

Chapter 3

Morphogenetic Robotics: A New Paradigm for Designing Self-Organizing, Self-Reconfigurable and Self-Adaptive Robots

Yaochu Jin and Yan Meng

Abstract By *morphogenetic robotics*, we mean a class of methodologies for designing self-organizing, self-reconfigurable and self-adaptive robots inspired by biological morphogenesis. We categorize these methodologies into three areas, namely, morphogenetic swarm robotic systems, morphogenetic modular robots and morphogenetic co-design of body and brain for robots. We also discuss the relationship between morphogenetic robotics and a few closely related areas in robotics, such as *epigenetic robotics*, which focuses on cognitive development in robotic systems, and *evolutionary robotics*, which is concerned with evolutionary design of robot controllers. A few examples are provided to illustrate the main ideas underlying the morphogenetic approaches to robotics.

3.1 What is Morphogenetic Robotics?

In biology, morphogenesis is a general term denoting the biological process that causes the creation of the form of multi-cellular organisms [60], including the body plan and the nervous system. In recent years, increasing evidence has been found in evolutionary developmental biology (evo-devo) that morphogenesis can be regarded as a self-organizing and self-assembling process through cellular and molecular interactions under genetic and environmental control [4, 52]. In addition, biological morphogenesis also shows a surprising degree of robustness [5].

Due to the attractive properties that biological morphogenesis exhibits, much attention has been paid to the use of genetic and cellular mechanisms for designing robotic

Y. Jin (✉)

Department of Computing, University of Surrey, Guildford, Surrey GU2 7XH, UK
e-mail: yaochu.jin@surrey.ac.uk

Y. Meng

Department of Electrical and Computer Engineering, Stevens Institute of Technology,
Hoboken, NJ 07030, USA
e-mail: yan.meng@stevens.edu

systems, in particular self-organizing swarm robotic systems and self-reconfigurable modular robots. Meanwhile, a large body of research has been conducted in artificial life and robotics to simultaneously design the body plan and neural controller of robots or animats using an evolutionary developmental approach [53, 55, 57].

With the recent and rapid advances in evolutionary developmental biology and systems biology, we expect that increasing research interest will focus on solving various problems in robotics inspired from biological morphogenesis. To promote research activities along this line, we advocate the term of *morphogenetic robotics* as a new research field in robotics that concentrates on the application of biological morphogenetic mechanisms to the design of self-organizing, self-configurable and self-adaptive robots.

From our perspective, morphogenetic robotics may include, but is not limited to the following three main topics:

- Morphogenetic swarm robotic systems that deal with the self-organization of swarm robots using genetic and cellular mechanisms underlying early biological morphogenesis [17, 31, 50].
- Morphogenetic modular robots, where modular robots adapt their configuration autonomously using morphogenetic principles [63].
- Evolutionary developmental approaches to the design of the body plan of robots, including sensors and actuators, and/or the controller of robots [18, 28].

Morphogenetic robotics is related to, but differs from, epigenetic robotics, often known as developmental robotics [30, 59], or developmental cognitive robotics [26]. The main difference between morphogenetic robotics and epigenetic robotics is that the former focuses on self-organization, self-reconfiguration, self-assembly and self-adaptive control of robots using genetic and cellular mechanisms inspired from biological morphogenesis, whereas the latter emphasizes the development of robots' cognitive capabilities, such as language, emotion and social skills, through experience during their lifetime. Morphogenetic robotics is closely connected to developmental biology and systems biology, while epigenetic robotics is related to cognitive science, developmental psychology and neuroscience. As a whole, morphogenetic robotics and epigenetic robotics together lay the foundation of developmental robotics in a more general sense. Meanwhile, morphogenetic robotics can be seen as a sub-field of morphogenetic engineering [12].

Other closely related notions are evolutionary robotics [46] and morphological computation (also known as morphological robotics) [41]. Traditionally, evolutionary robotics is concerned with the design of robot controllers using evolutionary algorithms. Complementary to evolutionary robotics, where the role of the robot's morphology in realizing an intelligent behavior is overlooked to a large degree, morphological computation was employed for connecting brain, body and environment in robot design. Unfortunately, it turned out that morphological computation has put too much emphasis on the direct design of the morphology of robots in shaping intelligent robotic behaviors and has not paid sufficient attention to the developmental aspects.

A brief introduction to biological morphogenesis and its computational models are presented in Sect. 3.2. A metaphor linking swarm robotic systems to multi-cellular

systems is described in Sect. 3.3, where a gene regulatory model is used for self-organizing multiple robots to form complex shapes. Related issues such as how to represent and form complex shapes without a global coordinate system are also discussed. Section 3.4 presents our idea of using a gene regulatory network model to configure modular robots, followed by Sect. 3.5, where the developmental approach to evolutionary co-design of the body and controller of robots or robot parts (e.g., robot arm and hand for object grasping) is discussed along with a short overview of research work on brain-body co-evolution originated from both artificial life and evolutionary robotics. Concluding remarks of this chapter are provided in Sect. 3.6.

3.2 Multi-Cellular Morphogenesis and its Computational Modeling

3.2.1 *Biological Morphogenesis and Metamorphosis*

Animal morphogenesis can be divided into early embryonic development and later embryonic development [16]. Early embryonic development typically involves cleavage, gastrulation, and axis formation, while later embryonic development is mainly responsible for the development of the nervous system, starting with the segregation of neural and glial cells from the ectoderm germ layer [47].

Metamorphosis is another interesting stage of biological development during which both the shape and size of the organisms changes [7]. There are two types of metamorphosis, namely, incomplete and complete metamorphosis. For organisms underlying incomplete metamorphosis, there are three developmental stages, in which nymphs look similar to adults. By contrast, organisms that undergo complete metamorphosis have four developmental stages, in which the shape of the organisms changes drastically.

Both multi-cellular morphogenesis and metamorphosis are under the control of gene regulatory networks (GRN). When a gene is expressed, information stored in the genome is transcribed into mRNA and then translated into proteins. Some of these proteins are transcription factors that can regulate the expression of their own or other genes, thus resulting in a complex network of interacting genes, the GRN. In the following, we discuss computational modeling of GRNs and show how these models can be used both for understanding biology and solving engineering problems.

3.2.2 *Computational Modeling of Developmental Gene Networks*

In order to understand the emergent morphology resulting from the interactions of genes in a GRN, reconstruction of gene regulatory pathways using a computational

model has become popular in systems biology [9]. A large number of computational models for GRNs have been suggested [11, 14], which can be divided into discrete models, such as random Boolean networks and Markovian models, and continuous models, such as ordinary differential equations and partial differential equations. Sometimes, one can also distinguish among GRNs deterministic models and stochastic models according to their ability to describe stochasticity in gene expression.

Generally speaking, the regulatory dynamics in a unicellular cell can be described by a set of ordinary differential equations. The mathematical model of gene expression with autoregulation can be typically described by:

$$\frac{d[R]}{dt} = -\gamma_R [R] + \alpha_R H([P]), \quad (3.1)$$

$$\frac{d[P]}{dt} = -\gamma_P [P] + \alpha_P [R], \quad (3.2)$$

where $[R]$ and $[P]$ are the concentrations of mRNA and protein, respectively, γ_R and γ_P are the decay rates of mRNA and protein, α_R and α_P are the synthesis rates of mRNA and protein, $H(X)$ is the Hill function. If the autoregulation is a repression, also known as negative autoregulation, the Hill function can be described by

$$H_r(x) = \frac{\theta^n + \beta}{\theta^n + x^n}, \quad (3.3)$$

and if the autoregulation is an activation, the Hill function can be written as:

$$H_a(x) = \frac{\beta x^n}{\theta^n + x^n}, \quad (3.4)$$

where β is an activation coefficient, θ is a threshold, and n is the ‘‘Hill coefficient’’.

