Autonomous Self-Reconfiguration of Modular Robots by Evolving A Hierarchical Mechanochemical Model

Yan Meng[1], Yuyang Zhang[1], and Yaochu Jin[2]

[1]Department of Electrical and Computer Engineering
Stevens Institute of Technology, New Jersey, USA
[2]Department of Computing, University of Surrey, Guildford, Surrey, UK
yan.meng@stevens.edu, yzhang14@stevens.edu, yaochu.jin@surrey.ac.uk

I. Introduction

Self-reconfigurable modular robots are autonomous robots with a variable morphology, where they are able to deliberately change their own shapes by reorganizing the connectivity of their modules to adapt to new environments, perform new tasks, or recover from damages. Each module is an independent unit that is able to connect to or detach from other units to form various structures / patterns. Compared with conventional robotic systems, self-reconfigurable robots are potentially more robust and more adaptable under changing environments.

Modular robots can be generally classified into two groups, according to their geometric arrangements of the modules: the chain / tree-based architectures [1] [2] [3] [4] and the lattice-based architectures [5] [6] [7] [8] [9] [10] [11] [12] [13]. In the chain / tree-based architectures, the modules are connected in a topology of a chain or a tree, where the motion controls of the modules are executed sequentially. It is relatively easier to design and implement this kind of architectures. In the lattice-based architecture, modules of a robot are usually arranged and connected in 3D patterns, such as a cubical or hexagonal grid, and the motion control of modules are carried out in parallel. Therefore, compared to the chain / tree-based architectures, the lattice-based architectures are more flexible and efficient to form complex structures, although the design and implementation of this kind of architectures are more difficult. From this point of view, lattice-based modular robots are more suited for dynamic environments. However, most existing lattice-based modular robotic systems can only reconfigure themselves into a few predefined patterns by following a set of user-defined rules. Although self-reconfiguration is believed to be the most important feature of modular robots, the ability to adapt their configuration autonomously under environmental changes largely remains to be demonstrated.

In principle, both centralized and distributed controllers can be used for reconfiguring modular robots. However, centralized high-level controllers for lattice-based modular robots are vulnerable to system failures or malfunctions of robot modules. By contrast, decentralized controllers are more robust and flexible under uncertain environments. However, one major challenge in developing a decentralized controller for self-reconfigurable modular robots lies in the coordination of local behaviors of multiple modules to achieve the desired global pattern to adapt to environmental changes. To address this challenge, some researchers turn their attentions to biological systems, where rich examples can be found of which robust and complex emergent behaviors are generated through relatively simple local interactions in the presence of uncertainties [14][15]. Inspired by the biological concept of hormone, Shen et al. [16] proposed a hormone-inspired adaptive communication protocol and adaptive distributed control protocol to allow modules to use hormone-like messages to accomplish locomotion and self-reconfigurations. Later, Schmickl et al. [17] developed a control system called AHHS (artificial homeostatic hormone systems) for the self-reconfiguration of robotic organisms, where the parameterization of the underlying dynamical system was encoded into a “genome” or the organism which is subject to selection through evolutionary algorithms. Stoy [18] proposed a cellular automata and gradients to control the self-reconfigurable modular robots. Pfeiffer and Bongard [19] mainly focused on using the evolutionary algorithms to generate the robot configurations using an embodied system. Recently, Spröwitz et al. [20] developed reconfigurable modular robots called Roombots used as adaptive furniture, where Roombots can change shape and function adaptively according to the needs of human users in a daily life. A central pattern generator (CPG) network was applied for locomotion control of the Roombots to produce robust, synchronized patterns for oscillatory and rotational joint movements. A stochastic optimization algorithm was used to optimize the CPG configurations.

Inspired by the embryonic development of multi-cellular organisms [21] and a mechanochemical model for cell morphogenesis [22], we proposed a morphogenetic approach for a modular robot using a hybrid hierarchical model in our previous work [23]. In [23], there are two layers in the model, where the first layer is a look-up-table based method to predefine the target pattern, and the second layer is a gene regulatory network (GRN) based controller to move the modules to construct the target pattern. The major limitation of this work is that the target pattern has to be predefined which cannot autonomously adapt to dynamic environments. Therefore, in this paper, we extend our previous work and propose a two-layer hierarchical morphogenetic mechanochemical framework for self-reconfiguration of modular robots. Layer 1 is responsible for autonomous generation of chemical patterns in a changing environment, which is based on a mechanochemical model. Layer 2 aims to physically realize the target pattern for the modular robot, which is a gene regulatory network (GRN) based controller. Furthermore, to optimize the pattern design of modular robots, the covariance matrix adaptation evolution strategy (CMA-ES) [24][25] is employed to evolve the pattern parameters of the mechanochemical model.

The proposed modular robots are also called morphogenetic modular robots, belonging to a new emerging subfield of developmental robotics termed morphogenetic robotics [26]. Morphogenetic robotics (MR) refers to a class of methodologies in robotics for designing self-organizing, self-reconfigurable, and self-repairable single- or multi-robot systems using genetic and cellular mechanisms governing biological morphogenesis. MR can be categorized into three main areas: morphogenetic swarm robotic systems (MSR) [27][28][10], morphogenetic modular robots (MMR) [23][16], and morphogenetic body and brain design for robots (MBB) [29][30].

The rest of the paper is organized as follows. The basic mechanics and locomotion design of a simulated lattice-based modular robot are described in Section II. Section III presents the morphogenetic framework for self-organization of modular robots. Section IV provides extensive simulation results to evaluate the proposed morphogenetic approach to self-reconfiguration of modular robots under changing environments. Section V concludes the paper with future work.

II. CrossCube: A Simulated Modular Robot

CrossCube is a simulated lattice-based modular robot we developed in a robot simulator using a real time physics engine PhysX. Each module of CrossCube has a flexible single cubic shape like Molecube [13], which does not require much free space for modules to move around, similar to the mechanics of SUPERBOT [31][32][33] and MTRAN [8] [34] [35]. Each module of CrossCube is a cubical structure, having its own computing and communication resource and actuation capability. Like other modular robots, the connecting parts of the modules can easily be attached to or detached from other modules. Each module can perceive its local environment and communicate with its neighboring modules using on-board sensors.

Each CrossCube module consists of a core and a shell as shown in Fig. 1(a). The core is a cube with six universal joints. Their default heading directions include up, down, right, left, forward, and backward, respectively. Each joint can attach to or detach from the joints of its neighbor modules. The axis of each joint can be actively rotated, extended, and return to its default direction and length.

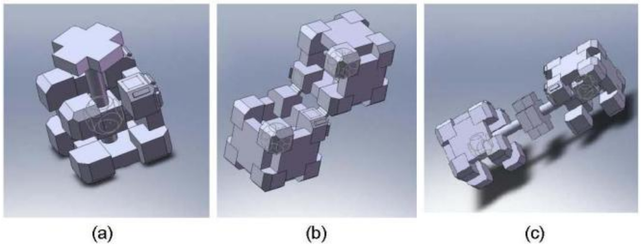


Fig 1. Mechanical demonstration of CrossCube. (a) The joints; (b) The locks on the boundaries of the modules. (c) Rotation and extension of the joints of the modules.

