An approach to time- and space-differentiated pattern formation in multi-robot systems

Tim Taylor[1], **Peter Ottery**[1] **and John Hallam**[2,1]

[1]Institute of Perception, Action and Behaviour,
School of Informatics, University of Edinburgh,
JCMB, The King's Buildings, Mayfield Road,
Edinburgh EH9 3JZ, U.K.

[2]The Maersk Mc-Kinney Moller Institute,
University of Southern Denmark,
Campusvej 55,
DK-5230 Odense M, Denmark

tim@tim-taylor.com

Abstract

We consider the problem of non-trivial pattern formation in decentralised multi-robot systems, and, in particular, how to achieve time- and space-varying behaviour. To tackle the problem, we explore the idea of evolving the fine-level regulation of an underlying self-organising controller. Results from simulation show the promise of the approach: we demonstrate a robot cluster that can stably maintain two different spatial patterns, switching between the two upon sensing an external signal; we also demonstrate a cluster in which individual robots develop differentiated states despite having identical controllers (which could be used as a starting point for functional specialisation of robots within the cluster). The controller was developed with a particular hardware platform in mind—the underwater HYDRON robots developed by the HYDRA consortium (an EU Fifth Framework project). We discuss the implementation of the controller on this and other multi-robot platforms comprising free-moving individual robots, and suggest possible simplifications of the design. This work could eventually have applications in various situations that require robust, complex self-organising behaviour in a collection of free-moving robots, e.g. in space, underwater and nano-scale systems.

1. Introduction

There has been much interest in decentralised multi-robot systems in recent years, due to their potential advantages in many applications over more traditional, monolithic architectures (Arai et al., 2002). The goal is to design systems that can accomplish their tasks more reliably, faster and/or cheaper than could be achieved by a single more complex robot. The general challenge is to develop controllers for the individual robots such that the group as a whole performs the desired higher-level task through the co-ordinated action of the individuals.

In the current work we are specifically interested in multi-robot systems comprising a large number of fairly simple, free-moving robots with limited individual capacity for sensing, actuation and communication. The target hardware is described in Section 2. A variety of decentralised controllers for free-moving systems have been proposed in the literature, e.g. (Holland and Melhuish, 1999, Fredslund and Matarić, 2002, Nembrini et al., 2002, Şahin et al., 2002, Quinn et al., 2003). In contrast to these previous studies, the present work concentrates on providing an underlying self-organising system for robust pattern formation (a similar approach was suggested by (Spears and Gordon, 1999)). We advance on previous work by extending the complexity of the goal tasks. This is achieved by allowing the behaviour of individual robots to be influenced by communication from neighbouring robots or by detection of signals from the environment. We use a genetic algorithm to evolve controllers to perform particular tasks.

There is insufficient space in this paper to provide all the details of the system; the interested reader is referred to (Taylor et al., 2007) for full details. Here we highlight the general approach taken, describe some representative results, and discuss possible simplifications of the approach and potential applications.

2. Robot Hardware and Simulation

This work was conducted as part of a project to develop distributed controllers for a group of real underwater robots called HYDRONs (see (Taylor, 2004, Østergaard et al., 2005)). Each HYDRON unit is a sphere of approximately 11cm diameter, with control of translational movement in three dimensions (provided by a system for impelling/ejecting water for horizontal movement, and a buoyancy control system for vertical movement). The units also have integrated depth

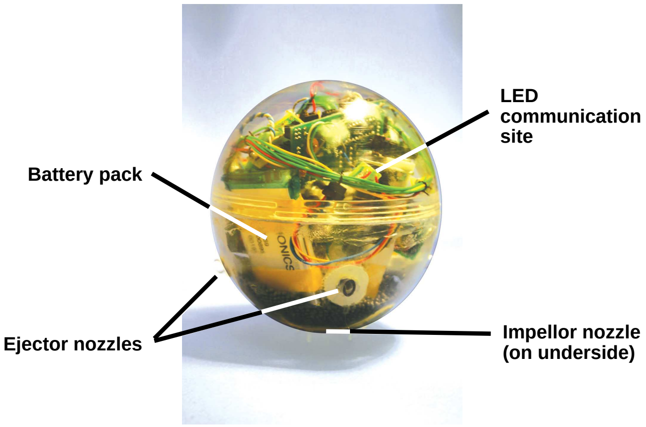

Figure 1: A single HYDRON robot

sensors. Inter-unit communication is provided by a set of eight directional optical transmitters and receivers arranged around the HYDRON's surface. Computing power is provided by an on-board Intel 33MHz 386EX processor with 512Kb RAM and 1Mb Flash disk. Power for all on-board systems is provided by two lithium-ion-polymer battery packs, making the units completely autonomous. The hardware design is shown in Figure 1. We did not have access to the finished hardware at the time the experiments reported here were conducted; the current experiments were therefore conducted in simulation, with the expectation that future work will be conducted on the real robots. For speedy execution we used a simplified two-dimensional simulation (see (Ottery and Hallam, 2004, Ottery, 2006)), although this still models fluid forces on the robots with reasonable accuracy.[1] The sensory, communication and actuation abilities of the simulated robots are based upon the specifications of the real HYDRONs, although in the 2D case we model four communication sites, rather than eight, equally spaced around the robot's circumference.

