A Genetic Regulatory Network-Inspired Real-Time Controller for a Group of Underwater Robots

Tim TAYLOR
Institute of Perception, Action & Behaviour (IPAB)
School of Informatics, University of Edinburgh, Scotland
tim.taylor@ed.ac.uk

Abstract. A decentralised real-time controller for a group of robots is presented, the design of which is inspired by biological genetic regulatory networks. A genetic algorithm (GA) is used to automatically evolve controllers for specific tasks. Results of initial experiments are presented and analysed, which demonstrate that it is possible to successfully evolve the controllers to achieve a simple clustering task. Performance is robust under a variety of parameter choices for the GA and controller.

1 Introduction

This work concerns the control of a group of agents to perform coordinated tasks, where each agent possesses its own local controller and can only communicate with other agents over a limited distance. Multiagent systems have become a popular subject of study in recent years due to their perceived advantages over more traditional, monolithic systems (e.g. [1]). Potential advantages include features such as lower production costs, improved fault tolerance, graceful degradation of behaviour and the ability to self-repair.

The current study is part of a research effort to develop a group of underwater robots (called Hydrons) which possess independent controllers but are able to achieve high level, coordinated tasks through local communication.[1] The hardware specifications are described in Section 2. In addition to developing novel hardware, a complementary goal of the project is to elucidate biological mechanisms for properties such as self-reconfiguration and self-repair (as demonstrated, for example, by organisms such as the hydra, a freshwater polyp, after which the project was named). The biological cases provide inspiration for the development of control algorithms (as well as the design of the hardware itself) to give the robots similar characteristics of adaptability and robustness. The work reported here concerns the design of one such algorithm, and results of initial experiments. The real Hydrons are still at the prototype stage, so the experiments were conducted using a detailed simulation of the robots.

A growing number of studies have been reported over the last few years on decentralised controllers for multi-robot systems (e.g. [2, 3, 4, 5]). However, most of these controllers are specific for a particular task; to perform a different task the controller would have to be

[1]This work forms part of the Hydra project, funded by the EU Information Society Technologies (Future and Emerging Technologies) programme. Details of the design of the Hydron hardware prototypes and associated simulation software can be found at the project website `http://www.hydra-robot.com`

Sensory	Depth sensor, Docking sensor
Signalling	Optical transmitters and receivers
Actuation	Move up, down, left, right, forward, backward

Table 1: Agent capabilities used by the controller

redesigned by hand. One of the goals of the current study was to develop a general controller which could be automatically evolved to achieve specific tasks.

A recent article showed that bacteria organise into coordinated groups using inter-organism communication which controls the expression of specific genes in individual organisms [6]. Furthermore, Reil recently reported work on a model of genetic regulatory networks (GRNs), which suggests that they possess desirable properties for acting as an evolutionary substrate [7]. Reil's model has subsequently been adapted for simulations of morphogenesis [8].

In the current study, a new GRN controller is developed, the design of which is based upon these existing models. However, the major novelty of the new controller is that it can be applied not to the morphogenesis of a single organism, but rather to the real-time continuous control of a group of robots (inspired by communication and coordinated action in bacteria).[2]

The rest of the paper is organised as follows. The next section presents a description of the GRN controller (Section 2). The application of a genetic algorithm to evolve the controller for performing a specified task is then discussed (Section 3). Next, the design of initial experiments to test the performance of the algorithm is presented (Section 4), together with results and analysis (Section 5). This is followed by a summary of the paper and discussion of future research directions (Section 6).

2 GRN Controller Design

The two major factors that influenced the controller design were the specifications of the real hardware and the analogy to biological genetic regulatory networks.