The cross-concaves on each side of the shell restrict the movement trajectory of the joints, as shown in Fig. 1(a). The borders of each module can actively be locked or unlocked with the borders of other modules, as shown in Fig. 1(b). Basic motions of modules in CrossCube include rotation, climbing and parallel motion. Fig. 1(c) illustrates a rotation movement of two modules. Parallel motion means that a module moves to an immediate neighboring position. Climbing motion means that a module moves to a diagonal neighboring position. Parallel motion and climbing motion allow a module of CrossCube to move to any position within the modular robot as long as the modules are connected. Since the major focus of this paper is the self-reconfiguration control algorithm, please refer to [23] for the detailed mechanical design of CrossCube.

III. The Morphogenetic Approach

A. Computational Modeling of Gene Regulatory Networks

Multi-cellular morphogenesis is under the control of gene regulatory networks (GRNs). These GRNs control a number of important cellular behaviors, such as responding to the environment, regulating the cell cycle and governing the development of an organism. A large number of computational models for GRNs have been suggested [36][37][38], which can largely be divided into discrete models such as random Boolean networks and Markov models, and continuous models such as ordinary differential equations and partial differential equations. GRN models can also distinguish themselves between deterministic models and stochastic models according to their ability to describe stochasticity in gene expression.

B. The Metaphor between the Self-Reconfigurable Modular Robots and Multi-cellular Organisms

The metaphor between reconfigurable modular robots and multi-cellular organisms is straightforward. We can treat each unit in modular robots as a single cell. The similarities in control, communication and physical interactions between cells in multi-cellular organisms and modules in modular robots are obvious. For example, the control in both modular robots and multi-cellular organisms is decentralized. Furthermore, the global behaviors of both modular robots and multi-cellular organisms emerge through local interactions of the units, which include mechanic, magnetic and electronic mechanisms in modular robots, and chemical diffusion and cellular physical interactions such as adhesion in multi-cellular organisms.

Each unit of the modular robots contains a chromosome consisting of several genes that can produce different proteins. The local communication between the modules can be setup by diffusing the proteins into neighboring modules. The concentration of diffused proteins decays over time and distance. The space where each module located in a physical 3D world is divided into a set of discrete grids, where each grid is occupied by one module of modular robots. The target configuration of modular robots is defined by the morphogen level of each grid. A higher value of morphogen level indicates a higher priority for the grid to be filled by a module.

C. The Proposed Hierarchical Framework

To develop a morphogenetic approach to self-reconfiguration of modular robots, two steps must be achieved. First, the target pattern that a modular robot needs to form has to be generated automatically based on the current environment, which is to be achieved by the layer 1 in the proposed hierarchical framework. Second, the modular robot needs to self-organize their modules to physically realize the target pattern generated by layer 1, which is the task of layer 2 in our hierarchical framework. Fig. 2 shows a block diagram of this hierarchical morphogenetic framework.

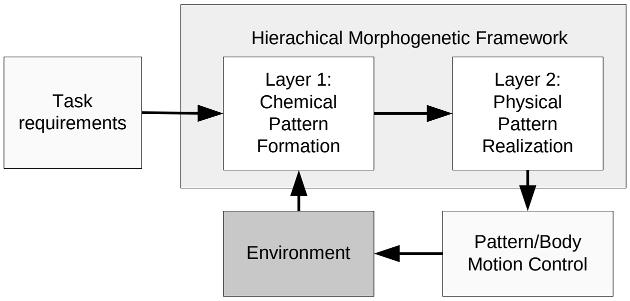

Hierachical Morphogenetic Framework
Task requirements
Layer 1: Chemical Pattern Formation
Layer 2: Physical Pattern Realization
Environment
Pattern/Body Motion Control

Fig. 2. A block diagram of the hierarchical morphogenetic framework.

The objective of this hierarchical morphogenetic model is to develop a more general distributed lattice-based modular robot which can reconfigure itself into any arbitrary patterns for locomotion tasks according to the current environment. For example, a vehicle-like pattern with wheels will be reconfigured if the robot detects that it is currently in an open space on a flat surface since this is the most efficient pattern to move faster. Similarly, a walking robot with legs will be reconfigured if the robot finds out that it has to traverse a very rough terrain with lots of obstacles on a soft flat surface. A climbing robot or a snake-like robot will be developed if the robots need to climb stairs or walls.

D. Layer 1: Chemical Pattern Formation

Adaptation to environmental changes is of paramount importance in reconfigurable modular robots. A mechanism is needed to adaptively define and modify the target configuration of the modular robot in a changing environment. Adaptation of the global configuration of the modular robot can be triggered by local sensory inputs. For such tasks, it is assumed that each module is equipped with a sensor to detect the distance between the module and obstacles in the environment. Once a module receives such sensory feedback, the information will be passed on to its neighbors through local communication. In this way, a global change in configuration can be achieved.

Interestingly, morphogenesis of multi-cellular systems provides a nice example in which self-organizing pattern formation is realized through cell-cell and cell-environment interactions. One cellular environment is a complex mixture of non-living material that makes up the extracellular matrix (ECM) [22]. In biological systems, the ECM is the extracellular part of animal tissue secreted by the cells that usually provides structural support to the cells. Meanwhile, the shape of ECM is also affected by the behaviors of the cells through the interactions between ECM and the cells. The interaction dynamics between the cells and the ECM can be described by a morphogenetic mechanochemical model proposed by Murray et al. [22][39]. In this model, the simultaneous development of pattern formation and morphogenesis is a closed-loop system where the extracellular matrix (ECM) can be treated as environmental constraints, which is, therefore, well suited for constructing layer 1 in a hierarchical GRN framework.

Main challenges in applying this mechanochemical model to self-reconfiguration of modular robots include: (1) the modules can only locate at discrete positions, while cells in organisms can move continuously. (2) In the mechanochemical model, the macro-level behavior of a large number of cells is considered in terms of density change, while in modular robots, the micro-level behavior of a much smaller number of modules is considered.

In this paper, we will tackle these issues using a virtual-cell (v-cell) based approach. The basic hypothesis and assumptions of this approach are listed as follows:

- There are three basic components in this model: v-cells, ECM, and environment (including task requirements and local environmental constraints). At the beginning, a predefined number of v-cells start to proliferate from a fixed grid. V-cells can proliferate, interact with other v-cells, and diffuse in a 3D space. The ECM and environment provide constraints on the v-cell behaviors. The difference between the density of the v-cells and the ECM value defines the morphogen level of each grid of a modular robot, which will be read in by the GRN of layer 2.
- Each discrete grid can contain multiple v-cells. The v-cells can only move to the grids that are either occupied by modules, or empty grids that are the modules' immediate neighbors. By following these rules, the v-cells will not move to the grids that are occupied by obstacles or disconnected from the robot system.
- Each grid is associated with one ECM value, which is determined by the density of v-cells of the immediate neighboring grids. If more than one ECM value is generated on the same grid by v-cells from different grids, the final ECM value is the sum of all the generated ECM values.

Based on the above assumptions, a new mechanochemical model for self-organization of modular robots is developed. As shown in Fig. 3, the v-cells (red dots) can move freely in the modules of a modular robot and their neighboring discrete grids, while the ECM and task requirements (for example, a locomotion task to traverse through different terrains) can produce environmental and task constraints to the behavior of the v-cells.

