Morphogenetic Robotics: An Emerging New Field in Developmental Robotics

Yaochu Jin, *Senior Member, IEEE*, and Yan Meng, *Member, IEEE*

Abstract—Developmental robotics is also known as epigenetic robotics. We propose in this paper that there is one substantial difference between developmental robotics and epigenetic robotics, since epigenetic robotics concentrates primarily on modeling the development of cognitive elements of living systems in robotic systems, such as language, emotion, and social skills, while developmental robotics should also cover the modeling of neural and morphological development in single- and multirobot systems. With the recent rapid advances in evolutionary developmental biology and systems biology, increasing genetic and cellular principles underlying biological morphogenesis have been revealed. These principles are helpful not only in understanding biological development, but also in designing self-organizing, self-reconfigurable, and self-repairable engineered systems. In this paper, we propose *morphogenetic robotics*, an emerging new field in developmental robotics, is an important part of developmental robotics in addition to epigenetic robotics. By morphogenetic robotics, we mean a class of methodologies in robotics for designing self-organizing, self-reconfigurable, and self-repairable single- or multirobot systems, using genetic and cellular mechanisms governing biological morphogenesis. We categorize these methodologies into three areas, namely, morphogenetic swarm robotic systems, morphogenetic modular robots, and morphogenetic body and brain design for robots. Examples are provided for each of the three areas to illustrate the main ideas underlying the morphogenetic approaches to robotics.

Index Terms—Developmental robotics, epigenetic robotics, evolutionary developmental robotics, morphogenesis, morphogenetic robotics.

I. WHAT IS AND WHY MORPHOGENETIC ROBOTICS?

DEVELOPMENTAL *robotics* is an interdisciplinary field of robotics that employs simulated or physical robots to understand natural intelligence on the one hand, and to design better robotic systems using principles in biological development, on the other hand [57]. The term *developmental robotics* is often used interchangeably with other two terms, namely, *epigenetic robotics* [64] and *ontogenetic robotics*, which focuses on modeling the postnatal development of cognitive behaviors in living systems, such as language, emotion, anticipation, and

social skills. Note, however, that the meaning of *epigenetic* is not unambiguous in biology. It can either be derived from *epigenesis* that describes morphogenesis and postnatal development of organisms, or from *epigenetics* referring to the changes in gene expression that are not caused by genetic changes. Although the difference between developmental robotics and epigenetic robotics has been noticed and topics that are far beyond cognitive development have been discussed [57], researchers have so far refrained from stressing the difference between developmental robotics and epigenetic robotics. Nevertheless, other terminologies such as developmental cognitive robotics [50] and autonomous mental development [95] have also been suggested to refer to the research efforts on modeling cognitive development, which we believe may be seen as attempts to clarify the confusion caused by treating epigenetic robotics equivalent to developmental robotics.

The physical development of animals includes the processes that cause the creation of both the body plan and nervous system, including cleavage, gastrulation, neurulation, organogenesis [96]. Some living organisms, such as amphibians, also undergo a biological process known as metamorphosis, during which both the shape and size of the organisms change [9]. The past decade has witnessed rapid technical and theoretical advances in evolutionary developmental biology [28] (often known as *evo-devo*) and systems biology in understanding molecular and cellular mechanisms that control the biological morphogenesis. These advances have not only helped us in understanding biological processes such as human deceases, but also provided us new powerful tools for designing engineered systems. For example, increasing evidence has been revealed that biological morphogenesis can be regarded as a self-organizing and self-assembling process through cellular and molecular interactions under the genetic and environmental control [6], [86]. In addition, biological morphogenesis has also shown a surprising degree of robustness [7]. Due to the attractive properties that biological morphogenesis exhibits, much attention has been paid to employing genetic and cellular mechanisms for designing robotic systems, in particular for self-organizing swarm robotic systems and self-reconfigurable modular robots. In addition, a large body of research has been performed in artificial life and robotics to design the body plan and neural controller of robots, using an evolutionary developmental approach [87], [89], [92].

In this paper, we propose that it is high time that the difference between developmental robotics and epigenetic robotics be stressed. To this end, we suggest that a new term, namely, *morphogenetic robotics*, be used to denote research efforts dedicated to the application of morphogenetic mechanisms to robotics,
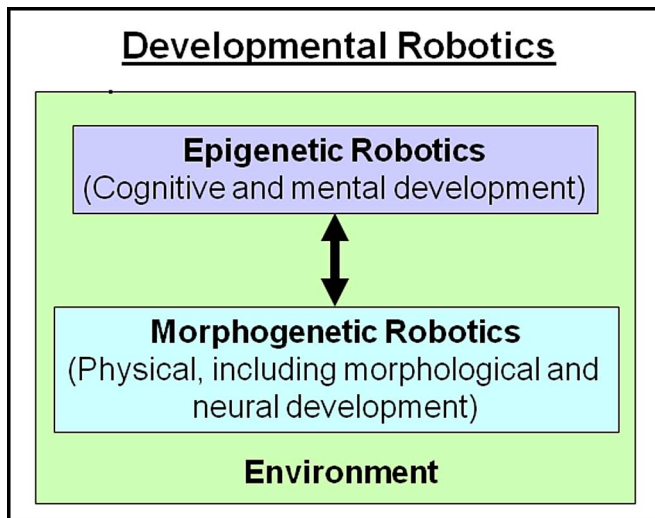
Fig. 1. Relationships between morphogenetic robotics, epigenetic robotics, and developmental robotics. Developmental robotics includes both morphogenetic robotics and epigenetic robotics, and morphogenetic robotics and epigenetic robotics are closely coupled not only directly in that the body plan and nervous system are the basis of cognitive development, but also indirectly through the environment.

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

1) Morphogenetic swarm robotic systems that deal with the self-organization of swarm robots using genetic and cellular mechanisms underlying the biological early morphogenesis [32], [58], [84].
2) Morphogenetic modular robots where modular robots adapt their configurations autonomously based on the current environmental conditions using morphogenetic principles [62], [63].
3) Developmental approaches to the design of the body or body parts, including sensors and actuators, and/or, design of the neural network-based controller of robots [35], [52]. Note that in epigenetic robotics, autonomous mental development can be seen as an incremental, on-line, and open-ended autonomous learning process situated in physical and social environments. The first neural structure, as well as a basic intrinsic motivation system, which enables robots to learn autonomously, is genetically hard wired as a result of neural morphogenesis (neurogenesis).

We believe that developmental robotics should include both morphogenetic robotics and epigenetic robotics. The former is mainly concerned with the physical development of the body and neural control, whereas the latter focuses on the cognitive and mental development. The body morphology, as well as the neural structure of the robots is a result of morphogenetic development, on which mental development is based through interaction with the environment. The relationship between morphogenetic robotics, epigenetic robotics, and developmental robotics is summarized in Fig. 1.

Other closely related terms are evolutionary robotics [79] and morphological computation (also known as morphological
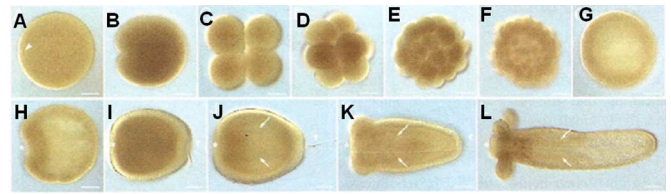

Fig. 2. Morphogenesis of *Nematostella vectensis*. The development stages are: egg (A), morlula (B–F), blastula (G), gastrula (H), planula (I–J), and polyp (K–L). Taken from [53].

robotics) [73]. Traditionally, evolutionary robotics is concerned with the design of robot controllers, using evolutionary algorithms, which takes the phylogeny into account. Complementary to evolutionary robotics, where the role of robot morphology is largely neglected in designing intelligent behaviors, morphological robotics was targeted for connecting brain, body, and environment in robot design. Unfortunately, it appeared that morphological computation advocates to generate intelligent behaviors with the help of the robot morphology using a simple controller and has not paid sufficient attention to the developmental aspects of morphology.

A brief introduction to biological morphogenesis and its computational models are presented in Section II. A metaphor between swarm robotic systems and multicellular systems is described in Section III, 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 IV presents our idea of using a gene regulatory network model for self-reconfigurable modular robots, followed by Section V, where the developmental approach to evolutionary codesign of the body and controller of robots or robot parts (e.g., a robot arm/hand for object grasping) is discussed. Concluding remarks of this paper are provided in Section VI.

II. MULTICELLULAR MORPHOGENESIS AND ITS COMPUTATIONAL MODELING

A. Biological Morphogenesis and Metamorphosis

Morphogenesis of animals can be divided into early embryonic development and later embryonic development [27]. Early embryonic development typically involves cleavage, gastrulation, and axis formation, while later embryonic development is mainly responsible for the development of the nervous systems, starting with the segregation of neural and glial cells from the ectoderm germ layer [80]. An example of morphogenesis of nematostella vectensis is illustrated in Fig. 2.

Metamorphosis is another interesting stage of biological development. 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. In contrast, organisms that undergo complete metamorphosis have four developmental stages, in which the shape of the organisms changes drastically. Fig. 3 illustrates incomplete and complete metamorphosis of insects.
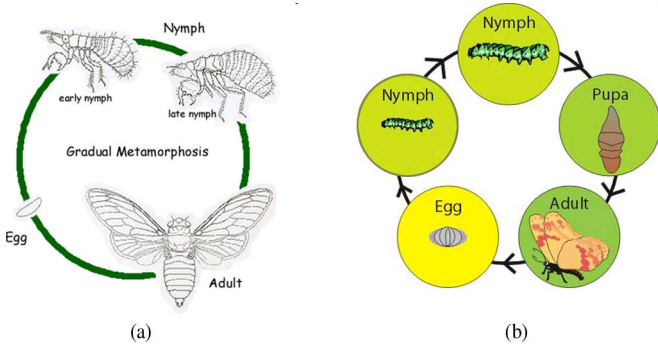
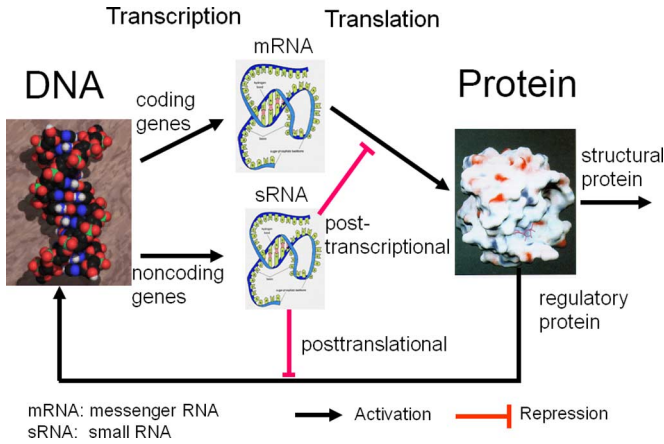Fig. 3. (a) Incomplete and (b) complete metamorphosis.