The basic hardware specifications are as follows. Each Hydron is a sphere of approximately 11cm diameter, with a syringe intake and an impeller both located on its underside, and four outflow nozzles located around its equator (each separated by 90 degrees). The Hydron can suck in water using its impeller and eject it from one of the outflow nozzles, thereby generating horizontal thrust in a particular direction. In addition, the syringe is used to change the Hydron's relative density by injecting or ejecting water from an internal chamber, thereby giving control over movement in the vertical plane. The centre of mass of the Hydron is offset below its geometric centre, which introduces a self-righting torque to keep the robot upright. Eight "interface sites" are positioned on the Hydron's surface, at the corners of an imaginary cube placed inside the robot such that each corner just touches the surface. An optical transmitter and receiver is located at each interface site (for inter-Hydron communication), together with a weak magnetic dipole (to enable two nearby Hydrons to physically dock).

In this work, the particular set of sensory, signalling and actuation capabilities that the robots were assumed to possess matches those of the Hydron hardware (Table 1). However, the controller design is such that this set can be easily modified for use on different hardware.

The controller for each agent comprises: a *genome* (a variable length integer string which

[2] Another GRN model for real-time robot control has recently been published [9], but this controls a single robot rather than a group.

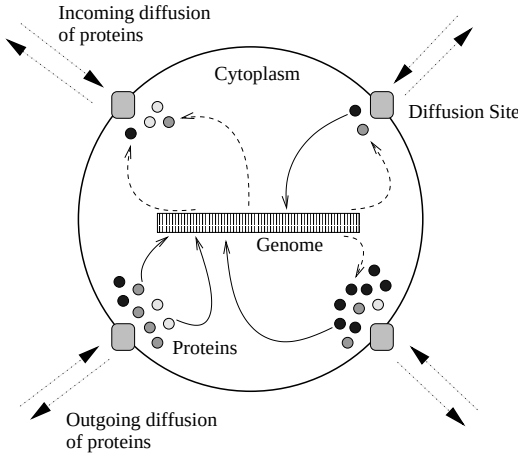

Figure 1: Summary of the GRN Controller design. For clarity, only four of the eight diffusion sites are shown.

may encode information about a number of *genes*); and a *cytoplasm* (containing a variety of *proteins* located at eight diffusion sites). Each gene produces a specific type of protein when expressed. The expression of each gene is controlled by a set of enhancer proteins and a set of inhibitor proteins. This sets up the essential ingredients of the regulatory network; genes produce proteins, and proteins control the expression of genes.

Proteins act as the interface between the genome and the physical environment. In addition to controlling gene expression, some types of protein also interface with the agent's sensory, signalling and actuation capabilities. Protein action is described in more detail in Section 2.2. When a gene produces a protein, it is released into the cytoplasm at one of eight diffusion sites (the specific site of deposition being under genetic control). The diffusion sites correspond to specific sites on the physical agent's body. For the Hydron robots, each site corresponds to the location of the optical transmitters and receivers.

A summary diagram of the controller design is shown in Figure 1. Further details are presented in the remainder of this section.

2.1 Genome and Gene Structure

A section from an example genome is shown in Figure 2. When a new genome is created, it is scanned from beginning to end in order to identify what genes it encodes. Specific consecutive sequences of digits in the genome (called *gene promoter sequences*) signify the start of a gene. The region starting at the end of the previous gene (or, in the case of the first gene, the start of the genome) up to the digit immediately preceding the promoter sequence is the *regulatory region* for that gene.

The regulatory region is scanned for information regarding enhancer and inhibitor proteins for the gene. The presence of enhancer proteins in the cytoplasm increases the activation of a gene, while inhibitor proteins decrease activation. An enhancer protein is encoded by a fixed length sequence of digits in the regulatory region, starting with a specific *enhancer promoter sequence*, followed by an encoding of the particular protein. Likewise, inhibitors are identified by a specific *inhibitor promoter sequence*, followed by the protein specification.