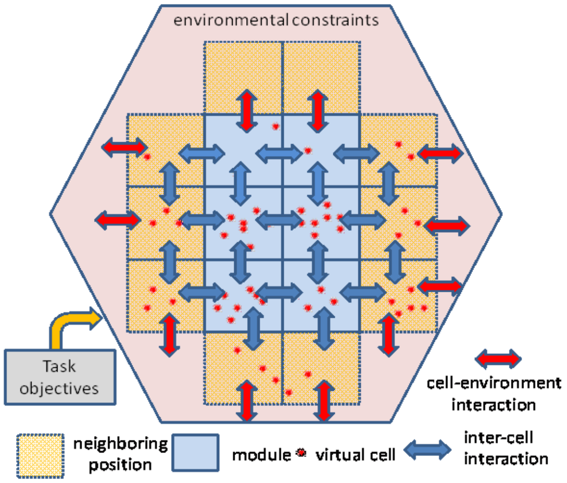


Fig. 3. The virtual-cell based mechanochemical model at a macro-level view.

The density of the v-cells and the ECM value in a grid can be defined as follows:

$$\frac{dn_i}{dt} = r \cdot n_i (N - n_i) - d \cdot K_i \cdot M_i - a \cdot \frac{\rho_i}{n_i + \rho_i} + \sum_k n_k^{rec} \tag{1}$$

$$\frac{d\rho_i}{dt} = -b \cdot \frac{n_i}{n_i + \rho_i} + e \cdot \sum_j f_{ji}(n_j) \tag{2}$$

where n_i and ρ_i represent the density of the v-cells and the ECM value in grid *i*, respectively. The first term in Eqn. (1) denotes the proliferation rate of the v-cells, where N is the predefined maximum number of v-cells allowed in the grid, and ($r \cdot N$) is the maximum value of linear proliferation rate of the v-cells in the grid. The proliferation rate of the v-cells will be expedited when the local density of v-cells is low, and will be reduced when the density of v-cells approaches to *N*.

$K_i \cdot M_i$ denotes the random dispersal of the v-cells, which depends on a dispersal control vector K_i and a density gradient vector M_i for grid *i*. Since the v-cells are allowed to move only to their immediate neighbors, v-cells can have up to six directions to move to: up, down, left, right, forward and backward. The dispersal control vector K_i is defined as $K_i = \left[k_i^{up}, k_i^{down}, k_i^{left}, k_i^{right}, k_i^{forward}, k_i^{backward}\right]$, where each element of K_i represents the dispersal rate for each direction. A higher dispersal rate means a faster dispersal of the v-cells in that direction. A zero dispersal rate means that the v-cells cannot move to that direction. To consider environmental constraints, for example, when an obstacle is detected, each dispersal control variable in vector K_i can further be defined as:

$$k_i^s = \begin{cases} 0 & \text{if neighbor position } s \text{ is occpied by obstacles} \\ h & \text{otherwise} \end{cases} \tag{3}$$

where k_i^s is the dispersal control variable in direction *s* (i.e., up, down, left, right, forward, or backward) in grid *i*. *h* is a predefined constant. $M_i = \left[m_i^{up}, m_i^{down}, m_i^{left}, m_i^{right}, m_i^{forward}, m_i^{backward}\right]^T$ is a vector that is affected by the density difference of v-cells between grid *i* and its 6 neighboring grids. For example, m_i^{up} is defined as $m_i^{up} = \max(0, n_i - n_{up})$,where n_{up} and n_i are v-cell densities of the upper neighboring grid and current grid *i*, respectively.

The flow control vector K_i plays an important role in dispersal of virtual cells. Expanding the second item of Eqn. (1), we have

$$\begin{aligned} & d * (k_i^{left} \cdot m_i^{left} + k_i^{right} \cdot m_i^{right} + k_i^{up} \cdot m_i^{up} + k_i^{down} \cdot m_i^{down} \\ & + k_i^{forward} \cdot m_i^{forward} + k_i^{backward} \cdot m_i^{backward}) \end{aligned} \tag{4}$$

which is the total number of v-cells that will flow out of the local grid *i*. The diffusion to each direction is scaled by the corresponding element in K_i . The third item in Eqn. (1) describes how the density of v-cells is decreased by the ECM value on the same grid. $\sum_k n_k^{rec}$ denotes the total densities of v-cells received by grid *i* from all the neighboring grids. *r*, *d*, and *a* are predefined constants.

Eqn. (2) describes the dynamics of the ECM value. The first item of Eqn. (2) describes how the density of local v-cells suppresses the ECM value. The higher the density of v-cells, the more the ECM value is reduced. The second item in Eqn. (2) denotes the process how the ECM value is generated in grid *i* based on the sum of densities of v-cells of neighboring grids. $f_{ji}(n_j)$ is the generated ECM value on grid *i* by grid *j*, where n_j is v-cell density in grid *j*.

The detailed rules of $f_{ji}(n_j)$ depends on the target pattern. Here, we use a vehicle-like pattern as an example to explain how to define $f_{ji}(n_j)$. This definition will be used for case study 1 in the experiments. We first provide the following design guidelines for the vehicle-like patterns. (1) The chassis of a vehicle pattern is normally designed as a rectangle. The total width of the vehicle is denoted by $W = x_{\max} - x_{\min} + 1$, where $x_{\max}$ and $x_{\min}$ are the rightmost and leftmost positions of the vehicle pattern. (2) Wheel modules are needed on the left and right boundary of the chassis, and cannot be immediate neighbors, otherwise they cannot rotate freely. (3) The target pattern should be built from bottom up. Only when the bottom floor is filled up and there are still modules left, the top floor will be filled. Based on these guidelines, $f_{ji}(n_j)$ can be calculated based on the following pseudo code.

```
Function $f_{ji}(n_j)$:
Initialization: $f_1 = 0$, $f_2 = 0$, and $n_{j\_last} = 0$.
if ( grid j is at the left side boundary and grid i is not on the right side of grid j)
    then $f_1 = 100 * n_j$;
if ( grid j is at the right side boundary and grid i is not on the left side of grid j)
    then $f_1 = 100 * n_j$;
if ( $n_j > 0$ and $n_{j\_last} = 0$, and grid i is located above grid j )
    then $f_2 = ECM_{up}$;
$n_{j\_last} = n_j$;
```

Return $f_1 + f_2$;

To optimize the chemical pattern formation using the mechanochemical model, the covariance matrix adaption evolution strategy (CMA-ES) is employed for tuning the parameters of $f_{ji}(n_j)$ in the model. CMA-ES is a stochastic, iterative optimization method belonging to the class of evolutionary algorithms (EAs) proposed by Hansen and Ostermeier [25][26]. For example, in the vehicle-like pattern, the parameters to be tuned are: the width of the vehicle W and ECM_{up}. More details are discussed in case study 1 in the experiments. Based on different target patterns, different $f_{ji}(n_j)$ is applied. Therefore, different parameters will be tuned by CMA-ES.

Based on the CMA-ES [25], λ offspring are generated from μ parents, where μ≤λ. Offspring individuals are generated by adding a random number generated from a normal distribution to a parent, or a recombination of a number of parents:

$$x_i^{(g+1)} \sim m^{(g)} + \sigma^{(g)} N_i(0, C^{(g)}) \qquad \text{for } i = 1, \ldots, \lambda \tag{5}$$

where "~" denotes the same distribution on the left and right side. $x_i^{(g+1)}$ is the *i*-th offspring individual in generation $g+1$, and x is a vector $x \in R^n$, which consists of the parameters need to be evolved. Vector $m^{(g)} \in R^n$ represents a recombination of a few parent individuals of generation g. $\sigma^{(g)}$ is the step-size that controls the step length of mutation in generation g. $C^{(g)}$ denotes the covariance matrix in generation g and $N_i(0, C^{(g)})$ is the *i*-th multivariate normal distribution with zero mean and covariance matrix $C^{(g)}$.

