

Meta-module Self-configuration Strategy for Modular Robotic System

Zhen Yang, Zhuang Fu^(✉), Enguang Guan, Jiannan Xu, and Hui Zheng

State Key Laboratory of Mechanical System and Vibration, Shanghai Jiao Tong University,
Shanghai 200240, People's Republic of China
zhfu@sjtu.edu.cn

Abstract. Modular robotic system (MRS) consisting of several identical modules is able to adapt to situations by configuring to the morphology which best suits the environment. Utilizing this characteristic, this paper presents a self-configuration strategy for the M-Lattice MRS. Based on this strategy, the construction task under complex environment is converted to a self-configuration task accomplished by modular robots. Meta-module method is utilized to avoid system cracking. Meanwhile, a gradient greedy method in a virtual gravity field is selected to indicate the locomotion of meta-modules. Computational simulations demonstrate the feasibility and scalability of the proposed strategy.

Keywords: Self-configuration · Meta-module · Modular robot · M-Lattice

1 Introduction

Over recent years, applications of MRS have increasingly drawn the attention of the robotics and control communities. MRS made up of a given number of modules can attain different configurations based on their relative coupling. Using this property, this paper introduces a self-configuration strategy for the M-Lattice MRS. Under this self-configuration strategy, the traditional construction task under complex environment is converted to a self-configuration task for MRS.

Modular robots can be geometrically classified into three types: chain-type, lattice-type and hybrid-type. And besides compact modular robots, there are several metamorphic modular robots. Chirikjian [1] develops a metamorphic MRS. The metamorphic module consists of six-bar linkages. And two linkages are connected by a rotation joint. The whole system is an aggregation of several hexagon modules. The metamorphic property makes the system have better performance in self-reconfiguration than compact modular systems. Inspired by the design of metamorphic module and the unit-compressible modular robot, Crystal [2], the system grid of M-Lattice is designed to be a hexagon which consists of six unit-compressible modules.

Using a MRS to solve the construction task is first proposed by Terada and Murata [3]. In their work, the construction task is accomplished by heterogeneous modules, the cubic structure modules and the assembler robots. In our work, a homogeneous modular robot is chosen to finish the construction task.

The control strategies for self-reconfiguration have referential meanings for self-configuration. In a typical centralized control strategy such as the simulated annealing method [4], current configuration and target configuration are compared in real time. The controller drives specific modules into several target positions. As the centralized control strategy suffers from the system scale, distributed control strategy attracts more attention. Golestan [5] uses a graph signature method to analyze relationships between module connections and target configurations. Zhu [6] simplifies the cellular automata method in self-reconfiguration task, using UBot. In our self-configuration strategy, modules moves along the system boundary without presetting their final positions. And the global message is unnecessary. Enlightened by Park [7] and Arney [8], a finite state machine is used to change modules from movable status to stable status.

Meta-module can be defined as a group of modules which act as one unit. During the locomotion, the modules that compose a meta-module keep the connection with each other. Compared with single module, meta-module gets more flexible mobility. Several studies have been reported in literature on the meta-module applications. One primary motion mode for meta-modules is crawling along the system boundary like a tiny chain MRS, such as Telecube [9] and ATRON [10]. Dewey [11] introduces a control strategy that has a good performance in flexible meta-module systems. In our work, meta-module is designed to be capable of crawling along the system boundary and then entering into the optimal position. Owing to the feature of self-configuration task, a distributed control strategy is designed to indicate the locomotion of meta-modules. And inspired by Christensen [12], a virtual gravity gradient field is used to evaluate the possible target positions for meta-modules.

2 Design of M-Lattice Modular Robot

The most challenging task in self-configuration is how to deal with the motion constrains. In a MRS, the motion constrains can be classified into two types: local constrains and global constrains.