A fixed number of digits immediately following the gene promoter sequence encode the gene itself (see Figure 3 for an example). The gene encodes information about the type of protein it produces when expressed, its output function (see below), and a specification of how

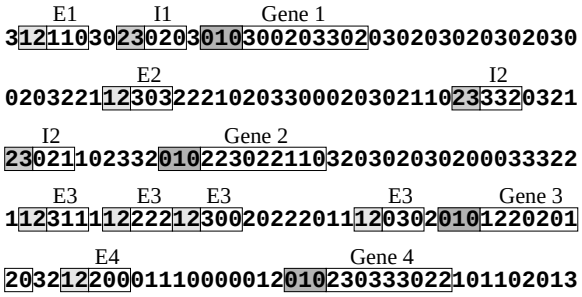

Figure 2: An example of a section of the genome. The genome is a string of base-4 integers, and proteins are specified by a string of 3 digits (so 64 different kinds of protein are available). The gene promoter sequence is 010, the enhancer promoter sequence is 12 and the inhibitor promoter sequence is 23. E*n* and I*n* are labels for enhancers and inhibitors, respectively, for Gene *n*. So the expression of Gene 1, for example, is enhanced by protein 20 (110 in base-4 notation), and inhibited by protein 8 (020 in base-4 notation).

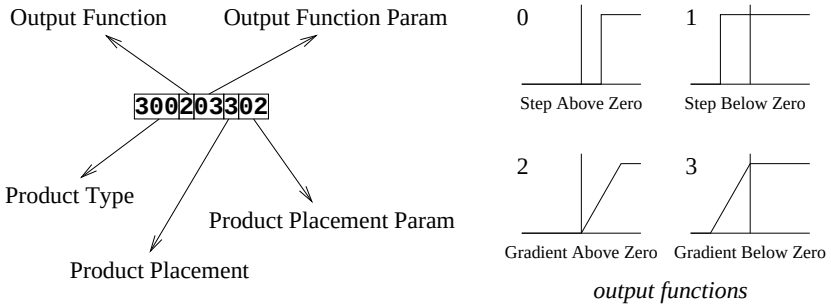

Figure 3: The structure of a gene. In this example, the gene produces protein 48 (300 in base 4) when expressed. It has a Gradient Above Zero output function (2) with gradient 3 (03). The product is deposited at a specific diffusion site (3), which is site number 2 (02).

the product protein is to be distributed across the different diffusion sites in the cytoplasm.

Four different output functions are available (see Figure 3), which relate the total concentration of regulators (i.e. total current cytoplasmic concentration of all enhancers for this gene minus total concentration of all inhibitors, summed across all diffusion sites) to the degree of gene activation. The form of the function is further refined by the *Output Function Parameter*, which specifies the position of the step or the gradient of the function as appropriate.

Four different product placement schemes are available: place product at the site with the highest concentration of a specific signal protein (Scheme 0), place product at the site with the lowest concentration of a specific signal protein (Scheme 1), distribute product across all sites (Scheme 2), or place product at a specific diffusion site (Scheme 3). The *Product Placement Parameter* specifies the specific site or the specific signal protein as appropriate.

2.2 Protein Action

Any protein can potentially act as an enhancer or an inhibitor of gene expression. Additionally, some proteins act as interfaces to the physical agent, either as actuation proteins or sensory proteins. The number of different actuation and sensory proteins, and the external events to which they are related, will depend upon the actual agent being controlled. In the present context, the agents are simulated Hydron units, and the actuation and sensory proteins used are as follows:

Only one type of actuation protein is used, for movement. When these collect at a diffusion site, they cause the agent to attempt to move in that direction. The agent's overall movement depends on the distribution of movement proteins across each of its diffusion sites. In current experiments, proteins 40–47 act as movement proteins.

Two types of sensory protein are used. These proteins are produced in the cytoplasm according to particular conditions in the environment, independent of any gene activity. One is produced in the cytoplasm (evenly across all diffusion sites) such that its concentration is proportional to the current depth of the agent. The other is produced at a particular diffusion site if the agent is physically docked with another unit at that site. In current experiments, proteins 48–55 act as docking sensors, and 56–63 as depth sensors.