To describe the morphogenesis of multi-cellular organisms, interaction among the cells in their gene expression dynamics must also be taken into account. Mjolsness et al. [35] have suggested a generalized GRN that considers diffusion of transcription factors among the cells:

$$\frac{\partial g_{ij}}{\partial t} = -\gamma_j g_{ij} + \phi \left[\sum_{l=1}^{n_g} W^{jl} g_{il} + \theta_j \right] + D_j \nabla^2 g_{ij}, \quad (3.5)$$

where g_{ij} denotes the concentration of j -th gene product (protein) in the i -th cell. The first term on the right-hand side of Eq.(3.5) represents the degradation of the protein at a rate of γ_j , the second term specifies the production of protein g_{ij} , and the last term describes protein diffusion at a rate of D_j . ϕ is an activation function for the protein production, which is usually implemented by a sigmoid function $\phi(z) = 1/(1 + \exp(-\mu z))$. Interactions between genes are described by an interaction matrix W^{jl} , whose elements can be either active (positive value) or repressive (negative value). θ_j is a threshold for the activation of gene expression. n_g is the number of

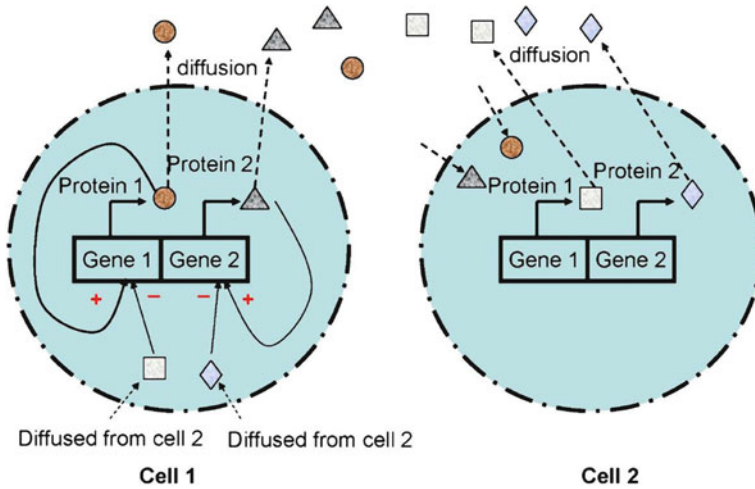


Fig. 3.1 Illustration of cell signaling in a multi-cellular system

proteins. An illustration of cell-cell interactions is provided in Fig. 3.1, where gene 1 of cell 1 is activated by its own protein and repressed by the protein produced by gene 1 of cell 2 through diffusion. Similarly, gene 2 of cell 1 is activated by its own protein, and repressed by the protein of gene 2 of cell 2 through diffusion.

3.2.3 Applications of Computational Models of GRN

In addition to the reconstruction of gene regulatory pathways based on biological data [51], computational models have been widely used for analyzing the dynamics of GRNs, particularly regarding the robustness of GRN motifs, the *in silico* synthesis of typical regulatory dynamics such as bistability and sustained oscillation, and the design of engineered systems [12], including morphogenetic robotics that we discuss in greater detail in the following sections.

3.2.3.1 Analysis of GRN Motifs

It is believed that GRNs can be analyzed by examining the structure and function of a number of wiring patterns, known as *network motifs*, such as auto-regulation, feedforward loops and feedback loops [1]. Recently, the role of feedback loops, in particular their coupling and its relationship to the robustness of resulting dynamics in the network motifs, has received increasing attention [10, 27, 58].

The *cis*-regulation logic also plays an important role in the dynamics and functionality of GRNs. A systematic investigation of control logic in gene regulation has

been reported in Schilstra and Nehaniv [48], who suggested that networks consisting of competitively binding activators and repressors can be controlled more robustly.

3.2.3.2 *In Silico* Synthesis

In Silico synthesis of typical regulatory dynamics can offer insight into the way nature has shaped the evolution of regulatory motifs [15, 40]. In Knabe et al. [25], a GRN was used for evolving biological clocks in the presence of periodic environmental stimuli, where both the number and activation type (activating or repressive) of regulatory units of each gene were subject to evolution. They reported that the evolved clock tended to be robust to perturbations that evolution has experienced. Jin and Sendhoff [22] investigated the influence of the genetic encoding scheme as well as the activation function used in the gene regulatory model of a relaxation oscillation circuit. Their results suggested that evolving sustained oscillation using a step function as the activation function was much easier than using a Hill function. Most recently, it has been found that robust motifs can emerge from *in silico* evolution without an explicit selection pressure [20].

3.2.3.3 Artificial Embryogeny

A large body of research has been reported on simulating biological development in computational environments [55]. Motivations for building models of artificial embryogeny include understanding biological development in artificial life, designing complex structures [56], and amorphous computing [6], to name a few. Obviously, all three areas of morphogenetic robotics involve computational models of biological morphogenesis, which will be elaborated upon in the following sections.

3.3 Morphogenetic Swarm Robotic Systems

3.3.1 *Swarm Robotic Systems*

A swarm robotic system is a multi-robot system consisting of a large number of homogeneous simple robots. Swarm robots are often used to fulfill tasks that are difficult or even impossible for a single robot, especially in the presence of uncertainties, or with incomplete information, or where a distributed control or asynchronous computation is required. Compared with centralized systems, swarm robotic systems with a distributed control are believed to be flexible, robust, and adaptive for tasks that are inherently distributed in space and/or time. Typical applications of swarm robotic systems include group transport, foraging, shape formation, and boundary coverage. However, designing a decentralized control algorithm for swarm robotic systems has been a challenging task [32].

3.3.2 A Metaphor Linking Swarm Robotic Systems to Multi-Cellular Systems

3.3.2.1 Cell-Robot Mapping

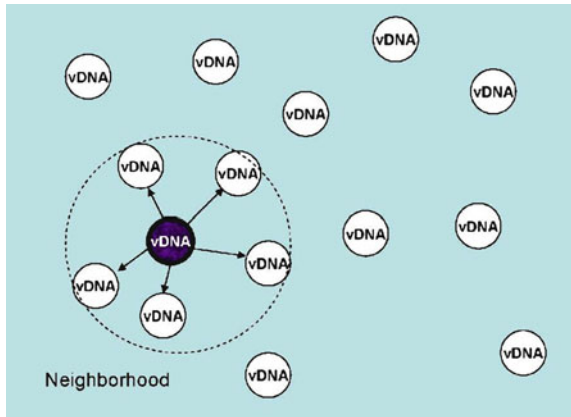
The basic idea in applying the genetic and cellular mechanisms of biological morphogenesis to the self-organized control of swarm robots is to establish a metaphorical link between a cell and a robot. In other words, it is assumed that the motion dynamics of each robot can be modeled by the regulatory dynamics of a cell. In [17, 19, 33], the location and velocity of the robots are described by the protein concentration of a few genes whose expression is influenced by each other. Typically, for a robot in a three-dimensional space, three proteins are used for denoting the robot's position, and three for the velocity. Note however that the mathematical definition of the protein concentrations standing for position and velocity of the robots do not satisfy the exact physical relationship between position and velocity. Figure 3.2 shows multiple robots in a field, each represented by a cell, where the robots are represented by cells containing a virtual DNA in a field. The dashed circle indicates the neighborhood of the shaded cell (robot) into which its proteins can diffuse and can influence the gene expression of these cells.

Keeping the metaphor between the cells and the robots in mind, the motion dynamics of each robot can be described by a GRN model, where the concentrations of two proteins of type G represent the x and y position of a robot, respectively, and that of the proteins of type P represent the analog of the velocity.

$$\frac{dg_{i,x}}{dt} = -az_{i,x} + mp_{i,x} \quad \frac{dg_{i,y}}{dt} = -az_{i,y} + mp_{i,y}, \quad (3.6)$$

$$\frac{dp_{i,x}}{dt} = -cp_{i,x} + kf(z_{i,x}) + bD_{i,x} \quad \frac{dp_{i,y}}{dt} = -cp_{i,y} + kf(z_{i,y}) + bD_{i,y}, \quad (3.7)$$

Fig. 3.2 A swarm robotic system represented by a multi-cellular system



where $i = 1, 2, \dots, n$ and n is the total number of robots (cells) in the system. $g_{i,x}$ and $g_{i,y}$ are the x and y position of the i -th robot, respectively, which correspond to the concentrations of two proteins of type G. $z_{i,x}$ and $z_{i,y}$ are spatial gradients (to be defined later). $p_{i,x}$ and $p_{i,y}$ are the concentrations of two proteins of type P, which denote the velocity-like property of the i -th robot along the x and y coordinates, respectively. $D_{i,x}$ and $D_{i,y}$ are the sum of the distances between the i -th robot and its neighbors. In the language of the multi-cellular system, it is the sum of the concentrations of protein type G diffused from neighboring cells. Mathematically, we have:

$$D_{i,x} = \sum_{j=1}^{N_i} D_{i,x}^j, \quad D_{i,y} = \sum_{j=1}^{N_i} D_{i,y}^j, \quad (3.8)$$

where N_i denotes the number of neighbors of robot i , and $D_{i,x}^j$ and $D_{i,y}^j$ are the protein concentrations diffused from neighboring robot j received by robot i , which are defined as:

$$D_{i,x}^j = \frac{(g_{i,x} - g_{j,x})}{\sqrt{(g_{i,x} - g_{j,x})^2 + (g_{i,y} - g_{j,y})^2}}, \quad (3.9)$$

$$D_{i,y}^j = \frac{(g_{i,y} - g_{j,y})}{\sqrt{(g_{i,x} - g_{j,x})^2 + (g_{i,y} - g_{j,y})^2}}. \quad (3.10)$$

The diffusion term in the regulatory model simulates cell-cell signaling in multi-cellular systems. For a swarm robotic system, this only entails that each robot should be able to detect the distance to its neighboring robots, which is practical and easy to realize.