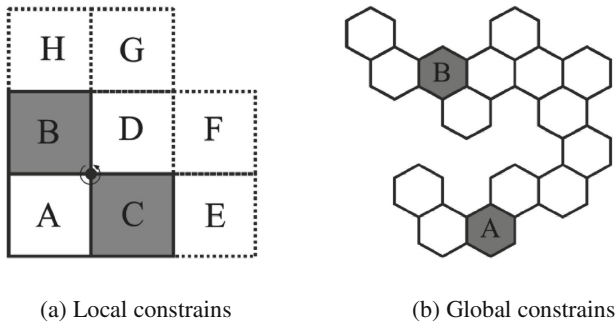


Fig. 1. The motion constrains. (a) Local constrains. If a square module wants to move from C to B around module A, the positions of D, E, F, G and H must be empty; (b) Global constrains. During the configuration process, the hexagon module A and B can't move. Otherwise, the system will break apart.

As shown in Fig. 1, local constrains is mainly determined by the module structure and the motion mode. Global constrains is a critical problem in the development of path planning algorithms. Thus at the stage of module structure design, besides of the efficient module transportation and reliable connection, there is an extra essential requirement that the module structure should release the local constrains as much as possible.

2.1 Basic Module

M-Lattice modular robot is designed to consist of two parts: a centre prism frame and three identical mechanical arms. As shown in Fig. 2, three mechanical arms are located on the symmetrical sides of the prism frame. Each mechanical arm contains two DOF which are the rotation DOF and expansion/contraction DOF. The rotation DOF drives one module connect with different modules and then changes module's position. The expansion/contraction DOF can change the rotation radius which could release the local constrains significantly. In addition, one module can keep the connection with three adjacent modules simultaneously [13].

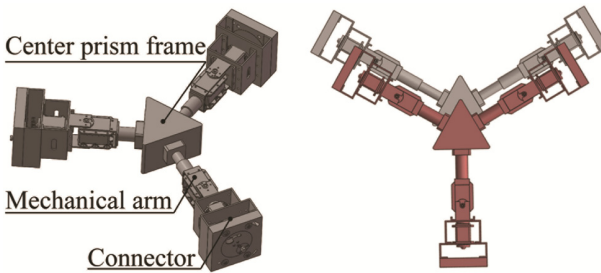


Fig. 2. M-Lattice modular robot (TFR-ISO view and expansion/contraction DOF)

2.2 Basic Motion

The locomotion of M-Lattice module is realized through the attach/detach operations and the rotations of mechanical arms within a sequence of connected modules. During the locomotion, one module in the module sequence called *Base* keeps the connection with system while the rest modules of sequence called *Assistants* are disconnected with system. Thus the module locomotion can be marked as a single-base and multi-assistants (SBMA) locomotion. The SBMA locomotion can be described in detail as follows. (1) Based on the initial position and target position, the module sequence can be determined. One specific module is chosen as *Base*. (2) Except for *Base*, all the modules within the sequence break the connections with system, then all the modules contract their mechanical arms. Based on the number of modules within the sequence, mechanical arms rotate a certain angle. Then module sequence forms a closed polygon. (3) One module breaks its former connection within the sequence. All the mechanical arms rotate back to their initial status. Then modules expand their mechanical arms and reconnect themselves with system.

An example of SBMA locomotion is shown in Fig. 3. The number of modules within the sequence is three and only one module is ordered to change its position. It can be found that owing to the expansion/contraction DOF, whether the positions adjacent to the module sequence are occupied or not, there is no collision risk.