2.3 Protein Dynamics

The proteins in the cytoplasm are subject to *attenuation* and *diffusion* dynamics. The concentration $C_p(t)$ of any protein p at any diffusion site decays over time,

$$C_p(t+1) = kC_p(t) \tag{1}$$

where k is the attenuation constant ($0 < k < 1$).

There are two types of diffusion, *internal* (within the cytoplasm of a single agent) and *external* (between agents). All proteins undergo internal diffusion, where the concentration at one site diffuses to neighbouring sites over time. For each protein p at site i, the change in concentration at that site, and at each of its N neighbouring sites j is given by:

$$\begin{aligned} C_p(i,t+1) &= (1-d)C_p(i,t) \\ C_p(j,t+1) &= C_p(j,t) + \frac{dC_p(i,t)}{N} \end{aligned} \tag{2}$$

where d is the internal diffusion constant ($0 < d < 1$).

A subset of proteins also undergoes external diffusion, where they diffuse out of the agent into the environment, and potentially diffuse into neighbouring agents:

$$\begin{aligned} C_p(i,t+1) &= (1-e)C_p(i,t) \\ E_p(i,t) &= eC_p(i,t) \end{aligned} \tag{3}$$

where $E_p(i,t)$ is the amount of protein p externally diffusing from diffusion site i at time t, and e is the external diffusion constant ($0 < e < 1$). External diffusion relies upon the existence of a communication system between agents; in the case of the simulated Hydron units, the optical communication system is utilised. It is assumed this system can transmit signals which digitally encode the protein type and diffusion amount. If a neighbouring agent receives such a signal, it will decode it and introduce the specified protein into the cytoplasm at the site of reception. The amount of protein introduced is a function of the amount encoded in the signal and the intensity of the signal itself, so the total amount introduced depends upon the distance between the transmitting and receiving agents.

At the beginning of a run, the cytoplasm may be initialised with proteins at some or all of the diffusion sites. Such initialisation may be useful to provide the controller with information about spatial orientation (see Section 3 for details of the initialisation scheme used in the current experiments). Note, however, that genes with Step Below Zero or Gradient Below Zero output functions (Figure 3) will be active in the absence of any proteins, so genomes containing such genes will become active even in an initially empty cytoplasm.

3 Evolving GRN Controllers

In order to produce GRN controllers which will cause a group of Hydron units to achieve a particular task, a genetic algorithm (GA) was employed to evolve a population of genomes. A standard generational GA was used, with tournament selection and elitism. Two-point crossover was applied, using different crossover points on each parent, which therefore allowed the length of an offspring genome to be shorter or longer than that of its parents (i.e. genome length can evolve over time).

To get a fitness value for each genome, it was used to construct a GRN controller for each of the Hydron robots in the group. The cytoplasm of each agent was initialised with six types of protein (proteins 1–6), placed at the diffusion sites on the Hydron's upper, lower, front, back, left and right sides respectively. This gives the controller information about the spatial axes of the robot; this is the only *a priori* knowledge given to each controller. When the controllers of each of the robots have been initialised in this way, they are allowed to execute for a fixed duration, and the overall behaviour of the group is then evaluated in terms of the given task and a fitness score assigned.

For the initial experiments reported here, the task was for the group to form a cluster. The fitness function was the mean square distance of each Hydron at the end of the evaluation period from the centroid of the group at the start of the period. The fitness score was the negative of the mean square distance, so the better the controller, the closer this negative number was to zero.[3]

4 Experimental Design

A series of experiments was designed to investigate the ability of the system to produce controllers which could perform the clustering task. A total of 160 experiments were conducted in four batches, in order to test the sensitivity of the results to the main parameters of the GA and the GRN controller.[4] For each combination of parameters, 10 separate runs were conducted using different random number seeds.

