A Developmental Genetics-Inspired Approach to Robot Control

Sanjeev Kumar
Krasnow Institute for Advanced Study
George Mason University
Fairfax, VA
kumars@cs.gmu.edu

ABSTRACT

The need to build modular, scalable, and complex technology capable of adaptation, self-assembly, and self-repair has fuelled renewed interest in using approaches inspired by developmental biology. To meet this need, a new field, called Computational Development (CD), has emerged. Its focus is on adapting processes and mechanisms from developmental biology so as to help us build scalable, complex technology. Due to the embryonic nature of the field, however, research investigating the potential of such approaches for different problem domains is crucial to its success. In this paper, the plausibility of applying a developmental biology-inspired approach to the demanding problem domain of reactive robot control is explored. Using developmental genetics as a source of inspiration, a model of genetic regulatory networks is used in conjunction with a spatially distributed evolutionary algorithm to evolve real-time robot controllers for tasks such as general purpose obstacle avoidance.

Categories and Subject Descriptors

F.1.1 [**Computation by Abstract Devices**]: [Models of Computation]; I.2.8 [**Artificial Intelligence**]: [Problem Solving, Control Methods, and Search]; I.2.9 [**Artificial Intelligence**]: [Robotics]

General Terms

Algorithms, Experimentation, Performance

Keywords

Development, Genetic Regulatory Networks, Reactive Robots, Control

1. INTRODUCTION

The need to build modular, scalable, and complex technology capable of, for example, adaptation, self-assembly, and self-repair, has, over recent years, fuelled renewed interest in using approaches inspired by developmental biology [11, 7, 3, 5, 9]. For example, in the field of robotics, aspects of developmental biology have been modeled with much success for distributed robot control [15, 6], while the problem domain seeing the most success with the application of the developmental metaphor is evolutionary design. In Evolutionary Robotics (ER), however, robot control using developmental biology inspired approaches has typically been achieved indirectly, for example, through the development of a controller such as a neural network. Several researchers have investigated such an approach [3, 5]. The focus of these investigations, however, were on evolving a developmental process which specifies how the topology of a neural network (NN) is to be grown using concepts from developmental biology in which, for example, cells are able to divide, differentiate and form connections with other cells. After a period of growth, the resulting network of cells is interpreted as a neural network used to control robot behaviour and navigation. Common to all such works is that once the neural network is constructed the genome is discarded [12]. This does not occur in biology, where developmental processes and mechanisms continue to operate long after embryonic development has ceased in order to maintain the highly dynamic state of organisms.

In biology, the ability to react fast to changing situations and circumstances is crucial to the success of an organism, for example, fleeing a predator. As Marcus [10] points out, neurons react on a faster time-scale than genes, typically on the order of milliseconds, whereas genes are relatively slower. However, this should not detract from the fact that the genome is an immensely complex, real-time control system that builds bodies, brains, and immune systems capable of remarkable abilities.

As computer scientists and engineers we are not constrained by the same issues as biology; consequently, Genetic Regulatory Networks (GRNs) would operate at the same time-scale as Neural Networks (NNs). Furthermore, in using GRNs as the control system for generating reactive robot behaviours, in which current sensor readings are mapped into actions, future research can begin to explore how one might harness their powerful regenerative abilities to construct robust, fault-tolerant robotic behaviours.

Work by Kumar [8] has shown GRNs to be a viable control architecture for reactive robot control for the problem domain of evolutionary robotics, along with other works such as [1, 12, 16]. Alternative, and related, approaches to robot

GECCO'05, June 25–29, 2005, Washington, DC, USA.

control and behaviour generation that have seen much success are rule systems [14]. Still, problems with rule systems exist: although they can be made to have state, traditionally state was lacking. This work explores the ability of GRNs to specify reactive robot behaviours through the evolution of a general purpose obstacle avoidance GRN controller.