3. Controller Design

The approach reported here came about through the combination of two previous lines of research on biologically-inspired controllers. One line concerned the evolution of Genetic Regulatory Network (GRN) controllers that interfaced directly with the robots' control systems, and the other concerned a Cellular Adhesion Molecule (CAM)-based controller for pattern formation. In this section we summarise the designs of the individual GRN and CAM controllers (Sections 3.1 and 3.2 respectively), then describe the combined controller (Section 3.3). Full details are given in (Taylor et al., 2007).

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

Table 1: Robot capabilities used by the controller. Square brackets indicate capabilities used in 3D simulations, but not in the 2D simulations reported here.

3.1 GRN Controller

The Genetic Regulatory Network (GRN) controller was a development of previous work by (Bongard, 2002), adapted for free-moving multi-robot systems. The two major factors that influenced the controller design were the specifications of the real hardware, and the analogy to biological genetic regulatory networks. It was assumed that each robot has a basic set of sensory, signalling and actuation capabilities. The particular set of capabilities assumed in the current work—which match the specifications of the real HYDRON units—is detailed in Table 1, although the design of the controller is such that the set can be easily modified.

The controller for each robot comprises:

- A *genome*: a variable length string of digits which may encode information about a number of *genes*.
- A *cytoplasm*: this contains a variety of *proteins* located at discrete *diffusion sites*. The locations of these sites correspond with the optical communication sites on the robot's surface, so the real robots have eight, and the 2D simulated robots used in these experiments have four.

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 robot's sensory, signalling and actuation capabilities. When a gene produces a protein, it is released into the cytoplasm at one of the diffusion sites (the specific site of deposition being under genetic control). A subset of protein types is able to diffuse from the diffusion sites into the cytoplasm of neighbours (a process implemented by the inter-robot communication system); in this way, behaviour can be influenced by the activity of neighbouring robots. A summary diagram of the controller design is shown in Figure 2.

3.2 CAM Controller

The Cellular Adhesion Molecule (CAM) controller was inspired by the robust self-organisation which is observed

[1] Subsequent work with the CAM controller has demonstrated that the self-organising behaviour produced in the 2D model transfers successfully to a 3D model (Ottery, 2006).

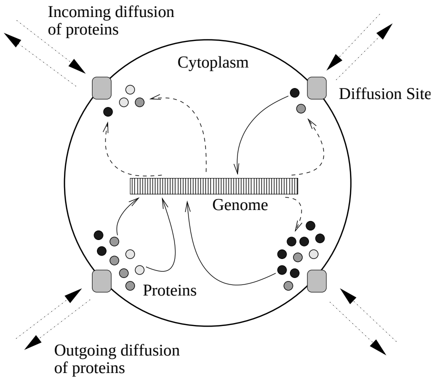
Incoming diffusion of proteins
Cytoplasm
Diffusion Site
Genome
Proteins
Outgoing diffusion of proteins

Figure 2: Schematic of the GRN Controller design.

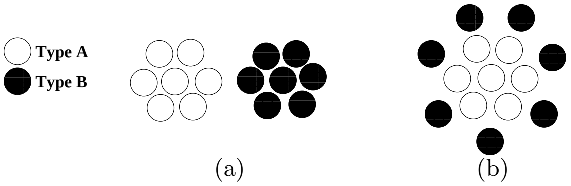

Figure 3: Two possible equilibrium states of a heterogeneous aggregate as given by the Differential Adhesion Hypothesis.

when aggregates of certain cell types are combined. Instead of remaining distributed at random, a degree of sorting takes place, causing the cells to form hierarchical patterns of coherent homotypic cells.

An explanation for this behaviour was proposed by (Steinberg, 1963). His "differential adhesion hypothesis" assumes that cells rearrange to minimise their free energy and thus form the most thermodynamically stable configuration. For example, if two cell types (A and B) are mixed, initially the bonding between the cells will be random. However, if the bonds are of different strengths, then a gradual selection of the strongest bonds will cause the cells to rearrange into a more stable configuration. If homotypic (A-A, B-B) bonds are stronger than heterotypic bonds (A-B), the cells will attempt to sort into pure populations of the different types (Figure 3(a)). If, however, the A-A bonds are stronger than the A-B bonds, which are in turn stronger than the B-B bonds, then the A cells will migrate centrally while the B cells form a shell around them (Figure 3(b)).