The final target pattern is determined by the morphogen level in each grid. In each discrete grid, the ECM value is inversely proportional to the density of the v-cells. As a result, v-cells tend to flow to the grids with lower ECM values. Therefore, the morphogen level mp_i of grid *i* can be defined as the difference between the density of the v-cells and the ECM value in each grid as $mp_i = n_i - \rho_i - z_i$, where z_i is a threshold which is determined by $(n_i - \rho_i)$ so that all the grids with positive morphogen levels belong to the target pattern, and others have negative morphogen levels. A higher value of morphogen level indicates a higher priority for the grid to be filled by a module. This mechanochemical process stops when the number of the grids with positive morphogen levels reaches the number of the available robot modules.

Both v-cells and ECM are virtual objects which don't physically exist. They are represented by variables which are stored in the memory of each module and can be exchanged between the neighboring modules through diffusion. For those grids occupied by modules, the density of v-cells and ECM value are estimated by the hosting modules. For those empty grids where no module exists, the density of v-cells and ECM values are maintained by neighboring modules in a distributed way. (Please be noted that only those empty grids which are immediate neighbors of the modules will be considered here). For example, one empty grid has two neighboring - modules A and B. When module A updates the density value of v-cells and ECM value on one empty grid, module A will broadcast this change to its neighbors so that module B can update this change in its own memory, and vice versa.

E. Layer 2: Physical Pattern Realization

The physical configuration of the modular robots to the target pattern generated by layer 1 is controlled by a GRN model. To this end, we need to build a metaphor between a multi-cellular organism and a modular robot. In this metaphor, each unit of the modular robot is considered as a cell containing a DNA. The DNA consists of a number of genes that can produce different proteins. The proteins can diffuse into neighboring modules (cells), through which local communication between the modules can be established. The concentration of the diffused proteins decays over time and distance.

As described above, the target pattern of the modular robot is defined by morphogen levels associated to the grids. To place the modules in the grids with positive morphogen level and keep grids with negative morphogen level empty, a local coordinate system must be built up. This can be achieved by setting an arbitrary module to be the origin of the coordinate system [40], and thus, all other modules can figure out their relative positions to the module on the origin through local communication. With the help of relative positions and the morphogen levels of grids in the target pattern, each module can decide: (1) if attracting proteins should be produced to attract other modules to fill in its neighboring grids, or (2) if repelling proteins should be produced to remove neighboring modules, or (3) if the module should respond to the attraction and repellence requests sent out by other modules. Thus, the reconfiguration of the modular robot is controlled through changing the state of the individual modules.

In the following subsections, we first describe the finite state machine of a module. Then, we introduce a GRN model that has two gene-protein pairs to control the desired state transitions of modules needed for realizing the target pattern.

(1) *Finite States of Modules:* Each module can have five different states, namely, 'stable', 'unstable', 'attracting', 'repelling', and 'repelled'. The "stable" state means the final state of the module. The "unstable" state means the module can respond to

attractions. The “attracting” state means the module is attracting other modules to fill in some of its neighboring grids. The “repelling” state means the module is repelling unwanted neighboring modules away. The “repelled” state means that the module responds to repelling requests and moves away from the current grid. The finite state machine of these five states of modules is given in Fig. 4.

When an ‘unstable’ module arrives at the destination grid, it changes its state to “stable” (arrow *a* in Fig. 4). When a ‘stable’ module has neighboring grids need to be filled, it will change its state to ‘attracting’ (arrow *b* in Fig. 4) to attract ‘unstable’ modules to fill in those grids. Once those neighboring grids are occupied, the ‘attracting’ module returns to the ‘stable’ state (arrow *c* in Fig. 4). A ‘stable’ module may also give up its current position to fill in a grid with higher morphogen level in the pattern by turning its state to ‘unstable’ (arrow *d* in Fig. 4).

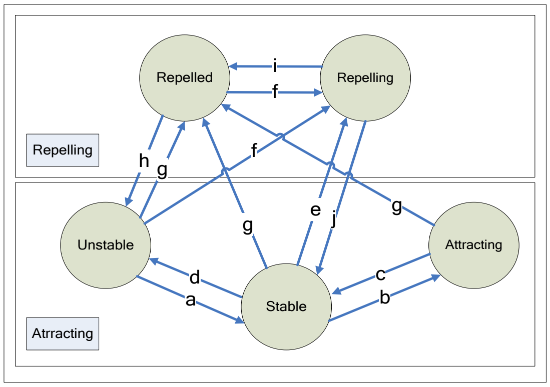


Fig. 4. State transition of each module in CrossCube.

A ‘repelling’ module rejects unwanted neighbor modules from their current positions by setting their states to ‘repelled’(arrow *g* in Fig. 4). Then the ‘repelled’ module moves away from its current position, and switches its state to ‘unstable’ (arrow *h* in Fig. 4). A module can be triggered to the ‘repelling’ state in two situations. The first one is when a ‘stable’ module finds out that some of its neighboring modules need to be removed (with a negative morphogen level), it changes its state to ‘repelling’ (arrow *e* in Fig. 4) and switches the state of those neighbors to ‘repelled’ (arrow *g* in Fig. 3). When all the ‘repelled’ modules have left, the ‘repelling’ module returns to the ‘stable’ state (arrow *j* in Fig. 4). The second situation is when a deadlock happens, where a moving module is blocked by its neighboring modules. To resolve this deadlock, the blocked module switches its state to ‘repelling’ (arrow *f* in Fig. 4), and attempts to change the state of all its neighbors to be ‘repelled’ (arrow *g* in Fig. 4). This removes some of its neighboring modules to make room for the blocked module to move away. Then the ‘repelling’ module returns to the ‘repelled’ state (arrow *i* in Fig. 4). To prevent the deadlock situations, the repellent states always have a higher priority than the attracting states.

The state transitions of each module are controlled by a GRN model having two gene-protein pairs: an attracting gene-protein pair (g_A, p_A) and a repelling gene-protein pair (g_p, p_p) .

(2) *Gene-Protein Pair for Attraction:* The attracting gene-protein pair (g_A, p_A) is used to control the transitions between ‘attracting’, ‘stable’ and ‘unstable’ states in Fig. 4 based on the morphogen levels of the target pattern generated by layer 1. Initially, all modules are set as ‘unstable’. After they are initialized with the target pattern and the relative position information to the origin, modules that are located in the grids with a positive morphogen level become ‘stable’. A ‘stable’ module initializes the gene expression level of its attracting gene g_A to zero.

Basically the expression level of g_A affects the state of the module as follows:

$$\text{state} = \begin{cases} \text{'unstable'} & \text{if } g_A < G_{A\_L} \\ \text{'stable'} & \text{if } G_{A\_L} < g_A < G_{A\_H} \\ \text{'attracting'} & \text{if } g_A > G_{A\_H} \end{cases} \tag{6}$$

where G_{A_L} is a negative threshold and G_{A_H} is a positive threshold.

Each ‘stable’ module generates attracting protein p_A for all the empty neighboring grids with a positive morphogen level. The local generated p_A and received p_A from other modules will regulate the expression level of g_A . When g_A is higher than G_{A_H} , the state changes to “attracting” , and the attracting protein p_A is diffused to other modules. p_A is defined as

$$p_A^{ij} = k_1 \cdot mp_j \tag{7}$$

where p_A^{ij} is the concentration of attracting protein generated by module *i* for its neighbor grid *j*. k_1 is the discount rate of p_A^{ij} over distance, which is a defined as 0.8 in experiments. mp_j is the morphogen level of grid *j*.