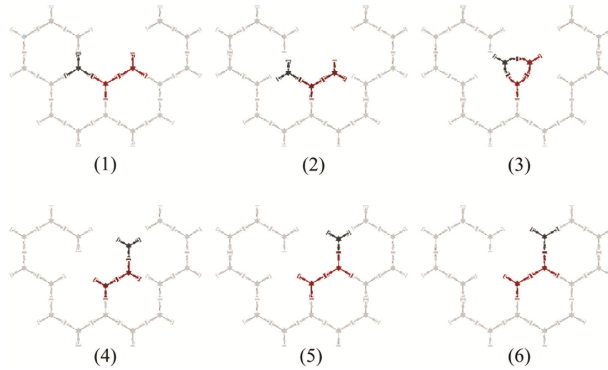


Fig. 3. An example of SBMA locomotion with three modules

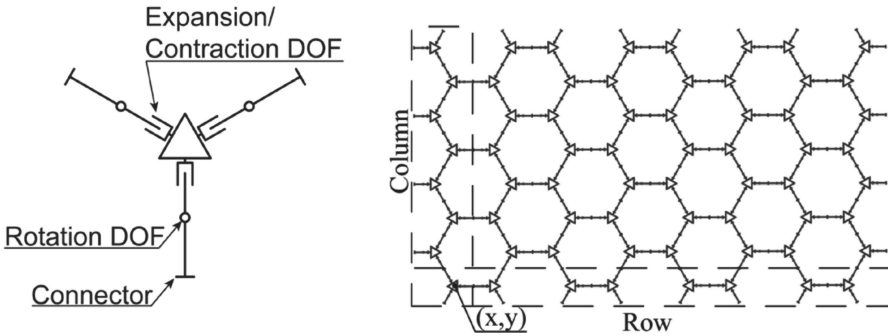
As mentioned before, mechanical arms need to rotate a certain angle to form a closed polygon. According to the geometric relationship, every module can be treated as three vertices in the closed polygon. The number of edges can be found as $E = 3n$. To compose a closed polygon, the rotation angle changes with the number of modules within the sequence as follows:

$$\frac{2\pi}{3}n + (\pi - \theta) \times 2n = \pi \times (E - 2) \Rightarrow \theta = \pi \times \frac{6 - n}{6n}$$

where n is the number of modules within the sequence; E is the number of edges of the closed polygon; θ is the rotation angle of mechanical arms.

2.3 System Structure

Six interconnected modules compose a hexagon grid and many hexagon grids connected together will form a huge mesh plane. In order to express modules' positions and connection status clearly in the following sections, the module is described in its topological form, as shown in Fig. 4(a). In our work, the position of each module is marked under a special X - Y index frame, as shown in Fig. 4(b). Under this index frame, each module's position is represented by an ordered pair of integers (x, y) , where x represents the column number of module's location and y represents the row number. In order to analyze the self-configuration task, some definitions are given firstly.



(a) Topological form of M-Lattice module frame (b) The X-Y index

Fig. 4. System structure of M-Lattice MRS

Definition 1. The configuration space of M-Lattice MRS is represented as a vertex index matrix, marked as VI . The form of the matrix VI is shown as follows:

$$VI = \begin{bmatrix} 1 & \cdots & X \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}$$

where each element $VI(x, y)$ represents a position that can be occupied by a module and (x, y) is the index representation introduced in Fig. 4(b).

- $VI(x, y) = 1$: the current vertex is occupied by a module;
- $VI(x, y) = 0$: the current vertex is empty;
- $VI(x, y) = X$: the current vertex can't be occupied by a module. This could be caused by the reserve for special shape buildings.

Definition 2. The connection status of M-Lattice MRS in the configuration space is represented by the adjacent edge matrix, marked as AE . If the position of module is (x, y) , the connection status of this module is marked as follows:

$$AE = \begin{bmatrix} x & y + 1 & 1 \\ x & y - 1 & 0 \\ x \pm 1 & y & X \end{bmatrix}$$

where each row represents a connection status for module (x, y) .

- $AE(i, 3) = 1$: the i -th adjacent position is occupied by a module;
- $AE(i, 3) = 0$: the i -th adjacent position is empty;
- $AE(i, 3) = X$: the i -th adjacent position can't be occupied by a module. This could also be caused by the reserve for special shape buildings.

Under the X-Y index frame of M-Lattice MRS, modules have two possible gestures. The only difference between these two gestures is the relative location of the adjacent

module which has the same index value along Y -axis as module (x, y) . Thus in the matrix \mathbf{AE} , if the position of module is (x, y) , the first row represents the connection status of upper adjacent position $(x, y+1)$, the second row represents the connection status of under adjacent position $(x, y-1)$ and the third row represents the connection status of left/right adjacent position whose index representation is $(x-1, y)$ or $(x+1, y)$ determined by the gesture of module (x, y) .