The CAM controller approximates this behaviour by using the robots' communication systems to model virtual membranes and adhesions between these membranes. The adhesions are modelled by Artificial Cellular Adhesion Molecules (A-CAMs) distributed on the membranes. The resulting attractions and repulsions experienced by the units are then modelled as physical movements which drive the virtually bonded aggregate to some desired equilibrium configuration. Full details of the system can be found in (Ottery, 2006).

3.3 Combined GRN-CAM Controller

The GRN and CAM controllers were combined as shown schematically in Figure 4. The design of both controllers was essentially unchanged; the only major difference is that the set of proteins in the GRN controller that act as an interface to the robot's actuation system now determine the expression of specific A-CAMs on the robot's virtual membrane, rather than directly controlling the robot's movement.

Specifically, 12 proteins are associated with the level of expression of unique A-CAMs on the robot's virtual membrane. Therefore, the GRN actually varies the adhesion between neighbouring robots and any real actuation is generated as a result of modelling the adhesions. The bonding relationships between the A-CAMs are predefined such that four of them are capable of homotypic bonding while the remaining eight are capable of heterotypic bonding. This provides eight different bonding relationships of various strengths for the GRN to exploit.

Furthermore, in the GRN controller, particular types of sensory protein were produced in the cytoplasm when the robot was in close vicinity to another robot. In the combined controller these proteins are now produced in proportion to the number of A-CAMs on the membrane that are currently bound to A-CAMs on neighbouring membranes. This is achieved by linking the 12 A-CAM types to the concentration of 12 unique proteins. When one of the A-CAM types bonds with an A-CAM expressed on another robot's membrane, the protein's concentration is increased by a level proportional to the number of bonds.

4. Evolution of Controllers

4.1 The Genetic Algorithm

To produce controllers which will cause a group of robots to achieve a particular task, a genetic algorithm (GA) was employed to evolve a population of genomes. A 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).

4.2 Smart Mutations

The successful evolution of a GRN with non-trivial dynamics requires that (some of) the proteins in the cytoplasm bind to the regulatory regions of other genes; it is only in this way that proteins can act as signals to influence the dynamics of the robot, thereby allowing the creation of complex regulatory networks. There is therefore a matching problem during evolution—to successfully evolve a new regulatory gene it is insuffi-

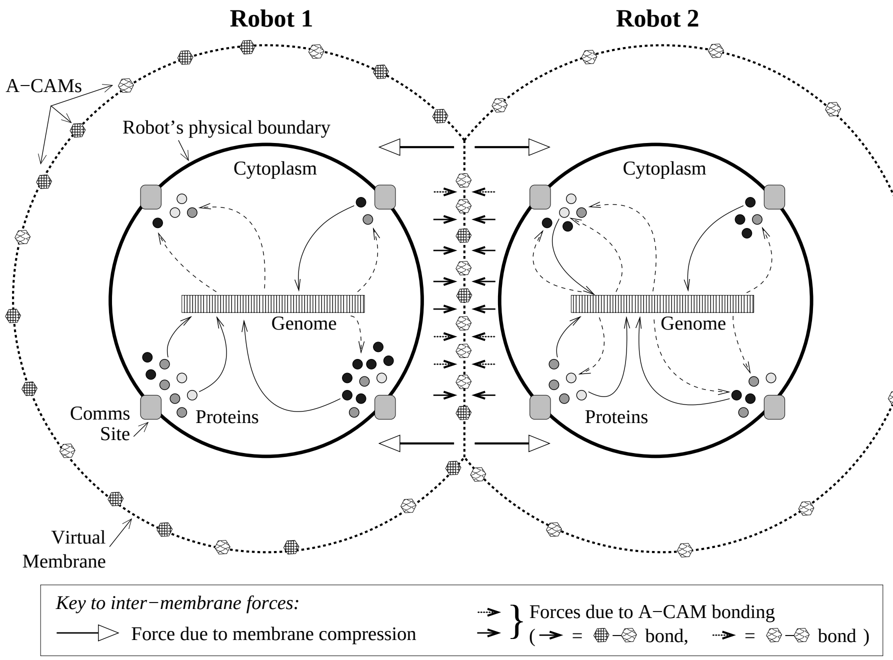
Robot 1
Robot 2
A−CAMs
Robot's physical boundary
Cytoplasm
Cytoplasm
Genome
Genome
Proteins
Proteins
Comms Site
Virtual Membrane
Key to inter−membrane forces:
Force due to membrane compression
Forces due to A−CAM bonding
(= bond, = bond)

Figure 4: Schematic of the combined GRN-CAM controller design

cient just to add a new gene to the genome; its product must actually be able to bind to the regulatory region of at least one other gene if it is to have any effect. In the original design of the GRN controller (Taylor, 2004) this correspondence between gene products and regulatory regions could only evolve by fortuitous mutations. However, if we are willing to relax the biological realism of the analogy, we can improve the GA to make these correspondences much more likely to occur. This is achieved by introducing an additional type of mutation, called a *smart mutation*; these ensure that, in general, genes are responsive to the specific proteins that are produced in the cytoplasm during the operation of the controller, rather than to randomly chosen proteins which may never actually be expressed. Further details are given in (Taylor et al., 2007).

