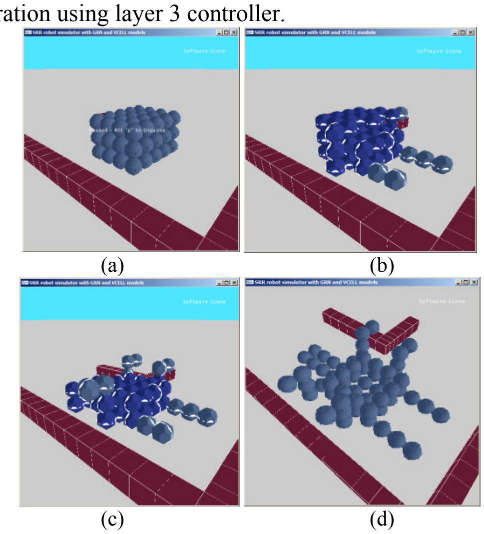


Fig. 11. Simulation of Cross-Ball on building a complex legged robot configuration using layer 3 controller.

ACKNOWLEDGMENT

This project is partially supported by Honda Research Institute Europe GmbH, 63073 Offenbach/Main, Germany.

REFERENCES

[1] D.Brandt, D.J.Christensen,A New Meta-Module for Controlling Large Sheets of ATRON Modules, in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.

[2] K. Gilpin,K. Kotay, and D. Rus, Miche: Modular Shape Formation by Self-Dissasembly, in *Proc. of IEEE International Conference on Robotics and Automation*, 2007.

[3] N. Hansen, S.D. Müller, and P. Koumoutsakos, Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES), *Evolutionary Computation*, 11(1) pp.1-18. 2003.

[4] F. Hou, and W. Shen, Distributed, Dynamic, and Autonomous Reconfiguration Planning for Chain-Type Self-Reconfigurable Robots, in *Proc. 2008 IEEE International Conference on Robotics and Automation,* May 2008.

[5] Y. Jin and Y. Meng. Morphogenetic robotics: An Emerging New Field in Developmental Robotics. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Application and Reviews.* (in press)

[6] M.W. Jorgensen, E.H. Ostergaard, and H.H. Lund, Modular ATRON: Modules for a Self-reconfigurable Robot, in *Proc. of IEEE International Conference on Intelligent Robots and Systems*, 2004.

[7] H. Kurokawa, K. Tomita, A. Kamimura,S. Kokaji, T. Hasuo, and S. Murata, Distributed Self-reconfiguration of M-TRAN III Modular Robots, *The International Journal of Robotics Research*, 27(3-4):373–386, 2008.

[8] Y. Meng, Y. Zhang,and Y. Jin, A Morphogenetic Approach to Self-Reconfigurable Modular Robots using a Hybrid Hierarchical Gene Regulatory Network, in *Proc. 12th International Conference on the Synthesis and Simulation of Living Systems (ALIFE XII)*, August 19-23, 2010.

[9] Y. Meng, Y. Zhang, and Y. Jin, Autonomous Self-Reconfiguration of Modular Robots using a Hierarchical Mechanochemical Model, *IEEE Computational Intelligence Magazine.* Vol. 6, no. 1, pp.43-54, 2011.

[10] S. Murata, E. Yoshida, H. Kurokawa, K. Tomita, and S. Kokaji, Self-repairing Mechanical Systems, *Autonomous Robots,* vol. 10, pp. 7–21,2001.

[11] S. Murata, and H. Kurokawa,Self-Reconfigurable Robot: Shape-Changing Cellular Robots Can Exceed Conventional Robot Flexibility, *IEEE Robotics & Automation Magazine,* March 2007.

[12] J.D. Murray, Modeling Biological Pattern Formation in Embryology, *ISI Atlas of Science: Animal and Plant Sciences,*1: 270-274, 1988.

[13] B. Salemi, M. Moll, and W. Shen, SUPERBOT: A Deployable, Multi-Functional, and Modular Self-reconfigurable Robotic System, in *Proc. of the IEEE International Conference on Intelligent Robots and Systems IROS '06*, 2006.

[14] J. Sastra, S. Chitta, and M. Yim, Dynamic Rolling for a Modular Loop Robot, *Int. Symposium on Experimental Robotics*, July 2006.

[15] W. Shen, B. Slemi, and P. Will, Hormone-inspired Adaptive Communication and Distributed Control for CONRO Self-reconfigurable Robots, *IEEE Trans. on Robotics and Automation*, vol. 18, pp. 700-712, Oct. 2002.

[16] K. Stoy and R. Nagpal, Self-Reconfiguration Using Directed Growth, in *Distributed Autonomous Robotic Systems 6*, R. Alami, *et al.*, Eds., ed: Springer Japan, 2007, pp. 3-12.

[17] M. Yim, A Reconfigurable Modular Robot with Many Modes of Locomotion, in *Proc. JSME Int. Conf. Advanced Mechatronics*, pp.283-288, 1993.

[18] M. Yim, Y. Zhang, K. Roufas, E. Duff, and C. Eldershaw, Connecting and Disconnecting for Chain Self-reconfiguration with Polybot, in *Proc. IEEE/ASME Trans. Mechtron*, vol. 7, pp.442-451.Dec. 2002.

[19] H. Ueno, Y. Wakabayashi, Y. Ohkami, S. Matunaga, R. Hayashi, and T. Yoshida, Ground Testbed of a Reconfigurable Brachiating Space Robot, *Advanced robot*, vol. 14, no. 5, pp.355-358, 2000.

[20] C. Unsal, H. Killiccote, and P.K. Kholsa, A Modular Self-Reconfigurable Bipartite Robotic System: Implementation and Motion Planning, *Autonomous Robots*, vol. 10, pp.23–40, 2001.

[21] P. White, V. Zykov, J. Bongard, and H. Lipson, Three Dimensional Stochastic Reconfiguration of Modular Robots, in *Proc. of Robotics: Science and Systems Conference*, pp. 161–168, 2006.

[22] L. Wolpert, *Principles of Development*, Oxford Unviersity Press, 2002.