Batches A, B and C looked at GA-related parameters. Batch A addressed the question of whether, for a given amount of computational resource, it was better to run a small GA population for a large number of generations, or a large population for a small number of generations. Set A1 evolved a population of 500 controllers for 100 generations, and set A2 evolved a population of 100 controllers for 500 generations. Batch B looked at the GA parameters of tournament size and elite group size (i.e. the top n controllers from one generation which were passed unaltered to the next). Tournament sizes of 5 and 25 were used, along with elite group sizes of 1 and 50, giving 4 different sets of runs in total for Batch B. Batch C looked at the probabilities of crossover and mutation. Crossover probabilities per selected parent of 0.5 and 0.9 were used, with mutation probabilities per genome digit of 0.01 and

[3] During analysis of these runs, a bug in the controller code came to light which meant that the Hydrons were unable to move in the vertical plane. This meant that the maximum possible fitness score achievable was -0.2231m^2, achievable if the Hydrons formed a perfectly aligned vertical column.

[4] The default parameter values, which were used in all runs unless otherwise stated, were as follows: Population Size 100, Generations 200, Elite Group Size 1, Tournament Size 25, Crossover Probability 0.9, Mutation Probability 0.005, Initial Genome Size 1000, Enhancer:Inhibitor Ratio 1:1. Each controller was evaluated for 30 seconds of simulated time; an identical copy of the controller was placed into five Hydrons, and each trial started with them in the same initial spatial configuration.

0.005, giving another 4 sets of runs for Batch C.

Batch D looked at parameters associated with the GRN controller. In particular, the length of the randomly-generated genomes in the initial population was varied from 1000 digits to 5000 digits. Also, the range of sequences in the genome that qualified as enhancer and inhibitor promoters was varied, to change the relative degree of gene enhancement and gene inhibition. Specifically, the number of sequences qualifying as enhancer promoters was varied from 3 to 2 to 1, while at the same time the number qualifying as inhibitor promoters was varied from 1 to 2 to 3. This gave 3 sets of experiments, where the relative probability of gene enhancement over gene inhibition was 3:1, 1:1 and 1:3 respectively. Combined with the 2 different genome lengths, this gave a total of 6 sets of runs in Batch D.

5 Results and Analysis

The results of the 160 experiments suggest that the performance of the system is remarkably resilient to the particular choices made for the main parameter values. In each batch of experiments, the GA was able to improve the performance of the GRN controllers over the run, and at least some of the 10 trials in each set achieved near-optimal fitness (i.e. a mean-square distance of within 0.01m^2 of the highest possible fitness).

Due to space limitations, a detailed statistical analysis of the full set of results will be presented elsewhere. Instead, in this section results from a typical set of runs are investigated, in order to shed light on the operation of the GRN controller and on its evolutionary potential.

This particular set of 10 runs were from Batch A, using a population size of 100, evolved for 500 generations. All other parameters took their default values as described in Section 4. The increase in fitness over time is shown in Figure 4a. The final fitness achieved by the best of the 10 runs was -0.2275 and the mean was -0.2363 (compared with the maximum achievable fitness of -0.2231). As the population evolved, the genome length tended to increase (not shown), as did the number of genes encoded (Figure 4b).