The paper is structured as follows: first a brief introduction to the field of developmental genetics is given, followed by a description of the genetic regulatory network (GRN) model used in this work. This is follwed by a section detailing the spatially distributed evolutionary algorithm employed in this work [13], while the following section introduces the robot environment used. The three sets of experiments performed to evolve real-time robot controllers are then described. The first set of experiments shows controllers that are able to memorise routes through the environment using no sensors. The second set shows controllers that are able to interact with the environment through sensors for both navigation and behaviour generation. However, analysis revealed that these controllers, although appearing to display obstacle avoidance behaviour, memorised routes through the environment. In order to address this, a third set of experiments is detailed that evolves GRN controllers to perform general purpose obstacle avoidance while being subjected to small amounts of sensory noise.

2. DEVELOPMENTAL GENETICS

Developmental genetics is the field of study that looks at how DNA specifies and controls genetic and cellular mechanisms that govern cell behaviour and permit a sophisticated program of development to emerge [17]. Central to genetics is the concept of *regulation*, in which the products of genes feedback into the system controlling the synthesis or decay of other gene products, which in turn regulate the expression of other genes and consequently other gene products, etc. The frequency with which regulation occurs and, more importantly, the number of stages centred around gene expression and development that afford the opportunity for regulation serves to underscore its importance.

This complex web of regulation in both space and time constitutes genetic regulatory networks. The power of these networks is made more apparent when the bigger multicellular picture is taken into account. Each cell in a multicellular organism contains a copy of the genome and thus of the GRNs. Additional complexity, is afforded by cell apparatus and STNs, Signal Transduction Networks.

STNs are essentially cascades of chemical reactions that occur when proteins bind to cell receptors (a rather crude analogy to the receptor-protein binding is that of a robot using its sensors). Typically, a chemical reaction occurs outside the cell at the receptor sites when and where a protein is detected. The receptor and incoming protein form what is known as a protein-receptor complex which acts as a signal. The signal conveyed by the protein-receptor complex is then transduced across the cell membrane and is allowed to enter the cell. At this point specific reactions occur in a cascade like manner. After completion of the cascade, a resulting signal in the form of mRNA or a protein is produced that is then permitted to target a particular gene or subset of genes within the cell's nucleus. Alternatively, the resulting mRNA or protein complex may not target any genes at all, but rather is decayed away. STNs therefore are able to provide a mechanism by which signal amplification or attenutation can be achieved. Here, again, we encounter the notion of regulation, which is not only prevalent in this synopsis of signal transduction, but pervades development, cell biology, and genetics.

Table 1: List of Proteins and their Purposes

Protein ID	Purpose Experiment
0	Move Forwards
1	Move Backwards
2	Rotate Counter-Clockwise
3	Rotate Clockwise
4	No Purpose
5	Rear Left Sensor
6	Left Sensor
7	Front-Left Sensor 1
8	Front-Left Sensor 2
9	Front-Right Sensor 2
10	Front-Right Sensor 1
11	Right Sensor
12	Rear-Right Sensor

Since both GRNs and STNs are housed within cells, this work, which focuses on genetic regulatory networks and not STNs or the cell, uses a highly abstracted and limited view of the cell as it pertains to GRNs. To this end, the biological cell offers a useful analogy for work in autonomous, distributed systems research. A cell may be viewed as an autonomous agent that has inputs and outputs as well as sophisticated processing abilities. This view of the cell translates to an agent with sensors (e.g. sonar) and effectors (e.g. a gripping arm). The analogy is attractive for many reasons, one of which is that cells can have many different types of sensors for sampling the environment, just as an agent can have different sensors.

3. THE GENETIC REGULATORY NETWORK MODEL

The model of genetic regulatory networks used in this work is a modified version of the GRN model used in the EDS (the Evolutionary Developmental System) [9] and is based on the connectionist framework proposed by Mjolsness (1996).