Based on these definitions, the network composed by M-Lattice modules can be represented as an undirected weighted graph $G = (\mathbf{VI}, \mathbf{AE}, w)$, where \mathbf{VI} is the vertex index matrix which means a set of modular robots, \mathbf{AE} is the adjacent edge matrix representing how the modules are connected, and w is a weight function. Thus the self-configuration task is now transformed to an abstract problem that is how to convert the vertex index matrix \mathbf{VI} to a nonzero matrix via the locomotion of modules. Nonzero matrix \mathbf{VI} means all the positions in the configuration space are occupied by modules or reserved for special shape buildings.

3 Self-configuration Strategy Based on Meta-module Locomotion

During the SBMA locomotion, the disconnection between *Assistant* and system is critically dangerous for maintaining the system connectivity. This problem can be solved as follows. It can be found that the primary problem is that it's hard to make sure whether *Assistant* is the only module connected with two local systems. But if the locomotion of single module does not involve any existing system module, system connectivity can be guaranteed with no doubt. Although the SBMA locomotion determined by the module structure and connection manner is unchangeable, we can give a new definition for 'single module'. That is the meta-module method.

Meta-module is a unit which consists of several basic modules. Although meta-module increases the granularity of the system, it has a significantly better mobility than the single module. Utilizing a meta-module method, the SBMA locomotion may not involve any existing system module and a single meta-module is able to move itself from one position to another. In our work, meta-module is designed to consist of four M-Lattice modules. During the locomotion of meta-module, the connection relations of meta-module are not fixed. The locomotion of meta-module along the system boundary is a succession of cycles of sense-think-act. These three parts can also be called as path searching, path planning and polymorphic locomotion.

3.1 Path Searching

As a special module unit, the locomotion of meta-module is realized through connecting with different modules. Path searching is executed by the meta-module members connected with system. In the M-Lattice MRS, each grid is a hexagon containing six modules. In order to search all the possible positions, the depth of path searching is set as four. All the possible positions are saved as follows:

$$Path = \begin{bmatrix} TgX_1 & TgY_1 & SysX_1 & SysY_1 & MetaX_1 & MetaY_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ TgX_n & TgY_n & SysX_n & SysY_n & MetaX_n & MetaY_n \end{bmatrix}$$

where $(TgXi, TgYi)$ is the location of the possible target position; $(SysXi, SysYi)$ is the location of the system module which is connected with the searching meta-module member; $(MetaXi, MetaYi)$ is the location of the searching meta-module member.

During the path searching, each meta-module member sends a message to its adjacent system module. If there is an empty place adjacent to it, the system module will send a message back to the meta-module member. If there exists another adjacent system module, it sends a message to the subsequent adjacent module and asks for empty place information. The search depth is four, and then meta-module gets all the empty places within this range.

3.2 Path Planning

From the grid geometry, it can be found that a hexagon grid is confirmed by three connected modules while two adjacent grids is confirmed by two connected modules. For each row of data in **Path**, the target position $(TgXi, TgYi)$ is filtered whether it is contained in the grid which is confirmed by the searching meta-module member and its adjacent system module. As each meta-module consists of four M-Lattice modules, the target position contained by a grid which also contains a meta-module member at least is reachable for the meta-module in theory.

A gradient greedy method in a virtual gravity field is utilized to choose the optimal target position from all the reachable positions. In the configuration space, the direction of virtual gravity is set along the X -axis and the position which has the largest gradient value is selected as the optimal target position. In each locomotion round, the meta-module member with the smallest index representation is set to stand for meta-module. This module determines a possible target position vectors \vec{V}_i from its position to the possible target position. The projection gradient $grad(i)$ along the X -axis vector \vec{V}_m for each possible target position vector is calculated. If the projection gradient is positive, the probability P of taking $grad(i)$ as the target gradient is given by the Boltzmann's law. And the modules with negative projection gradients are recorded with a small probability.

$$grad(i) = \left\langle \vec{V}_i, \frac{\vec{V}_m}{|\vec{V}_m|} \right\rangle \quad P(gradT = grad(i)) = \begin{cases} \frac{e^{grad(i)/\tau}}{n} & grad(i) > 0 \\ \sum_{i=1} e^{grad(i)/\tau} & \\ \epsilon & grad(i) < 0 \end{cases}$$

where $grad(i)$ is the projection gradient of position i ; τ is a positive temperature parameter in the Boltzmann distribution; ϵ is a small greedy probability for negative gradient; $gradT$ is the target gradient.