The gene expression level of the attracting gene $g_A(t)$ is defined by:

$$\frac{dg_A^i(t)}{dt} = -k_2 \cdot g_A^i(t) + k_3 \cdot \sum_j p_A^{ij} - k_4 \cdot \sum_k p_A^{rec} \tag{8}$$

where $g_A^i(t)$ is the gene expression level of modular robot *i*. The first term in Eqn. (8) means that $g_A^i(t)$ will decay over time. $\sum_j p_A^{ij}$ represents the sum of all the generated attracting proteins by robot module *i* for its empty neighboring grids. The more proteins the module generates for its empty neighboring grid, the higher the g_A expression level it will have, which means it will have better chance to change its state from "stable" to "attracting". $\sum_k p_A^{rec}$ is the sum of the protein concentrations received from other modules by robot module *i*. The module may turn to 'unstable' if other modules have stronger attractions so that $g_A(t)$ is reduced to a value less than G_{A_L}. Unstable modules choose to fill in the attracting grid with the highest P_A from all the received attracting proteins. k_2, k_3, and k_4 are constant coefficients.

(3) *Gene-Protein Pair for Repelling:* The 'repelling' and 'repelled' states are controlled by the repelling gene-protein pair (g_P, p_P). A 'repelling' module generates p_P and diffuses it to the neighboring modules that need to be repelled. The repellent protein p_P is defined as a positive constant 0.7. Each module has a gene g_P whose expression level decides the state transition to "repelled" state. The gene expression level of g_P is initialized as 0 and can be regulated by p_P through Eqn. (9)

$$\frac{dg_P^i(t)}{dt} = -k_5 \cdot g_P^i(t) - k_6 \cdot \sum p_p^{rec} \tag{9}$$

$$\text{state of grid } i = \text{repelled when} \quad g_P^i < -mp_i \tag{10}$$

where $g_P^i(t)$ is the gene expression level of the repellent gene at time *t*. p_P^{rec} is the concentration of the received repellent protein. mp_i is the morphogen level of the current grid. k_5 and k_6 are constant coefficients. The first item in Eqn. (9) denotes that $g_p(t)$ will decay over time. $\sum p_P^{rec}$ indicates the sum of all p_P received by module *i* from its immediate neighbors. g_P is reduced with more received p_P. Eqn. (1) indicates that modules with a lower morphogen level are more likely to be repelled.

F. Integration of the Hierarchical Framework

Layer 1 model takes task requirements and environmental constraint as the system inputs. By using the mechanochemical model, the target pattern is generated with the assembly of grids with positive morphogen levels, which is the input of layer 2 model. The GRN model coordinates robot modules to fill in the target pattern in a distributed manner. This procedure is interleaved because the reconfiguration of robot modules controlled by layer 2 model may change available grids for v-cells and bring new environmental constraints to the modules. As a result, the target pattern is expanded gradually which will eventually meet task requirements and adapt to the environment.

IV. Experimental Results

To evaluate the efficiency and robustness of the morphogenetic framework for self-reconfiguration of the modular robot, CrossCube, several case studies have been conducted in a robot simulator. This simulator is used to simulate the behaviors and interaction of CrossCube with a physical world using C++ and the PhysX engine from nVidia (http://en.wikipedia.org/wiki/PhysX). In the following experiments, the system parameters of the hierarchical model are set up as follows. The parameters of the GRN model in layer 2 are: $k_2 = 0.7, k_3 = 1, k_4 = 1$, $k_5 = 0.5, k_6 = 2$, G_{A_L} = -1, G_{A_H} = 1, and $G_{P\_L} = -2$. The parameters of the model in layer 1 are: *r* = 0.005, *N* = 200, *d* = 1, *a* = 0.05, *b* = 20 and *e* = 1. For case 4, the CMA-ES method is employed to evolve the pattern parameters in the mechanochemical model to achieve optimal pattern formation.

A. Case Study 1: Vehicle-like Pattern Formation

In case study 1, we focus on vehicle-like pattern formation. Three different experiments are carried out. The detailed rules of $f_{ji}(n_j)$ for vehicle-like pattern has been described in Section III.D. The parameters in $f_{ji}(n_j)$ are defined as W = 4 and ECM_{up} =130. The flow control vector of v-cells is defined as

$$\left[k_i^{up}=1, k_i^{down}=1, k_i^{left}=1, k_i^{right}=1, k_i^{forward}=1, k_i^{backward}=1\right].$$

In the first experiment, 16 robot modules are initially deployed in a 4x4 square. The space available for robot modules for self-reconfiguration is a 4x5 space as shown in Fig. 5(a). Fig. 5(a) is the initial configuration. Figs. 5(b)-(f) demonstrate the procedure of the self-reconfiguration of a modular robot using the hierarchical morphogenetic model. First, the v-cell based mechanochemical model of layer 1 is applied to create the target pattern based on the environmental constraints and target pattern requirements. Then, the GRN-based layer 2 model is employed to self-reconfigure the modules to physically realize the target pattern defined by chemical concentrations.

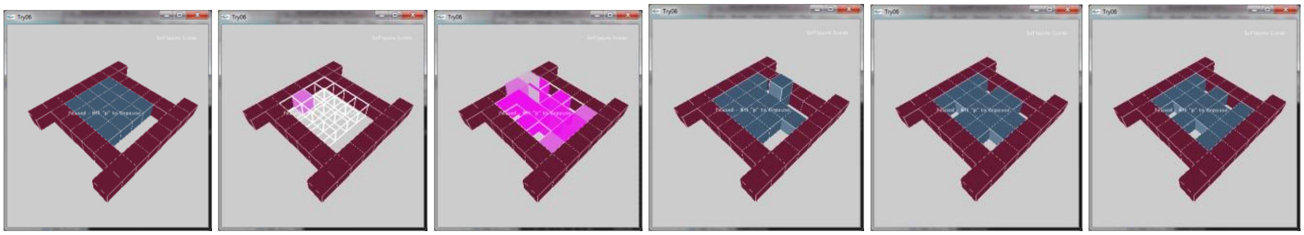

(a) (b) (c) (d) (e) (f)

Fig. 5. Snapshots of the self-reconfiguration process using the proposed hierarchical morphogenetic model in a 4x5 space. (a) The initial configuration of a modular robot in a 4x5 space. (b) 100 v-cells start to proliferate from a randomly-picked starting grid. (c) The final state of the distribution of the v-cells, where the candidate grids of the target pattern have been selected and the morphogen level of each grid is defined. (d)(e) The procedure of constructing the target pattern using the GRN-based layer 2 model. (f) The final target pattern has been constructed by the modular robot.

Please note that the blue grids in Fig. 5 represent the real robot modules in the stable state. The pink grids represent the density distribution of v-cells in the robot modules, where the darker the color, the higher the density of the v-cells in the modules is. The grids in magenta color represent the obstacles in the environment. These representations apply to all of the following simulation results, unless otherwise stated.