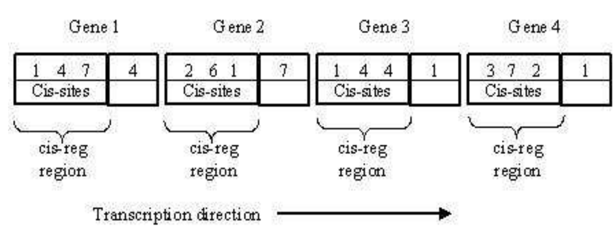


Figure 1: Cis-Trans Genome Architecture

Figure 1 illustrates the cis-trans architecture of the genetic regulatory network contained within the cell. GRNs are comprised of N genes connected by virtue of the proteins they encode. Each gene has two sections: a cis regulatory region (preconditions) and a coding region (adjacent to the cis-region shown in fig 1) which specifies a protein to be emitted if the gene is expressed. Just as in biology, the genes in the EDS may be thought of as rules with preconditions to be satisfied before a rule can fire and actions to be performed upon firing. Each gene has an associated evolved activation threshold beyond which the gene will activate and emit the protein specified in the coding region. Since genes are essentially rules, gene expression and rule firing are one and the same thing. Gene expression is achieved by assessing the level of occupation of the cis-site region of a gene by other proteins (or transcription factors). In the context of rule firing, this translates to assessing the contribution of proteins in satisfying the precondition requirements of the gene (rule) encoded in the form of protein cis-sites [2], see fig 1.

The number of proteins were varied for each experiment. However, in the case of the third experiment - which had the most proteins - a total of thirteen proteins were used in the model, the purpose of each protein is shown in table 1. Speed, s, and rotation angle, θ, are specified by the concentration of the corresponding protein (which may take up values in the interval 0..1) and scaled appropriately. For example, if only protein 0 is present at a concentration of, say, 0.5 then this value is multiplied by a constant which defines a ceiling for the robot's velocity, which in this case was set, arbitrarily, to 400. Protein 1 functioned in the same manner as protein 0, however, it specified the velocity of the robot but in reverse. Proteins 2 and 3 operated in a similar manner to specify the direction and degree of rotation with a ceiling set at 90.

3.1 Noise

The GRNs used in this work employed noise at two different levels:

- gene activation (rule firing) and
- sensory input (in the third experiment)

At the gene activation level, the firing of genes (rules) was performed probabilistically, thus introducing noise at the level of rule firing making the task of evolving robot controllers more difficult. The incorporation of noise at this level is justified on the basis that it enables the system to evolve robust rule sets. Noise at the level of sensory input (only employed in experiment 3) was provided by adding a small degree of Gaussian noise (see tables 2 and 3 for σ) to the sonar values before being input to the GRN controller.

4. EVOLUTION: THE DISTRIBUTED GENETIC ALGORITHM

In this work, a spatially Distributed Genetic Algorithm (DGA) was employed. The DGA used a single population distributed over a 2-D toroidal grid. A single generation in the DGA involved iterating over each individual in the population. Parents were selected by sampling from the Moore neighbourhood centred around the current individual.

After selecting a neighbourhood, the two best individuals from the neighbourhood are selected and the two worst individuals of the neighbourhood are replaced by the resulting two offspring. Offspring are generated through the application of the genetic operators: 2-point crossover (always applied) and mutation. In particular, all floating point parameters underwent Gaussian mutation, while the cis-trans architecture (i.e. the portion of the genome that specifies which proteins comprise which genes) underwent mutation at a rate of 1/L, where L is the length of the cis-trans architecture. Since protein synthesis, decay, gene interaction, and affinity values are continuous, all parameters were encoded as floating-point numbers on a haploid genome. For fuller details of the representation used in this work see [9].