4.3 Fitness Evaluation

To evaluate a controller, it was copied (with identical initial cytoplasmic state) into each of the robots in the group. The robots were allowed to run for a fixed duration, and the overall behaviour of the group was then evaluated in terms of the given task. In these experiments we used a group size of 36 robots, and each experiment was run on three separate trials starting from different initial configurations. The final fitness assigned to the genome was the mean fitness from the three trials. A series of tasks was designed to compare the quality of solutions when using the GRN-only controller (Section 3.1) and when using the combined GRN-CAM controller (Section 3.3).[2] Two tasks are reported here, described in Sections 5.1–5.2. Five evolutionary runs were performed for each of the four conditions (2 tasks and 2 controller types). Full details of the experiments' configuration are given in (Taylor et al., 2007).

5. Experiments

In each of the experiments described below, a sample of five evolutionary runs was carried out for both the GRN-only controller and the combined controller.

During the execution of each task it is clearly desirable that, at the very least, the robots maintain a single connected aggregate (i.e. each robot must always be within communication range of at least one other). Therefore, the fitness value awarded to each controller (F_t) was divided into two separate components. The first of these (F_1) simply reflects the proportion of the simulation during which the connected aggregate was maintained:

$$F_1 = \frac{T_M}{T_E} \qquad (1)$$

where T_E is the total evaluation time and T_M is the time the connected aggregate was maintained for. The second (F_2) was the fitness score achieved in relation to the specific task (defined in Sections 5.1–5.2), and therefore was only awarded when the aggregate was maintained for the entire simulation period. These components were given relative weightings of 0.1 and 0.9 respectively,[3] giving

[2] An extensive investigation into the properties of the CAM controller by itself has already been reported in (Ottery, 2006).

[3] These weightings were selected experimentally.

the following two cases:

$$F_t = \begin{cases} 0.1F_1 & \text{if aggregate not maintained} \\ 0.1 + 0.9F_2 & \text{if aggregate maintained for the} \\ & \text{duration of simulation } (F_1 = 1) \end{cases} \tag{2}$$

As mentioned above, to determine a more general fitness value, each controller was evaluated from three different initial configurations, and the individual fitness scores were averaged to give an overall fitness.

5.1 Response to External Signals

This task was designed to investigate whether the controllers could easily switch between different functions in response to some environmental trigger. In practice, such a trigger could take the form of a change in any environmental condition that the robots are able to detect and would result in a corresponding change in concentration of one or more sensory protein(s). Therefore, this was represented in the simulations by increasing the concentration of a small number of proteins in each robot's cytoplasm at some random time T_S, within a predefined period. The fitness value for the task was based on the robot aggregate's ability to demonstrate different behaviours before and after the signal was introduced. In this case, initially the aggregate was rewarded for maintaining a constant minimum enclosed area. However, once the signal was introduced, it was instead rewarded for maintaining a constant maximum enclosed area. The fitness function therefore had two components relating to the difference between the area values D and the variance V of each period of the simulation.

$$m_a = \frac{1}{T_S}\sum_{i=0}^{T_S} a_i \quad (3) \qquad m_b = \frac{1}{T_E - T_S}\sum_{i=T_S+1}^{T_E} a_i \quad (4)$$

$$D = \frac{m_b}{m_a + m_b} \tag{5}$$

where a_i is the area of the aggregate's concave hull at time i and T_E is the full evaluation time.

$$v_a^2 = \frac{1}{T_S}\sum_{i=0}^{T_S}(a_i - m_a)^2 \tag{6}$$

$$v_b^2 = \frac{1}{T_E - T_S}\sum_{i=T_S+1}^{T_E}(a_i - m_b)^2 \tag{7}$$

$$V = \frac{min(v_a v_b, k_v m_a m_b)}{k_v m_a m_b} \tag{8}$$

where the constant k_v is used to control the maximum level of variance that impacts on the reward (0.25 in these experiments). As both D and V have different levels of importance the final fitness is defined as:

$$F_2 = k_f D + (1 - k_f)(1 - V) \tag{9}$$

where k_f determines the relative weighting.