Fig. 4. Illustration of the gene expression process. Coding genes are first transcribed into mRNA, and then translated into proteins, which can have cellular and/or regulatory functions. Some so-called noncoding genes can also transcribed into small RNAs that can downregulate or fully switch off the expression of some genes through transcriptional or posttranslational interferences.

Both multicellular morphogenesis and metamorphosis are under the control of gene regulatory networks. When the DNA is expressed, information stored in the genome is transcribed into mRNA and then translated into proteins. Some of these proteins are transcription factors (TFs) that can regulate the expression of their own or other genes, thus resulting in a complex network of interacting genes termed as a gene regulatory network (GRN). New findings in the recent years suggest that some previously so called noncoding genes are transcribed into small RNAs [59], which can downregulate the expression of genes through e.g., RNA interference, as sketched in Fig. 4. However, most computational GRN models have not yet taken such effects explicitly into account with few exceptions [26], [29]. In the following, we will discuss computational modeling of GRNs and show how these models can be used for understanding biology and solving engineering problems.

B. Computational Modeling of Developmental Gene Networks

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 [11]. A large number of computational models for GRNs have been suggested [15], [20], which can largely

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, GRN models also distinguish themselves as deterministic models and stochastic models according to their ability to describe stochasticity in gene expression. Note that in artificial life, a few high-level abstraction models have also been used for modeling development, such as the L-systems [55] and grammar trees [30].

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

$$\frac{d\,[R]}{d\,t} = -\gamma_R\,[R] + \alpha_R H([P]) \tag{1}$$

$$\frac{d\,[P]}{d\,t} = -\gamma_P\,[P] + \alpha_P\,[R] \tag{2}$$

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

$$H_r(x) = \frac{\beta}{\theta^n + x^n} \tag{3}$$

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

$$H_a(x) = \frac{\beta\,x^n}{\theta^n + x^n} \tag{4}$$

where β is the activation coefficient, θ is the threshold, and n is the Hill coefficient.

For describing the morphogenesis of multicellular organisms, the interaction between the cells and its influence on gene expression dynamics must be taken into account. Mjolsness et al. [66] has suggested a generalized GRN model that considered diffusion of TFs among the cells:

$$\frac{dg_{ij}}{dt} = -\gamma_j g_{ij} + \phi\left[\sum_{l=1}^{n_g} W^{jl} g_{il} + \theta_j\right] + D_j \nabla^2 g_{ij} \tag{5}$$

where g_{ij} denotes the concentration of jth gene product (protein) in the ith cell. The first term on the right-hand side of (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 defined as a sigmoid function $\phi(z) = 1/(1 + \exp(-\mu z))$. The interaction between the genes is described with an interaction matrix W^{jl}, the element of which can be either active (a positive value) or repressive (a negative value). θ_j is a threshold for activation of gene expression. n_g is the number of proteins. An illustration of cell–cell interactions is provided in Fig. 5, 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
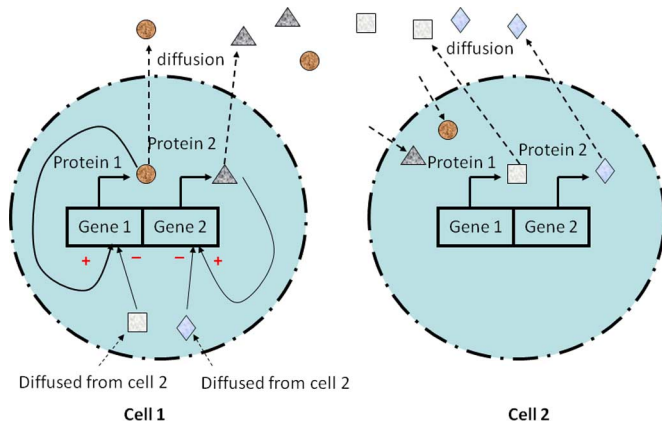
Fig. 5.   Illustration of cell signaling in a multicellular system.

2 of cell 1 is activated by its own protein and repressed by the protein of gene 2 of cell 2 through diffusion.

C. Applications of Computational Models of GRN

In addition to reconstruction of gene regulatory pathways based on biological data [85], computational models have widely been used for analyzing the dynamics of GRNs, particularly regarding robustness of GRN motifs, synthesizing *in silico* typical regulatory dynamics such as bistability and sustained oscillation, and designing engineered systems [16], including morphogenetic robotics that will be discussed in greater detail in the following sections.

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 autoregulation, feedforward loops, and feedback loops [1]. Recently, the role of feedback loops, in particular, the coupling of feedback loops and its relationship to the robustness of resulting dynamics of the network motifs has received increasing attention [14], [51], [94].

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 [81], which suggested that networks consisting of competitively binding activators and repressors can be controlled more robustly.

2) In Silico Synthesis: In silico synthesis of typical regulatory dynamics can offer us insights into how nature has shaped the evolution of regulatory motifs [25], [72]. In Knabe *et al.* [46], 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 tends to be robust to perturbations that evolution has experienced. Jin *et al.* [41] 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 is 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 [40]. In [40], it has been found that there is an inherent tradeoff between innovation and robustness in a feedforward Boolean model of genotype–phenotype mapping [38], which, interestingly can be resolved in a dynamic gene regulation model [42].

3) Artificial Embryogeny: A large body of research has been reported on simulating biological development in computational environments [89]. Motivations of building models for artificial embryogeny include understanding biological development in artificial life [89], designing complex structures [91], and amorphous computing [8], to name a few. Obviously, all the three areas of morphogenetic robotics involve in computational models of biological morphogenesis, which will be elaborated in the following sections.

III. MORPHOGENETIC SWARM ROBOTIC SYSTEMS

A. Swarm Robotic Systems

A swarm robotic system is a multirobot 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, boundary coverage, urban search and rescue, and unknown environment exploration. However, designing a decentralized control algorithm for swarm robotic systems has been a challenging task [12], [60].

B. Metaphor Between Swarm Robotic Systems and Multicellular Systems

1) Cell-Robot Mapping: The basic idea in applying genetic and cellular mechanisms in biological morphogenesis to self-organized control of swarm robots is to establish a metaphor between a cell and a robot. In other words, it is assumed that the movement dynamics of each robot can be modeled by the regulatory dynamics of a cell. In [32], [39], [61], 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 3-D 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. Fig. 6 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.

Keeping the metaphor between the cells and the robots in mind, the movement dynamics of each robot can be described by a GRN model, where the concentration of two proteins of type G represents the x and y position of a robot, respectively,
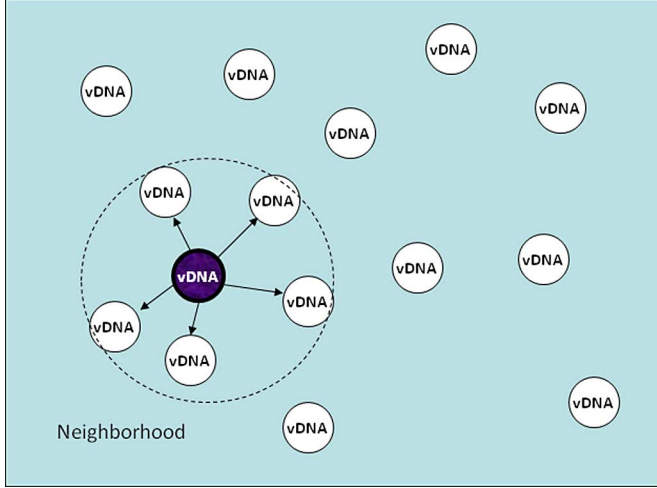
Fig. 6. Swarm robotic system represented by a multicellular system. Each circle represents a cell or a robot, and the dashed large circle denotes the neighborhood of a cell/robot (shaded), to which information can be passed.

and that of the proteins of type P representing the analog of the velocity.

$$\frac{dg_{i,x}}{dt} = -az_{i,x} + mp_{i,x}$$

$$\frac{dg_{i,y}}{dt} = -az_{i,y} + mp_{i,y} \qquad (6)$$

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

$$\frac{dp_{i,y}}{dt} = -cp_{i,y} + kf(z_{i,y}) + bD_{i,y} \qquad (7)$$

where $i = 1, 2, ..., n$, 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 ith robot, respectively, which corresponds to the concentration of two proteins of type G. $p_{i,x}$ and $p_{i,y}$ are the concentration of two proteins of type P, which denotes the velocity-like property of the ith robot along the x and y coordinates, respectively. $D_{i,x}$ and $D_{i,y}$ are the sum of the distances between the ith robot and its neighbors. In the language of the multicellular system, it is the sum of the concentration of protein type G diffused from neighboring cells. Mathematically, we have

$$D_{i,x} = \sum_{j=1}^{N_i} D_{i,x}^j \quad D_{i,y} = \sum_{j=1}^{N_i} D_{i,y}^j \qquad (8)$$

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 is defined as

$$D_{i,x}^j = \frac{(g_{i,x} - g_{j,x})}{\sqrt{(g_{i,x} - g_{j,x})^2 + (g_{i,y} - g_{j,y})^2}} \qquad (9)$$

$$D_{i,y}^j = \frac{(g_{i,y} - g_{j,y})}{\sqrt{(g_{i,x} - g_{j,x})^2 + (g_{i,y} - g_{j,y})^2}}. \qquad (10)$$

The diffusion term in the regulatory model simulates the cell–cell signaling in multicellular systems. For a swarm robotic

system, this entails that each robot is able to detect the distance to its neighboring robots, which is practical and easy to realize.

2) Morphogen Gradients for Target Shape Representation: In biological morphogenesis, morphogen concentration gradients control cell fate specification and play a key role in understanding pattern formation [5]. 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 (7). For 2-D target shape, $f(z_i)$ can be defined as follows:

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

$$f(z_{i,y}) = \frac{1 - e^{-z_{i,y}}}{1 + e^{-z_{i,y}}} \qquad (11)$$

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

$$z_{i,x} = \frac{\partial h}{\partial g_{i,x}} \quad z_{i,y} = \frac{\partial h}{\partial g_{i,y}} \qquad (12)$$

where h defines the shape the robots should form.