In the second experiment, the same number of robot modules is initialized in the same way as in the first experiment but in a smaller 4x4 space. The parameters of $f_{ji}(n_j)$ are the same as in the first experiment. Based on the new environmental constraints (i.e., a 4x4 square space) with the same pattern constraints (i.e., a vehicle pattern with wheels on both sides), a two-floor vehicle pattern is constructed following the same procedure of the previous experiment, as shown in Fig. 6. The requirements of vehicle pattern are still well-met.

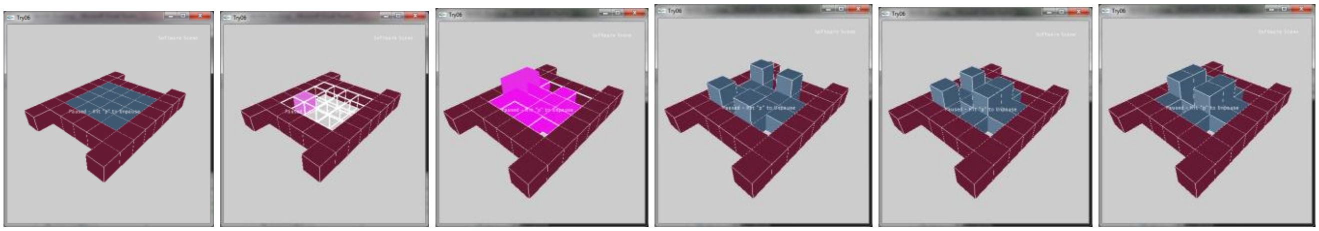
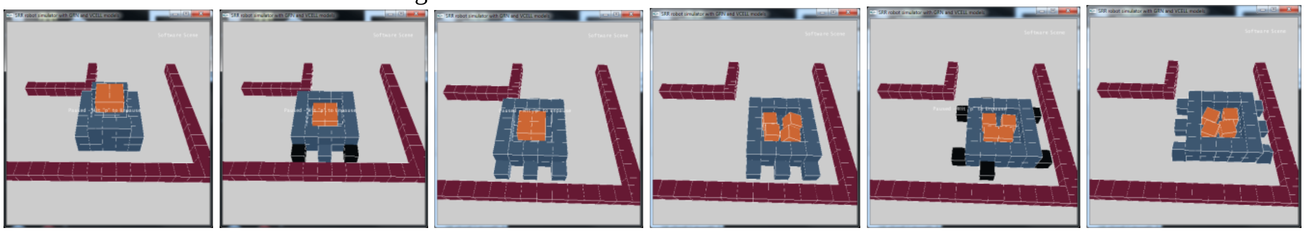

(a) (b) (c) (d) (e) (f)

Fig. 6. Snapshots of the self-reconfiguration process using the proposed hierarchical morphogenetic model in a 4x4 space. The explanations of Fig. 6(a) to Fig. 6(f) are similar to Fig. 5.

In the third experiment, a vehicle-like pattern carrying cargo is generated using the proposed hierarchical model, and the vehicle pattern can also reconfigure the wheels to avoid obstacles in the environment by enabling them to make turns. Snapshots of this experiment are shown in Fig. 7. Since the robot needs to make a sharp 90 degree turn, the robot reconfigures the positions of wheels to make turns to avoid colliding with obstacles.



(a) (b) (c) (d) (e) (f)

Fig. 7.Snapshots of the reconfiguration of a vehicle robot carrying cargo, and r the positions of wheels to make a sharp 90 degree turn in order to avoid obstacles in the environment.

B. Case Study 2: Self-Repairing

One important feature of reconfigurable modular robots is their ability to self-repair malfunctioned or damaged modules. For example, if some of the modules are damaged, the remaining modules will disconnect from those damaged modules and release new attracting proteins to attract existing modules on the grids with lower morphogen levels to fill in the grids of the damaged modules.

To evaluate the self-repairing performance of the GRN-based control in layer 2, another experiment is conducted here. As shown in Fig. 8 below, it is assumed that a vehicle-like pattern has been generated by layer 1, where the bottom-floor modules are functional modules and the top-floor modules are backups in the vehicle pattern. The backup modules have lower morphogen levels than those functional modules. When the vehicle is moving, an “explosion” occurs and some functional modules are blown away. The backup modules then move to fill in the damaged modules. Fig. 8 shows a snapshot of this self-repairing procedure using the GRN-based layer 2 model. This experiment demonstrates that the proposed approach is able to self-repair a modular robot in case of failed modules.

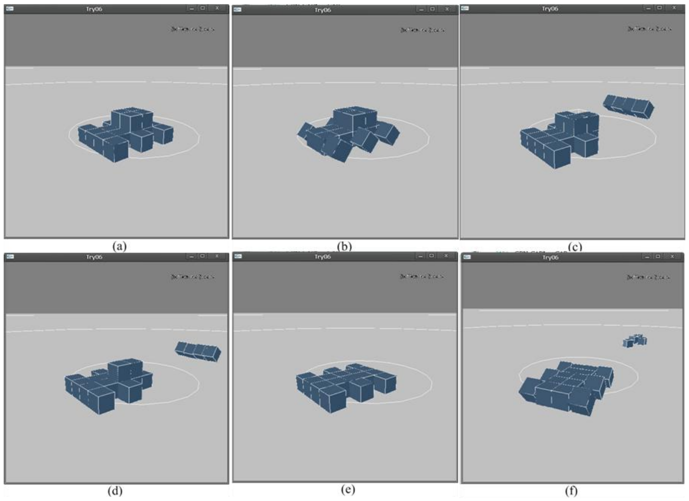


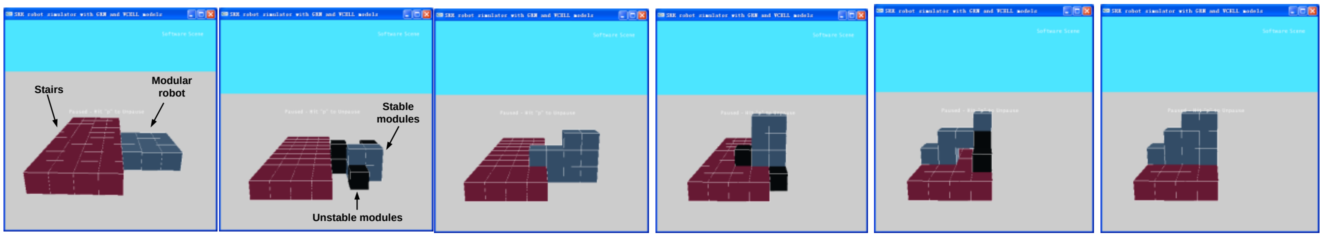
Fig. 8. Snapshots of the self-repairing of CrossCube using the GRN-based layer 2 model. (a) A vehicle pattern is formed. (b) The vehicle pattern moves forward. (c) Some modules are blown off when the explosion happens. (d) The failed part is filled up by the backup modules. (e) The vehicle is repaired. (f) The repaired vehicle continues moving.

C. Case Study 3: Pattern Adaptation in a Changing Environment – Climbing Stairs

This case study is to verify the efficiency and robustness of the hierarchical model for a modular robot to adapt to environmental changes. More specifically, we want to show that a modular robot using the proposed hierarchical model is able to climb stairs by changing its pattern autonomously. The starting grid of the robot is initialized with 4000 v-cells and the robot is distributed in a 3x3 square.