If several target positions have the same probability, the one with largest index representation along *Y*-axis is chosen as the optimal target position. This choice guarantees that the subsequent incoming meta-modules will have further search space and increases the success rate of self-configuration.

3.3 The Polymorphic Locomotion of Meta-module

The polymorphic locomotion of meta-module is a sequence of module’s motions to connect the meta-module with target position, absolutely based on meta-module members. In order to adapt to various target positions, the structure of meta-module is polymorphic just like a water-flow. In order to explain the locomotion clearly, some definitions are given firstly.

Definition 3. Target grid is the grid which contains the searching meta-module member, its adjacent system module and the target position.

Definition 4. Tail module is a meta-module member. It belongs to the target grid, and it is the tail of the meta-module sequence which means that Tail’s adjacent position should be empty.

Definition 5. Transitional grid can be confirmed by the tail module, its adjacent empty position and its adjacent position which does not belong to the target grid.

The relations of target grid, tail module and transitional grid are shown in Fig. 5.

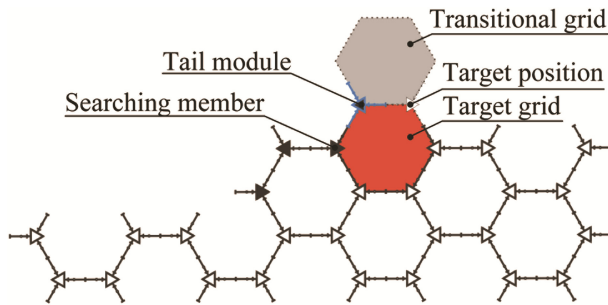


Fig. 5. The presentation of target grid, tail module and transitional grid

The polymorphic locomotion of meta-module is designed as a succession of stable grid transitions where meta-module members are driven into the target grid until the target position is occupied by a meta-module member. The polymorphic locomotion consists of two stages: combination to the transitional grid and separation to the target grid.

In the combination stage, all the meta-module members which do not belong to the target grid are relocated into the transitional grid. During the combination process, if a

meta-module member is outside of both target grid and transitional grid, it will be relocated into the transitional grid under the SBMA locomotion. One representation of combination is shown in Fig. 6.

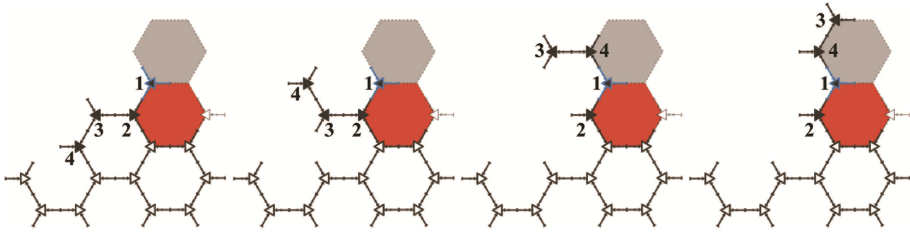


Fig. 6. One representation of combination process

In the separation stage, meta-module members in the transitional grid are relocated into the target grid. If the target position is occupied by a meta-module member, the polymorphic locomotion ends. The separation process is similar with the combination while the main difference is that, in the separation, the amount of meta-module members which should move into the target grid is determined by the occupied status of target grid. At the beginning of the separation, target grid is assumed as a connected half-occupied grid and the occupied amount can be obtained. Then meta-module members in the transitional grid start to move into target grid from the empty position adjacent to the tail module. One representation of separation is shown in Fig. 7.