The aforementioned 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 Bézier, B-Spline, and nonuniform rational B-Spline (NURBS) can be used.

The NURBS [75] 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 multirobot shape formation. First, it provides a common mathematical form 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 parameter in the NURBS curve is 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 $B_{j,q}(u)$, where i corresponds to the jth control point, and q 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 $\mathbf{p}_j$ and the associated weights w_j as follows:

$$\mathbf{c}(u) = \frac{\sum_{j=1}^{N} \mathbf{p}_i w_j B_{j,q}(u)}{\sum_{j=1}^{N} w_j B_{j,q}(u)} \qquad (13)$$

where N is the number of control points, u is the parametric variable, and $B_{j,q}(u)$ are B-spline basis functions.

With the NURBS model for shape description, complex shapes can be formed, refer to Section III-C for examples.
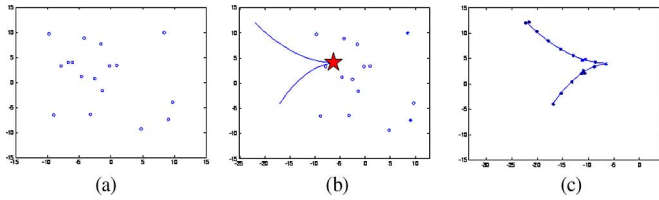
Fig. 7. Snapshots showing the emergence of a pattern from 17 robots similar to bird flocking [31]. (a) Random initialization; (b) determination of a reference robot (denoted by a star) through competition; (c) emergence of the target shape.
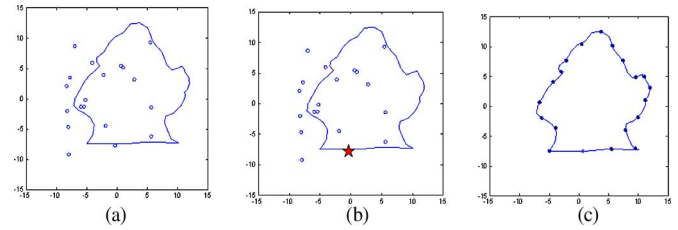

Fig. 8. Snapshots of 20 robots covering a boundary simulating that of the Brooklyn Borough of New York City [31]. (a) Random initialization; (b) robot that first detects the boundary is chosen as the reference robot; (c) coverage of the boundary.
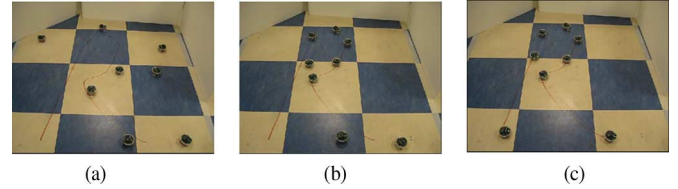

Fig. 9. Snapshots of 8 E-Puck robots forming the letter 'R' [61]. (a) $t = 0$ s. (a) $t = 6$ s. (a) $t = 11$ s.

C. Illustrative Results on GRN-Based Swarm Robot Self-Organization

In the experiments, the parameters in (6) and (7) need to be determined. A straightforward way is to define the parameters heuristically with the following condition being satisfied so that the states of the dynamic system can converge to the target shape:

$$mk \leq ac, a, c, k, m > 0. \qquad (14)$$

The proof of the convergence is provided in [32]. In addition, b should also be larger than zero so that collisions between the robots, possibly between a robot and an obstacle, can be avoided.

A more desired approach to setting up these parameters is to find a set of optimal parameters, using an evolutionary algorithm by minimizing the time and/or the total travel distance for all robots to converge to the target shape, as done in [32].

If NURBS is used for representing the target shape, parameter u in (13) should be determined for each robot. During the self-organization process, each robot will randomly pick a value from $\{0, 1/n, 2/n, \ldots, 1\}$, where n is the number of robots in the system, which is either assumed to be known, or can be estimated as described in [31]. It is therefore possible that different robots pick the same u value in the beginning and thus will compete the same point on the target shape. In this case, robots arrive later will try another u value until it converges to a point on the target shape where no other robot exists.

Simulation results where 17 robots are used to form a bird-flocking shape are given in Fig. 7. The robots are randomly distributed in the area in the beginning. A reference robot is chosen through a competition process, during which the robot that has the maximum number of neighbors wins. Driven by the GRN-based dynamics, the robots will then autonomously form the target shape. Snapshots showing 20 robots covering a boundary simulating that of the Brooklyn Borough of New York City are provided in Fig. 8. A proof-of-concept experiment has also been performed using E-Puck robots. Fig. 9 shows a few snapshots of 8 E-Puck robots converging to a capital letter "R," refer to [61] for details.

One important concern in designing decentralized algorithms for self-organizing swarm robots is their robustness in the presence of disturbances in the system as well as in the environment. Extensive empirical results show that the morphogenetic self-organization algorithm is robust to changes in the number of robots, insensitive to noise in the model parameters sensory measurements, and adaptable to environmental changes such as moving obstacles [39].

D. Intermediate Summary

Compared to existing approaches [36], the morphogenetic approach to swarm robotic systems has the following advantages. First, the global behavior, i.e., the target shape in the context of pattern formation, can be embedded in the robot dynamics in the form of morphogen gradients. In pattern formation, the global shape can be described using parametrized models such as a NURBS model that can represent both analytical and free-form shapes. The GRN model can then generate implicit local interaction rules automatically to generate the global behavior, which can be guaranteed through a rigorous mathematical proof. Second, the morphogenetic approach is robust to perturbations in the system and in the environment. Third, it has also shown that the morphogenetic approach can provide a unified framework for multirobot shape formation and boundary coverage [31], since the representation of the target shape is independent of a specific global coordination system. Morphogenetic approaches to self-organization of collective systems can potentially be applied to solving other engineering problems such as the topology self-reconfiguration of communication networks [54].

IV. MORPHOGENETIC MODULAR ROBOTS FOR SELF-ORGANIZED RECONFIGURATION

A. Reconfigurable Modular Robots

Self-reconfigurable modular robots consist of a number of modules and are able to adapt their shape (configuration) by rearranging their modules to changing environments [68]. Each module is a physical or simulated 'body' containing a controller. Both physical modular robots, such as M-TRAN [67] and Molecube [69], and simulated 'animats,' such as Karl Sims' virtual
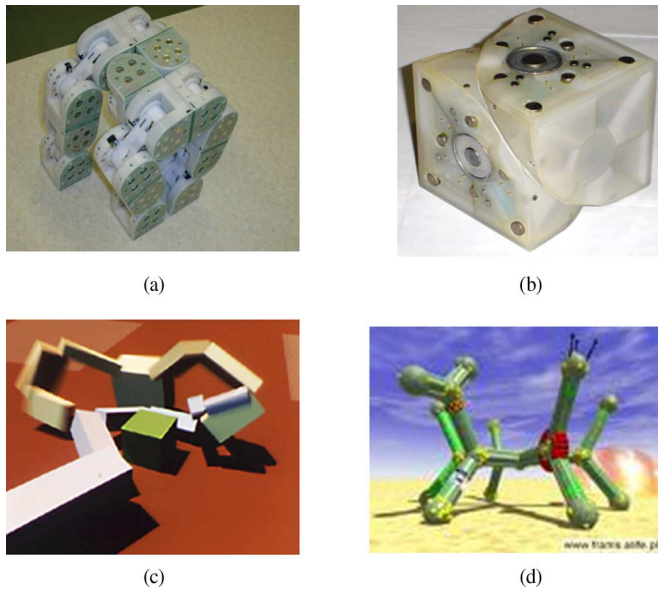
Fig. 10. Examples of physical and simulated modular robots. (a) M-TRAN, (b) molecube, (c) Karl Sims' virtual creature, and (d) Framsticks.
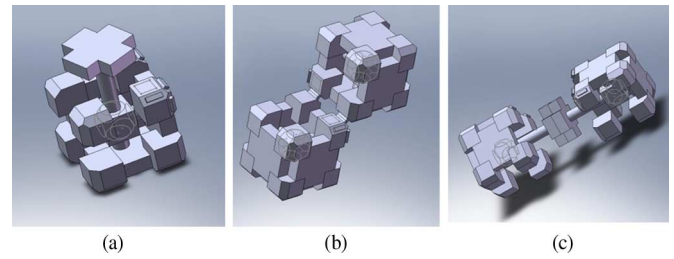

Fig. 11. Mechanical demonstration of CrossCube [63]. (a) Joints; (b) Locks on the boundaries of the modules. (c) Rotation and extension of the joints of the modules.

creature [87] and Framsticks [49] have been constructed for reconfigurable robotic systems, refer to Fig. 10. 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 cubic mechanism, 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 or detached to 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 believed to be the most important feature of self-reconfigurable robots, the ability to adapt their configuration autonomously under environmental changes remains to be demonstrated.

B. CrossCube—A Simulated Modular Robot

CrossCube [63] adopts a lattice-based cube design. Each module in CrossCube is a cubical structure having its own computing and communication resource and actuation capability. Like all modular robots, the connection part of the modules can easily be attached or detached to 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. 11(a). 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 return to its default direction.

The cross-concaves on each side of the shell restrict the movement trajectory of the joints, as shown in Fig. 11(a). The borders of each module can actively be locked or unlocked with the borders of other modules, as shown in Fig. 11(b). The length and angle of the lock mechanism can also be adjusted on the boarders of the modules.

Basic motions of modules in CrossCube include rotation, climbing, and parallel motion. Fig. 11 (c) illustrates a rotation movement of two modules. Parallel motion 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 will 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 motion allow a module of CrossCube to move to any position within the modular robot as long as the modules are connected.