3.3.2.2 Morphogen Gradients for Target Shape Representation

In biological morphogenesis, morphogen concentration gradients control cell fate specification and play a key role in pattern formation [3]. In the present gene regulatory model for shape formation of swarm robots, the target shape information is also provided in terms of morphogen gradients, which is defined by $f(z)$ in Eq. (3.7). For two-dimensional target shape, $f(z_i)$ can be defined as follows:

$$f(z_{i,x}) = \frac{1 - e^{-z_{i,x}}}{1 + e^{-z_{i,x}}}, \quad f(z_{i,y}) = \frac{1 - e^{-z_{i,y}}}{1 + e^{-z_{i,y}}} \quad (3.11)$$

where $z_{i,x}$ and $z_{i,y}$ are the gradients along the x -axis and y -axis, respectively, of an analytic function h , which is described as:

$$z_{i,x} = \frac{\partial h}{\partial g_{i,x}}, \quad z_{i,y} = \frac{\partial h}{\partial g_{i,y}} \quad (3.12)$$

where h defines the shape the robots should form. For example, if the robots are required to form a unit circle, h can be defined as

$$h = (g_{i,x}^2 + g_{i,y}^2 - 1)^2. \quad (3.13)$$

The above GRN makes it possible for the swarm robots to form shapes that can be described by an analytical function. There are potentially three problems with this way of shape representation. First, the complexity of the shapes is limited. Second, the system needs a global coordinate system for describing the shapes, which poses a big problem for decentralized systems. Third, the shape can be formed only on a predefined location. To address these issues, parametrized shape representation models, such as are Bézier, B-Spline and non-uniform rational B-Spline (NURBS) can be used.

NURBS [43] is a mathematical model commonly used in computer graphics and design optimization for generating and representing curves and surfaces. NURBS can offer two unique features for multi-robot shape formation. First, it provides a common mathematical formalism for both standard analytical shapes and free-form shapes. Second, it is a parametrized representation that is independent of an absolute coordinate system. Once the parameters in the NURBS curve are fixed, a corresponding point on the NURBS curve can be determined without a global coordinate system. The basis functions used in NURBS curves are defined as $B_{i,k}(u)$, where i corresponds to the i -th control point, and k is the degree of the basis function. A NURBS curve can be defined as the combination of a set of piecewise rational basis functions with $n+1$ control points \mathbf{p}_i and the associated weights w_i as follows:

$$\mathbf{c}(u) = \frac{\sum_{i=1}^n \mathbf{p}_i w_i B_{i,k}(u)}{\sum_{j=1}^n w_j B_{j,k}(u)} \quad (3.14)$$

where n is the number of control points, u is the parametric variable, and $B_{i,k}(u)$ are B-spline basis functions. Assuming that the basis functions are of degree $k-1$, a NURBS curve has $n + k + 1$ knots t_i in non-decreasing sequences: $t_0 \leq t_1 \leq \dots \leq t_{n+k}$. The basis functions are defined recursively as:

$$B_{i,1}(u) = \begin{cases} 1 & \text{if } t_i \leq u \leq t_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

with

$$B_{i,k}(u) = \frac{u - t_i}{t_{i+k} - t_i} B_{i,k-1}(u) + \frac{t_{i+k+1} - u}{t_{i+k+1} - t_{i+1}} B_{i+1,k-1}(u)$$

The parametric domain is $t_{k-1} \leq u \leq t_{k+1}$.

With the NURBS model for shape description, complex shapes can be formed, please refer to Sect. 3.3.3 for examples.

3.3.3 Illustrative Results

Simulation results where 20 robots are used to form a circle are shown in Fig. 3.3. The robots are randomly distributed in the area in the beginning. Driven by the GRN-based dynamics, the robots move toward a circle and distribute on it evenly. Snapshots showing 56 virtual robots forming a NURBS shape are provided in Fig. 3.4. A proof-of-concept experiment has also been performed using real e-puck robots. Figure 3.5 shows a few snapshots of eight e-puck robots converging to a capital letter “R”, see [33] for details.

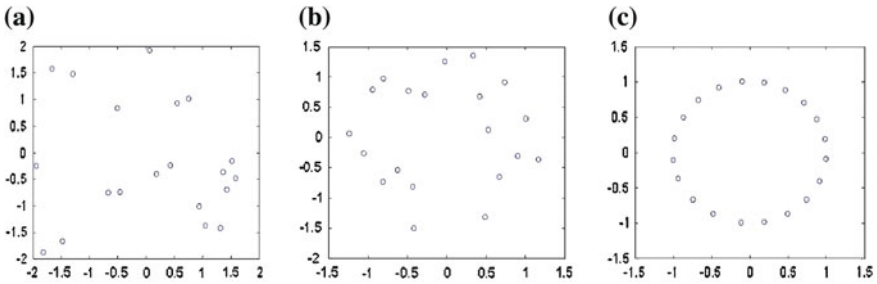


Fig. 3.3 Autonomous formation of a circle using 20 robots. **a** Initial positions. **b** Intermediate positions. **c** Converged positions

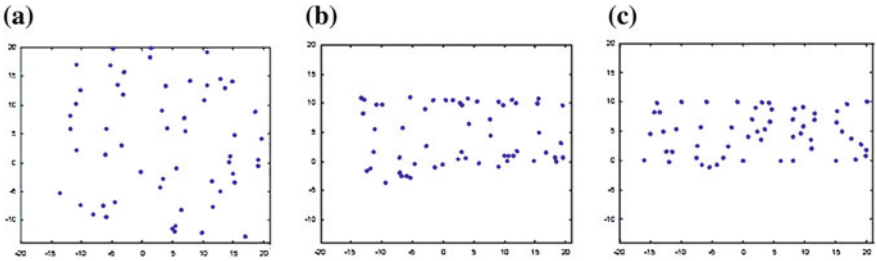


Fig. 3.4 Formation of ‘NURBS’ using 56 robots. **a** Initial positions. **b** Intermediate position. **c** Final shape



Fig. 3.5 Formation of letter ‘R’ using eight e-puck robots. **a** $t = 0s$, **b** $t = 6s$, **c** $t = 11s$

3.4 Morphogenetic Modular Robots for Self-Organized Reconfiguration

3.4.1 Reconfigurable Modular Robots

Self-reconfigurable modular robots consist of a number of modules and are able to adapt their shape (configuration) to changing environments by re-arranging their modules [37]. Each module is a physical or simulated “body” containing a controller. Physical modular robots, such as M-TRAN [36] and Molecube [38], have been constructed for reconfigurable robotic systems. Modules in M-TRAN comprise two connected cubic parts. The connection mechanism between the two cubic parts allows the modules to perform basic motions such as lifting or rotating. However, compared to single-cube mechanisms, the mandatory connection between the two cubic parts may become a mechanical constraint. The modules in Molecube are composed of two half-cubes on a diagonal plane. Each half-cube can swivel with respect to the other half, which is inspired by the swiveling action. An advantage of Molecube modules is their single cubic shape that can freely be attached to or detached from neighboring modules. However, the motion of the modules often requires more free space around the module so that the movement is not blocked. Although self-reconfiguration is the most important feature of self-reconfigurable robots, the ability to adapt their configuration autonomously under environmental changes remains to be demonstrated.

3.4.2 CrossCube: A Simulated Modular Robot

We have developed CrossCube [63], which adopts a lattice-based cube design. Each module in CrossCube is a cubical structure having its own computing and communication resources and actuation capabilities. Like all modular robots, the connection part of the modules can easily be attached or detached to/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. 3.6a. The core is a cube with six universal joints. Their default heading directions are bottom, up, right, left, front, and back, 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 can return to its default direction.

The cross-concaves on each side of the shell restrict the movement trajectory of the joints, as show in Fig. 3.6a. The borders of each module can actively be locked or unlocked with the borders of other modules, as shown in Fig. 3.6b. The length and angle of the lock mechanism can also be adjusted on the borders of the modules.

Basic movements of modules in CrossCube include rotation, climbing and parallel motion. Figure 3.6c illustrates a rotation movement of two modules. Parallel motion

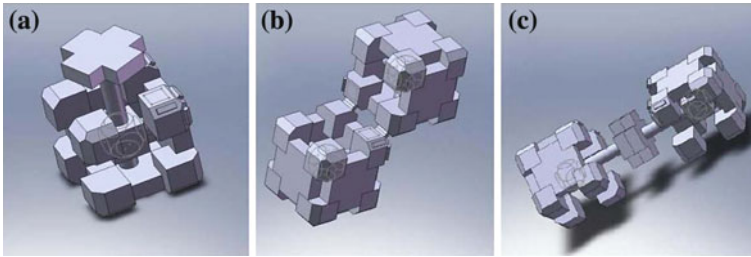


Fig. 3.6 Mechanical demonstration of CrossCube. **a** The joints. **b** The locks on the boundaries of the modules. **c** Rotation and extension of the joints of the modules

means that a module moves to a next position which is parallel to its current position. During a parallel motion, a module moves from its current position to a parallel position on its right. All joints of the modules stick out slightly to make enough free space for modules to move. Climbing motion means that a module moves to a diagonal neighboring position. Parallel motion and climbing allow a module of CrossCube to move to any position within the modular robot as long as the modules are connected.