In the simulations presented in this work the evaluation period was set to 100s and the signal was introduced at a random time in the period 50 ± 20s. In addition, the control parameter k_f was set to 0.75 to add extra weight to the level of change in the aggregate's behaviour.

5.2 Spatial Differentiation of Function

Differentiation is the process which allows a collection of initially homogeneous robots to divide into a number of distinct subsets, each of which is capable of performing a more specific function. The aim of this task was to determine whether the controllers could achieve this by generating two equally sized subsets of robots ($\mathcal{A}$ and $\mathcal{B}$), each exhibiting a unique pattern of protein expression. In the actual simulations, the patterns that were selected involved two ranges of 16 proteins α and β, such that one pattern was achieved by maximising the concentration of the α proteins while minimising the concentration of the β proteins and the second pattern was achieved by performing the converse. To encourage the population to differentiate as quickly as possible, the mean difference between these concentrations during the entire evaluation period was used to calculate the final fitness value. Therefore, it was possible to determine the degree to which any single robot was a member of either $\mathcal{A}$ or $\mathcal{B}$ using the following d_r values:

$$g_a = \frac{1}{U|\alpha|}\sum_{t=0}^{T_E}\sum_{i\in\alpha} c_i^t \quad (10) \qquad g_b = \frac{1}{U|\beta|}\sum_{t=0}^{T_E}\sum_{i\in\beta} c_i^t \quad (11)$$

$$d_r = \frac{1}{2}\left(g_a + (1 - g_b)\right) \tag{12}$$

where c_i^t is the cytoplasmic concentration of protein i at time t, and U is a normalisation constant representing the maximum summed concentration of a single protein over the entire evaluation period.

Therefore, both g_a and g_b and thus d_r will be in the range $[0.0, 1.0]$, with:

$$d_r \begin{cases} > 0.5, & \text{if there are more set } \alpha \text{ proteins than set } \beta \\ = 0.5, & \text{if both sets of proteins are equal} \\ < 0.5, & \text{if there are more set } \beta \text{ proteins than set } \alpha \end{cases} \tag{13}$$

Thus, if we sort the robots by their d_r values and assign the top half to group $\mathcal{A}$ and and bottom half to group $\mathcal{B}$, we can evaluate how well they have achieved the differentiation task with the following fitness function:

$$F_2 = \frac{1}{n}\left(\sum_{i\in\mathcal{A}} d_i + \sum_{j\in\mathcal{B}}(1 - d_j)\right) \tag{14}$$

Therefore, if the robots are completely undifferentiated (all the d_r values are the same), $F_2 = 0.5$. However, if as desired the robots in $\mathcal{A}$ express more of the set α proteins than set β proteins and the reverse is true for the robots in $\mathcal{B}$, $F_2 > 0.5$ indicating that the robots have partially differentiated.

6. Results and Analysis

6.1 Controller Evolution

The controller evolution results for both controllers are shown in Figure 5. In each case, the mean and maximum fitness values are given for each generation of a full evolutionary run. Most significantly, the results show that the combined GRN-CAM controller achieved better final fitness values for both tasks. In addition, it also appears far easier to find solutions for the combined controller, which often starts at generation 0 with a higher fitness than the best controllers evolved for the GRN by itself. Additionally, the performance of the combined controller is more consistent, as indicated by the smaller error bars.

In the signal response task, the combined GRN-CAM controller finds a solution relatively quickly (even though the CAM-only controller would have been unable to solve the task). Figure 6 shows screen shots taken at regular intervals from a sample simulation—these show that the cluster is performing the task well. In contrast, the evolved GRN-only controllers show no noticeable change in behaviour, and in many cases the aggregate even fails to remain connected (data not shown).

In the differentiation task, it can be seen that the GRN-CAM controller rapidly achieves differentiation, as indicated by the fitness in excess of 0.55.[4] The fitness continues to improve over the run; the magnitude of fitness improvement is very small, but this is mainly due to the very large upper bound (U) used in the fitness calculation (see Section 5.2). An example of the evolved differentiation behaviour from a GRN-CAM controller is shown in Figure 7. In contrast, the GRN-only controller fails to find any solution.

6.2 Controller Quality

The best controllers obtained for each of the tasks were subjected to a series of tests for scalability with larger numbers of robots, specificity to initial configuration, and ability to cope with perturbations in the form of random forces and torques applied to the individual robots. The GRN-CAM controller outperformed the GRN-only controller, showing a better level of performance, and more consistency, in almost all cases (data not shown; full details in (Taylor et al., 2007)).

7. Discussion

The reported experiments demonstrate that we have succeeded in our goal of moving beyond static pattern formation; we have produced decentralised controllers that show both temporally-differentiated patterns (Figure 6) and spatially-differentiated patterns (Figure 7).