C. Self-Reconfiguration as Morphogenesis

The connection between reconfigurable modular robots and multicellular organisms appears more straightforward. Each unit in modular robots can be seen as a cell, and there are similarities in control, communication, and physical interactions between cells in multicellular organisms and modules in modular robots. For example, control in both modular robots and multicellular organisms is decentralized. In addition, global behaviors of both modular robots and multicellular 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 multicellular organisms. Therefore, it is a natural idea to develop control algorithms for self-reconfigurable modular robots, using biological morphogenetic mechanisms [63], [100]. In the following, we describe briefly 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, through which local communications between the modules can be established. The concentration of the diffused proteins decays over time. The target configuration of the modular robot is also defined by morphogen
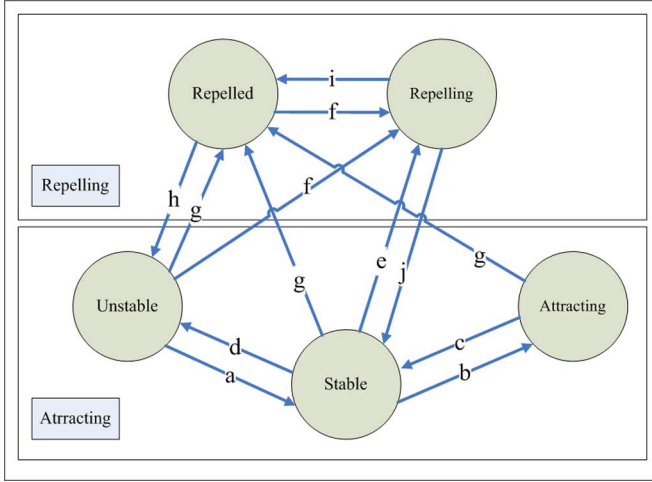
Fig. 12.   State transition of each module in CrossCube [63].

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

Different from the morphogenetic swarm robotic system described in Section III, in which the target shape is fully defined by a kind of maternal morphogen, each unit in the morphogenetic modular robot system can modify the morphogen gradients by secreting either positive or negative morphogen gradients, which is indispensable 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 repellent behaviors 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 states, namely, stable, unstable, attracting, repellent, and repelled. The transition relationships between the five states of modules are given in Fig. 12. Refer to [63] for details of state transitions.

1) GRN-Based Pattern Transition: The state transitions are controlled by a GRN model having two gene–protein pairs, an attracting gene–protein pair ($G_A - P_A$) and a repellent gene–protein pair ($G_P - P_P$). We assume that the repellent states always have a higher priority than the attracting states. As a result, all the states triggered by attracting behaviors can be overwritten by the states triggered by repellent behaviors. The reason for this is that a grid having a repellent (negative) morphogen gradient should be kept empty as long as migration modules is still in need during reconfiguration.

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. 12. At the initial stage of shape configuration, all modules are set as unstable. After they are initialized with the target configuration, modules located in the grids with an attracting morphogen gra-

dient 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 that has an attracting morphogen gradient. These grids become attracting to attract unstable modules to occupy them. Here, P_A is defined as

$$P_A^{ij} = \{AP^{ij}, S^i, C_A^{ij}\} \tag{15}$$

where P_A^{ij} is the attracting protein generated by the ith module for its jth neighbor. AP^{ij} is the jth neighboring attracting grid of the ith module. S^i is the identification label of the ith module, and C_A^{ij} is the concentration of the protein P_A^{ij}, which equals 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 G_A of neighbors as well.

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

$$\frac{dg_A(t)}{dt} = -ag_A(t) + b\sum p_{A\_local} - c\sum p_{A\_rec} \tag{16}$$

where $g_A(t)$ is the gene expression level of G_A at time t. p_{A_local} and p_{A_rec} are protein concentrations of locally generated protein and received protein from other modules, respectively. a, b, and c are constant coefficients, which can be determined, e.g., using an evolutionary algorithm.

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

$$\text{state} = \begin{cases} \text{unstable when } g_A < G_{A\_L} \\ \text{stable when } G_{A\_L} < g_A < G_{A\_H} \\ \text{attracting when } g_A > G_{A\_H} \end{cases} \tag{17}$$

where G_{A_L} is a negative threshold and G_{A_H} is a positive threshold. According to (16), g_A falls below a negative threshold G_{A_L} with the increase of $c\sum p_{A\_rec}$. A higher value of $c\sum p_{A\_rec}$ means that there are some more important grids to be filled in. 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.

The expression level of g_A will be enhanced when $b\sum p_{A\_local}$ increases, which means that the module has some 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 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 by the GRN-based model described in (16) and (17). More specifically, P_A can regulate G_A through (16), while G_A

can determine when P_A is allowed to diffuse into neighboring grids based on (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) Gene-Protein Pair for Repelling: The repellent states are controlled by the repellent gene–protein pair ($G_P - P_P$). The repelling modules produce P_P, which is defined as

$$P_P^{ij} = \{RP^{ij}, S^i, C_P^{ij}\} \tag{18}$$

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

$$\frac{dg_P(t)}{dt} = dg_P(t) - e\sum p_{P\_\text{rec}}$$
$$\text{state} = \text{repelled when } g_P < G_{P\_L} \tag{19}$$

where $g_p(t)$ is the gene expression level of the repellent gene G_P at time t, p_{P_rec} is the concentration of the received repellent 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 to be repelled.

To summarize, P_P can regulate G_P through (19). G_P can produce P_P under the condition that G_P is below G_{P_L} and the module is blocked.

4) Lookup Table-Based Configuration Representation: 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 I
VEHICLE CONFIGURATION

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)	

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 I. In the table, x, y, and z are 3-D coordinates of grid positions, ML denotes morphogen level and PID stands for position identification. Additionally, we define some joints' behaviors to enable the vehicle to move, once the configuration is completed. Joints can be identified by its PID and RD means joint rotate direction.

D. Illustrative Examples of GRN-Based Self-Reconfiguration

This section describes briefly a case study on using GRN-based controller to coordinate CrossCube modules for a locomotion task, in which the modular robot needs to traverse through different environmental fields. A software is developed to simulate the behaviors and interaction of CrossCube in a physical world using C++ and the Physx engine from nVidia. In the following experiment, the parameters of the GRN models are setup 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 a new grid.

Before showing the self-reconfiguration ability of the system in a changing environment, we first perform a simple experiment to verify the effectiveness of the model. The modular robot has a "block" configuration consisting of 16 modules, which should convert into a vehicle-like pattern defined in terms of morphogen level, as shown in Table I. A set of snapshots is provided in Fig. 13 to show a few intermediate configurations toward the vehicle configuration realized by the GRN-based model.

To verify the model's ability to reconfigure the modular robot to adapt to different environments, a simulation has been performed where a vehicle needs first to move through a narrow passage. Then, the robot must climb up a step to move forward. The environment is defined using the size of the modular robots as a unit: the wider passage is seven units in width, while the narrower passage is of 5.5 unit in width. The height of the step is one unit. The reconfiguration is triggered when any of the modules in the front detect obstacles. If all front modules detect obstacles, a climbing reconfiguration process will be activated. A number of snapshots showing the reconfiguration processes
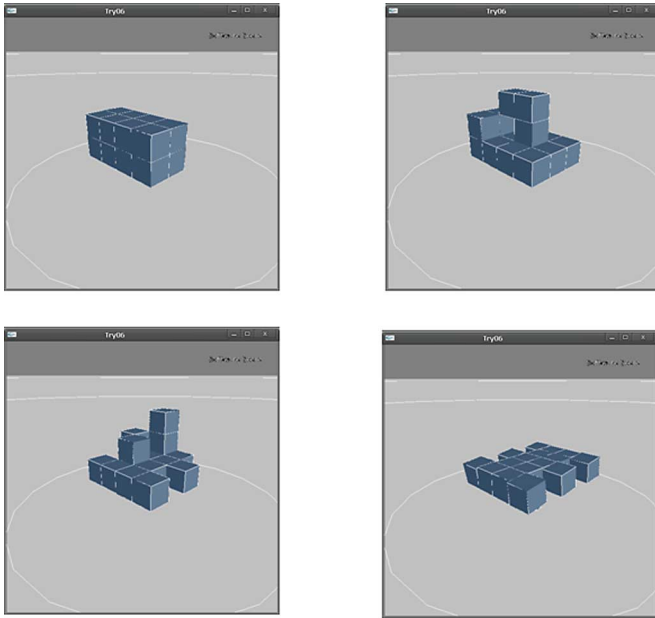
Fig. 13. Autonomous configuration of a vehicle from a rectangle based on the GRN model [63].
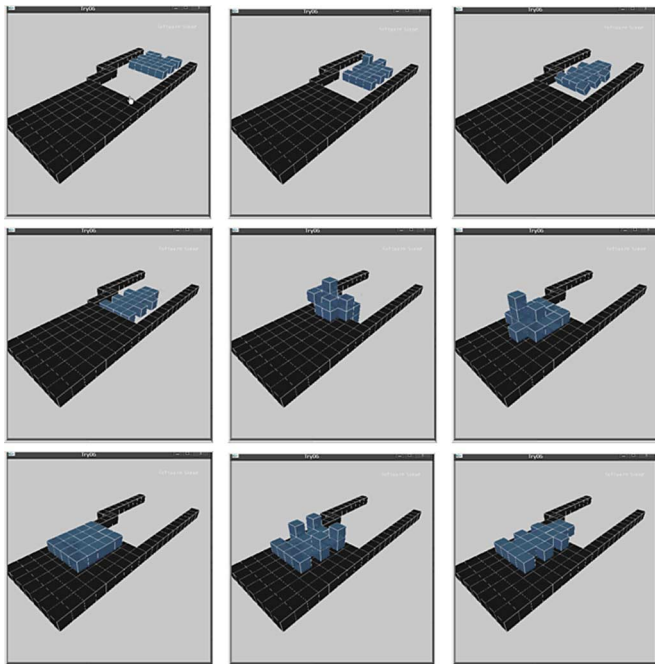

Fig. 14. Set of snapshots demonstrating a series of reconfigurable processes during locomotion and climbing. The robot first adapted its width to the narrow passage, then changed its configuration for climbing up a step, and finally reconfigured itself into a vehicle again to move forward.

are given in Fig. 14. More details of the GRN-based modular robot system can be found in [63].

E. Intermediate Summary

The GRN model described earlier represents a hierarchical approach to self-reconfiguration of modular robots, where one layer defines the desired configuration of the modular robots while the other layer organizes the modules autonomously to achieve the desired configuration. 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 gene regulatory networks [21]. In response to the environment changes, the layer for defining the robot configuration is able to adapt the target configuration, based on which the second layer can reorganize the modules autonomously to realize the target configuration.

V. MORPHOGENETIC BRAIN AND BODY DESIGN FOR INTELLIGENT ROBOTS

A. Why Development?

The role of neural and morphological development in designing intelligent robots has largely been neglected in evolutionary robotics [56], although coevolution of brain and body has long been recognized both in robotics [76] and artificial life [92]. This situation has not been changed a lot to date due to various difficulties in coevolving the development of body and brain. First, there are a lack of knowledge about the developmental mechanisms in biology and a lack of physically realistic environment [74]. Second, the influence of artificial development on the systems performance is not well understood. Although it is believed that the developmental mechanism offers the possibility to evolve complex systems, the performance advantage of such developmental systems over nondevelopmental ones remains unclear. Finally, necessary hardware, which is of particular importance in robotics, such as growing materials, adaptable structures, adaptable sensors, and actuators are still lacking.