3.4.3 *Self-Reconfiguration as Morphogenesis*

The link between reconfigurable modular robots and multi-cellular organisms appears more straightforward than in the case of collective robotics. Each unit in modular robots can be seen as a cell, and there are similarities in control, communication and physical interactions between cells in multi-cellular organisms and modules in modular robots. For example, control in both modular robots and multi-cellular organisms is decentralized. In addition, the global behavior of both modular robots and multi-cellular organisms emerges 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. Therefore, it is a natural idea to develop control algorithms for self-reconfigurable modular robots inspired from biological morphogenetic mechanisms [62, 63]. In the following, we briefly describe a recently proposed morphogenetic approach to designing control algorithms for reconfigurable modular robots.

Similar to morphogenetic swarm robotic systems, each unit of the modular robot contains a chromosome consisting of several genes that can produce different proteins. The proteins can diffuse into neighboring modules, thus establishing local communications between the modules. The concentration of the diffused proteins decays over time. The target configuration of the modular robot is also defined by

morphogen gradients. The space in which the modular robot is located is divided in a grid, where each grid location is occupied by one CrossCube module. The morphogen gradient can be either positive or negative. A positive morphogen gradient means that the grid location should be occupied by a module, while a negative gradient suggests that the module at that location, if any, should be removed. A higher value of morphogen gradient indicates a higher priority for the grid location to be filled by a module.

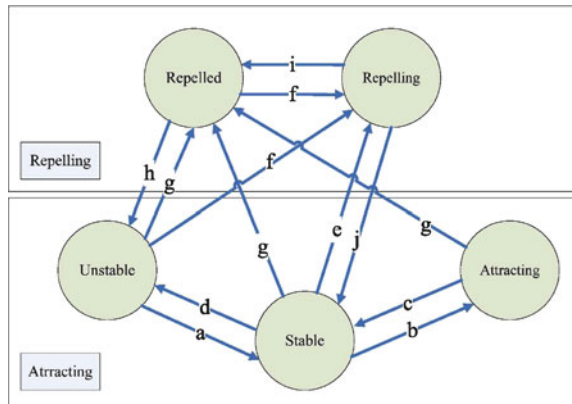
Different from the morphogenetic swarm robotic system described in Sect. 3.3, in which the target shape is fully predefined by a virtual “maternal” morphogen, each unit in the morphogenetic modular robot system can modify the morphogen gradients by secreting either positive or negative morphogen gradients, essential for adapting its configuration to the current environment or task. As a result, each module is able to attract or repel neighboring modules.

The attraction and repelling types of behavior of the modules are regulated by a GRN-based controller, which can adaptively set the state of the modules to one of the following five situations, namely, “stable”, “unstable”, “attracting”, “repelling”, and “repelled”. The transition relationships between the five states of modules are given in Fig. 3.7. Refer to [63] for details of state transitions.

3.4.3.1 GRN-Based Pattern Transition

State transitions are controlled by a GRN model containing two gene-protein pairs, an attracting gene-protein pair ($G_A - P_A$) and a repelling gene-protein pair ($G_P - P_P$). We assume that the repelling states always have a higher priority than the attracting states. As a result, all the states triggered by attracting behavior types can be overwritten by the states triggered by repelling behavior types. The rationale for this is that a grid location with a repelling (negative) morphogen gradient should be kept empty as long as migrating modules are still in need during reconfiguration.

Fig. 3.7 State transition of each module in CrossCube



3.4.3.2 Gene-Protein Pair for Attraction

The attracting gene-protein pair ($G_A - P_A$) is used to control the transition between “attracting”, “stable” and “unstable” states as shown in Fig. 3.7. At the initial stage of shape configuration, all modules are set as unstable. After they are initialized with the target configuration, modules at grid locations with an attracting morphogen gradient become stable. For a newly stabilized module, the gene expression level of its attracting gene G_A is initialized to be zero. Meanwhile, this module generates an attracting protein P_A for each empty neighboring grid location that has an attracting morphogen gradient. These locations become “attracting” to attract unstable modules to occupy them. Here, P_A is defined as:

$$P_A^{ij} = \{AP^{ij}, S^i, C_A^{ij}\}, \quad (3.15)$$

where P_A^{ij} is the attracting protein generated by the i -th module for its j -th neighbor. AP^{ij} is the j -th neighboring attracting grid location of the i -th module. S^i is the identification label of the i -th module, and C_A^{ij} is the concentration of the protein P_A^{ij} , which is equal to the morphogen gradient of AP^{ij} . P_A can regulate G_A in the same cell and can also diffuse into neighboring modules to regulate the G_A of neighbors as well.

The dynamics of G_A and P_A can be described by the following GRN model:

$$\frac{dg_A(t)}{dt} = -a \cdot g_A(t) + b * \sum p_{A,\text{local}} - c * \sum p_{A,\text{rec}}, \quad (3.16)$$

where $g_A(t)$ is the gene expression level of G_A at time t . $p_{A,\text{local}}$ and $p_{A,\text{rec}}$ are protein concentrations of locally generated protein and received protein from other modules, respectively. a , b , and c are constant coefficients.

Based on the expression level of g_A , the state of the module can be regulated according to the following rules:

$$\text{state} = \begin{cases} \text{unstable for } g_A < G_{A,L} \\ \text{stable for } G_{A,L} < g_A < G_{A,H} \\ \text{attracting for } g_A > G_{A,H} \end{cases} \quad (3.17)$$

where $G_{A,L}$ is a negative threshold and $G_{A,H}$ is a positive threshold. According to Eq. (3.16), g_A falls below a negative threshold $G_{A,L}$ with the increase of $c * \sum p_{A,\text{rec}}$. A higher value of $c * \sum p_{A,\text{rec}}$ means that there are more important grid locations to be filled. So the module needs to change its state from stable to unstable and move to a more important position following the attracting morphogen gradient. An unstable module chooses a P_A with the highest concentration value from all the received attracting proteins. Then the module migrates to the attracting position requested by that P_A . In order to guide the unstable modules to migrate to their destination, each module can detect the proteins within its local environment and choose the

position with the highest protein concentration as its destination. Once they reach their destination, the unstable modules become stable.

On the contrary, the expression level of g_A will be enhanced when $b * \sum p_{A,local}$ increases, which means that the module will have important neighboring positions to fill. So the module changes its current state to the attracting state. The attracting modules emit attracting proteins in the grid location in which they sit, and the emitted proteins will then diffuse into other modules. The attracting module will become stable again once its neighboring attracting positions are all occupied.

In summary, the gene-protein pair ($G_A - P_A$) can regulate each other through the GRN-based model described in Eqs. (3.16) and (3.17). More specifically, P_A can regulate G_A through Eq.(3.16), while G_A can determine when P_A is allowed to diffuse into neighboring grid locations based on Eq. (3.17). That is to say, only if the expression level of G_A is between $G_{A,L}$ and $G_{A,H}$, P_A can be generated; and only if the expression level of G_A is above $G_{A,H}$, P_A is allowed to diffuse.

3.4.3.3 Gene-Protein Pair for Repulsion

The repelling states are controlled by the repelling gene-protein pair ($G_P - P_P$). The repelling modules produce P_P , which is defined as

$$P_P^{ij} = \{R^{ij}, S^i, C_P^{ij}\}, \quad (3.18)$$

where P_P^{ij} is the repelling protein generated by the i -th module for its j -th neighbor. R^{ij} is the j -th repelling grid position around the i -th module. S^i is the identification label of repelling module i , and C_P^{ij} is the concentration of the protein P_P^{ij} , which is equal to a predefined positive constant. As we mentioned before, when a stable module finds out that some of its neighbors are located in a position with repelling morphogen gradient, it changes its state to “repelling” and switches the state of its neighbors to “repelled”. If the repelling module is triggered under this situation, R^{ij} is reset such that P_P can only repel the specific neighboring module that is located in R^{ij} . If the repelling module is triggered by a deadlock, R^{ij} is not reset because P_P should be detected by all the neighboring modules of the repelling module.

The gene expression level of g_P is initialized to be the morphogen gradient of the current grid position of the module. It can be regulated by P_P through the following equation:

$$\frac{dg_P(t)}{dt} = d \cdot g_P(t) - e * \sum p_{P_{rec}} \text{state} = \text{“repelled” when } g_P < G_{P_L} \quad (3.19)$$

where $g_P(t)$ is the gene expression level of the repelling gene G_P at time t , $p_{P_{rec}}$ is the concentration of the received repelling protein, G_{P_L} is a negative constant threshold and d and e are constant coefficients.

When a module receives P_P , the concentration of g_P will be reduced. If $g_P < G_{P_L}$, the module changes its state to “repelled”. Obviously, modules with a lower morphogen gradient are more easily repelled.

To summarize, P_P can regulate G_P through Eq. (3.19). G_P can produce P_P under the condition that G_P is below G_{P_L} and the module is blocked.