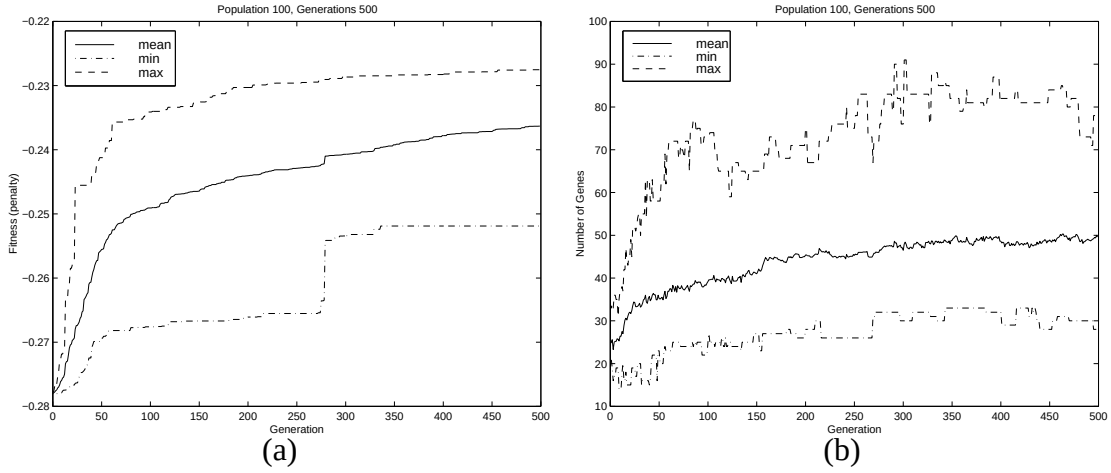

Figure 4: (a) Fitness of best individual in population over time (generations) for runs in Batch A2. For each generation, the mean, maximum and minimum fitness for the 10 runs is plotted. (b) Number of genes contained in the genome of the best individual in population over time.

To get a better idea of the operation of the genetic regulatory network, the evolved controller from the final generation of the best run was analysed. Figure 5a shows the gene regulation matrix for this controller. The genome contained 30 genes. Each of these is plotted on

the vertical axis, and the corresponding horizontal row shows which other genes it regulates. Of the 30 genes, all but 3 are involved in the regulation of other genes. Some regularities are evident in the matrix. For example, Genes 1, 8, 13 and 22 all inhibit Genes 29 and 30. It is unlikely that this combination spontaneously arose four times during evolution; rather, these genes probably arose by duplication during the crossover process. Likewise, Genes 4, 9, 10 and 14 all regulate the same genes, as do Genes 6, 18, 19 and 30. Further investigation of these genes reveals that they are not all exactly identical, but some differ in details such as their output function and product placement parameters. It is likely that these differences arose by mutations subsequent to duplication. Other groups of genes are similar but show greater divergence (e.g. Genes 2 and 4). This pattern of gene duplication and divergence is very desirable in evolution, as it allows an evolved function to be retained while variations of the function are simultaneously tested. A similar pattern of duplication and divergence of genes is found in biological evolution, and has been suggested as one of the main mechanisms by which novel structures and organismic complexity evolve [10].

The evolution of the connectivity of the gene regulation matrix was analysed. In particular, the distributions of k_{in} (the number of genes which regulate a given gene) and k_{out} (the number of genes which a given gene regulates) were plotted for the best controllers at various generations throughout the evolutionary run. Recent research has suggested that metabolic networks in a wide range of biological organisms exhibit a power-law distribution for k_{in} and k_{out} (i.e. they are scale-free networks) [11]. Analysis of the evolved GRN controllers showed that the networks had similar power-law distributions for k_{in} and (not so clearly) for k_{out} in early generations, but actually evolved away from these to more multimodal distributions in later generations. Further analysis is underway to elucidate this behaviour.

The pattern of gene expression over time, as recorded in one of the Hydrons during simulation, is shown in Figure 5b. Note that genes with the Step Below Zero or Gradient Below Zero output functions (see Figure 3) are on by default, unless actively inhibited; various genes in this category can be seen. Although most genes show a steady level of activation throughout, some (e.g. Genes 3, 20, 21, 26 and 27) fluctuate significantly over time. Further investigation reveals that Genes 3, 7, 11, 20, 21, 26 and 27 all produce protein 47, which is one of the movement proteins (Section 2.2). Most of these genes are the ones whose expression fluctuates significantly over time, which demonstrates that the controller has evolved a fairly complex pattern of control of the robot's movement in order to achieve the task.