Nevertheless, the role of development in brain–body coevolution cannot be overestimated, simply because in natural evolution, development is an indispensable phase in which organisms have to interact with the environment constantly and find a way to survive. It has been found that development can bias the evolutionary path considerably, as illustrated in Fig. 15. In addition, it has been surmised that development can also open up new opportunities for evolution [4], [78], which has partly been verified in computational developmental systems [42].

B. Computational Models for Neural and Morphological Development

Various computational models have been suggested for neural and morphological development [34], [89]. A large body of research on modeling the growth of nervous systems was based on grammatical re-writing rules such as L-systems [10], [13], [44] and grammar trees [30], [48]. In [17], Kauffmann's Boolean network was used for modeling the structural development of dynamic neural networks. More low-level models that consider cell–cell chemical interactions through reaction and diffusion of morphogens have also been employed for modeling neurogenesis [18], [23], [45], [65], [70]. A recurrent artificial neural network was used for modeling the development of a spiking neural network for the control of a Khepera robot [22]. In [24], the matrix rewriting scheme suggested in [44] was applied to modeling neural morphogenesis, which was coevolved
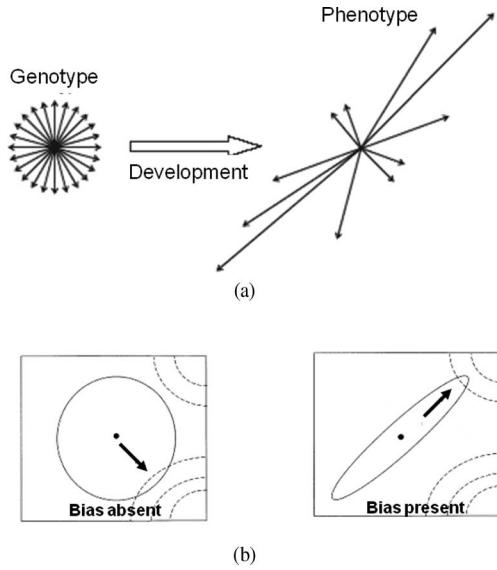
Fig. 15. Influence of development on selection directions. (a) Developmental bias due to nonlinear genotype–phenotype mapping (taken from [77]) and (b) change of selection directions under developmental bias (taken from [4]).
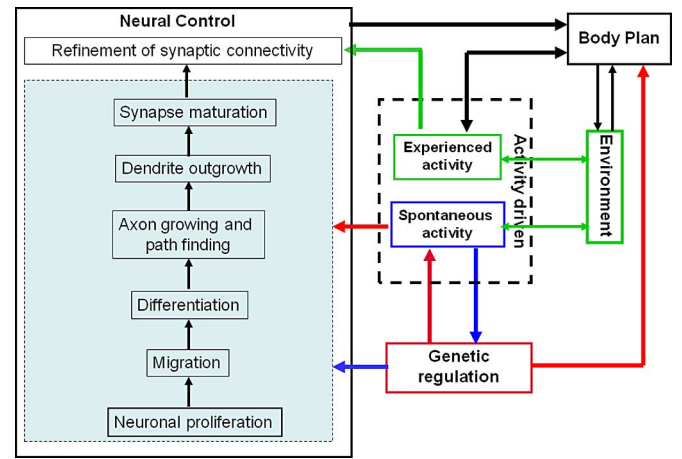

Fig. 16. Main processes in neural development driven by genetic and environmental control situated in a physical environment together with the development of the body plan.

RU⁺	RU⁻	SU^D	SU^M	RU⁺	RU⁺	RU⁻	SU^A	SU^TF	RU⁺	SU^TF	SU^D

RU⁺: Activating regulatory unit   RU⁻: Inhibitory regulatory unit

SU^D: Cell division   SU^M: Cell migration

SU^A: Axon growth   SU^TF: Producing transcription factor

Fig. 17. Example of chromosome for neural development.

with neural plasticity rules for controlling a mobile robot. A low-level GRN model with chemical diffusion was adopted for evolving neurogenesis for a hydra-like animats [41]. The weights of the developed spiking neural networks were then further adapted using an evolution strategy for generating food-catching behavior.

Similar computational models have also been suggested for simulating morphological development. The main differences may lie in the following aspects compared to those for neural development. First, increasing attention has been paid to 3-D models [19], [33], [90], [98], which plays an important role in modeling morphological development. Second, physical cellular interactions are also modeled in addition to the genetic regulatory mechanisms, such as adhesion repulsion between the cells. Both sphere models [91] and spring-mass-damper models [82] have been used.

Finally, efforts have also been made to simulate both neural and morphological development, using the same developmental model, though most of them use a high-level developmental model, such as the L-system [49], [87]. A Boolean genetic regulatory network has been used for modeling both morphological parts such as sensor and actuator, and control part such as control-neurons at a very high abstraction level [2].

So far, most neural networks generated by a developmental model have very limited functionalities. In addition, most developmental models of neural networks take only genetic mechanisms, i.e., activity-independent development, into account. For the neural network to function, activity-dependent development, which is responsible for synapse refinement, is essential. The main processes in neural development is summarized in Fig. 16. From the figure, we can see that early development of nervous systems, such as neuron axon growth, dendrite outgrowth, and synapse maturation, which was usually thought to be genetically
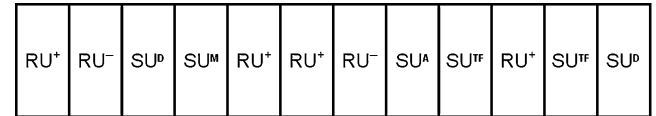
regulated, are also considerably driven by neural activity [83], [88], [93].

C. GRN Model for Neural and Morphological Development

The growth of the animat morphology is under the control of GRNs and cellular physical interactions. Extended from the cellular growth model for structural design, GRN models for the development of a nervous system [41] and body plan [82] of primitive animals have been proposed. In the genome of the GRN models, each gene consists of a number of structural units (SUs) proceeded by a number of regulatory units (RUs). RUs can be activating (RU^+) or repressive (RU^-). When SUs are activated, they will produce proteins either responsible for cellular behaviors such as cell division, cell death, cell migration, and axon growth, or proteins regulating the activation of the structural units, which are also known as 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 out of the cell and regulate the genes of other cells, it is termed as 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. 17. From the figure, we note that single or multiple RUs may regulate the expression of a single or multiple SUs.

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 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 ith TF and jth RU is defined by

$$\gamma_{i,j} = \max\left(\epsilon - \left|\mathrm{aff}_i^{\mathrm{TF}} - \mathrm{aff}_j^{\mathrm{RU}}\right|, 0\right). \tag{20}$$

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

Thus, the activation level contributed by this RU (denoted by $a_j, j = 1, \ldots, N$) 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 kth gene, that is regulated by N RUs, can be defined by

$$\alpha_k = 100 \sum_{j=1}^{N} h_j a_j (2s_j - 1) \tag{21}$$

where $s_j \in (0, 1)$ denotes the sign (positive for activating and negative for repressive) of the jth RU and h_j is a parameter, representing the strength of the jth RU. If $\alpha_k > 0$, then the kth 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:

$$A = \beta \frac{2}{1 + e^{-20 \cdot f \cdot \alpha}} - 1 \tag{22}$$

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

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 ($p^{\mathrm{ex}} \in (0, 1)$) is also encoded in the corresponding gene. Thus, $p^{\mathrm{ex}} A$ is the amount of external TF and $(1 - p^{\mathrm{ex}})A$ 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 into a number of grids. External TFs are put on four grid points around the center of the cell, which undergoes first a diffusion (23) and then decay process (Eqn. 24):

$$\mathbf{u}_i(t) = \mathbf{u}_i(t-1) + 0.1 D_i^f (\mathbf{G} \cdot \mathbf{u}_i(t-1)) \tag{23}$$

$$\mathbf{u}_i(t) = \min\left((1 - 0.1 D_i^c)\, \mathbf{u}_i(t), 1\right) \tag{24}$$

where $\mathbf{u}_i$ is a vector of the concentrations of the ith 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 time step.

The aforementioned cellular model has been applied to simulate both morphological and neural development [41], [82]. In the experiment to generate an animat like *C. elegans*, two predifused, external TFs without decay and diffusion 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. In the experiments, the GRN model is initialized randomly, and the target of the evolution is to evolve an elongated animate, whose morphology is defined by a rectangular shape. Without any hard constraints for stopping cell division,
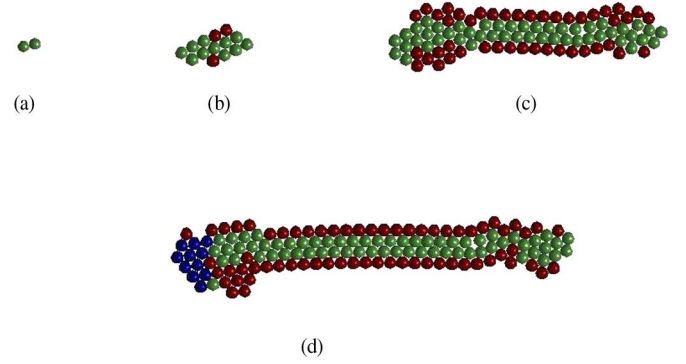

(a)   (b)   (c)

(d)

Fig. 18. Self-stabilized cellular growth under the control of a GRN model presented in [82]. (a) System is initialized with two cells. (b)–(c) System grows as cells divide. (d) Growth is self-stabilized dynamically, where the cells on light grey color will divide and those in dark will die. A balance of cell division and cell death is accomplished under the control of the GRN.

we are able to evolve a GRN that results in self-stabilized cellular growth [82]. A few snapshots of the self-stabilized cellular growth process is provided in Fig. 18, where the cellular system starts from two cells sitting in the middle of the simulation area and reaches a dynamic stability at the end of the development. In the figure, cells in light color are going to divide in the next developmental step, and those in dark are going to die in the next step. In another experiment, we use a similar GRN model for simulating neural growth in hydra. In the beginning, a few simulated stem cells are randomly distributed in the body plan of the hydra-like animat. Then, cells divide, migrate, and axons grow so that the neurons are connected [41]. Snapshots showing this growth process are given in Fig. 19. The evolved GRN resulting in the neural development in Fig. 19 is presented in Fig. 20, which is able to generate the correct temporal activation sequence for cell division, cell migration, and axon growth.