3.4.3.4 A Representation of Robot Configurations Based on Lookup Tables

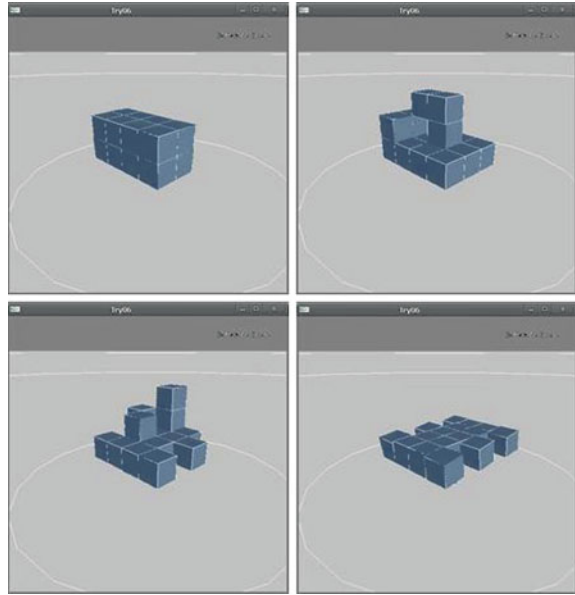
Adaptation to environmental changes is of paramount importance in reconfigurable modular robots. Similar to analytical or parametrized representation of the target shape in morphogenetic swarm robots, a mechanism is needed to define and modify the target configuration of the modular robot. Adaptation of the global configuration of the modular robot, i.e., change in morphogen gradients, can be triggered by local sensory feedback. Once a module receives such sensory feedback, this information will be passed on to its neighbors through local communication. In this way, a global change in configuration can be achieved.

For the sake of simplicity, a number of basic configurations for different environments can be predefined in terms of a lookup table for a given mission, for instance locomotion. For such tasks, it is also assumed that each module is equipped with a sensor to detect the distance between the module and obstacles in the environment. An example of defining the configuration of a vehicle is provided in Table 3.1. In this table, x , y , and z represent 3D coordinates of grid positions, ML denotes a morphogen level and PID stands for position identification. Additionally, we define a few joints’ movements to enable the vehicle to move, once the configuration is completed. Joints can be identified by their PID, and RD means the joint’s rotate direction.

Table 3.1 Vehicle configuration lookup table

Positions (x , y , z , ML, PID)		Joints (PID1, PID2, RD)
(0, 0, 0, 10, 0)	(1, 0, 3, 10, 10)	(0, 1, 0)
(1, 0, 0, 10, 1)	(2, 0, 3, 10, 11)	(2, 3, 1)
(2, 0, 0, 10, 2)	(0, 0, 4, 10, 12)	(6, 7, 0)
(3, 0, 0, 10, 3)	(1, 0, 4, 10, 13)	(8, 9, 1)
(1, 0, 1, 10, 4)	(2, 0, 4, 10, 14)	(12, 13, 0)
(2, 0, 1, 10, 5)	(3, 0, 4, 10, 15)	(14, 15, 1)
(0, 0, 2, 10, 6)	(0, 0, 1, -1, 16)	
(1, 0, 2, 10, 7)	(3, 0, 1, -1, 17)	
(2, 0, 2, 10, 8)	(0, 0, 3, -1, 18)	
(3, 0, 2, 10, 9)	(3, 0, 3, -1, 19)	

Fig. 3.8 Autonomous configuration of a vehicle from a rectangle



3.4.4 Illustrative Examples for Self-Reconfiguration

This section briefly describes a case study using a GRN-based controller to coordinate CrossCube modules for a locomotion task. A software is developed to simulate the behavior of CrossCube and its interactions with a physical world using C++ and the PhysX engine from Nvidia. In the following experiment, the parameters of the GRN models are set as follows: $a = 0.7$, $b = 1$, $c = 1$, $G_{A,L} = -1$, $G_{A,H} = 1$, $G_{P_L} = 2$, and $C_P^{ij} = 0.7$. The concentration of each protein decays to 80 % of its previous level when it diffuses into new grid locations.

A set of snapshots is provided in Fig. 3.8 to show how a vehicle configuration can be realized using the GRN-based control model and the lookup table in Table 3.1.

To verify the model's ability to adapt the configuration, a simulation was performed where a vehicle needs to go through a narrow tunnel whose width is smaller than that of the vehicle generated above. In this case, self-reconfiguration of the vehicle is necessary to accomplish the task. A number of snapshots showing this adaptation process is given in Fig. 3.9. More details about the GRN-based modular robot system can be found in [63].

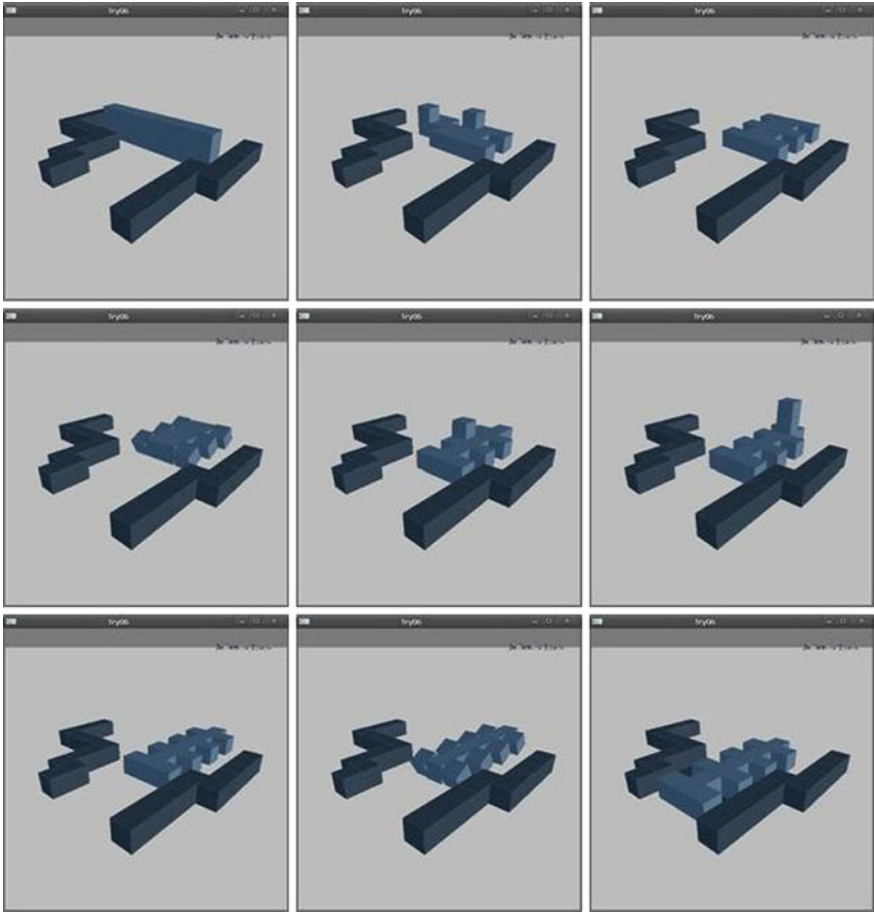


Fig. 3.9 A set of snapshots demonstrating a reconfiguration process during a locomotion task

3.5 Morphogenetic Brain-Body Co-Design of Robots

3.5.1 Brain-Body Co-Evolution

Brain-body co-evolution has attracted much attention in the research field of artificial life [57] since the seminal work of Karl Sims [53], refer to Fig. 3.10 for an overview. The most attractive aspect of the work is that a developmental model using a directed graph has been adopted for both neural controller and body plan. However, no significant progress in the understanding of biological principles have been achieved since Sims' work due to the following two facts. First, the power of the models for brain-body co-evolution remains practically bounded [18, 34, 54]. While detailed

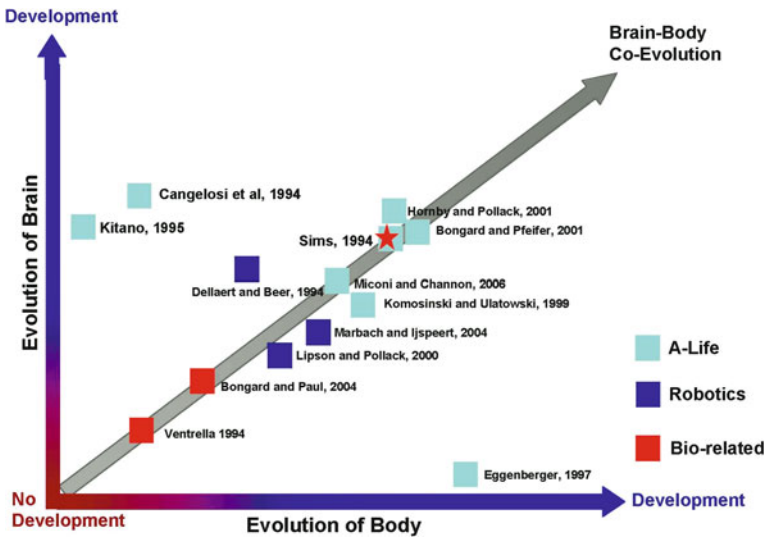


Fig. 3.10 A summary of brain-body co-evolution research

models of either neural development [24] or morphology [13] have been suggested, few models can achieve a balanced and biologically plausible abstraction level in modeling the development of both neural controller and body plan, and most of them are not able to perform biologically meaningful behaviors. Second, most works in brain-body co-evolution were meant mainly for improving the efficiency of generating a specific behavior, rather than understanding biological principles. An exception can be found the work by Bongard and Paul [8], who studied the correlation between morphological symmetry and locomotive efficiency.