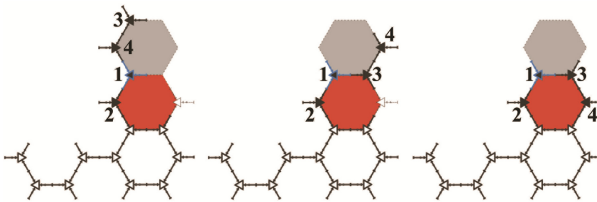


Fig. 7. One representation of separation process

4 Simulations of Self-configuration

A simulator to evaluate the properties of meta-module self-configuration strategy is built on MATLAB, running on a desktop with CPU Pentium E8400, RAM 4 GB.

The simulation scene is shown in Fig. 8. M-Lattice modular robot is represented as a node consisting of a filled circular and three bold lines for simplicity. For arbitrary configuration structure, the initial configuration space is assumed as rectangle and the system is initialized under the presentation of an undirected weighted graph $G = (VI, AE, w)$. Module supply area is on the left. Meta-module consisting of four modules is taken into

configuration space from the module supply area and then crawls along the system boundary step by step. After a meta-module is terminated, the next meta-module starts to move. If the entrance is blocked by system modules, one simulation round is finished.

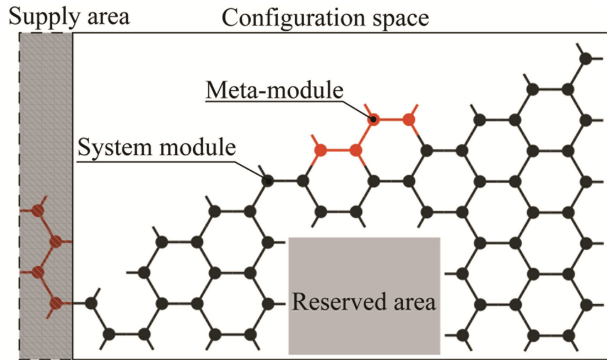


Fig. 8. Simulation scene

In order to test the scalability of self-configuration strategy, the system is set with different scales from 10×10 to 60×60 . And two structures are chosen as the target configurations which are rectangle and L-shaped geometric structure. During the self-configuration process, path searching, path planning and polymorphic locomotion are operated by the meta-module in a distributed manner. If a collision happens, the simulation would be shut down. If system cracks because of SBMA locomotion, the self-configuration task is considered as a failure, and the simulation is also terminated. The self-configuration process of a configuration space with 400 (20×20) modules is shown in Fig. 9.

The simulations are repeated twenty times for every system scale. For each experiment, time consumption is recorded. As shown in Fig. 10, time consumption grows approximately exponentially as the number of modules increases. There are some reasons. First, that is because in the simulation, the local and global constrains are monitored by the simulator. And this operation consumes extra time which is not pertinent to the self-configuration process. Second, combination and separation process cost extra time because of the transitional grid. We are working on a modified locomotion manner where the transitional grid is unnecessary. Third, in a huge configuration space, it is possible to realize parallel self-configuration while multiple meta-modules entry into the configuration space at regular intervals and self-configure at the same time. Parallel process shortens the time consumption and is achievable because of the distributed property of self-configuration strategy where the involved modules of path searching, path planning and polymorphic locomotion are limited.