4.1 Fitness Functions

Three sets of experiments were performed; all experiments involved using Euclidean distance as part of the fitness function. For the first set of experiments, the task set for the robots was to maximise Euclidean distance – to move as far as possible – within a set period of time, approximately 12 seconds with no sensory information. In the second experiment, the robot was allowed to use sensors. See equation 1 for the fitness function used.

$$EuclideanDistance = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (1)$$

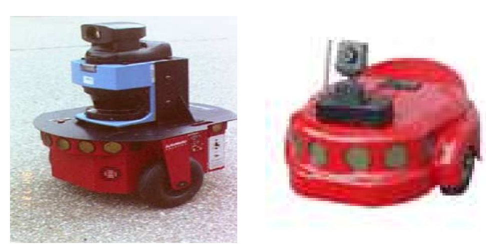
The third set of experiments used equation 1 as one of the terms, the full fitness function is shown in equation 2. This equation is an adapted version of the obstacle avoidance fitness function used by Floreano, et al. (1996), it defines an equation to be maximised. It is, however, lacking an additional term, which Floreano included: a term to measure whether or not the robot is moving in a straight line. This was included in Floreano's work to prevent evolution awarding high fitness values to controllers that cause the robot to traverse a circle as fast as it can. In place of this term, the Euclidean distance was used.

$$Avoidance = \Sigma(e + s + SonarValue_{MAX}) \quad (2)$$

Where e is the Euclidean distance provided by eqn 1, s is the speed of the robot and $SonarValue_{MAX}$ is the reading from the sonar with the most activity.

5. ROBOTS

This work employed two different robots – a Pioneer 2DX (see fig. 2 (a)), and an Amigobot (see fig. 2 (b)) – in simulation using the University of Southern California's robot simulator, player-stage [4]. This simulator was selected for many reasons of which the most salient was the wide range of robots supported, and the inherent noise in the system. This element of stochasticity aides the transition of the controller over, what has now become known as, the 'reality gap' from simulation to real hardware [5].



(a) (b)

Figure 2: The Pioneer 2DX (a) and Active Media's Amigobot (b)

Table 2: List of Evolutionary and Developmental Parameters for Expt 1

Evolution	Development
PopSize 100	LifeTime 12 sec
Generations 100	Cis-Sites 2
Num Genes 10	Num Proteins 4
Crossover 100%	
Gaussian Mutation 0.75	

6. EXPERIMENTS

This section details the experiments and results. In the first and second set of experiments, both robots were permitted to rotate either left or right and move forwards, but were prevented from moving backwards so as to encourage clear paths to emerge while avoiding obstacles. In the third set of experiments, however, the robots were allowed to reverse.

6.1 Objectives and Parameter settings

This work had two main objectives:

- Show that GRNs are a plausible technology for real-time reactive robot control and
- Generate reactive behaviours such as traversing an obstacle ladend world

However, during the course of the experiments, analysis of evolved solutions generated an additional objective:

- to evolve general purpose obstacle avoidance irrespective of the robot's position in the environment.

Due to the inherent noise in the GRN's gene activation mechanism and the addition of Gaussian noise to the sensory inputs, reactive robot control would be achieved in the face of noise at both levels. The number of proteins was changed between the experiments due to problem requirements such as the use of sonar proteins, for example.

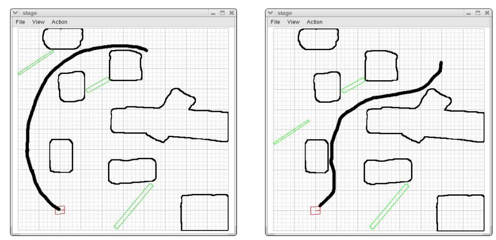
6.2 Experiment 1: Memorising a Path

The first experiment requires the memorisation of a path through an environment with obstacles using no sensors or start seeds (i.e. no maternal factors - solutions are evolved completely from scratch). Although this experiment does not class as 'reactive' robot control, a solution to this problem does require internal state, thus the experiment was deemed necessary in order to ascertain the level of state embodied within GRNs. The GRN could only use the first four proteins shown in table 1. Note, this experiment was only performed with the Pioneer 2DX in simulation. This task implicitly requires internal state in order to solve the problem. This set of experiment used 4 proteins, a population size of 100 evolving for 100 generations, additional evolutionary and developmental parameter settings are shown in table 2.

6.3 Experiment 1: Results

As can be seen from figure 3, the EDS evolved genetic regulatory network controllers that were successfully able to memorise a path through the environment without any sensors. The figures illustrate two different paths discovered by two different controllers evolved from scratch. Figure 3 (a) provides an example of the robot traversing a circular trajectory with minor deviations. To encourage a more intricate path to be found, the wall in the upper left corner of fig 3 (a) was brought down, thus blocking the robots path in the upper left direction, see fig 3 (b). With the wall now blocking the robot's path figure 3 (b) shows a different evolved GRN resulting in a more intricate path with no environment interaction. In this example, the robot is able to navigate, quickly, through the environment negotiating left and right turns past obstacles. This example reflects internal state in GRNs.

Using no sensors, a population size of 100 individuals and 100 generations, good solutions capable of discovering clear paths through the environment emerged at around the 54th generation.



(a) (b)

Figure 3: Best Pioneer 2DX GRN controllers using no sensors, run 1 (a) and run 2 (b)

6.4 Experiment 2: System-Environment Interaction

While memorising paths through an environment using no sensory feedback may seem to be of limited use, it does demonstrate that GRNs are capable of encoding such paths, and that these kinds of solutions are evolvable. In the second experiment, both the Pioneer 2DX and the Amigobot were used, again in simulation, but with different types of sensors. In the case of the Pioneer, SICK LMS 200 laser-range finder sensors were employed, while the Amigobot used its eight sonar sensors. The laser values were compiled into two values: one, which corresponded to a left side sensor and the other, which corresponded to a right side sensor. The amigobot has sonar emitters and receivers distributed around the robot with six arranged around the sides and front, while two are located at the rear of the robot. Since the robot was not permitted to reverse, the two sonars at the rear of the robot were not used (only in experiments 1 and 2); consequently, the remaining six sonars were split in two and provided left and right sonar sensing. The settings used were: 7 proteins with a population size of only 25 (5*5) individuals evolving for 50 generations (see table 3).

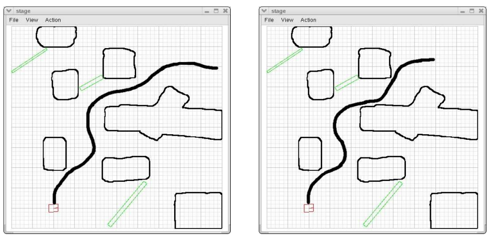
6.5 Experiment 2: Results

Figures 4 and 5 show that despite noisy gene transcription (noisy rule firing) GRNs are evolved that are able to control both a Pioneer 2DX robot equipped with SICK LMS lasers, and an Amigobot equipped with sonars through the same environment more convincingly. Additionally, coupling system and environment through sensors enables faster evolution of successful solutions, for example, all experiments using sensors resulted in good solutions, able to cope with noisy gene transcription, emerging within just four genera-

Table 3: List of Evolutionary and Developmental Parameters for expt 2

Evolution	Development
PopSize 25	LifeTime 12 sec
Generations 50	Cis-Sites 6
Num Genes 15	Num Proteins 7
Crossover 100	Sensor Noise σ 0.00
Gaussian Mutation 0.75	

tions using 25 individuals. Compare this to the performance of GRNs evolved with no sensors in experiment 1.



(a) (b)

Figure 4: Best Pioneer 2DX GRN controller with laser sensors, run 1 (a) and run 2 (b)

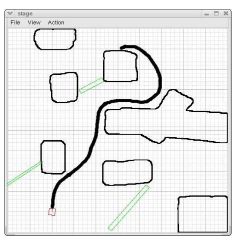


Figure 5: Best Amigobot GRN controller with 8 sonar sensors, run 1

6.6 Experiment 3: General Purpose Obstacle Avoidance (GPOA)

Analysis of the results in experiment two revealed that evolution had managed to cheat: the controllers had learned routes through the world just as in the first experiment; except this time, solutions were found much sooner since evolution was able to exploit sensory information from the laser and sonars. With this in mind a third objective was set: to evolve a general purpose obstacle avoidance GRN controller.

In this experiment, in order to ensure the controllers were using sensory information and not simply using additional proteins to memorise the route, only sensor information – in the form of protein concentrations – was used in the GRN. In other words, the GRNs could only use sensor proteins – proteins 5 through 12, see table 1 – to trigger gene expression. This was achieved by forcing cis-site inputs to genes to be sonar proteins. The total number of cis-sites per gene was increased for this experiment to six. Thus, of a total of eight sonar proteins only six were used per gene, note the particlar sonar proteins used were set by evolution. A total of thirteen proteins were used for this set of experiments: four proteins for movement, eight for sonar readings, and one protein with no purpose (evolution can use this protein as it sees fit).

Table 4: List of Evolutionary and Developmental Parameters for GPOA

Evolution	Development
PopSize 25	LifeTime 12 sec
Generations 25	Cis-Sites 6
Num Genes 7	Num Proteins 13
Crossover 100	Sonar Noise σ 0.02
Guassian Mutation 0.75	

Noise was added at the level of gene activation in the form of probabilistic gene activation. In addition, Gaussian noise was added to all sonar values with a small standard deviation (σ) of 0.02, arbitrarily selected. A single fitness evaluation consisted of controlling the robot for twelve seconds using the currently evolved GRN. The parameter settings for this experiment are shown in table 4. As figure 6 shows, a new world was created with corridors, open spaces, and obstacles in the form of walls.

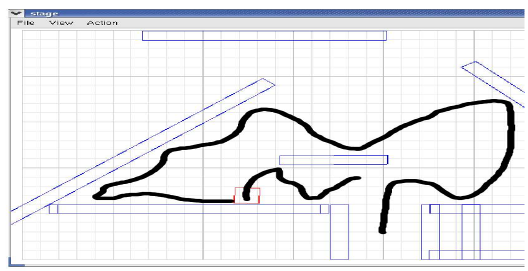


Figure 6: General Purpose Obstacle Avoidance world

6.7 Experiment 3: Results

A solution displaying general purpose obstacle avoidance behavior was found in generation five and gradually optimised over the subsequent twenty generations. As the robot moved around the world it tended to display a general anticlockwise turning motion while avoiding obstacles. This particular solution's approach to negotiating obstacles was as follows: on approaching an obstacle head-on, and fast, the controller caused the robot to reverse. On a slower approach the robot got within a certain distance of the object and either reversed or very slowly bumped into the object, upon which reverse is triggered immediately. An obstacle sensed behind the robot, however, always resulted in immediate forward movement away from the object.

It is worthwhile noting that despite being evolved in a static environment where each fitness assessment consisted of a single trial, the Amigobot can be moved to different areas of the environment and still maintain general purpose obstacle avoidance behaviour. Additionally, informal experiments have shown that evolved GRN controllers are quite robust with respect to the level of noise added, for example, this GPOA controller evolved using sonar noise with a std dev. σ of 0.02 (very small), yet as the noise is increased the controller manages to cope with no adverse effects.

7. CONCLUSIONS

This work has explored the application of the developmental metaphor to the field of evolutionary robotics by investigating the ability of genetic regulatory networks (GRNs) to specify reactive robot behaviours. Through the successful evolution of GRN robot controllers that provide general purpose obstacle avoidance, we have shown:

- that the developmental biology metaphor can be productively applied to the problem domain of evolutionary robotics;
- GRNs to be a plausible technology for real-time reactive robot control;
- GRNs have internal state and are capable of generating reactive behaviours such as traversing an obstacle ladend world; and
- evolved GRN controllers for general purpose obstacle avoidance that are able to cope with varying amounts of noise at two important levels: rule firing (or gene expression) and sensor noise.

In order to ensure GRNs remain a viable option for the field of robotics, as well as evolutionary robotics, further research is required into the ability of GRNs to specify, multiple low-level reactive robot behaviours in a modular manner. The ability of GRNs to construct modular networks while being robust to damage makes GRNs very well suited to the task of generating robust, reactive robot behaviours. The preliminary results shown here offer encouraging potential for future research into GRN-based controllers, and indeed developmental biology inspired approaches to evolutionary robotics and robotics in general.

8. ACKNOWLEDGMENTS

Many thanks to Jeff Bassett, Adrian Grajdea, Ken De Jong, Sean Luke, and Paul Wiegand for their helpful comments and critiques. This research is sponsored by SAIC.

9. REFERENCES

[1] P. Bentley. Adaptive fractal gene reguatory networks for robot control. *In Miller, J. (Ed.) Workshop on Regeneration and Learning in Developmental Systems, Genetic and Evolutionary Computation Conference (GECCO 2004).*

[2] E. Davidson. *Genomic Regulatory Systems: Development and Evolution*. Academic Press, 2001.

[3] F. Dellaert. Towards a biologically defensible model of development, 1995.

[4] B. Gerkey, R. Vaughan, and A. Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. *In Proceedings of the 11th International Conference on Advanced Robotics.*, pages 317–323, 2003.

[5] N. Jakobi. Harnessing morphogenesis. *International Conference on Information Processing in Cells and Tissues (IPCAT)*, pages 29–41, 1995.

[6] T.-X. Jiang, R. Wideltz, W.-M. Shen, P. Will, D. Wu, C. Lin, J. Jung, and C. Chuong. Integument pattern formation involves genetic and epigenetic controls operated at different levels: Feather arrays simulated by a digital hormone model. *International Journal on Developmental Biology*, 2004.

[7] H. Kitano. Designing neural networks using genetic algorithms with graph generation system. *Complex Systems*, 4:461–476, 1990.

[8] S. Kumar. A developmental biology inspired approach to robot control. *Workshop on Self-Organization and Development in Artificial and Natural Systems, In Proceedings of the Artificial Life 9 Conference (Alife 9)*, 2004.

[9] S. Kumar. *Investigating Computational Models of Development for the Construction of Shape and Form.* PhD thesis, University College London, 2004.

[10] G. Marcus. *The Birth of the Mind*. Basic Books, 2004.

[11] P. Prusinkiewicz, A. Lindenmeyer, and J. Hanan. Developmental models of herbaceous plants for computer imagery purposes. *In Computer Graphics (SIGGRAPH 88 Conference Proceedings)*, 22, 1988.

[12] T. Quick, K. Dautenhahn, C. Nehaniv, and G. Roberts. Evolving embodied genetic regulatory network-driven control systems. *In Proceedings of the 7th European Conference on Artificial Life (ECAL '2003)*, 2003.

[13] J. Sarma. *An Analysis of Decentralized and Spatially Distributed Genetic Algorithms.* PhD thesis, George Mason University, Fairfax VA, USA, 1998.

[14] A. Schultz and J. Grefenstette. Evolving robot behaviors. In *Proceedings of the 1994 Artificial Life Conference*, 1994. Poster paper.

[15] W.-M. Shen, B. Salemi, and P. Will. Hormone-inspired adaptive communication and distributed control for conro self-reconfigurable robots. *IEEE Transactions on Robotics and Automation*, 18(5), 2002.

[16] T. Taylor. Genetic regulatory network control for a team of underwater robots. *Artificial Life*, 2003.

[17] L. Wolpert, R. Beddington, J. Brockes, and E. Meyerowitz. *Principles of Development.* Oxford University Press, UK, 2002.