In the following section, we first emphasize the importance of including development into robot design, followed by the description of a cellular model for modeling morphological and neural development. Then, a conceptual example of co-evolution of shape development and hand control for object grasping is presented.

3.5.2 Developmental Bias in Brain-Body Co-Evolution

Although the importance of co-evolution of brain and body has long been recognized [44], the simultaneous development of both the body plan and its control system has largely been overlooked in evolutionary robotics [29]. This situation has not changed much over time due to various difficulties in co-evolving the development of body and brain. First, there is a lack of knowledge about developmental mechanisms in biology and a lack of physically realistic simulated environments [42]. Second, the influence of artificial development on system performance is not well understood. Although it is believed that developmental mechanisms offer the possibility to evolve

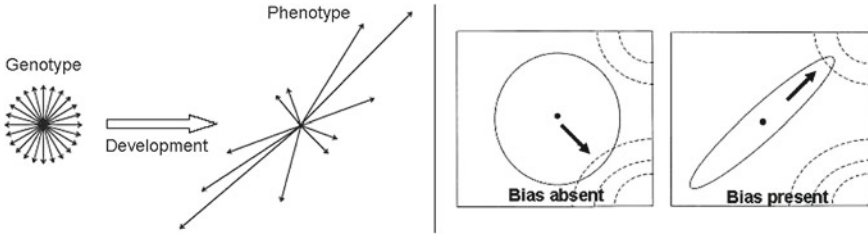


Fig. 3.11 Influence of development on selection directions. (*Left*) Developmental bias due to a nonlinear genotype-phenotype mapping (adapted from [45]). (*Right*) Change of selection directions in the presence of developmental bias (adapted from [2])

complex systems, the performance advantage of such developmental systems over non-developmental ones remains unclear. Finally, the necessary hardware, which is of particular importance in robotics, such as growing materials, adaptable structures, adaptable sensors and actuators, is still lacking.

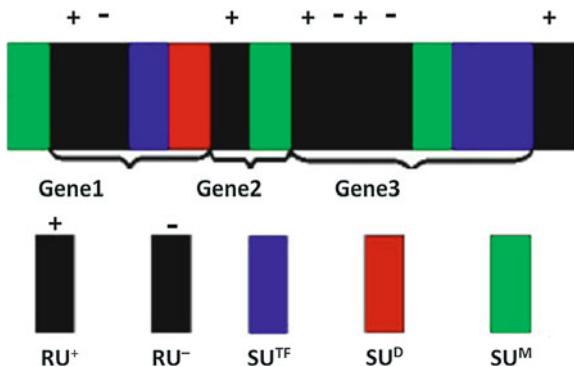
Nevertheless, the role of development in brain-body co-evolution cannot be over-estimated, simply because in natural evolution, development is an indispensable phase during which organisms have to interact with the environment constantly and find a way to survive. It is believed that development can bias the evolutionary path considerably, as illustrated in Fig. 3.11.

3.5.3 A GRN Model for Neural and Morphological Development

The growth of an animat morphology can also be put under the control of a GRN and cellular physical interactions. Elaborating upon the cellular growth model for structural design, we have proposed GRN models for the development of a nervous system [21] and body plan [49] of primitive animals. In the genome of the GRN models, each gene consists of a number of structural units (SUs) preceded by a number of regulatory units (RUs) on the genomic sequence. RUs can be activating (RU^+) or repressive (RU^-). When SUs are activated, they will produce either proteins responsible for cellular behaviors such as cell division, cell death, cell migration, and axon growth, or proteins regulating the activation of the SUs, also known as transcription factors (TFs). If a TF can only regulate the genes inside the cell, it is then called an internal TF. If a TF can also diffuse outside the cell and regulate the genes of other cells, it is called an external TF. A TF can be both intracellular and intercellular. An example of a chromosome in the cellular model for neural development is provided in Fig. 3.12. From the figure, we note that single or multiple RUs may regulate the expression of a single or multiple SUs, located just after them on the sequence.

Whether a TF can influence an RU is dependent on the degree of match between the affinity value of a TF and that of an RU. If the difference between the affinity values

Fig. 3.12 An example of chromosome for neural development, where RU^+ and RU^- denote activating and inhibitory regulatory units, respectively; SU^{TF} , SU^D and SU^M are structural units, once activated, producing transcription factors, resulting in cell division and cell migration, respectively



of a TF and a RU is smaller than a predefined threshold ε , the TF can bind to the RU to regulate. The affinity match ($\gamma_{i,j}$) between the i -th TF and j -th RU is defined by:

$$\gamma_{i,j} = \max \left(\varepsilon - \left| \text{aff}_i^{\text{TF}} - \text{aff}_j^{\text{RU}} \right|, 0 \right). \quad (3.20)$$

If $\gamma_{i,j}$ is greater than zero and the concentration c_i of the i -th TF is above a threshold (ϑ_j) defined in the j -th RU, then the i -th TF influences the j -th RU.

Thus, the activation level contributed by this RU (denoted by a_j , $j = 1, \dots, N$) amounts to $a_j = \sum_{i=1}^M |c_i, -\vartheta_j|$, where M is the number of existing TFs. The expression level of the k -th gene, that is regulated by N RUs, can be defined by

$$\alpha_k = 100 \sum_{j=1}^N h_j a_j (2s_j - 1), \quad (3.21)$$

where $s_j \in (0, 1)$ denotes the sign (positive for activating and negative for repressing) of the j -th RU and h_j is a parameter representing the strength of the j -th RU. If $\alpha_k > 0$, then the k -th gene is activated and its corresponding behaviors encoded in the SUs are performed.

A SU that produces a TF encodes all parameters related to the TF, such as the affinity value, a decay rate D_i^c , a diffusion rate D_i^f , as well as the amount of the TF to be produced:

$$A = \beta \frac{2}{1 + e^{-20 \cdot f \cdot \alpha}} - 1, \quad (3.22)$$

where f and β are both encoded in the SU^{TF} .

A TF produced by a SU can be partly internal and partly external. To determine how much of a produced TF is external, a percentage ($p^{\text{ex}} \in (0, 1)$) is also encoded in the corresponding gene. Thus, $p^{\text{ex}} A$ is the amount of external TF and $(1 - p^{\text{ex}}) A$ is that of the internal TF.

To make it easier for simulating the diffusion of TFs, cells are put in an environment that is divided in a grid. External TFs are put on four grid points around the center

of the cell, which undergoes first a diffusion and then decay process:

$$\text{Diffusion: } \mathbf{u}_i(t) = \mathbf{u}_i(t - 1) + 0.1 \cdot D_i^f \cdot (\mathbf{G} \cdot \mathbf{u}_i(t - 1)), \quad (3.23)$$

$$\text{Decay: } \mathbf{u}_i(t) = \min ((1 - 0.1 \cdot D_i^c) \mathbf{u}_i(t), 1), \quad (3.24)$$

where \mathbf{u}_i is a vector of the concentrations of the i -th TF at all grid points and the matrix \mathbf{G} defines which grid points are adjoining.

The SUs encode cellular behaviors and the related parameters. The SU for cell division encodes the angle of division, indicating where the daughter cell is placed. A cell with an activated SU for cell death will die at the end of the developmental phase.

The above cellular model has been applied to the simulation of both morphological and neural development [21, 49].