The controller quality experiments also demonstrate that the GRN-CAM controllers, compared to the GRN-only controllers, are more robust to environmental perturbations and scale better with larger clusters of robots. This is not a great surprise, because the GRN-CAM system has an underlying self-organising behaviour that allows the robots to better cope with environmental disturbances. This leads us to suspect that the GRN-CAM controllers evolved in simulation would also transfer to hardware better than would the GRN-only controllers.

Regarding the CAM-only controller, the addition of the GRN allows us to evolve robots that change their adhesive properties according to changes in protein concentrations (which may be due to the internal dynamics of the GRN controller, or to the detection of external signals). The combined GRN-CAM controller, unlike the CAM-only controller, can therefore produce both time- and space-differentiated patterns.

The described controllers were designed with the HYDRON hardware in mind, and could be implemented using the available on-board computing, sensing and actuation systems. An attractive feature of the GRN-CAM controller is that the GRN interfaces with the underlying CAM model rather than directly with the hardware—the CAM model provides an abstraction layer that hides the details of the hardware. A GRN-CAM controller could, in principle, therefore be transferred to any robotic system that implemented the CAM model, which requires only modest computing and communication capabilities.

Historically, the GRN-CAM controller came about by the fusion of two previous projects. In retrospect, some aspects of the controller, particularly those related to the GRN, appear overly complex. It is possible that a much simpler system (e.g. a finite state machine) might be able to achieve similar control of the underlying CAM system, at least for the kinds of tasks considered here.

Looking forward, this system, like any other type of evolved controller, will face issues of how to structure the evolutionary process to achieve progressively more complex behaviours. However, the demonstrated ability of individual robots to adopt stable differentiation of state could serve as a good starting point from which to build more complex functional specialisation.

The described approach of evolving the fine-level regulation of an underlying self-organising system has potential applications in any system of free-moving robots that is required to display a repertoire of robustly-maintained spatial patterns, with the ability to switch between patterns on demand. The most obvious underwater ap-

[4] Using only the differentiation fitness function a value of > 0.5 indicates differentiation. However, in the final fitness value this increases to 0.55 (i.e. $0.5 \times 0.9 + 0.1$, see Equation 2).

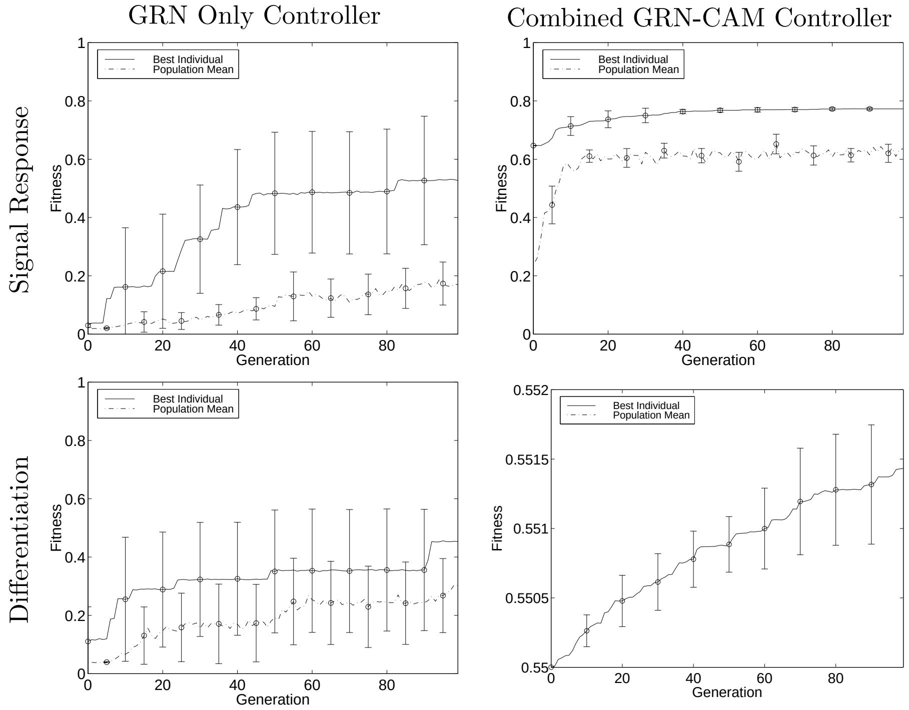

Figure 5: Mean and maximum fitness over time the four evolution conditions. For each evaluation, fitness is averaged over three different starting positions. Each experiment was repeated 5 times. (Note the different scale on the bottom-right graph. In this experiment, the population mean increased gradually but remained below 0.55 hence is not visible on the graph.)

plication is in the deployment of sensor networks for ocean sampling, where the system offers an alternative to more traditional, hand-designed controllers (e.g. (Leonard et al., 2007, Gerasimov et al., 2006)). Other potential application areas include nano-scale systems (where the CAM model may have to be implemented with physical forces rather than computationally), and, perhaps more likely in the shorter-term, on micro-satellite and unmanned aerial vehicle (UAV) systems.