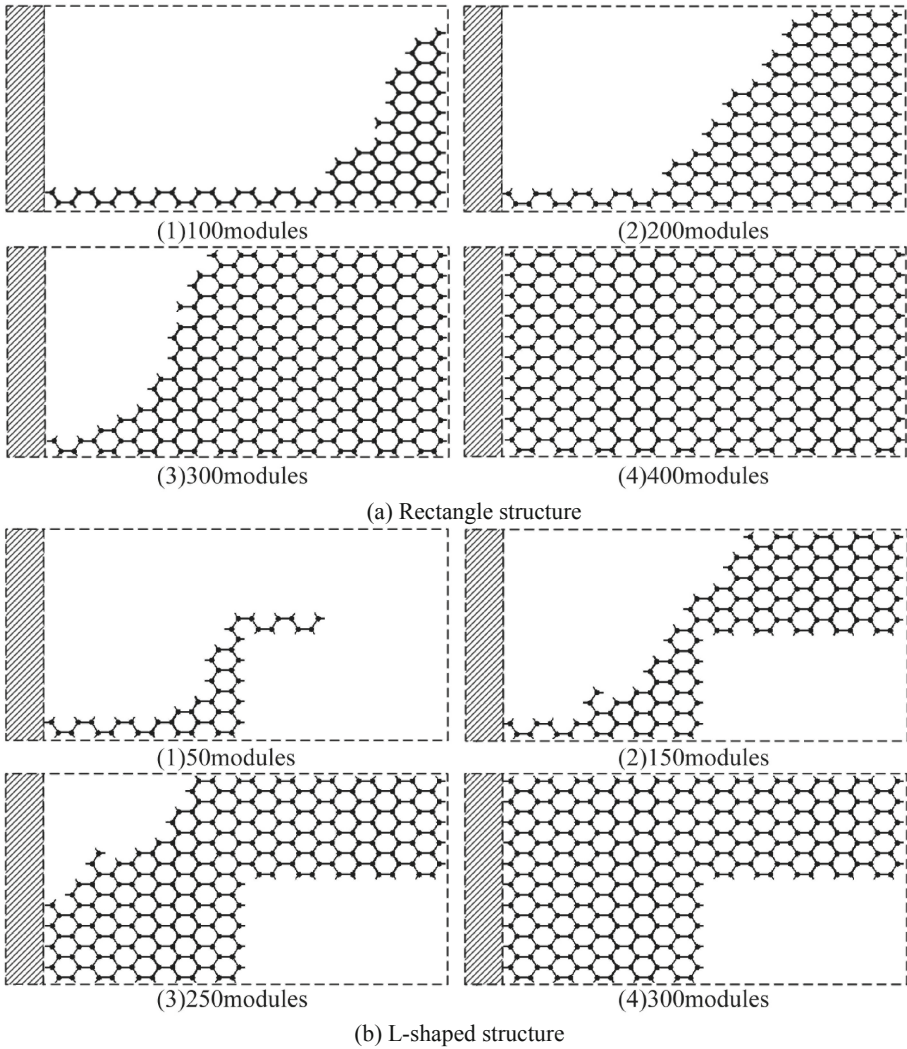


Fig. 9. Self-configuration processes of rectangle and L-shaped geometric structure

In all of the simulation rounds, the success rate of self-configuration always maintains 100%. Furthermore, through analysing the message load, the behaviours of meta-modules in the self-configuration process can be approximately evaluated. And this can be considered as a characterization factor of system scalability. It can be found that the message load grows approximately linearly as the number of modules increases. The scalability of self-configuration is advisable. And the property of scalability needs further research.