In one experiment, we generate an animat similar to *C. elegans*, on the basis of two prediffused, external TFs without decay and diffusion, which are deployed in the computation area (maternal morphogen gradients). The first TF has a constant gradient in the x -direction and the second in the y -direction. A few screenshots of the self-stabilized growth is provided in Fig. 3.13. Other images showing the growth of a nervous system distributed on a cylinder body simulating a hydra are given in

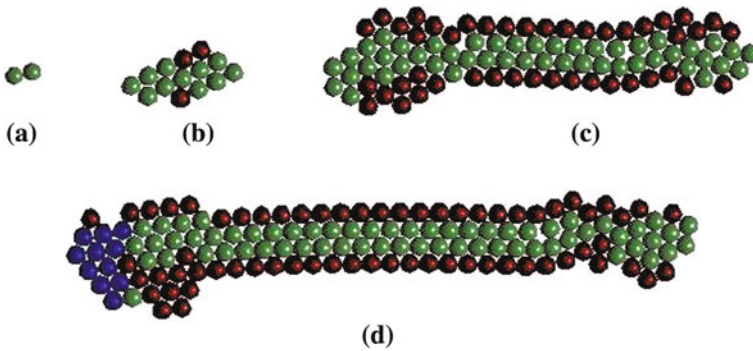


Fig. 3.13 Self-stabilized cellular growth under the control of a GRN model

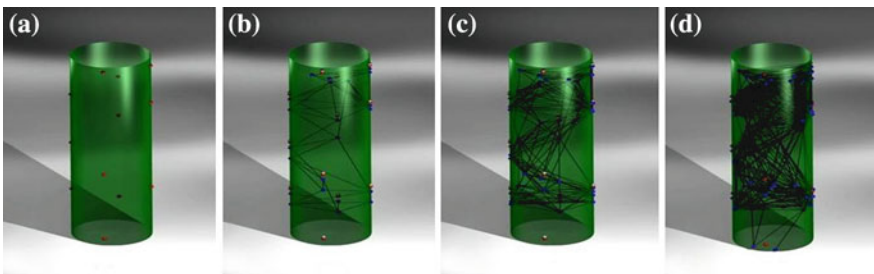


Fig. 3.14 Development of a nervous system

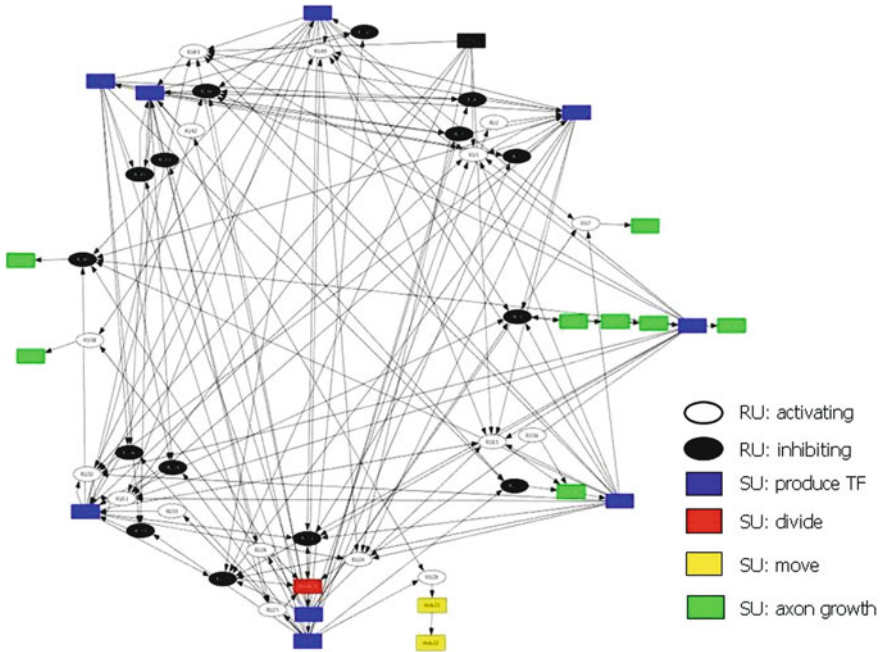


Fig. 3.15 An evolved GRN for artificial neural development

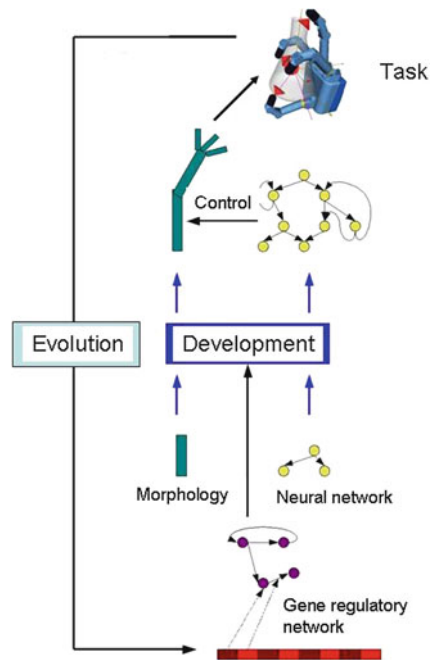
Fig. 3.14. A GRN resulting from the evolution of neural development is presented in Fig. 3.15.

3.5.4 Co-Evolution of Hand Morphology and Controller Development

One can observe that the morphology of animal hands kept changing dramatically during evolution. Animal hands distinguish themselves in both shape and length in the finger segments. Moreover, it has been hypothesized that the reproductive advantage of certain types of behavior, such as throwing and clubbing, played a key role in differences between human and chimpanzee hands [61].

The importance of co-evolving the development of hand morphology and control in robotics is twofold. First, object grasping and manipulation with a robot hand is in itself a challenging task as such systems are usually highly redundant. Existing work focuses on the design of the hand controller for a given morphology, which becomes inefficient when the shape of the objects changes noticeably. A better approach is to co-design the hand morphology and control in a developmental manner, as illustrated

Fig. 3.16 A diagram for co-evolving the development of hand/arm and control



in Fig. 3.16. In this way, the shape and number of finger segments, the number of fingers and even the number of arms can evolve together with their controller.

Second, co-evolution of the hand morphology and control in a computational environment gives us means for understanding the phylogenetic changes in evolution of animal hands. Brain-body co-evolution in computational environments has led to findings regarding the organizational principles of nervous systems and the emergence of bilateral symmetry in neural configuration [23, 39]. We expect that different hand morphologies will emerge by evolving the system toward different behaviors.

3.6 Concluding Remarks

This chapter suggests a new field of robotics termed *morphogenetic robotics*, which focuses on employing genetic and cellular mechanisms from biological morphogenesis toward the development of self-organizing, self-reconfigurable and self-adaptive robotic systems, covering a wide range of robotic systems such as swarm robotic systems, modular robots and intelligent robots. While epigenetic robotics concentrates on the cognitive development of robotic systems, morphogenetic robotics looks at on the growth process of the body plan and nervous system. Therefore, we believe that morphogenetic robotics is complementary to epigenetic robotics and fills the

gap between epigenetic robotics and developmental robotics in that developmental robotics should include neural, morphological and cognitive development. We also expect that we will benefit from the synergies between morphogenetic and epigenetic robotics, as neural and morphological development are laying down the neurophysiological foundations of cognitive development.

Acknowledgments The authors would like to thank Hongliang Guo, Yuyang Zhang, Till Steiner, Lisa Schramm and Benjamin Inden for the illustrative examples used in this chapter. YJ is grateful to Edgar Körner and Bernhard Sendhoff for their support.

References

1. Alon, U.: Network motifs: theory and experimental approaches. *Nat. Rev. Genet.* **8**, 450–461 (2007)
2. Arthur, W.: The effect of development on the direction of selection: toward twenty-first century consensus. *Evol. Dev.* **10**, 375–390 (2004)
3. Ashe, H., Briscoe, J.: The interpretation of morphogen gradients. *Development* **133**, 385–394 (2007)
4. Belousov, L.: Integrating self-organization theory into an advanced course on morphogenesis at Moscow State University. *Int. J. Dev. Biol.* **47**, 177–181 (2003)
5. Ben-Amor, H., Cadau, S., Elena, A., Dhouailly, D., Demongeot, J.: Regulatory networks analysis: robustness in morphogenesis. In: 2009 International Conference on Advanced Information Networking and Application Workshops, pp. 924–928, Bradford, UK (2009)
6. Bhattacharyya, A.: Morphogenesis as an amorphous computing. In: 3rd Conference on Computing Frontiers, pp. 53–64, ACM, New York (2006)
7. Bishop, C., Erezyilmaz, D., Flatt, T., Georgiou, C., Hadfield, M., Heyland, A., Hodin, J., Jacobs, M., Maslakova, S., Pires, A., Reitzel, A., Santagata, S., Tanaka, K., Youson, J.: What is metamorphosis? *Integr. Comp. Biol.* **46**, 655–661 (2006)
8. Bongard, J., Paul, C.: Investigating morphological symmetry and locomotive efficiency using virtual embodied evolution. In: From Animal to Animats: The Sixth International Conference on Simulated Adaptive Behavior, pp. 420–429 (2000)
9. Bower, J., Bolouri, H.: Computational Modeling of Genetic and Biochemical Networks. MIT Press, Cambridge (2001)
10. Crumiere, A., Sablik, M.: Positive circuits and d -dimensional spatial differentiation: application to the formation of sense organs in *Drosophila*. *BioSystems* **94**, 102–108 (2008)
11. DeJong, H.: Modeling and simulation of genetic regulatory systems: a literature review. *J. Comput. Biol.* **9**, 67–103 (2002)
12. Doursat, R.: Programmable architectures that are complex and self-organized: from morphogenesis to engineering. In: Bullock, S., Noble, J., Watson, R., Bedau, M. (eds.) 11th International Conference on the Simulation and Synthesis of Living Systems (ALIFE XI), pp. 181–188. MIT Press, Winchester (2008)
13. Eggenberger, P.: Evolving morphologies of simulated 3d organisms based on differential gene expression. In: Artificial Life IV, pp. 205–213 (1997)
14. Endy, D., Brent, R.: Modeling cellular behavior. *Nature* **409**, 391–395 (2001)
15. Francois, F., Hakim, V.: Design of genetic networks with specified functions by evolution *in silico*. *Proc. Natl. Acad. Sci. USA* **101**, 580–585 (2004)
16. Gilbert, S.: *Developmental Biology*. Sinauer Associates, Sunderland (2003)
17. Guo, H., Meng, Y., Jin, Y.: A cellular mechanism for multi-robot construction via evolutionary multi-objective optimization of a gene regulatory network. *BioSystems* **98**, 193–203 (2009)

18. Hornby, G., Pollack, J.: Creating high-level components with a generative representation for body-brain evolution. *Artif. Life* **8**, 223–246 (2002)
19. Jin, Y., Guo, H., Meng, Y.: Robustness analysis and failure recovery of a bio-inspired self-organizing multi-robot system. In: Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems, pp. 154–164. IEEE Press, Los Alamitos (2009)
20. Jin, Y., Meng, Y.: Emergence of robust regulatory motifs from in silico evolution of sustained oscillation. *BioSystems* **103**(1), 38–44 (2011)
21. Jin, Y., Schramm, L., Sendhoff, B.: A gene regulatory model for the development of primitive nervous systems. In: INNS-NNN Symposia on Modeling the Brain and Nervous Systems. LNCS, vol. 5506, pp. 48–55. Springer, Berlin (2009)
22. Jin, Y., Sendhoff, B.: Evolving in silico bistable and oscillatory dynamics for gene regulatory network motifs. In: Congress on Evolutionary Computation, pp. 386–391 (2008)
23. Jones, B., Jin, Y., Sendhoff, B., Yao, X.: Evolving functional symmetry in a three dimensional model of an elongated organism. In: *Artificial Life XI*, pp. 305–312 (2008)
24. Kitano, H.: A simple model of neurogenesis and cell differentiation based on evolutionary large-scale chaos. *Artif. Life* **2**(1), 79–99 (1995)
25. Knabe, J.F., Nehaniv, C.L., Schilstra, M.J.: Genetic regulatory network models of biological clocks: evolutionary history matters. *Artif. Life* **14**, 135–148 (2008)
26. Kuniyoshi, Y., Lungarella, M.: Developmental cognitive robotics: an emerging new field. In: Workshop on Cognitive Developmental Robotics, pp. 3–5, Kyoto, Japan (2006)
27. Kwon, Y.K., Cho, K.H.: Coherent coupling of feedback loops: a design principle of cell signaling networks. *Bioinformatics* **24**, 1926–1932 (2008)
28. Lee, J., Sitte, J.: Morphogenetic evolvable hardware controllers for robot walking. In: 2nd International Symposium on Autonomous Minirobots for Research and Edutainment (2003)
29. Lipson, H.: Uncontrolled engineering: a review of evolutionary robotics (book review). *Artif. Life* **7**, 419–424 (2001)
30. Lungarella, M., Metta, G., Pfeifer, R., Sandini, G.: Developmental robotics: a survey. *Connect. Sci.* **15**, 151–190 (2003)
31. Mamei, M., Vasirani, M., Zambonelli, M.: Experiments in morphogenesis in swarms of simple mobile robot. *Appl. Artif. Life* **18**, 903–919 (2004)
32. Meiner, D.: Swarm robotics algorithm: a survey. Technical report, University of Maryland (2007)
33. Meng, Y., Guo, H., Jin, Y.: A morphogenetic approach to flexible and robust shape formation for swarm robotic systems. *Auton. Robots* (2009) (Submitted)
34. Miconi, T., Channon, A.: An improved system for artificial creatures evolution. In: *Artificial Life X*, pp. 255–261 (2006)
35. Mjolsness, E., Sharp, D.H., Reinitz, J.: A connectionist model of development. *J. Theor. Biol.* **52**, 429–453 (1991)
36. Murata, S., Kakomura, K., Kurokawa, H.: Toward a scalable modular robotic system—navigation, docking, and integration of M-TRAN. *IEEE Robot. Autom. Mag.* **14**, 56–63 (2008)
37. Murata, S., Kurokawa, H.: Self-reconfigurable robots. *IEEE Robot. Autom. Mag.* **14**, 71–78 (2007)
38. Mytilinaios, E., Marcus, D., Desnoyer, M., Lipson, H.: designed and evolved blueprints for physical artificial life. In: Ninth International Conference on Artificial Life (ALIFE IX), pp. 15–20 (2004)
39. Oros, N., Steuber, V., Davey, N., Caamero, L., Adams, R.: Evolution of bilateral symmetry in agents controlled by spiking neural networks. In: 2009 IEEE Symposium on Artificial Life, pp. 116–123 (2009)
40. Paladugu, S., Chickarmane, V., Dekard, A., Frumkim, J., MaCormarck, M., Sauro, H.: *In silico* evolution of functional modules in biochemical networks. *IEE Proc. Syst. Biol.* **153**, 223–235 (2006)
41. Pfeifer, R., Iida, F., Gomez, G.: Morphological computation for adaptive behavior and cognition. *Inter. Congr. Ser.* **1291**, 22–29 (2006)
42. Pfeifer, R., Knoll, A.: Intelligent and cognitive systems. *ERCIM News* **64**, 11–12 (2006)

43. Piegel, L., Tiller, W.: *The NURBS Book*. Springer, Berlin (1997)
44. Pollack, J., Lipson, H., Funes, P., Ficici, S., Hornby, G.: Coevolutionary robotics. In: *Evolvable Hardware*, pp. 208–216 (1999)
45. Psujek, S., Beer, R.: Developmental bias in evolution: evolutionary accessibility of phenotypes in a model of evo-devo systems. *Evol. Dev.* **10**(3), 375–390 (2008)
46. Nolfi, S., Floreano, D.: *Evolutionary Robotics*. MIT Press, Cambridge (2004)
47. Sanes, D., Reh, T., Harris, W.: *Development of Nervous Systems*, 2nd edn. Academic Press (2006)
48. Schilstra, M., Nehaniv, C.: Bio-logic: gene expression and the laws of combinatorial logic. *Artif. Life* **14**, 121–133 (2008)
49. Schramm, L., Jin, Y., Sendhoff, B.: Emerged coupling of motor control and morphological development in evolution of multi-cellular animates. In: *10th European Conference on Artificial Life* (2009)
50. Shen, W., Will, P., Galstyan, A.: Hormone-inspired self-organization and distributed control of robotic swarms. *Auton. Robots* **17**, 93–105 (2004)
51. Sima, C., Hua, J., Jung, S.: Inference of gene regulatory networks using time-series data: a survey. *Curr. Genomics* **10**, 416–429 (2009)
52. Simons, K., Karsenti, E., St Johnston, D., Wijer, C., Swaminathan, S. (eds.): *Self-Organization and Morphogenesis in Biological Systems*. Schloss Ringberg, Tegernsee, Germany (2006)
53. Sims, K.: Evolving 3D morphology and behavior by competition. *Artif. Life* **1**, 353–372 (1994)
54. Spector, L., Klein, K., Feinstein, M.: Division blocks and the open-ended evolution of development, form, and behavior. In: *Genetic and Evolutionary Computation Conference*, pp. 316–323 (2007)
55. Stanley, K.O., Miikkulainen, R.: A taxonomy for artificial embryogeny. *Artif. Life* **9**, 93–130 (2003)
56. Steiner, T., Trommler, J., Brenn, M., Jin, Y., Sendhoff, B.: Global shape with morphogen gradients and motile polarized cells. In: *Congress on Evolutionary Computation*, pp. 2225–2232 (2009)
57. Taylor, T., Massey, C.: Recent developments in the evolution of morphologies and controllers for physically simulated creatures. *Artif. Life* **7**(1), 77–87 (2001)
58. Tsai, T.Y.C., Choi, Y.S., Ma, W., Pomerening, J., Tang, C., Ferrell Jr, J.: Robust tunable biological oscillations from interlinked positive and negative feedback loops. *Science* **321**, 126–129 (2008)
59. Weng, J., McClelland, J., Pentland, J., Sporns, O., Stockman, I., Sur, M., Thelen, E., : Autonomous mental development by robots and animals. *Science* **291**, 599–600 (2001)
60. Wolpert, L.: *Principles of Development*. Oxford University Press, New York (2002)
61. Young, R.: Evolution of human hand: the role of throwing and clubbing. *J. Anat.* **202**, 165–174 (2003)
62. Yu, C.H., Haller, K., Ingber, D., Nagpal, R.: Morpho: A self-deformable modular robot inspired by cellular structure. In: *Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3571–3578. IEEE Press, Los Alamitos (2008)
63. Zhang, Y., Meng, Y., Jin, Y.: CROSSCUBE—a morphogenetic modular robot. In: *International Conference on Robotics and Automation* (2010)