References

Arai, T., Pagello, E., and Parker, L. E. (2002). Guest editorial: Advances in multirobot systems. *IEEE Transactions on Robotics and Automation*, 18(5):655–661.

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

Fredslund, J. and Matarić, M. J. (2002). A general, local algorithm for robot formations. *IEEE Trans.on Rob. and Auto., Special Issue on Multi-Robot Systems*, 18(5):837–846.

Gerasimov, V., Healy, G., Prokopenko, M., Wang, P., and Zeman, A. (2006). Symbiotic sensor networks in complex underwater terrains: A simulation framework. In Gabrys, B., Howlett, R. J., and Jain, L. C., (Eds.), *KES (3)*, volume 4253 of *Lecture Notes in Computer Science*, pages 315–323. Springer.

Holland, O. and Melhuish, C. (1999). Stigmergy, self-organisation, and sorting in collective robotics. *Artificial Life*, 5(2):173–202.

Leonard, N., Paley, D., and Lekien, F. et al. (2007). Collective motion, sensor networks, and ocean sampling. *Proceedings of the IEEE*, 95(1):48–74.

Nembrini, J., Winfield, A., and Melhuish, C. (2002). Minimalist coherent swarming of wireless networked autonomous mobile robots. In *Proceedings of SAB*, pages 373–382.

Østergaard, E. H., Christensen, D. J., Eggenberger, P., Taylor, T., Ottery, P., and Lund, H. H. (2005). Hydra: From cellular biology to shape-changing artefacts. In *Proc 15th Int Conf Art Neural Nets (ICANN 2005)*, pages 275–281. Springer.

Ottery, P. (2006). *Using Differential Adhesion to Control Self-Assembly and Self-Repair of Collections of Modular Mobile Robots*. PhD thesis, School of Informatics, Edinburgh Univ.

Ottery, P. and Hallam, J. (2004). Steps toward self-reconfigurable robot systems by modelling cellular adhesion mechanisms. In Groen, F., Amato, N., Bonarini, A., Yoshida, E., and Kröse, B., (Eds.), *Proceedings of IAS8*, pages 720–728. IOS Press.

Quinn, M., Smith, L., Mayley, G., and Husbands, P. (2003). Evolving controllers for a homogeneous system of physical robots: Structured cooperation with minimal sensors. *Phil. Trans. Royal Soc. London, Series A*, 361:2321–2344.

Şahin, E., Labella, T., and Trianni, V. et al. (2002). SWARM-BOT: Pattern formation in a swarm of self-assembling mobile robots. In El Kamel, A., Mellouli, K., and Borne, P., (Eds.), *Proc. IEEE Int. Conf. on Systems, Man and Cybernetics*.

Spears, W. M. and Gordon, D. F. (1999). Using artificial physics to control agents. In *IEEE International Conference of Information, Intelligence and Systems*.

Steinberg, M. S. (1963). Reconstruction of tissues by dissociated cells. *Science*, 141(3579):401–408.

Taylor, T. (2004). A genetic regulatory network-inspired real-time controller for a group of underwater robots. In Groen, F., Amato, N., Bonarini, A., Yoshida, E., and Kröse, B., (Eds.), *Proceedings of IAS8*, pages 404–412, Amsterdam. IOS Press.

Taylor, T., Ottery, P., and Hallam, J. (2007). Pattern formation for multi-robot applications: Robust, self-repairing systems inspired by genetic regulatory networks and cellular self-organisation. Technical Report EDI-INF-RR-0971, School of Informatics, University of Edinburgh, http://www.inf.ed.ac.uk/publications/report/0971.html.