First, in order to trigger the robot to move forward, a forward “force” needs to be applied on v-cells. To update the density of v-cells in each grid, this forward “force” can be obtained from $\sum_k n_k^{rec}$ in Eqn. (1). Second, we need to define $f_{ji}(n_j)$ for climbing patterns. Since we want the pattern to be able to move forward and climb up the stairs, the major environmental constraint is that the v-cells should not diffuse in right, left, or backward directions. Diffusion of v-cells in the right or left direction will increase the width of the pattern during the traveling, which may eventually make the climbing impossible. Therefore, if the grid *j* is at the leftmost or rightmost position of the pattern, $f_{ji}(n_j) = 100 * n_j$, where *i* can be the left or right grid of *j*, the width should be kept the same during the climbing.

Third, since the robot has to climb up on the stairs, the v-cells moving up should be encouraged and moving down should be restricted. Therefore the dispersal control vector is defined as $k_i^{up} = 1, k_i^{down} = 0$, $k_i^{left} = 0.5$, $k_i^{right} = 0.5$, $k_i^{forward} = 0.7$, and $k_i^{backward} = 0.7$.



(a) (b) (c) (d) (e) (f)

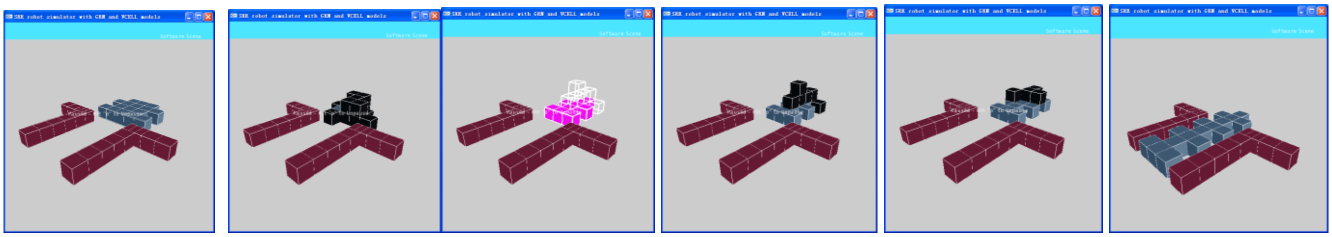
Fig. 9. Snapshots demonstrating the reconfigurable processes in the stair-climbing case using the proposed hierarchical model.

A set of illustrations showing the adaptation of the vehicle pattern to environmental changes is shown in Fig. 9. The black cubes are ‘unstable’ modules. When the stairs in the environment are detected in front of the robot, new target patterns are automatically generated to allow the robots to climb the stairs. Since this is a distributed system, where each module makes its own movement decisions based only on its local information, the dynamics of layer 1 and layer 2 is updated asynchronously. More specifically, the v-cells first flow to an intermediate grid (not yet the target grid) using the layer 1 model, then the modules will move to these positions controlled by the GRN-based layer 2 model. These two processes interleave until the target pattern is realized or the robot successfully climbs over up stairs.

D. Case Study 4: Pattern Adaptation in a Changing Environment – Traversing a Narrow Passage

To demonstrate that the modular robot using the proposed hierarchical morphogenetic model can autonomously adapt to changing environments, another experiment was conducted. To optimize the pattern design of the modular robot for a locomotion task, the CMA-ES is applied to evolve the pattern parameters of $f_{ji}(n_j)$, which include the width of the robot (*W)* and ECM_{up}.

Here, for a locomotion pattern, the fitness function of the CMA-ES method is defined as the travel distance within a certain time period. The longer the robot can travel within a certain time period, the better the current pattern is for the locomotion task. We set $\lambda = 20, \mu = 5$ and $\sigma = 50$. The parameters of the best evolved vehicle-like pattern are: $W = 3$ and ECM_{up} =201. Fig. 10 shows snapshots of the self-adaption procedure of the robot in a changing environment, where the robot changes its patternafter a narrower passage is detected.



(a) (b) (c) (d) (e) (f)

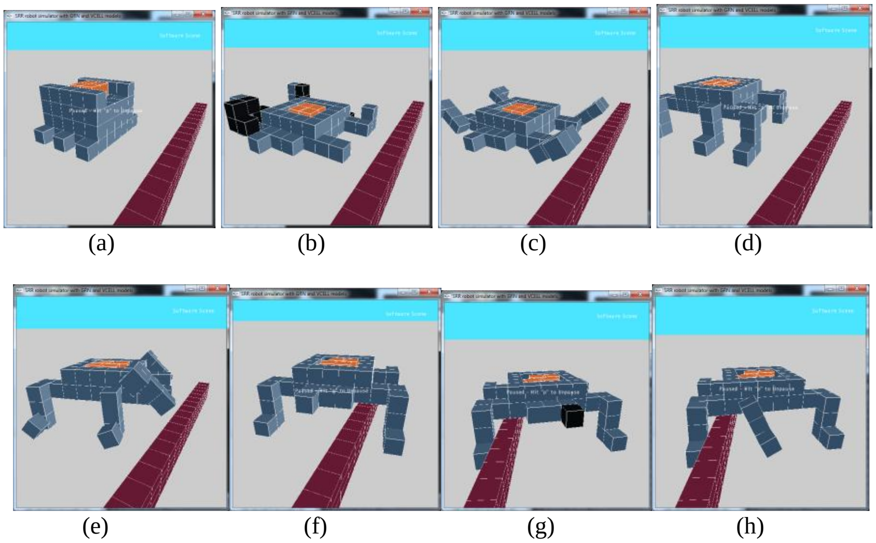
Fig. 10. Snapshots demonstrating a series of reconfigurable processes during locomotion in a changing environment, where the robot adapted its width to a narrow path.

E. Case Study 5: More Complex Patterns

It may be argued that snake-like robots can implement case studies 2, 3 and 4 with simple mechanical design and simple controllers. If so why do we need to develop such a complex model? As mentioned before, the objective of the proposed approach (both mechanical design and controllers) is to design a more general distributed-controlled modular robot which can reconfigure itself into any arbitrary patterns according to the current environment, such as a vehicle-like robot with wheels, walking robots with legs, climbing robots, or snake-like robots, etc. To this end, it is impossible for the chain-based architecture with simple sequential controllers to reconfigure more complex patterns besides snake-like robots.

To show a more general case study that the proposed model can conduct while other simpler robots cannot, another experiment has been conducted. As shown in Fig. 11, a vehicle-like pattern carrying cargo is moving forward. Then an obstacle is detected. The robot has to cross over the obstacle as it cannot be avoided. Therefore, the robot needs to reconfigure to a walking robot with six legs while still carrying the cargo on the top. The robot can detect the height of the obstacle and establish the length of the legs. After the walking robot is built, the robot crosses over the obstacle.

Another argument may arise. If there are any other simple models which can realize the same or similar functions of the proposed model? The simple ways of realizing the same or similar functions of the proposed model have to be some rule-based models. However, in the rule-based models, we have to predefine the rules to predict all the possible situations in the environment, which is unpractical when the environment changes. On the other side, the proposed model has a theoretical foundation from biological systems, which shows us how biological systems can self-organize in a distributed manner, while it is difficult to realize the similar functions in the rule-based systems.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 11. Snapshots demonstrating a series of reconfiguration from a vehicle pattern to a 6-legged walking robot pattern, and the walking robots cross over an obstacle. Please note that the feet of the middle legs are reconfigured in the last row.