D. Conceptual Framework for Coevolving the Development of Robot Hand Morphology and Controller

It has been found that the morphology of the animal hands has changed a lot to adapt to the needs of the animals during evolution. Fig. 21 (a) shows a few examples of primate hands. From the figure, we can see that they distinguish themselves in both shape and length in the finger segments. Besides, it has been hypothesized that particular behaviors, such as throwing and clubbing, have played a key role in differences between a hand of human beings and that of a chimpanzee [99], refer to Fig. 21 (b).

The importance of coevolving the development of hand morphology and control in robotics is twofold. On the one hand, object grasping and manipulation with a robot hand is in itself a challenging task in that such systems are usually highly redundant. Existing work focuses on the design of the hand controller for a given morphology [97], which is inefficient when the shape of the objects changes considerably. A better approach is to codesign the hand morphology and control in a developmental manner, as illustrated in Fig. 22. In this way, the shape
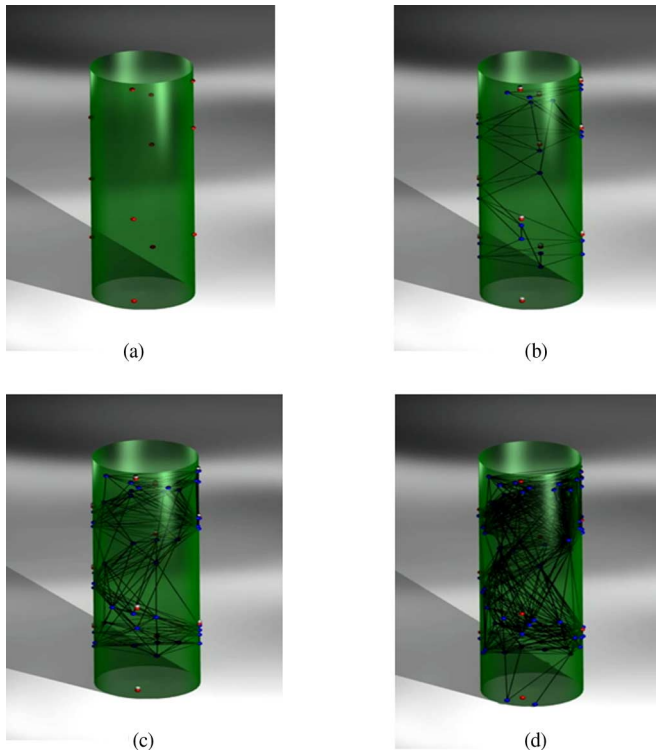
(a)

(b)

(c)

(d)

Fig. 19. Development of a nervous system using the GRN model in [41]. (a) A few stem cells are randomly distributed on a hydra-like body wall. (b)–(d) Cells divide, migrate, and axons grow. As the development goes on more connections are built up.
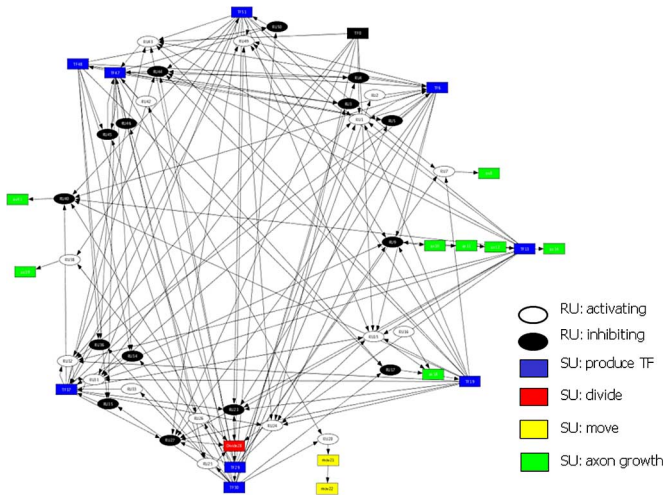

Fig. 20. Evolved GRN resulting in the neural development in Fig. 19. SU: structural genetic units; RU: regulatory genetic units.
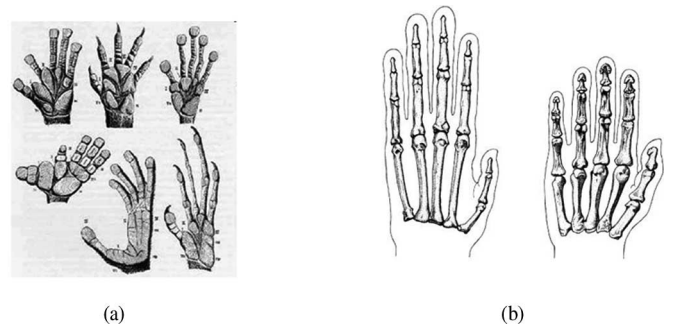

(a)

(b)

Fig. 21. (a) Examples of primate hands. (b) Differences between a human hand and a chimpanzee hand.
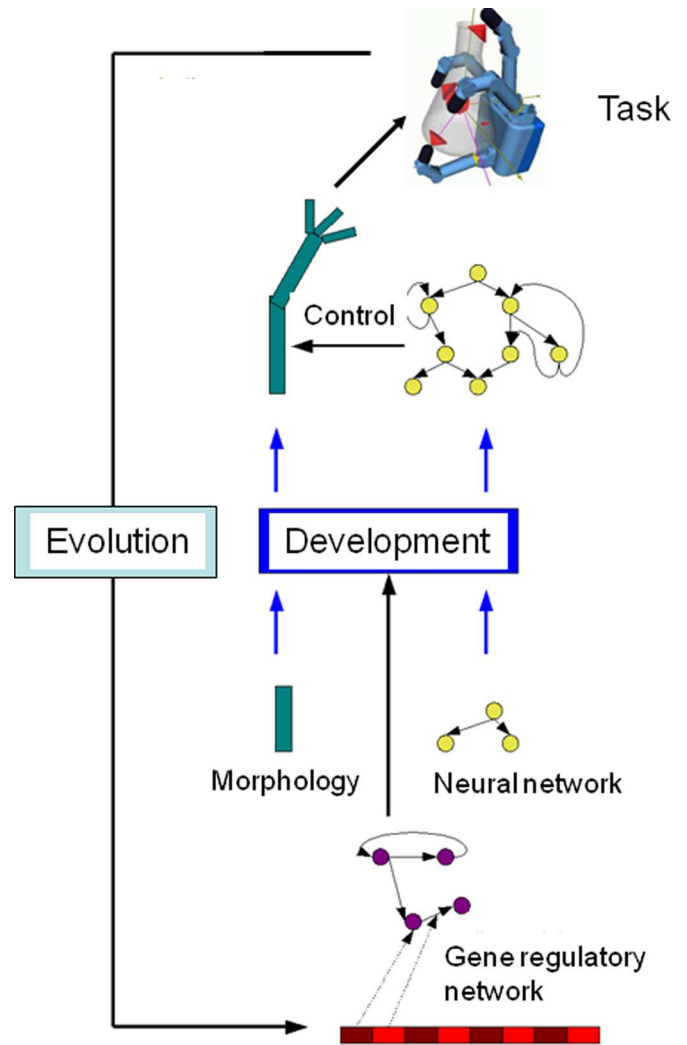
Fig. 22. Conceptual diagram for coevolving the development of hand/arm and control. Adapted from [37].

and number of finger segments, the number fingers, and even the number of arms can be evolved together with their controller.

Meanwhile, coevolution of the hand morphology and control in a computational environment provides us a means for understanding the phylogenetic changes in evolution of animal hands. So far, brain–body coevolution in computational environments has led to findings regarding the organizational principles of nervous systems and the emergence of bilateral symmetry in neural configuration [43], [71]. We expect that different hand morphologies will emerge by evolving the system for different behaviors.

VI. CONCLUDING REMARKS

This paper suggests a new field of robotics termed *morphogenetic robotics*, which focuses on employing genetic and

cellular mechanisms in biological morphogenesis for developing 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. Morphogenetic robotics, as epigenetic robotics, is a part of developmental robotics. While epigenetic robotics concentrates on the mental development of robotic systems, morphogenetic robotics focuses on the physical development of the body plan and nervous system of the robots. Therefore, we believe that developmental robotics should include both morphogenetic robotics and epigenetic robotics.

Research on morphogenetic robotics is still in its infancy and therefore many issues remain to be explored. First, many genetic and cellular mechanisms underlying biological morphogenesis still remain elusive, and much work needs to be done on reconstruction of spatiotemporal gene expression patterns using computational models based on biological data. In particular, the self-adaptation capability of the genetic and cellular models to environmental changes used in morphogenetic robotics needs to be improved. The introduction of hierarchical gene regulatory models suggests a promising step toward this goal, but many details on autonomous self-adaptation based on sensory input are still unclear. Second, the interactions between morphogenetic robotics and epigenetic robotics are largely unexplored. Obviously, the physical and mental development are closely coupled, since neural and morphological development lay the neurophysiological foundation for cognitive and mental development, and both are constrained by the environment in which the robots reside. Furthermore, research on developmental robotics should also be performed taking evolution into account, as development can not only bias the direction of evolution, but also enhance evolvability [42]. Finally, morphogenetic robotics is currently very much limited to computational simulations. Appropriate hardware for morphogenetic robotics, including programmable materials [3], [47], and adaptable sensors and actuators, is to be studied.

REFERENCES

[1] U. Alon, "Network motifs: Theory and experimental approaches," *Nature Rev. Genetics*, vol. 8, pp. 450–461, 2007.

[2] F. Dellaert and R. D. Beer, "Toward an evolvable model of development for autonomous agent synthesis," in *Proc. Artif. Life IV*, 1994, pp.246–257.

[3] J. Amend, Jr. and H. Lipson, "Shape-shifting materials for programmable structures," presented at the 11th Int. Conf. Ubiquitous Computing: Workshop on Architectural Robotics (Ubicomp Archibots 2009), Orlando, FL.

[4] W. Arthur, "The effect of development on the direction of selection: Toward twenty-first century consensus," *Evol. Dev.*, vol. 10, pp. 375–390, 2004.

[5] H. Ashe and J. Briscoe, "The interpretation of morphogen gradients," *Development*, vol. 133, pp. 385–394, 2007.