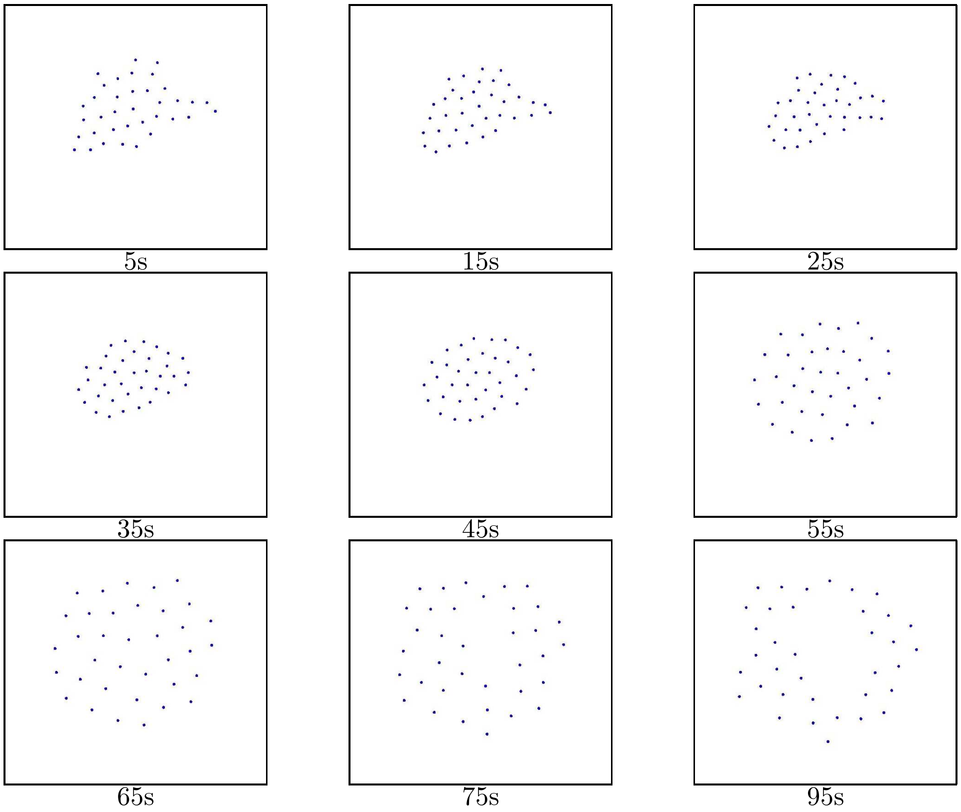

Figure 6: Screen shots from the signal response task with the GRN-CAM controller. The signal was introduced at 40s. Before the signal, the cluster's task was to adopt a configuration with minimum area. After the signal, the task was to adopt a configuration with maximum area while remaining locally connected.

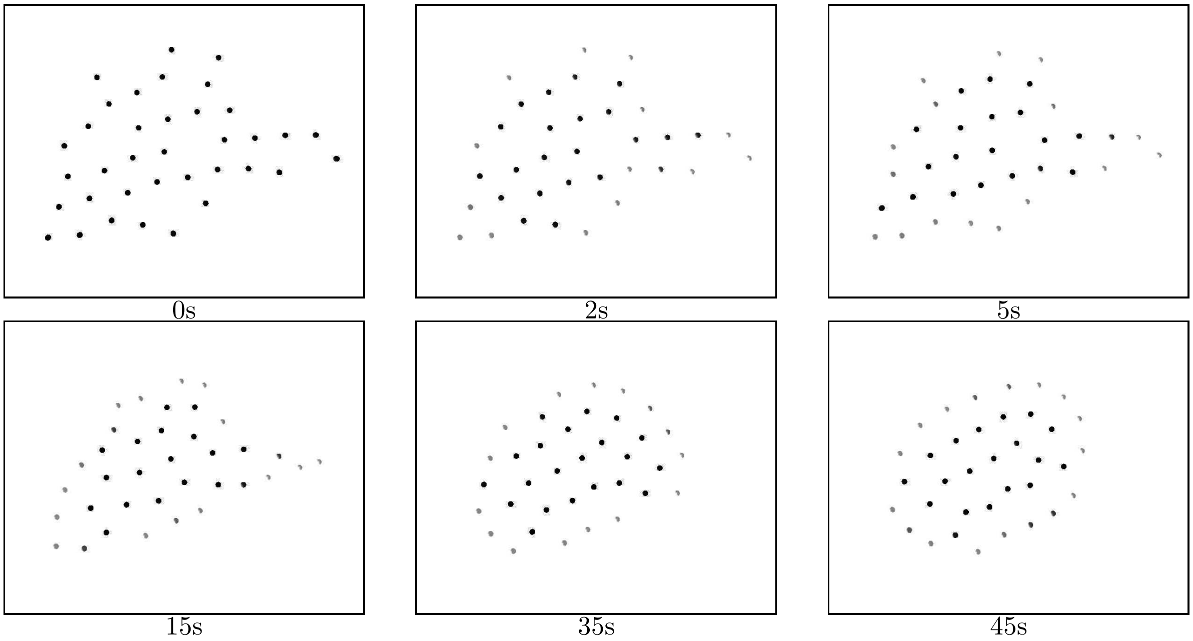

Figure 7: Screen shots from the differentiation task with the GRN-CAM controller. Shading indicates each robot's d_r value (dark is high, light is low). The task of the cluster was to differentiate such that half the robots had high d_r values and half had low values. The task is successfully completed by the end of the simulation: the central robots have high d_r values (dark shading) and the peripheral robots have low d_r values (light shading).