V. Conclusion and Future Work

In this paper, we presented a two-layer hierarchical morphogenetic approach to self-reconfiguration of modular robots, which is inspired by multi-cellular morphogenesis. Layer 1 of this morphogenetic framework is a virtual-cell based mechanochemical model that can generate appropriate target patterns for modular robots to adapt to environmental changes. Layer 2 is a gene regulatory network (GRN) based controller that can automatically self-reconfigure the modular robots to achieve the desired configuration generated by layer 1. In addition, the CMA-ES is employed to evolve the parameters of the mechanochemical model so that a modular robot can generate new patterns to adapt to environmental changes. Such a hierarchical structure makes it possible to separate the control mechanisms for defining a target configuration from those for realizing it, similar to biological morphogenesis. In response to the environmental changes, the layer for defining the target pattern of the modular robot is able to adapt the target configuration, after which the second layer can re-organize the modules autonomously to realize the target configuration.

In the future, we will investigate the following three issues. First, we need to consider the motion control of the generated patterns in a systematic way since different patterns require different motion control capabilities. For example, the motion control of a walking robot is much more complex than a vehicle-like robot. Second, we will develop a physical modular robot of CrossCube and implement the proposed hierarchical morphogenetic framework on real modular robots with the considerations of motion control issues. Last, more complex environmental changes will be implemented to evaluate the robustness of the proposed approach.

References

[1] Shen, W., Krivokon, M., Chiu, H., Everist, J., Rubenstein, M., and Venkatesh, J., 2006. Multimode locomotion for reconfigurable robots, Autonomous Robots, vol. 20, no. 2, pp. 165-177.

[2] Yim, M., Eldershaw, C., Zhang, Y., and Duff, D. G., 2004. Limbless conforming gaits with modular robots, in Proc. of International Symposium on Experimental Robotics.

[3] Yu, C.H., Haller, K., Ingber, D., and Nagpal, R., 2009. Morpho: A self-deformable modular robot inspired by cellular structure, in Proc. of International Conference on Robotics and Automation.

[4] Yu, C., and Nagpal, R., 2009. Self-Adapting Modular Robotics: A Generalized Distributed Consensus Framework, in Proc. of International Conference on Robotics and Automation.

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

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

[7] Kamimura, A., Kurokawa, H., Yoshida, E., Murata, S., Tomita, K., and Kokaji, S. 2004. Distributed adaptive locomotion by a modular robotic system, M-TRAN II, in Proc. of IEEE International Conference on Intelligent Robots and Systems.

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

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

[10] Sayama, H., Robust Morphogenesis of Robotic Swarms, IEEE Computational Intelligence Magazine, 5(3), pp.43-49.

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

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

[13] Zykov, V., Chan, A., and Lipson, H., 2007. Molecubes: An Open-Source Modular Robotics Kit, in Proc. of International Conference on Robotics and Automation.

[14] Kelly, K., 1994. Out of Control – The New Biology of machines, Social Systems and Economic World. Basic Books.

[15] Nolfi, S., and Floreano, D., 2004, Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines, MIT Press,

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

[17] Schmickl, T., Hamann, H., Stradner, J., and Crailsheim, K., 2010, Hormone-based control for multi-modular robotics. In Symbiotic Multi-Robot Organisms: Reliability, Adaptability, Evolution, P. Levi and S. Kernbach, Eds. Springer.

[18] Stoy, K, 2006, Using cellular automata and gradients to control self-reconfiguration, Robotics and Autonomous Systems, vol. 54, pp.135-141.

[19] Pfeiffer, R., and Bongard, J. C., How the Body Shapes the Way We Think. MIT Press, 2006.

[20] Spröwitz, A., Pouya, S., Bonardi, S., Kieboom, J., Möckel, R., Billard, A., Dillenbourg, P., and Ijspeert, A., Roombots: Reconfigurable Robots for Adaptive Furniture, IEEE Computational Intelligence Magazine, 5(3), pp.20-32.

[21] Wolpert. L, 2002. Principles of Development. Oxford University Press.

[22] Murray, J. D., 1988, Modelling biological pattern formation in embryology, ISI Atlas of Science: Animal and Plant Sciences 1: 270-274.

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

[24] Hansen, N, and Ostermeier, A., Completely derandomized self-adaptation in evolution strategies. Evolutionary Computation, 9(2) pp.159-195. 2001.

[25] Hansen, N., Müller, S. D., and Koumoutsakos, P., Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evolutionary Computation, 11(1) pp.1-18. 2003.

[26] Jin, Y., and Meng, Y., Morphogenetic Robotics: An Emerging New Field in Developmental Robotics, IEEE Trans. on Systems, Man, and Cybernetics, Part C. (Accepted)

[27] Guo, H., Meng, Y., and Jin, Y., 2009, A Cellular Mechanism for Multi-Robot Construction via Evolutionary Multi-Objective Optimization of a Gene Regulatory Networks, BioSystems, 98(3), pp.193-203.

[28] Guo, H., Meng, Y., and Jin, Y., 2010, Analysis of Local Communication Load in Shape Formation of a Distributed Morphogenetic Swarm Robotic System, IEEE Congress on Evolutionary Computation.

[29] Dürr, P., Mattiussi, C., and Floreano, D., 2010, Genetic Representation and Evolvability of Modular Neural Controller, IEEE Computational Intelligence Magazine, 5(3), pp. 10-19.

[30] Massera, G., Tuci, E., Ferrauto, T., and Nolfi, S. 2010, The Facilitatory Role of Linguistic Instructions on Developing Manipulation Skills, IEEE Computational Intelligence Magazine, 5(3), pp.33-42.

[31] Everist, J., Hou, F., and Shen, W., 2006. Transformation of Control in Congruent Self-Reconfigurable Robot Topologies, in Proc. of International Conference on Intelligent Robots and Systems.

[32] Salemi, B., Moll, M., and Shen, W., 2006. SUPERBOT: A Deployable, Multi-Functional, and Modular Self-Reconfigurable Robotic System, in Proc. of IEEE International Conference on Intelligent Robots and Systems.

[33] Moll, M., Will, P., Krivokon, M., and Shen, W., 2006. Distributed Control of the Center of Mass of a Modular Robot, in Proc. of IEEE International Conference on Intelligent Robots and Systems.

[34] Kurokawa, H., Tomita, K., Kamimura, A., Yoshida, E., Kokaji, S., and Murata, S., 2005. Distributed Self-reconfiguration Control of Modular Robot M-TRAN, in Proc. of International Conference on Mechatronics and Automation.

[35] Yoshida, E., Kurokawa, H., Kamimura, A., Tomita, K., Kokaji, A., and Murata, S., 2004. Planning Behaviors of a Modular Robot: an Approach Applying a Randomized Planner to Coherent Structure, in Proc. IEEE International Conference on Intelligent Robots and Systems.

[36] Alon, U., 2007, Network motifs: theory and experimental approaches, Nature Review Genetics, vol. 8, pp. 450–461.

[37] DeJong, H., 2002, Modeling and simulation of genetic regulatory systems: A literature review, Journal of Computational Biology, vol. 9, pp. 67–103.

[38] Endy D., and Brent, R, 2001, Modeling cellular behavior, Nature, vol. 409,pp. 391–395.

[39] Murray, J. D., and Maini, P. K., 1988, Mechanochemical models for generating biological pattern and form in development, Physics Reports, 171, no. 2, pp.59-84.

[40] Jin, Y., Meng, Y., and Guo, H., 2010, A Morphogenetic Self-Organization Algorithm for Swarm Robotic Systems using Relative Position Information, 10th Annual UK Workshop on Computational Intelligence (UKCI 2010).