[6] L. Beloussov, "Integrating self-organization theory into an advanced course on morphogenesis at Moscow State University," *Int. J. Dev. Biol.*, vol. 47, pp. 177–181, 2003.

[7] H. Ben-Amor, S. Cadau, A. Elena, D. Dhouailly, and J. Demongeot, "Regulatory networks analysis: Robustness in morphogenesis," in *Proc. 2009 Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Bradford, U.K., 2009, pp. 924–928.

[8] A. Bhattacharyya, "Morphogenesis as an amorphous computing," in *Proc. 3rd Conf. on Comput. Frontiers*, 2006, pp. 53–64.

[9] C. Bishop, D. Erezyilmaz, T. Flatt, C. Georgiou, M. Hadfield, A. Heyland, J. Hodin, M. Jacobs, S. Maslakova, A. Pires, A. Reitzel, S. Santagata, K. Tanaka, and J. Youson, "What is metamophosis?," *Integr. Comp. Biol.*, vol. 46, pp. 655–661, 2006.

[10] E. Boers, "Biological metaphors and design of modular artificial neural networks," Master's thesis, Dept. Comput. Sci. Exp. Psychol., Leiden Univ., The Netherlands, 1992.

[11] J. Bower and H. Bolouri, *Computational Modeling of Genetic and Biochemical Networks*. Cambridge, MA: MIT Press, 2001.

[12] L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Syst., Man, Cybern. C: Appl. Rev.*, vol. 38, no. 2, pp. 156–182, Mar. 2008.

[13] A. Cangelosi, S. Nolfi, and D. Parrisi, "Cell division and migration in a ' for neural networks," *Netw.—Comput. Neural Syst.*, vol. 5, pp. 479–515, 1994.

[14] A. Crumiere and M. Sablik, "Positive circuits and d-dimensional spatial differentiation: Application to the formation of sense organs in *Drosophila*," *BioSystems*, vol. 94, pp. 102–108, 2008.

[15] H. DeJong, "Modeling and simulation of genetic regulatory systems: A literature review," *J. Comput. Biol.*, vol. 9, pp. 67–103, 2002.

[16] R. Doursat, "Programmable architectures that are complex and self-organized: From morphogenesis to engineering," in *Proc. 11th Int. Conf. Simul. Synthesis of Living Systems (ALIFE XI)*, pp., 181–188.

[17] I. East and J. Rowe, "Abstract genetic representation of dynamical neural networks using kauffman networks," *Artif. Life*, vol. 3, pp. 67–80, 1997.

[18] P. Eggenberger, "Cell interactions as a control tool of developmental processes for evolutionary robotics," in *Proc. Anim. Animats*, 1996, pp. 440–448.

[19] P. Eggenberger, "Evolving morphologies of simulated 3d organisms based on differential gene expression," in *Proc. 4th Eur. Conf. Artif. Life*, 1997, pp. 205–213.

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

[21] D. Erwin and E. Davidson, "The evolution of hierarchical gene regulatory networks," *Nature Rev. Genet.*, vol. 10, no. 2, pp. 141–148, 2009.

[22] D. Federici, "Evolving developing spiking neural networks," in *Proc. Congr. Evol. Comput.*, 2005, vol. 1, pp. 543–550.

[23] K. Fleischer and A. Barr, "A simulation testbed for the study of multicellular development: The multiple mechanisms of morphogenesis," in *Proc. Artif. Life III*, 1993, pp. 389–416.

[24] D. Floreano and J. Urzelai, "Neural morphogenesis, synaptic plasticity, and evolution," *Theory Biosci.*, vol. 120, no. 3–4, pp. 225–240, 2001.

[25] F. Francois and H. Vincent, "Design of genetic networks with specified functions by evolution *in silico*," *Proc. Natl. Acad. Sci. USA*, vol. 101, pp. 580–585, 2004.

[26] N. Geard and J. Wiles, "Structure and dynamics of a gene regulatory network model incorporating small RNAs," in *Congr. Evol. Comput.*, 2003, pp.199–206.

[27] S. Gilbert, *Developmental Biology*. Sunderland, MA: Sinauer Associates, 2003.

[28] S. Gilbert,"The morphogenesis of evolutionary developmental biology," *Int. J. Dev. Biol.*, vol. 47, pp. 467–477, 2003.

[29] A. Goncalves and E. Costa, "A computational model for gene regulatory networks and its topological properties," in *Proc. Artif. Life XI*, 2008, pp.204–211.

[30] F. Gruau, "Genetic synthesis of modular neural networks," in *Proc. Int. Conf. Neural Netw*, 1993, pp. 318–325.

[31] H. Guo, Y. Jin, and Y. Meng, "A unified framework for self-organized multi-robot pattern formation and boundary coverage inspired from morphogenesis," *ACM Trans. Auton. Adaptive Syst.*, to be published.

[32] H. Guo, Y. Meng, and Y. Jin, "A cellular mechanism for multi-robot construction via evolutionary multi-objective optimization of a gene regulatory network," *BioSystems*, vol. 98, no. 3, pp. 193–203, 2009.

[33] P. Haddow and J. Hoye, "Achieving a simple developmental model for 3d shapes: Are chemicals necessary?," in *Proc. GECCO 2007*, pp. 1013–1020.

[34] S. Harding and W. Banzhaf, "Artificial development," in *Organic Computing*, R. Würtz, Ed. New York: Springer-Verlag, 2008, pp. 202–219.

[35] G. Hornby and J. Pollack, "Creating high-level components with a generative representation for body-brain evolution," *Artif. Life*, vol. 8, pp. 223–246, 2002.

[36] M. Hsieh and V. Kumar, "Pattern generation with multiple robots," in *Proc. Int. Conf. Robot. Autom*, 2006, pp. 2442–2447.

[37] B. Inden, "Co-evolution of neural and morphological development for grasping in changing environments," presented at the EL-TEC Meeting, Honda Research Institute Europe, Offenbach, Germany, Jun. 2009.

[38] Y. Jin, R. Gruna, I. Paenke, and B. Sendhoff, "Multi-objective optimization of robustness and innovation in redundant genetic representations," in *Proc. IEEE Symp. Multi-Criteria Decision-Making*, 2009, pp. 38–45.

[39] Y. Jin, H. Guo, and Y. Meng, "Robustness analysis and failure recovery of a bio-inspired self-organizing multi-robot system," in *Proc. Third IEEE Int. Conf. Self-Adaptive Self-organizing Systems*, 2009, pp. 154–164.

[40] Y. Jin and Y. Meng, "Emergence of robust regulatory motifs from *in silico* evolution of sustained oscillation," *BioSystems*, to be published.

[41] Y. Jin, L. Schramm, and B. Sendhoff, "A gene regulatory model for the development of primitive nervous systems." in *Proc. INNS-NNN Symposia Model. Brain Nerv. Syst.* (ser. LNCS 5506), 2009, pp. 48–55.

[42] Y. Jin and J. Trommler, "A fitness-independent evolvability measure for evolutionary developmental systems," in *Proc. IEEE Symp. Comput. Intell. Bioinf. Comput. Biol.*, 2010, pp. 69–76.

[43] B. Jones, Y. Jin, B. Sendhoff, and X. Yao, "Evolving functional symmetry in a three dimensional model of an elongated organism," in *Proc. Artif. Life XI*, 2008, pp. 305–312.

[44] H. Kitano, "Designing neural networks using genetic algorithms with graph generation system," *Complex Syst.*, vol. 4, pp. 461–476, 1990.

[45] H. Kitano, "A simple model of neurogenesis and cell differentiation based on evolutionary large-scale chaos," *Artif. Life*, vol. 2, no. 1, pp. 79–99, 1995.

[46] J. F. Knabe, C. L. Nehaniv, and M. J. Schilstra, "Genetic regulatory network models of biological clocks: Evolutionary history matters," *Artif. Life*, vol. 14, pp. 135–148, 2008.

[47] A. Knainan, "Design of programmable matter," Master's thesis, MIT, Boston, MA, 2008.

[48] J. Kodjabachian and J.-A. Meyer, "Evolution and development of neural controllers for locomotion, gradient-following, and obstacle-avoidance in artificial insects," *IEEE Trans. Neural Netw.*, vol. 9, no. 5, pp. 796–812, Sep. 1998.

[49] M. Komosinski and S. Ulatowski, "Framsticks: Towards a simulation of a nature-like world, creatures and evolution," in *Proc. Eur. Conf. Artif. Life*, 1999, pp. 261–265.

[50] Y. Kuniyoshi and M. Lungarella, "Developmental cognitive robotics: An emerging new field," in *Proc. Workshop Cogn. Dev. Robot.*, Kyoto, Japan, 2006, pp. 3–5.

[51] Y.-K. Kwon and K.-H. Cho, "Coherent coupling of feedback loops: A design principle of cell signaling networks," *Bioinformatics*, vol. 24, pp. 1926–1932, 2008.

[52] J. Lee and J. Sitte, "Morphogenetic evolvable hardware controllers for robot walking," presented at the 2nd Int. Symp. Autonomous Minirobots for Research and Edutainment, Brisbane, Australia, Feb. 2003.

[53] P. Lee, S. Kumburegama, H. Marlow, M. Martindale, and A. Wikramanayake, "Asymmetric developmental potential along the animal-vegetal axis in the anthozoan cnidarian, nematostella vectensis, is mediated by disheveled," *Dev. Biol.*, vol. 310, no. 1, pp. 169–186, 2007.

[54] Q. Liang, "Designing power aware self-reconfiguring topology for mobile wireless personal area networks using fuzzy logic," *IEEE Trans. Syst., Man, Cybern. C: Appl. Rev.*, vol. 33, no. 3, pp. 290–294, Aug. 2003.

[55] A. Lindenmayer, "Mathematical models for cellular interaction in developmental. Parts I and II," *J. Theor. Biol.*, vol. 18, pp. 280–315, 1968.

[56] H. Lipson, "Uncontrolled engineering: A review of evolutionary robotics (book review)," *Artif. Life*, vol. 7, pp. 419–424, 2001.

[57] M. Lungarella, G. Metta, and G. Pfeifer R., Sandini, "Developmental robotics: A survey," *Connect. Sci.*, vol. 15, pp. 151–190, 2003.

[58] M. Mamei, M. Vasirani, and M. Zambonelli, "Experiments in morphogenesis in swarms of simple mobile robot," *Appl. Artif. Life*, vol. 18, pp. 903–919, 2004.