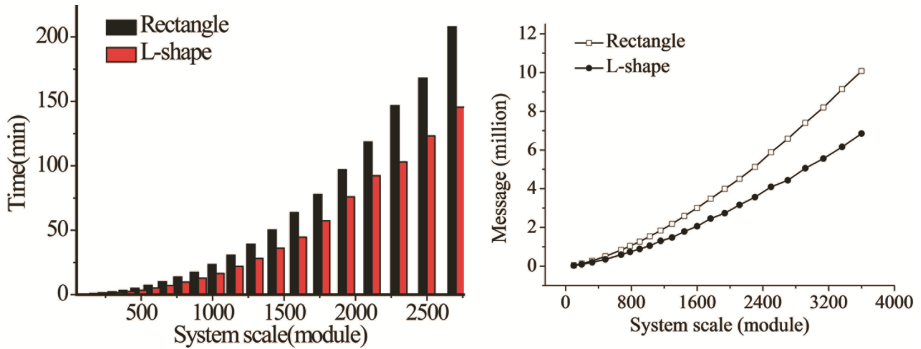


Fig. 10. Time consumption and message load of self-configuration process

5 Conclusions

This paper proposes a novel self-configuration strategy for the M-Lattice MRS. The validity and scalability of this strategy for M-Lattice MRS have been presented. The module design and topological features of the M-Lattice MRS have been described. The construction task under complex environment is converted to a self-configuration task accomplished by modular robots. A meta-module method is designed to realize the self-configuration process. Formed by four M-Lattice modules, each meta-module changes its position through connecting with different modules. While the connection status is changeable within the meta-module members, the connections within the system do not change during the self-configuration process, which reduces the global constraints successfully. Under our self-configuration strategy, single meta-module is able to finish path searching, path planning and polymorphic locomotion to form a specific target structure. Results from simulations have shown that the self-configuration strategy for the M-Lattice MRS has an advisable scalability for the regular target structures. However, it still needs further research about it. And it may be applied in the domain of scalable MRS.

Acknowledgments. This work is partially supported by the National Natural Science Foundation of China (Grant No. 61473192, U1401240, 61075086) and National Basic Research Program of China (2014CB046302).

References

1. Chirikjian, G.S.: Kinematics of a metamorphic robotic system. In: IEEE International Conference on Robotics and Automation, pp. 449–455 (1994)
2. Rus, D., Vona, M.: Crystalline robots: self-reconfiguration with compressible unit modules. *Autonom. Robots* **10**(1), 107–124 (2001)
3. Terada, Y., Murata, S.: Automatic modular assembly system and its distributed control. *Int. J. Robot. Res.* **27**(3), 445–462 (2008)

4. Chiang, C.J., Chirikjian, G.S.: Modular robot motion planning using similarity metrics. *Autonom. Robots* **10**(1), 91–106 (2001)
5. Golestan, K., Asadpour, M., Moradi, H.: A new graph signature calculation method based on power centrality for modular robots. In: Martinoli, A., Mondada, F., Correll, N., Mermoud, G., Egerstedt, M., Hsieh, M., Parker, L.E., Støy, K. (eds.) *Distributed Autonomous Robotic Systems*. STAR, vol. 83, pp. 505–516. Springer, Heidelberg (2013)
6. Zhu, Y., Bie, D., Iqbal, S., Wang, X., Gao, Y., Zhao, J.: A simplified approach to realize cellular automata for UBot modular self-reconfigurable robots. *J. Intell. Robot. Syst.* **79**(1), 1–18 (2014)
7. Park, H.W., Ramezani, A., Grizzle, J.W.: A finite-state machine for accommodating unexpected large ground-height variations in bipedal robot walking. *IEEE Trans. Robot.* **29**(2), 331–345 (2013)
8. Arney, D., Fischmeister, S., Lee, I., Takashima, Y., Yim, M.: Model-based programming of modular robots. In: *IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing*, pp. 66–74 (2010)
9. Aloupis, G., Collette, S., Damian, M., Demaine, E.D., El-Khechen, D., Flatland, R., Langerman, S., O'Rourke, J., Pinciu, V., Ramaswami, S., Sacristán, V., Wuhler, S.: Realistic reconfiguration of crystalline (and telecube) robots. In: Chirikjian, Gregory S., Choset, H., Morales, M., Murphey, T. (eds.) *Algorithmic Foundation of Robotics VIII*. STAR, vol. 57, pp. 433–447. Springer, Heidelberg (2009)
10. Shokri, A., Masehian, E.: A meta-module approach for cluster flow locomotion of modular robots. In: *Proceeding of the 2015 3rd RSI International Conference on Robotics and Mechatronics*, pp. 425–431 (2015)
11. Dewey, D.J., Ashley-Rollman, M.P., Michael, D.R., et al.: Generalizing metamodules to simplify planning in modular robotic systems. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1338–1345 (2008)
12. Christensen, D.J., Østergaard, E.H., Lund, H.H.: Metamodule control for the atron self-reconfigurable robotic system. In: *The 8th Conference on Intelligent Autonomous Systems*, pp. 685–692 (2004)
13. Guan, EG., Yan, WX., Jiang, DS., Fu, Z., Zhao, YH.: A novel design for the self-reconfigurable robot module and connection mechanism. In: *3rd International Conference on Intelligent Robotics and Applications*, pp. 400–408 (2010)