Similar plots for changes in cytoplasmic protein concentrations over time (not shown) reveal that some proteins maintain a fairly steady presence, whereas others vary more significantly as the genes that are producing them are enhanced and inhibited by other proteins. In this particular controller, only one type of protein diffuses between Hydron units over the course of the simulation, but this is sufficient for the directional transfer of information between robots. It is by this mechanism that differentiation of gene expression (and therefore behaviour) among the five Hydrons occurs. Further experiments are underway to study the importance of these inter-Hydron communications for the successful completion of the task.

6 Summary and Future Work

The distributed controller for a group of robots has been presented, inspired by biological genetic regulatory networks. It has been demonstrated that a genetic algorithm can be used to automatically evolve these controllers to achieve a simple task. Initial results and analyses

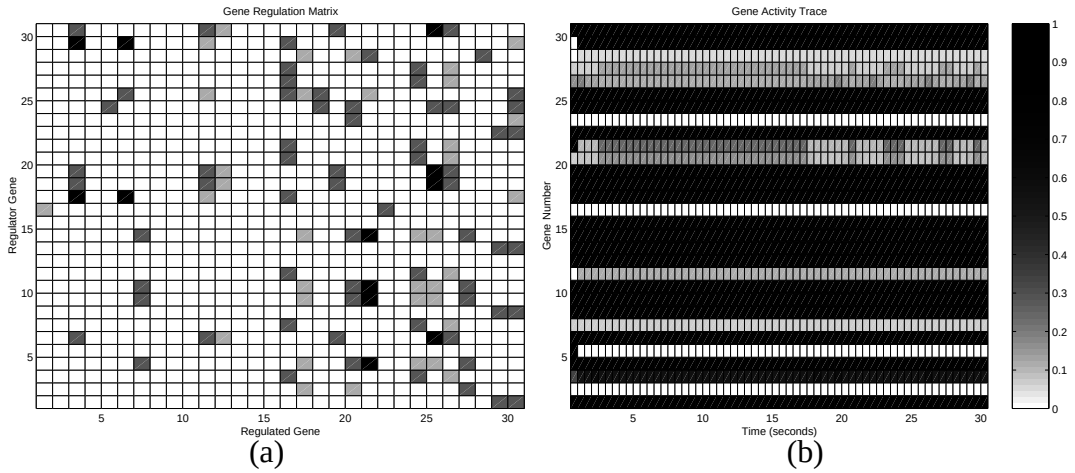

Figure 5: (a) Gene regulation matrix. Each square represents the regulation of the gene along the horizontal axis by the product of the gene on the vertical axis. A white square denotes no regulation, light grey denotes enhancement, dark grey denotes inhibition, and black denotes both enhancement and inhibition. See text for details. (b) Trace of gene activity over time, for the first 30 seconds of controller activity.

of the evolutionary procedure and the controller have been presented. These indicate that the procedure is fairly robust to choices of parameter values, but further, more extensive experiments and analysis are underway to confirm this. The evidence so far suggests that the GRN controller provides a suitable substrate for evolution, and indeed there is evidence that the same kind of duplication and divergence of genes is occurring as happens during biological evolution. Further experiments are underway to analyse the operation of the controller in more detail. An important and ongoing aspect of this analysis is the development of assessment criteria and evaluation metrics for comparing the performance of the algorithm with other types of controller.

One direction of future research is to evolve more complex behaviours. An approach to this would be to evolve a controller that will drive a group of robots into a particular configuration when an external signal (an optical signal masquerading as an incoming protein) is detected. Having evolved a response to one signal, the controller will be evolved to respond differently to another signal. In this way, a user could prompt the group to change configuration simply by supplying the appropriate optical signal.

Another direction is to develop the controller using more biologically-inspired ideas in order to improve its performance. Recent work has demonstrated the advantages of coupling the action of GRNs to physical processes that partially determine the action and interaction of cells [12]. To accommodate these ideas, an extension to the current GRN controller is planned, by coupling it with a system that generates differential attractive forces between Hydron units (following work on artifical cellular adhesion molecules, or A-CAMs [13]). In this way, the GRN is controlling the expression of A-CAMs on a Hydron's surface rather than directly controlling the hardware actuation systems. This would make the hybrid controller more general by abstracting away from the details of the Hydron's design, and would also be expected to work better, because the GRN would be fine-tuning a pre-existing self-organising dynamics (provided by the A-CAMs) rather than trying to control each and every movement.

Finally, the development of the Hydron hardware will soon be complete. Experiments will then commence on transferring controllers evolved in simulation onto the physical robots.

References

[1] Jeffrey S. Rosenschein, Michael Wooldridge, Tuomas Sandholm, and Makoto Yokoo, editors. *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM Press, 2003.

[2] Julien Nembrini, Alan Winfield, and Chris Melhuish. Minimalist coherent swarming of wireless networked autonomous mobile robots. In *Proceedings of the 7th International Conference on Simulation of Adaptive Behaviour (SAB 2002)*, 2002.

[3] Jens Wawerla, Gaurav S. Sukhatme, and Maja J. Matarić. Collective construction with multiple robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002)*, 2002.

[4] Jakob Fredslund and Maja J. Matarić. A general algorithm for robot formations using local sensing and minimal communication. *IEEE Transactions on Robotics and Automation, special issue on Multi Robot Systems*, 2002.

[5] E. Şahin, T.H. Labella, V. Trianni, J.-L. Deneubourg, P. Rasse, D. Floreano, L. Gambardella, F. Mondada, S. Nolfi, and M. Dorigo. SWARM-BOT: Pattern formation in a swarm of self-assembling mobile robots. In A. El Kamel, K. Mellouli, and P. Borne, editors, *Proceedings of the IEEE Int. Conf. on Systems, Man and Cybernetics*, 2002.

[6] E. Peter Greenberg. Bacterial communication: Tiny teamwork. *Nature*, 424:134, 2003.

[7] Torsten Reil. Dynamics of gene expression in an artificial genome – implications for biological and artificial ontogeny. In D. Floreano, J.-D. Nicoud, and F. Mondada, editors, *Proceedings of the 5th European Conference on Artificial Life (ECAL'99)*, number 1674 in Lecture Notes in Artificial Intelligence. Springer, 1999.

[8] J.C. Bongard. Evolving modular genetic regulatory networks. In *Proceedings of the IEEE 2002 Congress on Evolutionary Computation (CEC2002)*, pages 1872–1877. IEEE Press, 2002.

[9] Tom Quick, Chrystopher L. Nehaniv, Kerstin Dautenhahn, and Graham Roberts. Evolving embodied genetic regulatory network-driven control systems. In W. Banzhaf, T. Christaller, P. Dittrich, J. T. Kim, and J. Ziegler, editors, *Proceedings of the 7th European Conference on Artificial Life (ECAL'2003)*. Springer Verlag, 2003.

[10] John Maynard Smith and Eörs Szathmáry. *The Major Transitions in Evolution*. W.H. Freeman, Oxford, 1995.

[11] H. Jeong, B. Tombor, R. Albert, Z.N. Oltval, and A.-L. Barabási. The large-scale organization of metbolic networks. *Nature*, 407:651–654, 2000.

[12] Peter Eggenberger Hotz. Genome-physics interaction as a new concept to reduce the number of genetic parameters in artificial evolution. In *Proceedings of the IEEE 2003 Congress on Evolutionary Computation (CEC2003)*, 2003.

[13] Peter Ottery and John Hallam. Steps toward self-reconfigurable robot systems by modelling cellular adhesion mechanisms. In *Proceedings of the 8th Conference on Intelligent Autonomous Systems (IAS-8)*, 2004.