[59] J. Mattick, "A new paradigm for developmental biology," *J. Exp. Biol.*, vol. 210, pp. 1526–1547, 2007.

[60] D. Meiner, "Swarm robotics algorithm: A survey," Univ. Maryland, College Park, MD, Tech. Rep., May 2007.

[61] Y. Meng, H. Guo, and Y. Jin, "A morphogenetic approach to flexible and robust shape formation for swarm robotic systems," *Auton. Robots*, to be published.

[62] Y. Meng, Y. Zhang, and Y. Jin, "Autonomous self-reconfiguration of modular robots by evolving a hierarchical mechanochemical model," *IEEE Comput. Intell. Mag.*, to be published.

[63] ——, "A morphogenetic approach to self-reconfigurable modular robots using a hybrid hierarchical gene regulatory network," in *Proc. 12th Int. Conf. Synth. Simul. Living Syst. (ALIFE XII)*, to be published.

[64] G. Metta and L. Berthouze, "Epigenetic robotics: Modeling cognitive development in robotic systems," *Interact. Stud.*, vol. 7, pp. 129–134, 2006.

[65] O. Michel, M. Clergue, and P. Collard, "Artificial neurogenesis: Applications to the cart-pole problem and to an autonomous mobile robot," *Int. J. Artif. Intell. Tools*, vol. 6, no. 4, pp. 613–634, 1997.

[66] E. Mjolsness, D. H. Sharp, and J. Reinitz, "A connectionist model of development," *J. Theor. Biol.*, vol. 52, pp. 429–453, 1991.

[67] S. Murata, K. Kakomura, and H. Kurokawa, "Toward a scalable modular robotic system—navigation, docking, and integration of M-TRAN," *IEEE Robot. Autom. Mag.*, vol. 14, no. 4, pp. 56–63, Dec. 2008.

[68] S. Murata and H. Kurokawa, "Self-reconfigurable robots," in *Proc. IEEE Robot. Autom. Mag.*, Mar. 2007, pp. 71–78.

[69] E. Mytilinaios, D. Marcus, M. Desnoyer, and H. Lipson, "Designed and evolved blueprints for physical artificial life," in *Proc. 9th Int. Conf. Artif. Life (ALIFE IX)*, 2004, pp. 15–20.

[70] S. Nolfi and D. Parisi, "Growing neural networks," Institute of Psychology, C.N.R., Rome, Tech. Rep. PCIA-91-15, 1991.

[71] N. Oros, V. Steuber, N. Davey, L. Caamero, R. and Adams, "Evolution of bilateral symmetry in agents controlled by spiking neural networks," in *Proc. 2009 IEEE Symp. Artif. Life*, 2009, pp. 116–123.

[72] S. Paladugu, V. Chickarmane, A. Dekard, J. Frumkim, M. MaCormarck, and H. Sauro, "*In silico* evolution of functional modules in biochemical networks.," *IEE Proc. Syst. Biol.*, vol. 153, pp. 223–235, 2006.

[73] R. Pfeifer, F. Iida, and G. Gomez, "Morphological computation for adaptive behavior and cognition," *Int. Congr. Series*, vol. 1291, pp. 22–29, 2006.

[74] R. Pfeifer and A. Knoll, "Intelligent and cognitive systems," *ERCIM News*, vol. 64, pp. 11–12, 2006.

[75] L. Piegl and W. Tiller, *The NURBS Book*. New York: Springer-Verlag, 1997.

[76] J. Pollack, H. Lipson, P. Funes, S. Ficici, and G. Hornby, "Coevolutionary robotics," in *Proc. Evolvable Hardware*, 1999, pp. 208–216.

[77] S. Psujek and R. Beer, "Developmental bias in evolution: Evolutionary accessibility of phenotypes in a model of evo-devo systems," *Evol. Dev.*, vol. 10, no. 3, pp. 375–390, 2008.

[78] M. Ridley, "Evolutionary developmental biology," in *Evolution*, New York: Wiley, 2003, pp. 572–589.

[79] N. S. and D. Floreano, *Evolutionary Robotics*. Cambridge, MA: MIT Press, 2004.

[80] D. Sanes, T. Reh, and W. Harris, *Development of Nervous Systems*, 2nd ed. New York: Academic, 2006.

[81] M. Schilstra and C. Nehaniv, "Bio-logic: Gene expression and the laws of combinatorial logic," *Artif. Life*, vol. 14, pp. 121–133, 2008.

[82] L. Schramm, Y. Jin, and B. Sendhoff, "Emerged coupling of motor control and morphological development in evolution of multi-cellular animates," in *Proc. 10th Eur. Conf. Artif. Life*, 2009.

[83] F. Sengpiel and P. Kind, "The role of activity in development of the visual system," *Curr. Biol.*, vol. 12, no. 23, pp. R818–R826, 2002.

[84] W. Shen, P. Will, and A. Galstyan, "Hormone-inspired self-organization and distributed control of robotic swarms," *Auton. Robots*, vol. 17, pp. 93–105, 2004.

[85] C. Sima, J. Hua, and S. Jung, "Inference of gene regulatory networks using time-series data: A survey," *Curr. Genomics*, vol. 10, pp. 416–429, 2009.

[86] K. Simons, E. Karsenti, S. D., C. Wijer, and S. Swarminathan, Eds., *Self-Organization and Morphogenesis in Biological Systems*, Schloss Ringberg, Tegersee, Germany, Dec. 2006.

[87] K. Sims, "Evolving 3D morphology and behavior by competition," *Artif. Life*, vol. 1, pp. 353–372, 1994.

[88] N. Spitzer, "Electrical activity in early neuronal development," *Nature*, vol. 444, pp. 707–712, 2006.

[89] K. O. Stanley and R. Miikkulainen, "A taxonomy for artificial embryogeny," *Artif. Life*, vol. 9, pp. 93–130, 2003.

[90] T. Steiner, Y. Jin, and B. Sendhoff, "A cellular model for the evolutionary development of lightweight material with an inner structure," in *Proc. 10th Annu. Conf. Genet. Evol. Comput. (GECCO)*, 2008, pp. 851–858.

[91] T. Steiner, J. Trommler, M. Brenn, Y. Jin, and B. Sendhoff, "Global shape with morphogen gradients and motile polarized cells," in *Proc. Congr. Evol. Comput.*, 2009, pp. 2225–2232.

[92] T. Taylor and C. Massey, "Recent developments in the evolution of morphologies and controllers for physically simulated creatures," *Artif. Life*, vol. 7, no. 1, pp. 77–87, 2001.

[93] J.-P. Thivierge, "How does non-random spontaneous activity contribute to brain development?" *Neural Netw.*, vol. 22, no. 7, pp. 901–912, 2009.

[94] T. Y.-C. Tsai, Y. S. Choi, W. Ma, J. Pomerening, C. Tang, and J. Ferrell Jr., "Robust tunable biological oscillations from interlinked positive and negative feedback loops," *Science*, vol. 321, pp. 126–129, 2008.

[95] J. Weng, J. McClelland, J. Pentland, I. Sporns, O. Stockman, M. Sur, and E. Thelen, "Autonomous mental development by robots and animals," *Science*, vol. 291, pp. 599–600, 2003.

[96] L. Wolpert, *Principles of Development*. London, U.K.: Oxford University Press, 2002.

[97] B. H. B. Yeung and J. K. Mills, "Design of a six dof reconfigurable gripper for flexible fixtureless assembly," *IEEE Trans. Syst., Man, Cybern. C: Appl. Rev.*, vol. 34, no. 2, pp. 226–235, May 2004.

[98] O. Yogev and E. Antonsson, "Growth and development of continuous structures," in *Proc. 9th Annu. Conf. Genetic Evol. Comput. (GECCO 2007)*, New York: ACM, 2007, pp. 1064–1065.

[99] R. Young, "Evolution of human hand: The role of throwing and clubbing," *J. Anat.*, vol. 202, pp. 165–174, 2003.

[100] C.-H. Yu, K. Haller, D. Ingber, and R. Nagpal, "Morpho: A self-deformable modular robot inspired by cellular structure," in *Proc. 2008 IEEE/RSJ Int. Conf. Intell. Robots Syst*, 2008, pp. 3571–3578.

Yan Meng (M'03) received the B.S. and M.S. degrees in electrical engineering from Xian Jiaotong University, Xian, China, in 1991 and 1994, respectively, and the Ph.D. degree in electrical engineering from Florida Atlantic University, Boca Raton, FL, in 2000.

From 2001 to 2003, she was a Senior Engineer in several industrial companies. In 2003, she was an Assistant Professor in the Department of Computer Engineering, Jackson State University, Jackson, MS. In 2004, she joined the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, where she is currently an Assistant Professor. Her research interests include bio-inspired multirobot systems, self-reconfigurable modular robots, computational intelligence, machine learning, robot/computer vision, real-time embedded systems, and bioinformatics.

Yaochu Jin (M'98–SM'02) received the B.Sc., M.Sc., and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 1988, 1991, and 1996, respectively, and the Dr.-Ing. degree from Ruhr University, Bochum, Germany, in 2001.

He is currently a Professor and Chair in Computational Intelligence, Department of Computing, University of Surrey, Guildford, U.K., where he is also the Head of the Nature Inspired Computing and Engineering Group. He was a Principal Scientist and Project Leader at the Honda Research Institute Europe in Germany. He is a coeditor of four books and three conference proceedings, the author of a monograph, and a coauthor of more than 100 peer-reviewed journal and conference papers. He has delivered numerous invited plenary and keynote speeches on various topics, including morphogenetic robotics, analysis and synthesis of gene regulatory networks, evolutionary aerodynamic design optimization, and multiobjective learning at international conferences and symposia. His current research interests include computational approaches to a systems-level understanding of evolution, learning and development in biology, and bio-inspired methods for solving complex engineering problems.

Prof. Jin is an Associate Editor of *BioSystems*, the IEEE TRANSACTIONS ON NEURAL NETWORKS, the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, PART C: APPLICATIONS AND REVIEWS, and the IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE. He is also an Editorial Board Member of Soft Computing, Memetic Computing and Swarm Intelligence Research. He was the Tutorial Chair of the Congress on Evolutionary Computation (CEC) 2007 and CEC 2010, a Program Cochair of the IEEE Symposium on Computational Intelligence in Multicriteria Decision Making 2007, 2009, 2011, and a Program Chair of the Fuzzy Systems and Knowledge Discovery 2005. He was the recipient of the Best Paper Award of the 2010 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology.