

Informatics Research Report

**Pattern formation for multi-robot applications:
Robust, self-repairing systems inspired by
genetic regulatory networks and cellular
self-organisation**

Running title: Pattern formation for multi-robot applications

Tim Taylor[†], Peter Ottery[†] and John Hallam^{‡†}

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

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

**Corresponding author:
Tim Taylor
email: tim@tim-taylor.com**

January 30, 2006

Abstract

This work concerns a biologically-inspired approach to self-assembly and pattern formation in multi-robot systems. In previous work the authors have recently studied two different approaches to multi-robot control, one based upon the evolution of controllers modelled as genetic regulatory networks (GRNs), and the other based upon a model of self-organisation in aggregates of biological cells mediated by cellular adhesion molecules (CAMs). In the current work, a hybrid GRN-CAM controller is introduced, which captures the advantages, and overcomes the disadvantages, or both of the original controllers; it combines the adaptability of the evolutionary process with the robustness of an underlying self-organising dynamics. The performance of the new controller is investigated and compared with the previous ones. For example, one experiment involves the evolution of a robot cluster that can stably maintain two different spatial patterns, switching between the two upon sensing an external signal. Another experiment involves the evolution of 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 results show that the combined GRN-CAM controller is more flexible and robust than either the GRN controller or the CAM controller by itself, and can produce more complex spatiotemporal behaviours. The GRN-CAM controllers are also potentially portable to robotic systems other than those for which they were evolved, as long as the new system implements the underlying CAM model of self-organisation. Some technical issues regarding the implementation of the GRN and joint GRN-CAM systems are also discussed, including the use of “smart mutation” operators to improve the speed of evolution of GRNs, and evolving the rate of dynamics of the GRN controller to suit the particular task in hand.

Keywords: Multi-robot control, pattern formation, genetic regulatory networks, cellular adhesion molecules, evolution, self-organisation

1 Introduction

The work described here is part of our ongoing research into automated techniques for producing distributed controllers for multi-robot systems (see [Østergaard et al., 2005]).

There has been much interest in 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 achieve their task more reliably, faster and/or cheaper than could an individual robot. The general challenge for multi-robot control is to develop controllers for the individual robots such that the group as a whole is able to coordinate to perform the desired higher-level task.

In the current work we are specifically interested in multi-robot systems comprising a large number of fairly simple robots with limited individual capacity for sensing, actuation and communication. The target hardware is described in Section 2.

Several studies have been reported recently on the design of distributed controllers for systems of this type (e.g. [Holland and Melhuish, 1999, Nembrini et al., 2002, Wawerla et al., 2002, Fredslund and Matarić, 2002, Şahin et al., 2002]). However, most of these controllers are specific for a particular task; to perform a different task the controller would have to be redesigned by hand.

In previous work, we have investigated the use of genetic algorithms to evolve controllers for the robots, rather than designing them by hand [Taylor, 2004]. The controllers were partially inspired by the mechanisms for coordinated action in bacteria [Greenberg, 2003]; their design was based upon genetic regulatory networks (GRNs), which have a number of desirable properties in this context (e.g. the existence of multiple stable attractors in their dynamics, the ability to differentiate, and a high degree of robustness [Reil, 1999]). Other work has employed evolutionary algorithms to generate neural network controllers for multi-robot systems (e.g. [Quinn et al., 2003, Trianni et al., 2004]). Although each of these studies has achieved some success, there is no guarantee that these evolutionary methods will produce particularly general controllers which continue to work in conditions unseen during the evolutionary process (e.g. a change in the number of robots, or an increase in environmental perturbations).

In other recent work, we have also experimented with an alternative approach to developing distributed controllers for pattern formation [Ottery and Hallam, 2004]. This approach is modelled on the self-organising properties of biological cells mediated by the selective expression of cellular adhesion molecules (CAMs) on a cell's membrane. One type of CAM will selectively adhere to specific other types present on the membranes of neighbouring cells, with groups of cells self-organising to form the most thermodynamically stable configuration. Our CAM controller treats robots as artificial cells with virtual membranes and artificial cell adhesion molecules (A-CAMs). Our experiments have shown that the approach can generate robust pattern formation behaviour with the ability for self-repair.

Both of these approaches to the control of multi-robot systems (i.e. evolution and self-organisation) have distinct advantages, and disadvantages, which are in many ways complementary. In the work described here, we show how the two approaches can be combined in an attempt to capture the advantages, and overcome the disadvantages, of both. The idea is that the GRN controller no longer directly controls a robot’s movements, but instead controls the expression of A-CAMs. From the point of view of the GRN controller, by adding the CAM model we should be able to produce more robust controllers because we are evolving control for a system which already has self-organisational properties. In addition, the design of the controller becomes more portable, because the GRN is interfacing with the CAM model rather than directly with the robot’s actuation systems. From the perspective of the CAM controller, the addition of GRN-mediated expression of A-CAMs allows us to produce more complex patterns of behaviour (both spatially and temporally), including the possibility of reacting to external conditions.

The rest of the paper is organised as follows. A brief description of the target hardware platform, and of the simulation software used in the experiments reported here, is presented in Section 2. The controller design is described in Section 3, including a summary of the original GRN and CAM controllers (Sections 3.1 and 3.2), followed by a description of the combined controller (Section 3.3). Details of the genetic algorithm used to evolve the controllers, including the “smart mutation” mechanism used to improve results, are presented in Section 4. The experiments we have conducted to test the new controller are described in Section 5, followed by results and analysis in Section 6. Finally, a discussion of the results and of future research directions is presented in Section 7. An appendix lists the default parameter values used in the experiments.

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 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 onboard Intel 33MHz 386EX processor with 512Kb RAM and 1Mb Flash disk. Power for all onboard systems is provided by two lithium-ion-polymer battery packs, making the units completely autonomous. At the time the experiments reported here were conducted, the HYDRON hardware had been through a series of prototype stages, and 20 units of the final design were in the process of being manufactured. The final hardware design is shown in Figure 1. As we did not have access to the finished hardware, the current experiments were

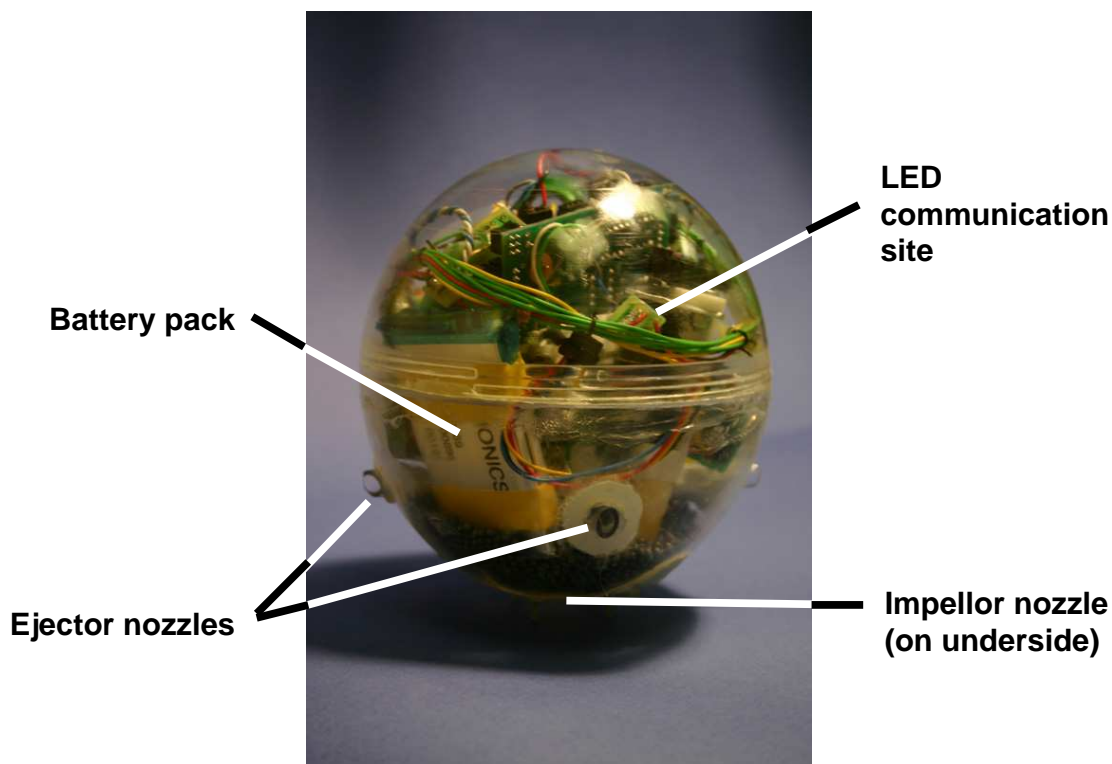


Figure 1: A single HYDRON robot.

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

As mentioned in Section 1, the combined GRN-CAM controller that is the focus of the current study came about through the combination of two previous lines of research. One line concerned the evolution of GRN controllers that interfaced directly with the robots’ control systems, and the other concerned a CAM-based controller for pattern formation. In the following sections 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).

¹Subsequent 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].

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

Table 1: Agent 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 two major factors that influenced the GRN controller design were the specifications of the real hardware, and the analogy to biological genetic regulatory networks. It was assumed that each agent 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 agent comprises:

- A *genome*. This is a variable length integer string 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 robots, 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. Protein action is described in more detail in Section 3.1.2. 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 summary diagram of the controller design is shown in Figure 2. Further details are presented in the following sections.

3.1.1 Genome and Gene Structure

A genome is a variable length string of digits from the set $\{0 \dots 3\}$. A subsection from an example genome is shown in Figure 3. 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,

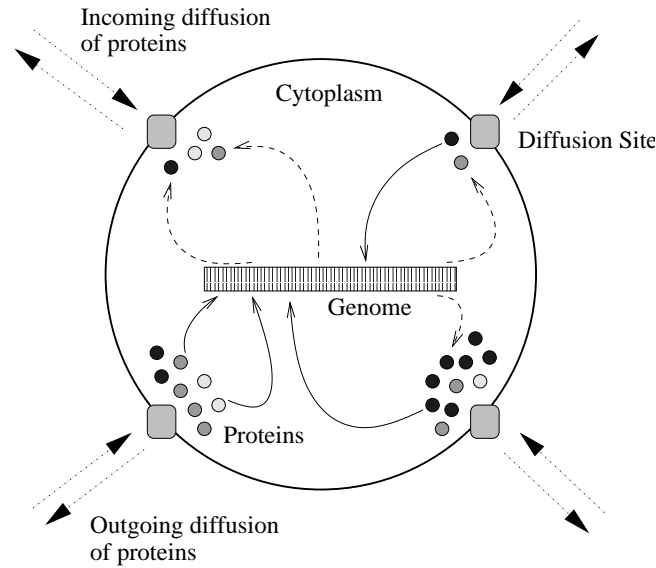


Figure 2: Summary of the GRN Controller design.

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

Four different output functions are available (as shown in Figure 4), 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) 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.

```

      E1      I1      Gene 1
31211030230203010300203302030203020302030
      E2      I2
02032211230322210203300020302110233320321
      I2      Gene 2
23021102332010223022110320302030200033322
      E3      E3      E3      E3      Gene 3
11231111222212300202220111203020101220201
      E4      Gene 4
20321220001110000012010230333022101102013

```

Figure 3: An example of a subsection 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).

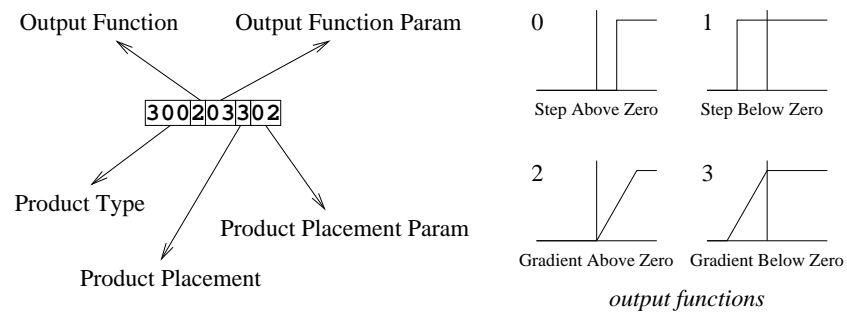


Figure 4: The structure of a gene. In this example, the gene produces protein 48 (300 in base 4 notation) when expressed. It has a Gradient Above Zero output function (2) with gradient 3 (03). The product is deposited at a specific diffusion site (3), which is site number 2 (02).

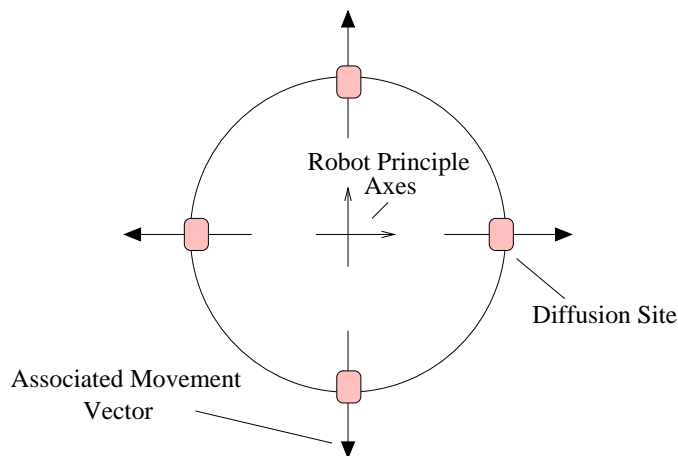


Figure 5: Movement vectors associated with the diffusion sites in the GRN-only controller.

3.1.2 Protein Action

As mentioned in the previous section, proteins are specified in the genome by a string of three digits, so 64 different kinds of protein are available (labelled 0–63).

Any of these proteins can potentially act as an enhancer or an inhibitor of gene expression if they are specified in the regulatory region for a gene. Additionally, some proteins act as interfaces to the physical robot, 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.

Actuation Proteins 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. Specifically, each diffusion site ($d \in \mathcal{D}$) is linked with movement in a particular direction along the principle axes of the robot (see Figure 5). These movement vectors (M) are then scaled by the summed concentration of 16 predefined actuation proteins (\mathcal{A}) at the respective diffusion site giving a desired movement in each direction. A final movement vector (F) is then calculated from these four scaled values, and the HYDRON’s actuation system is activated accordingly:

$$F = \sum_{d \in \mathcal{D}} M_d \sum_{a \in \mathcal{A}} Conc(a, d) \quad (1)$$

In the GRN-only experiments reported in Section 6, proteins 32–47 were used for actuation.

Sensory Proteins The sensing mechanisms made available to the GRN-only controller include external protein diffusion and a local neighbours sensor.² The external diffusion mechanism provides crude information about the distance and direction of neighbouring robots by using the local communication system to allow a specific protein (p) to diffuse between them. This is achieved by associating each diffusion site with one of the four transmitter/receiver pairs located on the robot’s surface. When the externally diffusing protein is expressed at one of the sites (i), a signal is broadcast from the associated transmitter to indicate the amount of the protein that is diffusing from the site (E). The local concentration of the protein (C_p) is also adjusted accordingly:

$$C_p(i, t + 1) = (1 - e)C_p(i, t) \quad (2)$$

$$E_p(i, t) = eC_p(i, t) \quad (3)$$

where e is the external diffusion constant ($0 < e < 1$).

Likewise, when one of the sensors detects an incoming signal from a neighbouring robot, the level of the protein is increased at the sensor’s associated diffusion site (j). The increase is a function of the concentration value encoded in the signal and the distance (d) that the signal has travelled:

$$C_p(j, t + 1) = C_p(j, t) + \frac{k_T}{d^2} E_p(i, t) \quad (4)$$

where k_T is the predefined transmission coefficient (0.0009 in these experiments).

The local neighbours sensor increases the concentration of a specific protein at all the diffusion sites, by an amount proportional to the number of robots (n) within communication range:

$$C_p(t + 1) = C_p(t) + \frac{n}{N} \quad (5)$$

where N scales the value appropriately.³ This sensor is included to help approximate the sensing abilities of the combined GRN-CAM controller such that both controllers have reasonably equivalent abilities.

The specific protein actions for the GRN controller are summarised in Table 2 (left-hand side).

3.1.3 Protein Dynamics

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

$$C_p(i, t + 1) = kC_p(i, t) \quad (6)$$

²In previous experiments with the GRN-only controller using a 3D simulator of the HYDRONS [Taylor, 2004], we also included a depth sensor mechanism. This was not included in the 2D simulations reported here.

³This value can be manually tuned to adjust the sensitivity. A value of 4 was used in these experiments.

GRN-Only Controller		Combined GRN-CAM Controller	
<i>Proteins</i>	<i>Description</i>	<i>Proteins</i>	<i>Description</i>
0	External diffusion protein		
32-47	Actuation proteins	36-47	A-CAM expression proteins
48	Local neighbours sensor	48-59	A-CAM bonding proteins

Table 2: Summary of the specific protein actions in each controller.

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 neighbouring sites j (where the set of neighbours to site i is denoted \mathcal{N}_i , with cardinality $|\mathcal{N}_i|$) is given by:

$$\begin{aligned}
C_p(i, t + 1) &= (1 - d)C_p(i, t) \\
C_p(j, t + 1) &= C_p(j, t) + \frac{dC_p(i, t)}{|\mathcal{N}_i|} \quad (j \in \mathcal{N}_i)
\end{aligned} \tag{7}$$

where d is the internal diffusion constant ($0 < d < 1$). In our experiments, the neighbours of each site were defined as the nearest site in a clockwise direction around the HYDRON's circumference, and the nearest site counterclockwise, so each site has two neighbours.

A subset of proteins also undergoes external diffusion, as explained in the previous section.

Combining all of the above, the overall dynamics of proteins in the cytoplasm can therefore be expressed as follows:

$$\begin{aligned}
C'_p(i, t + 1) &= k(1 - d)(1 - e)C_p(i, t) + \sum_{j \in \mathcal{N}_i} \frac{dC_p(j, t)}{|\mathcal{N}_i|} \\
&\quad + G_p(i, t) + S_p(i, t) + I_p(i, t)
\end{aligned} \tag{8}$$

$$C_p(i, t + 1) = \min(C'_p(i, t + 1), 1) \tag{9}$$

where $G_p(i, t)$ is the total amount of protein p produced at diffusion site i at time t by gene action, $S_p(i, t)$ is the amount produced by sensory proteins (which will be $\frac{n}{N}$ if protein p is a sensory protein, or zero otherwise; see Section 3.1.2), and $I_p(i, t)$ is the total amount diffusing into the agent at site i from external sources (i.e. the sum of all incoming E_p signals if protein p is an external diffusion protein, or zero otherwise; see Section 3.1.2). The *min* function in Equation 9 places a ceiling of 1 on the maximum amount of each type of protein stored at each site.

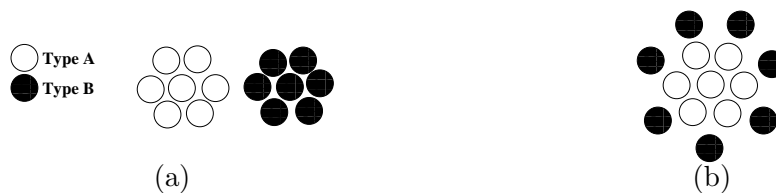


Figure 6: Two equilibrium states proposed by the Differential Adhesion Hypothesis.

3.2 CAM Controller

The CAM controller was inspired by the robust self-organisation which is observed 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.

The leading hypothesis which explains this behaviour is the “differential adhesion hypothesis” proposed by Steinberg [Steinberg, 1963]. This states 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 there exists a difference in the strength of the bonds then a gradual selection of the strongest bonds will cause the cells to rearrange into a more stable configuration. More specifically, 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 6(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 6(b)).

The CAM controller approximates this behaviour by using the communication system of simple robotic units with specifications similar to the HYDRONs (referred to as A-Cells or Artificial Cells), to model virtual membranes and adhesions between these membranes. The resulting attractions and repulsions the A-Cells experience are then modelled as physical movements which drive the virtually bonded aggregate to some desired equilibrium configuration.

The following sections summarise the key aspects of the controller and their relationship to the hardware. A fully detailed description of the system can be found in [Ottery, 2006].

3.2.1 Communication

The communication system employed by the A-Cells is relatively simple. Each A-Cell continually broadcasts a single transmission signal, with a limited range, from each of its transmitters. This signal contains both information relating directly to the broadcasting A-Cell and a list carrying information aimed at individual cells

in the system.

In addition to data transfer the transmission signal also allows A-Cells to determine an approximate position of their neighbours. The intensity of the signal gives a crude indication of the distance it has travelled while the set of detectors that are receiving the signal indicate the general direction of the source.

3.2.2 Membrane Model

One of the most important structures in living cells is the deformable membrane which surrounds the cell nucleus. The CAM controller provides a very simple approximation of such a membrane by using the A-Cells' communication system to create a virtual force field which extends some distance from the actual A-Cell hull. Like a fluid membrane this force field behaves like a viscous spring, resisting compression from similar force fields generated by neighbouring A-Cells with a force proportional to the speed of the compression.

$$R = (a * m) + (d * v) \quad (10)$$

where R is the repulsion force acting along the line joining the two A-Cell centres (Kgms^{-2}), a the area of membrane (m^2) that has been compressed, m the repulsion factor ($\text{Kgm}^{-1}\text{s}^{-2}$, a system parameter), d the damping coefficient (Kgs^{-1} , also a system parameter) and v is the estimated relative velocity (ms^{-1}) concerned along the line joining their centres.

Each A-cell transmits to its neighbours the radius of its physical hull and the distance its membrane (virtual force field) extends from this hull. Therefore, by comparing the approximate distance of neighbouring A-Cells with their relative sizes, an A-Cell can detect membrane collisions and estimate the area of membrane that has been compressed. Also, by monitoring how the distance of each neighbouring A-Cell changes over time it is possible to calculate their relative speeds, and thus the viscous component of the repulsive force.

To ensure that two colliding A-Cells experience an equal and opposite repulsive force some further communication is required. Each of the colliding A-Cells add the repulsive force they calculated to their transmission signal along with the ID of the A-Cell involved in the collision. Each A-Cell can then simply average its own calculated value with that of the other giving equivalent results.

One flaw with this simple model is that each membrane collision is considered separately. Therefore, it is possible for the same area of membrane to collide with more than a single other membrane simultaneously. However, the principal repulsive behaviours of the membrane remain consistent.

3.2.3 A-CAMs

The membranes of real cells use Cellular Adhesion Molecules (CAMs) to form bonds which are both reversible and selective. We attempt to model these key properties through the use of A-CAMs. Like CAMs, the A-CAMs are capable of

both homotypic or heterotypic binding. However, unlike some real CAMs each A-CAM may only bond with a single type. This greatly simplifies the implementation without restricting the power of the model.

When the A-CAMs form bonds they produce a small attractive force pulling the two areas expressing the A-CAMs closer together. Thus bonded A-Cell membranes are pulled together until the attractive force is balanced by the repulsive force generated by the compression.

When two areas of real cell membrane come into contact, only some subset of the CAMs on each membrane will be close enough to CAMs on the other membrane to allow them to bond. The size of this subset is dependent on both the number of CAMs on each area of membrane and the strength of the bond between the CAMs. Greater surface densities of CAMs will increase the chance that any two are facing each other and a greater bond strength will allow the CAMs to seek each other out more assiduously. To express these relationships with the A-CAMs we constructed the following equation which satisfies all the observed constraints when relating numbers of available A-CAMs to numbers that bond (N).

$$N = \min(A_{\alpha 1}, A_{\beta 2}) \text{ sigmoid} \left(m(\alpha, \beta) k_B + \frac{(A_{\alpha 1} - A_{\beta 2})^2}{A_{\alpha 1} * A_{\beta 2}} \right) \quad (11)$$

where $A_{\alpha 1}$ is the number of A-CAMs of type α on the contact area of membrane 1, $A_{\beta 2}$ is the corresponding number of type β A-CAMs on contact area of membrane 2, $m(x, y)$ is the attractive force between A-CAMs of type x and y and k_B is a system parameter that controls the influence of the bonding strength on bonding numbers.

3.2.4 Adhesion Sites

The A-Cell membrane model allows the same area of membrane to be in contact with more than a single other membrane simultaneously (see Section 3.2.2). However, to achieve more realistic selective bonding, the bonding of each area should be restricted to only one of the membranes in contact.

This is achieved by dividing the membrane surface into a number of equally sized patches or adhesion sites, such that each adhesion site can only bond with a single membrane. If an adhesion site comes into contact with more than a single membrane it makes a probabilistic decision as to with which it should bond. This decision is based on the estimated bond strength of a bond between a unit area of each membrane. To calculate this value, each A-Cell includes in its transmission signal the surface density of every A-CAM it is expressing. These values can then be compared with those of any A-Cell in contact and the estimated bond strength can be calculated.

Having established which adhesion sites wish to bond on each membrane it is then necessary to calculate the number of A-CAMs this represents. As we assume that any A-CAMs being expressed are evenly distributed across the membrane

surface this is relatively straightforward. Once calculated, these values are communicated between the colliding A-Cells by including them in their respective transmission signals. Each A-Cell then compares each pair of bonding A-CAMs in turn and calculates the total attractive force that will be produced.

This algorithm makes the following two simplifications: (1) it is assumed that adhesion sites that decide to bond on two colliding membranes appose each other. However, it is clear that this might not be the case; and (2) adhesion sites which decide to bond to a specific membrane are not reallocated if unused. Avoiding these simplifications would require far more complexity and would greatly increase the communication required between A-Cells.

3.2.5 Random Motion

It has been demonstrated that random cell movements resulting from membrane ruffling (localised expansions and contractions of the cell membrane) may increase local exploration and allow cells to discover membranes with which they may bond more strongly [Mombach and Glazier, 1996, Rieu et al., 1998]. As our simple model of the cell membrane is a uniform sphere, the exact nature of the ruffling cannot easily be reproduced. However, the result of this phenomena is simply that the cells make small random movements. Therefore, an equivalent behaviour can be achieved by adding some random component directly to the A-Cell movements. In the A-Cell model this component consists of an additional random movement force, the magnitude of which is scaled by a predefined system parameter. Therefore, both active and inactive membranes can be modelled.

3.2.6 CAM Controller Summary

Each A-Cell repeatedly cycles through the procedure of detecting membrane collisions and calculating the associated forces they generate. The algorithm is summarised in Figure 7.

3.3 Combined GRN-CAM Controller

The GRN and CAM controllers were combined as shown schematically in Figure 8. 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 A-Cell's membrane. Therefore, the GRN actually varies the adhesion between neighbouring A-Cells 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

1. Check which signals can be detected and calculate the approximate distance and direction of their source.
2. Determine if there are any membrane collisions.
3. For each A-Cell in contact:
 - (a) Compare the values in the A-Cell's transmission signal with the values being broadcast from this A-Cell and calculate any repulsive or attractive forces.
4. Sum any repulsive or attractive forces to calculate a single force vector.
5. Add any random force to the force vector.
6. For each A-Cell in contact:
 - (a) Calculate the repulsive force generated by compressing the membranes.
 - (b) Calculate the estimated bond strength.
7. For each adhesion site:
 - (a) Probabilistically decide which membrane to bond to.
8. Construct the new transmission signal.
9. Generate the final force.

Figure 7: A summary of the procedure employed by the CAM controller.

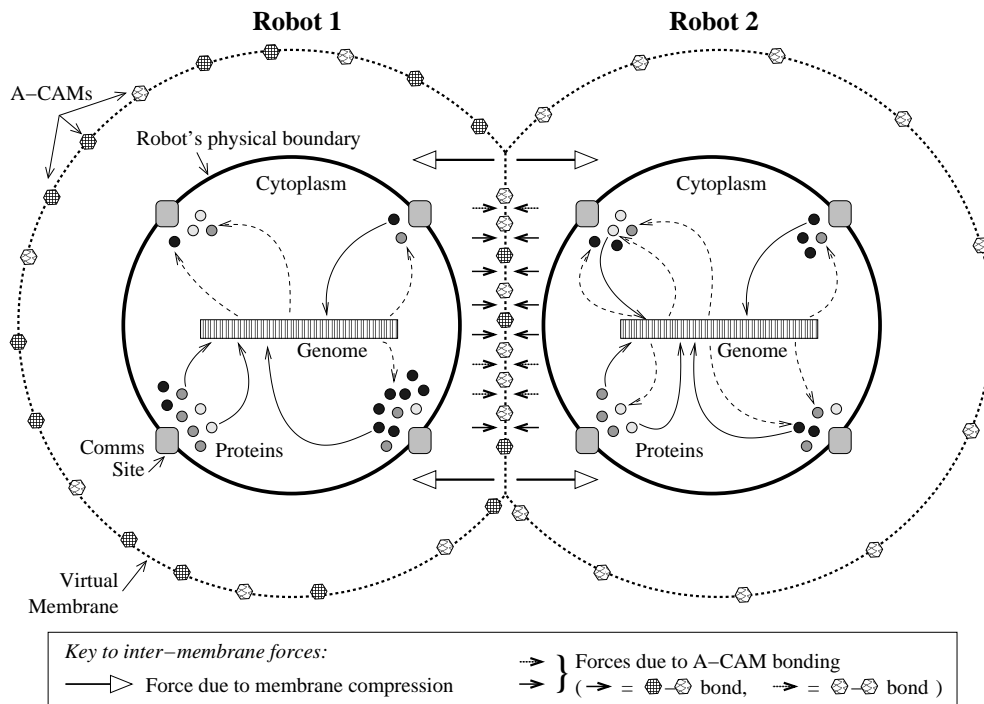


Figure 8: Schematic of the combined GRN-CAM controller design

A-CAM Bonding Relationships	
<i>Homotypic</i>	<i>Heterotypic</i>
A-CAM ₁ - A-CAM ₁	A-CAM ₅ - A-CAM ₆
A-CAM ₂ - A-CAM ₂	A-CAM ₇ - A-CAM ₈
A-CAM ₃ - A-CAM ₃	A-CAM ₉ - A-CAM ₁₀
A-CAM ₄ - A-CAM ₄	A-CAM ₁₁ - A-CAM ₁₂

Table 3: The available A-CAMs and the possible bonds between them.

remaining eight are capable of heterotypic bonding. This provides eight different bonding relationships for the GRN to exploit (Table 3).

To map each protein (p) to some level of A-CAM expression (A_p), its average concentration is calculated across all 4 diffusion sites (as A-CAMs are distributed evenly across the whole membrane). This gives a value between 0 and 1 which is then multiplied by the maximum level of A-CAM expression that is allowed (m)⁴.

$$L(A_p, t) = m \frac{1}{|D|} \sum_{d \in D} C_p(t) \quad (12)$$

⁴Given two A-Cells each expressing a single type of A-CAM, such that the two A-Cells adhere, the maximum level of expression was calculated as the lowest value which, if expressed on both A-Cells, would cause them to physically collide.

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 A-Cell’s membrane, the protein’s concentration is increased by a level proportional to the number of bonds:

$$C_p(t + 1) = C_p(t) + \frac{B(A_p)}{m} \quad (13)$$

where $B(A_p)$ is the number of bonds formed by the A-CAM linked to protein p .

The specific protein actions for the combined GRN-CAM are summarised in Table 2 (right-hand side).

For simplicity, in the experiments reported here we did not allow diffusion of proteins between neighbouring robots in the combined controller. Differentiation of gene expression could still occur due to the action of sensory proteins being produced in proportion to the number of bound A-CAMs on a robot’s membrane; these effectively signal how many other robots are in the immediate vicinity.

4 Evolution of Controllers

In order to produce 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 an identical controller for each of the HYDRON robots in the group. The controllers 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.

A series of tasks was designed to compare the quality of evolved controllers when using the GRN controller by itself (Section 3.1) and when using the combined GRN-CAM controller (Section 3.3).⁵ Three different tasks were used, of increasing difficulty, as described in Sections 5.1–5.3. Five evolutionary runs were performed for each of the six conditions (3 tasks and 2 controller types). The default parameter values, which were used in each case unless otherwise stated, are listed in the Appendix. To evaluate the fitness of a controller, an identical copy of

⁵Experiments with the CAM-only controller are not reported here, as the first of the tasks we investigate is easy for a (manually designed) CAM controller, while the other tasks are impossible for the CAM controller by itself. An extensive investigation into the properties of the CAM controller is reported in [Ottery, 2006].

it was placed into 36 robots, and fitness was recorded as the mean performance of the system over three separate trials starting from different initial configurations (with compactness values 0.4, 0.5 and 0.6 as defined in Equation 17, Section 5.1).

4.1 Smart Mutations

In addition to combining the GRN and CAM controllers, we also introduced a new type of mutation during the evolution experiments reported here.

The successful evolution of a GRN controller 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 insufficient 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 as reported in [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 design of the evolutionary algorithm in order to make these correspondences much more likely to occur. This is achieved by introducing an additional type of mutation, called a *smart mutation*, which operates as follows.

During the fitness evaluation of each genome, a record is kept of the highest instantaneous cytoplasmic concentration (in any of the robots in the cluster) achieved by each protein during the course of the evaluation. Similarly, for each gene, a record is kept of the minimum and maximum concentrations of regulators for that gene (i.e. the sum of concentrations of all enhancers minus the sum of concentrations of all inhibitors) at any point during the evaluation.

When each genome in the population has been evaluated and selection had been applied to generate a new population, smart mutations are applied to members of the new population, followed by the standard crossover and mutation operators. The smart mutations are applied to each non-elite genome with a fixed probability p_{SM}^{genome} ($= 0.75$ in the experiments reported here). The method of application is as follows.

Each gene on the genome is considered in turn. For each gene, three different types of smart mutation are considered: mutation of the output function; mutation of the regulators; and mutation of the signal protein (if appropriate). Each type of smart mutation is applied to the gene with probability p_{SM}^{gene} ($= 0.75$). The three types are now described in turn.

The goal of a smart mutation to a gene’s output function is to align the gene’s range of sensitivity to its regulator concentrations with the actual range of regulator concentrations observed when the controller is in operation. If the gene has an “above zero” output function (i.e. Step Above Zero or Gradient Above

Zero, Figure 4), and the observed maximum regulator concentration c_R^{obs} is greater than the threshold value specified by the gene, then the gene is edited to reset the threshold to a uniform-randomly chosen value between zero and c_R^{obs} . If the gene has an “above zero” output function and $c_R^{obs} < 0$, then the gene is edited to specify a “below zero” function instead (a Step Above Zero function is changed to Step Below Zero, and Gradient Above Zero is changed to Gradient Below Zero). If the gene has a “below zero” output function, then the opposite of the above procedure is applied.

The goal of smart mutation of a gene’s regulators is to ensure that each regulator could potentially have an effect on the gene’s activation in practice. For each regulator (be it either an inhibitor or an enhancer), if the observed maximum concentration of the corresponding protein c_P^{obs} was zero, then the regulator specification is edited to use a protein selected at random from the list of proteins that were observed to achieve non-zero concentrations during the operation of the controller.

Finally, if the gene specifies that its product should be placed at a diffusion site defined by the concentration of a specific signal protein (i.e. Product Placement Scheme 0 or 1, with signal protein specified by the gene’s Product Placement Parameter), and if the observed maximum concentration of the corresponding signal protein c_P^{obs} was zero, then the gene’s Product Placement Parameter is edited to specify a different signal protein, selected at random from the list of proteins that were observed to achieve non-zero concentrations during the operation of the controller.

In this way, the smart mutations 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.

5 Experiments

The tasks outlined below are presented in order of difficulty. In each case, a sample of five evolutionary runs were carried out for both the GRN-only controller and the combined controller. Unfortunately, the evaluation time required for each individual run (several days when running in parallel on a dedicated 16-node computer cluster) meant that a larger sample was not feasible. However, the bootstrapping method was applied to improve the error estimates [Efron and Gong, 1983]; each sample was resampled 5000 times to give a standard error estimate which is then used to determine the confidence error.

During the execution of each task it is clearly desirable that, at the very least, the robots maintain a single connected aggregate.⁶ Therefore, the fitness value

⁶In the combined GRN-CAM controller a connection is achieved through membrane contact whereas in the GRN-only controller it implies that robots are within communication range. During these experiments the two distances are equivalent.

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:

$$F_1 = \frac{T_M}{T_E} \quad (14)$$

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, and therefore was only awarded when the aggregate was maintained for the entire simulation period. These components were combined with the relative weightings of 0.1 and 0.9 respectively, giving the two following scenarios.

1. Aggregate not maintained:

$$F_t = 0.1F_1 \quad (15)$$

2. Aggregate maintained for the duration of the simulation ($F_1 = 1$):

$$F_t = 0.1 + 0.9F_2 \quad (16)$$

In addition, to determine a more general fitness value, each controller was evaluated from three different initial configurations. Each of these was selected from a population of randomly generated configurations in which the aggregates were fully connected, such that they had compactness values 0.4, 0.5 and 0.6 (see [Ottery, 2006] for details). Once all three fitness values were calculated, the average result was taken.

5.1 Clustering

The first task tests the controllers' ability to perform simple aggregation or rounding. We used the compactness C of the robot cluster as a measure of aggregation:

$$C = \frac{4\pi a}{p^2} \quad (17)$$

where a is the area of the concave hull defined by the robots' positions, p is its perimeter, and ($0 \leq C \leq 1$). However, to encourage the controllers to achieve maximum compactness as soon as possible, the final fitness is calculated as the mean compactness of the aggregate over the entire 100s evaluation period (T_E).

$$F_2 = \frac{1}{T_E} \sum_{i=0}^{T_E} C_i \quad (18)$$

5.2 Response to External Signal

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.⁷ 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.

$$m_a = \frac{1}{T_S} \sum_{i=0}^{T_S} a_i \quad (19) \quad m_b = \frac{1}{T_E - T_S} \sum_{i=T_S+1}^{T_E} a_i \quad (20)$$

$$D = \frac{m_b}{m_a + m_b} \quad (21)$$

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

$$v_a^2 = \frac{1}{T_S} \sum_{i=0}^{T_S} (a_i - m_a)^2 \quad (22) \quad v_b^2 = \frac{1}{T_E - T_S} \sum_{i=T_S+1}^{T_E} (a_i - m_b)^2 \quad (23)$$

$$V = \frac{\min(v_a v_b, k_v m_a m_b)}{k_v m_a m_b} \quad (24)$$

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:

$$F_2 = k_f D + (1 - k_f)(1 - V) \quad (25)$$

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 ± 20 s. 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.

⁷If there is no notion of maintaining a constant configuration, the controllers tend to exploit the dynamics of the simulation to produce solutions where the behaviour is fine tuned to the evaluation period T_E .

5.3 Differentiation of Function

Differentiation is the process which allows a collection of initially homogeneous agents to divide into a number of distinct subsets, each of which is capable of performing a more specific function. This aim of this task was to determine whether the controllers could achieve this by splitting an aggregate of A-Cells into two equally sized sets (\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:

$$g_a = \frac{1}{U|\alpha|} \sum_{t=0}^{T_E} \sum_{i \in \alpha} c_i^t \quad (26) \quad g_b = \frac{1}{U|\beta|} \sum_{t=0}^{T_E} \sum_{i \in \beta} c_i^t \quad (27)$$

$$d_r = \frac{1}{2} (g_a + (1 - g_b)) \quad (28)$$

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.⁸

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

$$d_r \begin{cases} > 0.5, & \text{if there are more set } \alpha \text{ proteins than set } \beta \\ = 0.5, & \text{if both sets of proteins are equal} \\ < 0.5, & \text{if there are more set } \beta \text{ proteins than set } \alpha \end{cases} \quad (29)$$

Thus, the robots can be assigned to \mathcal{A} and \mathcal{B} , such that \mathcal{A} consists of those with the highest 50% of the d_r values. The overall fitness is then defined as:

$$F_2 = \frac{1}{n} \left(\sum_{i \in \mathcal{A}} d_i + \sum_{j \in \mathcal{B}} (1 - d_j) \right) \quad (30)$$

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.

⁸Specifically, U was defined as $U = \frac{T_E}{T_s} * N_{diff} * P_{max}$, where T_s is the period of a single timestep, N_{diff} is the number of diffusion sites and P_{max} is the maximum concentration of a protein. This gives a value of 2000 for these experiments.

5.4 Robustness of Evolved Controllers

When the evolution experiments were complete, we took the best evolved controller for each of the six conditions (3 tasks and 2 controller types) and analysed their robustness under a variety of scenarios.

Scalability To evaluate the scalability of the evolved controllers, their performance was tested for a variety of different group sizes in the range 12–60 robots. For each group size, a random initial configuration with a compactness index of 0.5 ± 0.01 (the mid-range of those used during evolution) was selected.

Specificity of Initial Configuration A wider variety of initial configurations, with compactness values in the range $[0.0, 0.8]$, were used to examine how dependent the evolved behaviours were on the robots’ initial configuration.

Added Random Motion The final tests investigated how well the evolved controllers could cope with levels of environmental perturbations which differed from those under which they were evolved. Three sets of experiments were carried out in total; (1) Small random forces applied to each robot, (2) Small random torques applied, and (3) Small random forces and torques applied. The ranges used for these tests were derived from earlier simulations of the cell adhesion system.

5.5 Implementational Variations

We also conducted experiments to test the effect of a number of implementational issues, and extensions to the controller design, on the performance of the system.

No Smart Mutations We ran 5 independent evolutionary runs of the GRN controller in the Clustering task without smart mutations, in order to gauge the difference that the smart mutations made to the results achieved.

External Protein Diffusion In most of the experiments reported with the combined GRN-CAM controller, we did not allow proteins to diffuse between neighbouring robots (as explained in Section 3.3). We also ran 5 independent evolutionary runs of the GRN-CAM controller in the Differentiation of Function task, in which Protein 0 was allowed to diffuse between robots (as in the GRN-only experiments).

Evolution of Rate of Protein Dynamics There are a number of parameters controlling the protein dynamics rates in the GRN (and combined) controller. Specifically, one controls the internal diffusion rate of proteins in the cytoplasm of an individual robot, one controls the external diffusion rate of proteins between

robots, and a third controls the attenuation rate of proteins in the cytoplasm. In most of the experiments reported, these rates are externally defined and constant (see Appendix).

We also conducted a series of experiments in which each genome, in addition to its standard content, also specifies values for these three protein dynamics rates. These values were allowed to evolve to see if the system could find the most appropriate dynamics rates for the task in hand. At each generation, each rate specification on the genome was mutated with probability 0.05. If mutated, the value was changed by a value uniform-randomly chosen from the interval ± 0.15 (with the new value capped at 1.0 or 0.0 if necessary). 5 independent evolutionary runs of the GRN controller were conducted on each of the three tasks (Clustering, Response to External Signal, and Differentiation of Function).

6 Results and Analysis

6.1 Controller Evolution

The controller evolution results for both controllers are shown in Figure 9. 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 each of the three 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.

The difference in performance is most clearly marked in the initial clustering task. In this case, the combined controller can solve the task, to some degree, by simply expressing any two A-CAMs which adhere. The only real challenge for the GA is to find a GRN which sets the levels of A-CAM expression that generate the most appropriate level of adhesion for the environment.⁹ This is particularly visible in the results which show that even at generation 0 the best individual achieved a very high fitness which was then increased only slightly over the remainder of the run.

In the signal response task, the combined GRN-CAM controller again finds a solution relatively quickly. However, the benefit of combining the controllers is more obvious as the GA is able to find progressively better solutions during the course of the run. An example of an evolved genetic regulatory network for the combined controller for this task is shown in Figure 10. In this figure, each gene is plotted on the vertical axis, and the corresponding horizontal row shows which

⁹In practice, as the combined controller does not require any dynamic behaviour to achieve this task, the GRN actually adds unnecessary complexity. Instead, a GA could simply be used to directly set the levels of A-CAM expression which could remain fixed for the duration of the simulation.

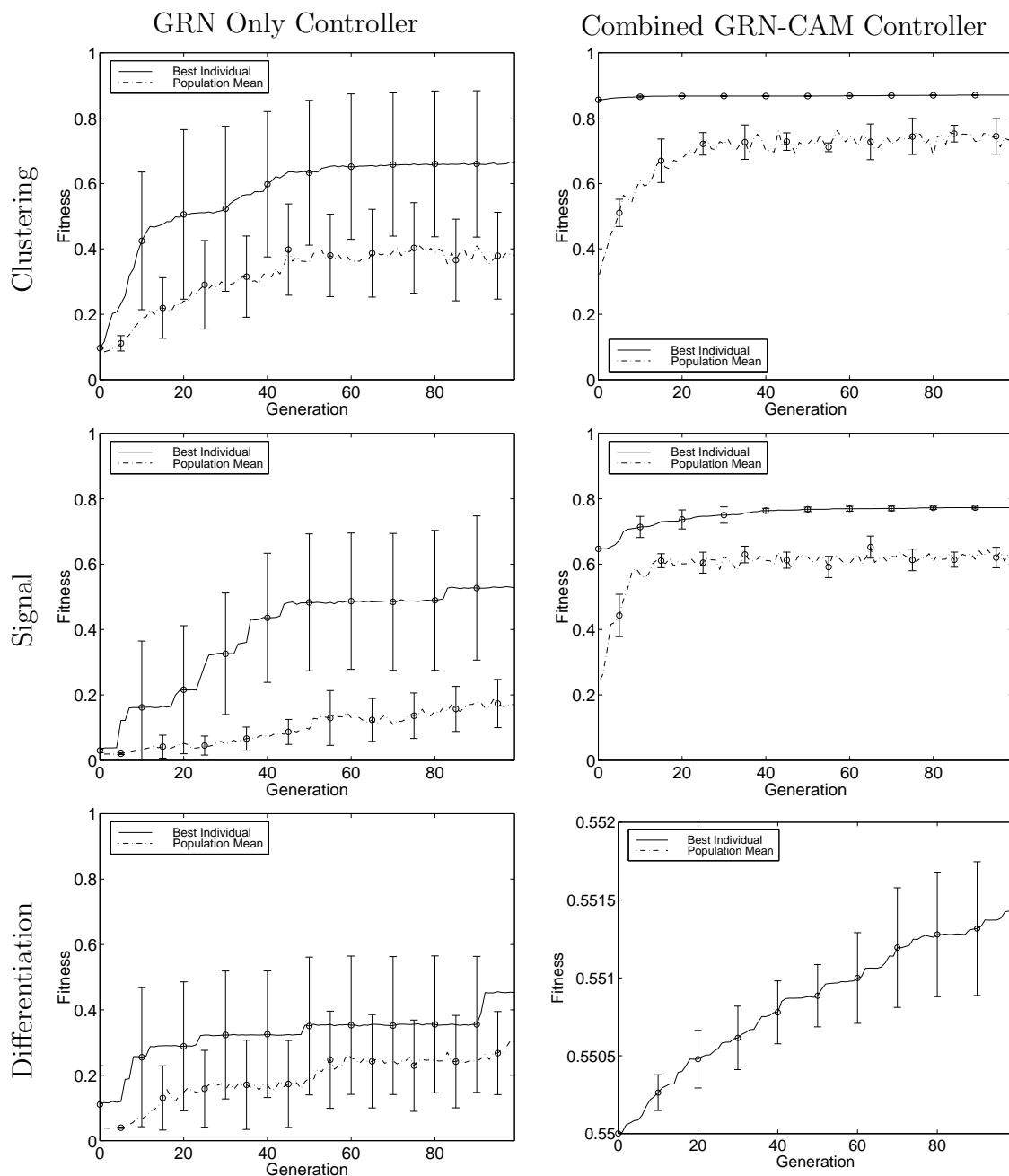


Figure 9: Mean and maximum fitness over time for each of the six evolution conditions. Note the different scale on the bottom-right graph (in this experiment, the population mean increased gradually but remained below 0.55 hence is not visible on the graph; see text for details).

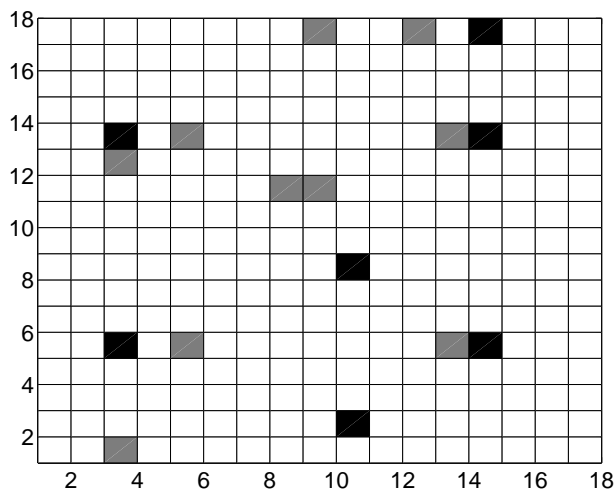


Figure 10: Gene regulation matrix of the best evolved GRN-CAM controller for the signal response task. Each square represents the regulation of the gene along the horizontal axis by the product of the gene on the vertical axis. A white square denotes no regulation, light grey denotes enhancement, dark grey denotes inhibition, and black denotes both enhancement and inhibition.

other genes it regulates. From this regulation matrix we can identify that 8 of the 17 genes in the genome are involved in regulation, with each of these inhibiting or enhancing up to 4 others. The introduction of the signal can clearly be identified on the protein trace (Figures 11 and 12) as the sudden production of the signal proteins (protein numbers 1-15) 40 seconds into the simulation. For the GRN-CAM controller (Figure 11), this signal causes a change in gene expression and the GRN quickly settles into a new stable state. In this new state, there is a clear change in the expression of proteins, particularly those proteins involved in the production of A-CAMs (protein numbers 36-47). Figure 13 shows screen shots taken at regular intervals during the simulation the traces were taken from. Viewing a simulation of the best solution for the GRN-only controller, no noticeable change in behaviour is apparent and in many cases the aggregate even fails to remain connected. The traces do show a momentary change in behaviour as a result of the signal (Figure 12). However, the GRN quickly reverts to its earlier pattern of expression.

Finally, in the differentiation task, it can be seen that the combined controller rapidly achieves differentiation, as indicated by the fitness in excess of 0.55.¹⁰ However, the GRN-only controller fails to find any solution. In addition, the results show that although the combined controller does continue to improve, the overall increase in fitness is actually very small. This is mainly due to the very large

¹⁰Using 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 16).

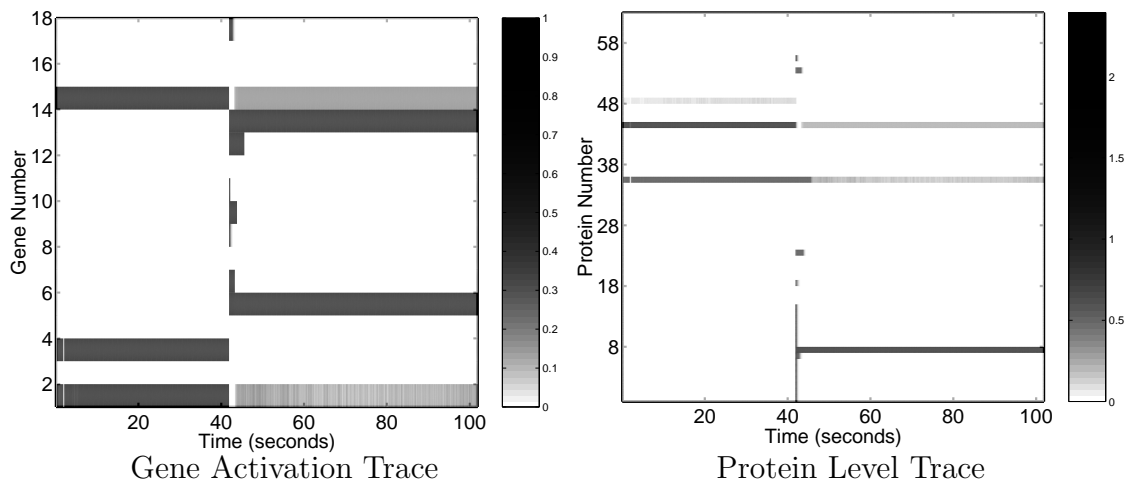


Figure 11: Traces obtained from a genetic regulatory network which was evolved to carry out the signal response task with the combined GRN-CAM controller.

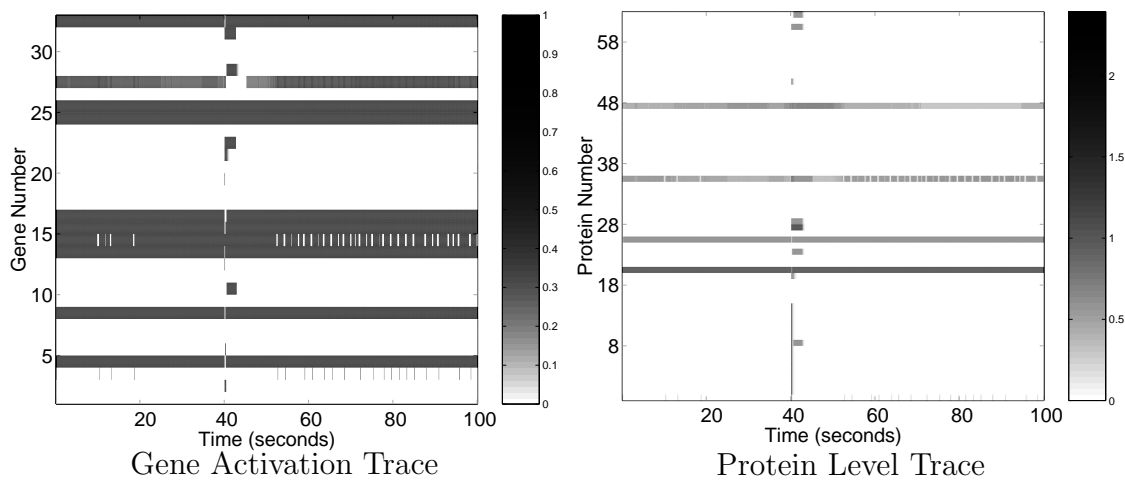


Figure 12: Traces obtained from a genetic regulatory network which was evolved to carry out the signal response task with the GRN-only controller.

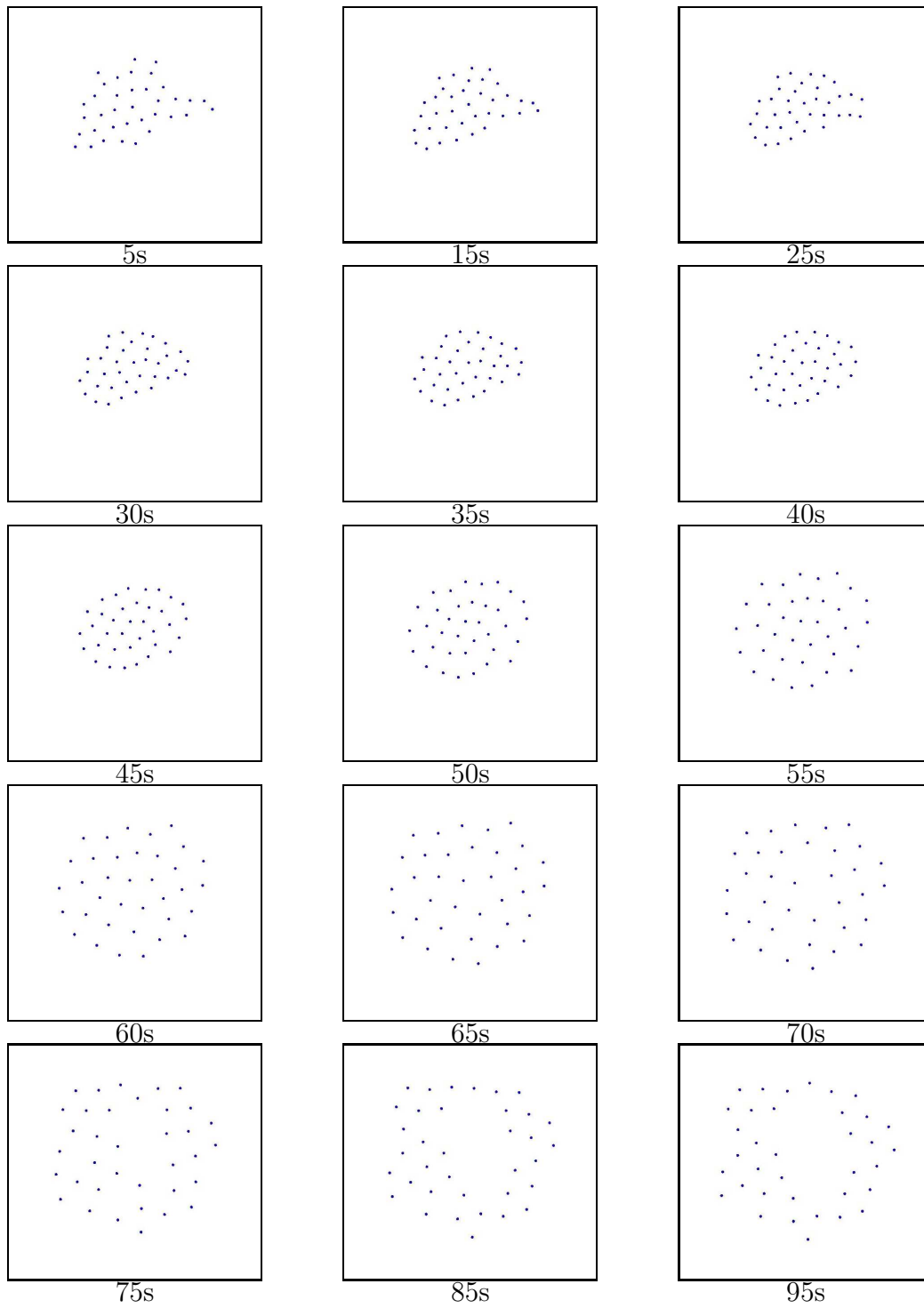


Figure 13: Screen shots of the combined controller carrying out the signal response task. The signal was introduced at exactly 40s.

upper bound (U) used in the fitness calculation (see Section 5.3).¹¹ An example of the evolved differentiation behaviour from a GRN-CAM controller is shown in Figure 14.

6.2 Controller Quality

To evaluate the robustness of the evolved controllers, the best combined GRN-CAM controller and the best GRN-only controller obtained for each of the previous tasks were then subjected to the robustness tests for scalability, specificity to initial configuration, and ability to cope with perturbations described in Section 5.4. For each scenario, each of the controllers was again evaluated for its respective task while the simulation setup was modified as described. A sample of 20 runs were carried out for each of the parameter values.

The results of these robustness experiments are shown in Figures 15–17. Again, the combined controller outperformed the GRN-only controller and in most cases it is clear that the scenarios had a significantly smaller effect on its performance. In addition, the combined controller still shows far greater consistency, significantly so for both the clustering and differentiation tasks.

The results show most clearly that the significant difference between the models appears to be their ability to cope with environmental changes. The different conditions have very little effect on the combined controller. However, when the level of the random movement forces increases above those that were present during evolution, the GRN-only controller’s performance begins to deteriorate. In addition, the presence of any torque appears to reduce the performance of the GRN-only controller (as can be seen by the difference between the simulations with only random movement forces and random movement forces with a small fixed torque) and an increase in torque has a disastrous effect. This is possibly due to a time lag in the GRN between the detection of externally diffusing proteins and the production of actuation proteins. For example, if a robot detects a neighbour in one direction and it rotates before the appropriate actuation proteins are produced, it will head off in the wrong direction.

There is also a significant effect on both types of clustering controller when the initial configuration is changed. However, this is understandable given the task as the robots will require a longer time to cluster from an initially less compact configuration.

¹¹In actual fact, due to an error, the U value that was used while the results were recorded was double the size of the true maximum. Therefore, the maximum achievable fitness in the task was actually 0.775.

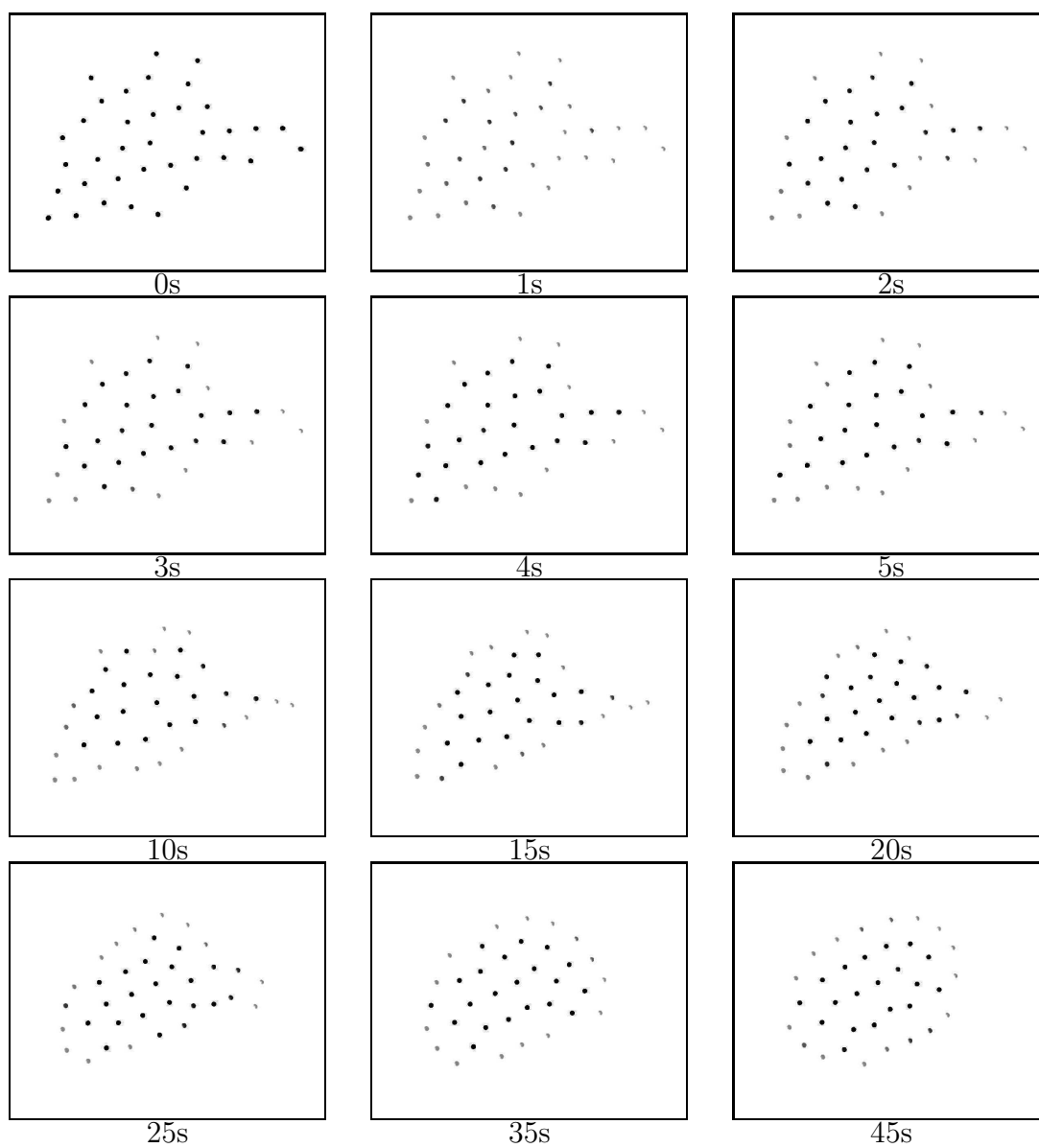


Figure 14: Screen shots of the combined controller carrying out the differentiation task. The shading indicates the level of difference between the two groups of indicator proteins at that time step, $\frac{1}{2}(g_a + (1 - g_b))$ (see Section 5.3).

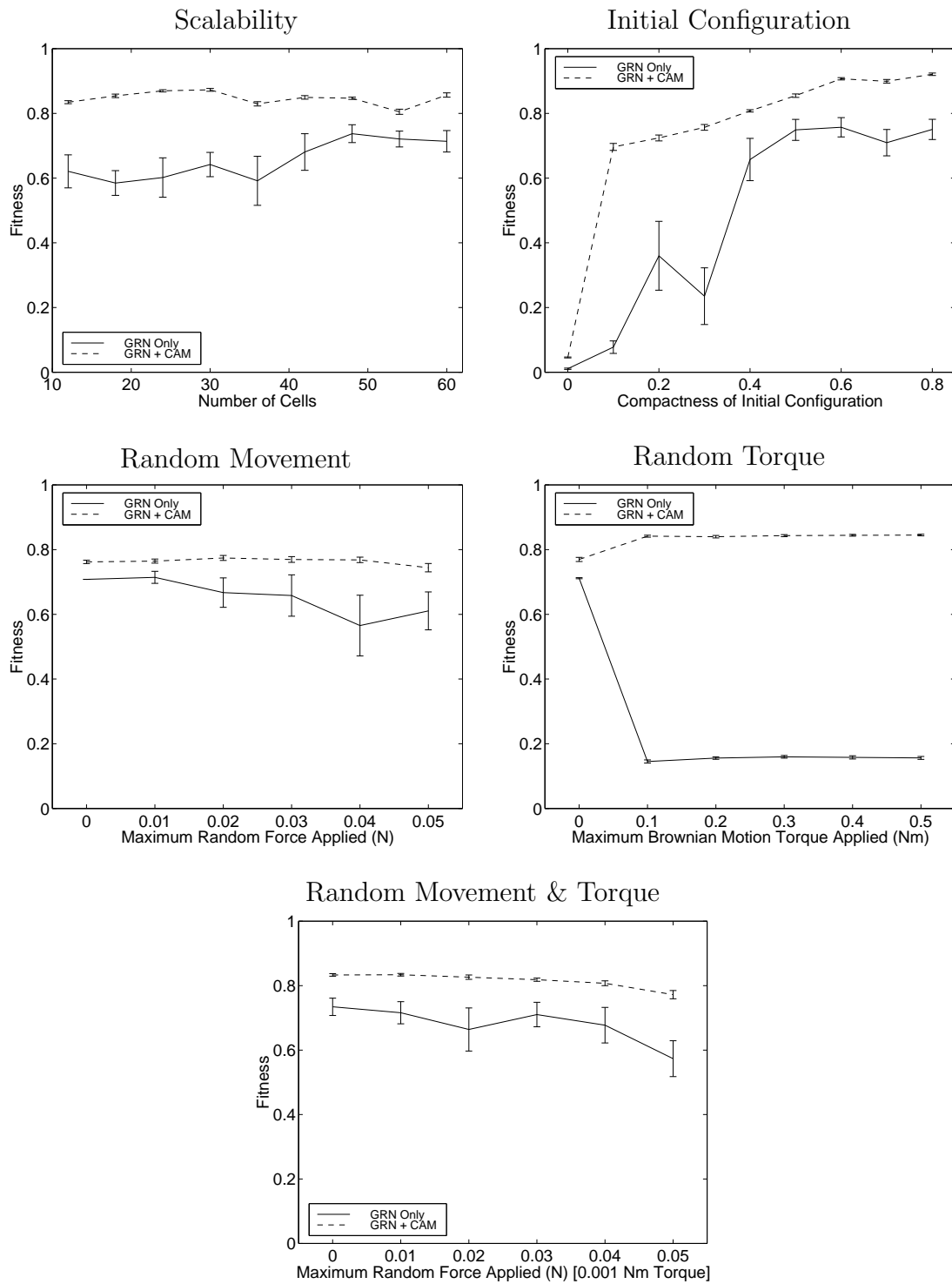


Figure 15: Robustness experiments carried out with the clustering controllers.

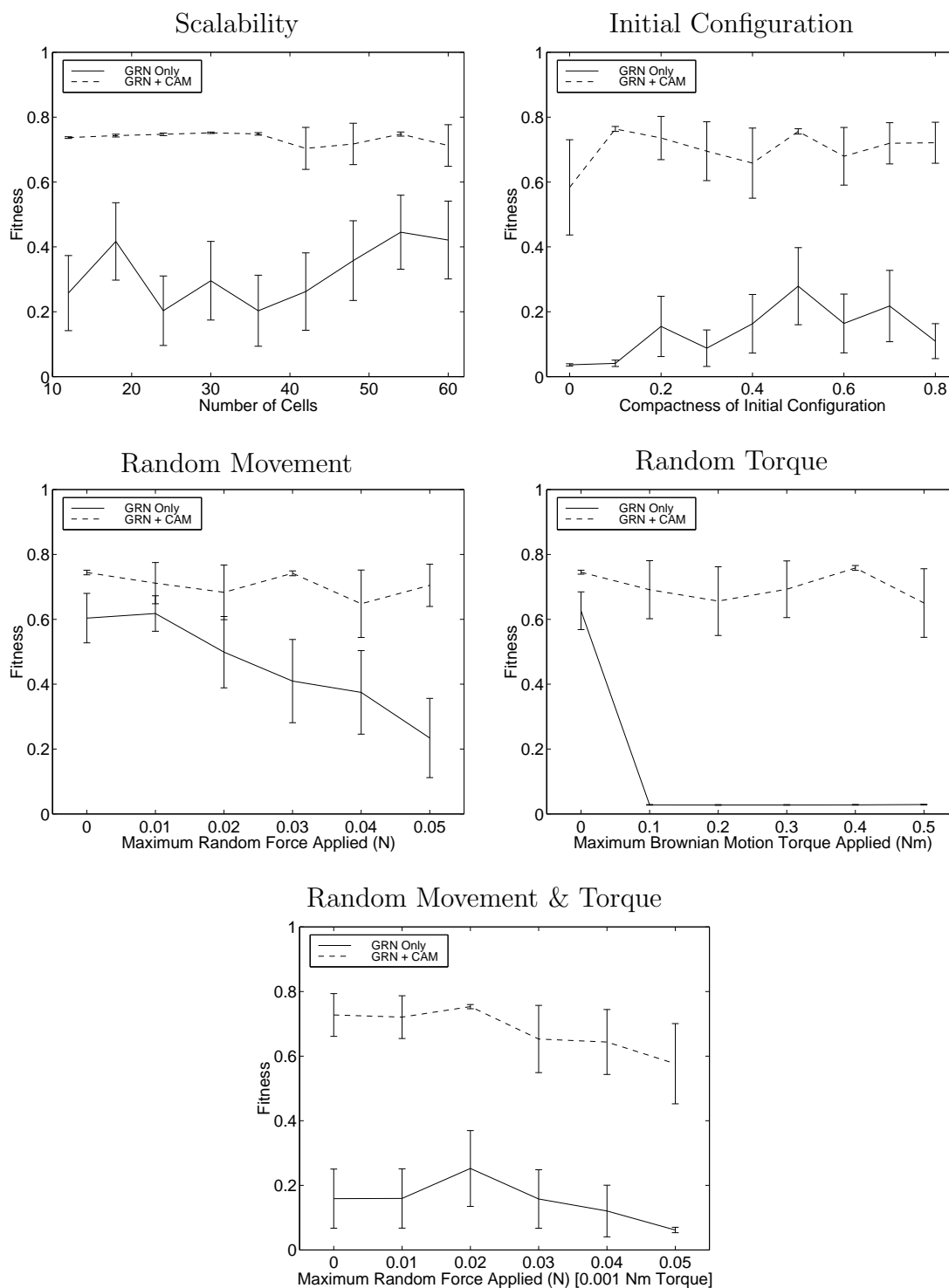


Figure 16: Robustness experiments carried out with the signal response controllers.

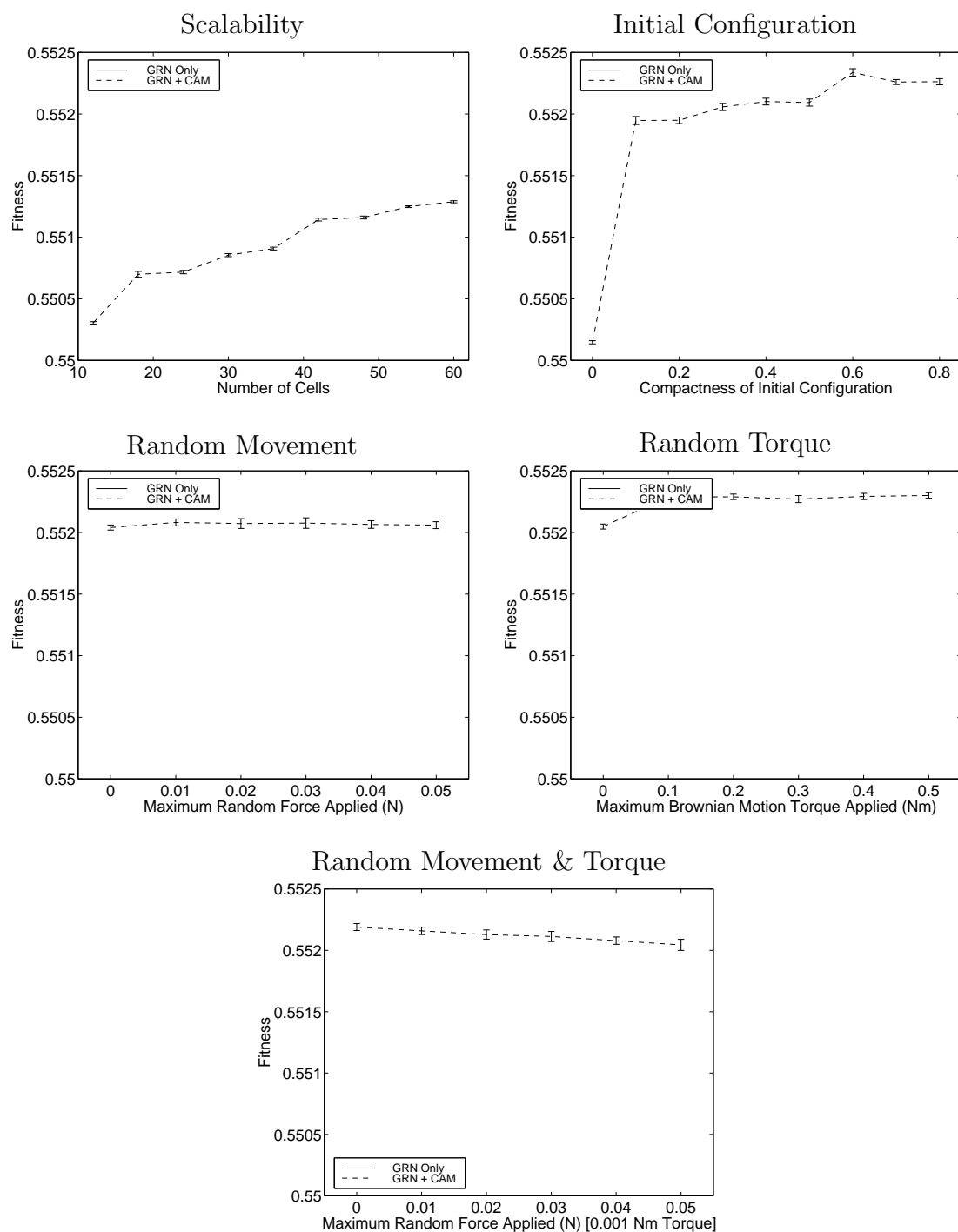


Figure 17: Robustness experiments carried out with the differentiation controllers. The results for the GRN-only controller cannot be seen as the mean values were generally below the lower bound of the scale.

6.3 Implementational Variations

6.3.1 No Smart Mutation

The results from the runs with no smart mutations are shown in Figure 18(a). Comparing these with the results from the GRN Clustering run with smart mutations (top left graph in Figure 9), we can see that the performance of the best individual and the population means are both consistently higher in the runs with smart mutations (although the difference is not significant at the 0.05 level due to the wide confidence limits).

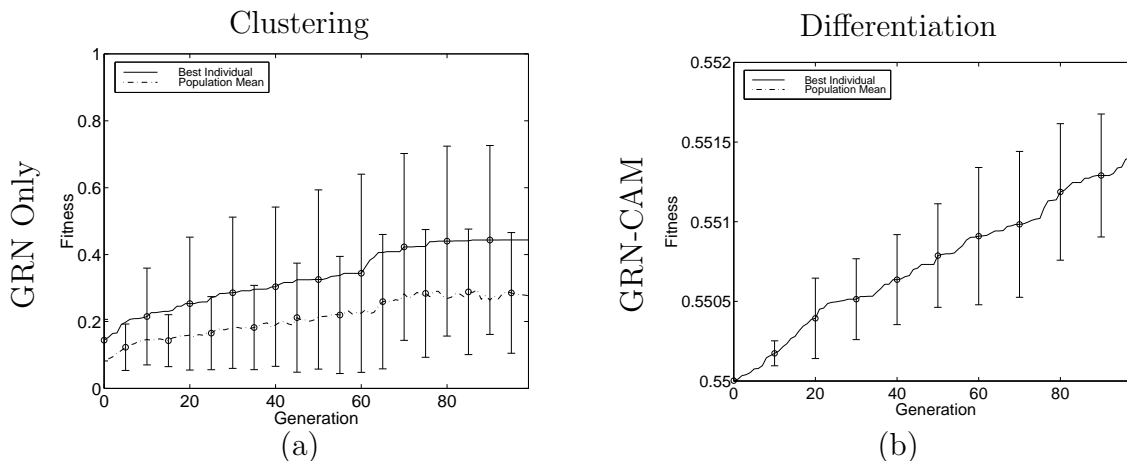


Figure 18: (a) Mean and maximum fitness over time for the GRN Clustering run with no smart mutations. (b) Mean and maximum fitness over time for the GRN-CAM Differentiation of Function run with external diffusion of Protein 0.

6.3.2 External Protein Diffusion

Figure 18(b) shows the results of the runs with the GRN-CAM controller where some external diffusion of proteins was allowed, for the Differentiation of Function task. If we compare these results with the corresponding run where external diffusion was not allowed (bottom right graph in Figure 9), we see that the results are virtually identical. This indicates that, as originally assumed, the addition of a sensory protein that is produced in proportion to the number of bound A-CAMs on the membrane is a suitable replacement to the external protein diffusion mechanism—both techniques allow an individual robot to react to the presence of other robots in their vicinity. Of course, the external protein diffusion mechanism is more general and potentially more powerful for complex tasks, but in the tasks tested here the “bound A-CAM sensor protein” mechanism was sufficient.

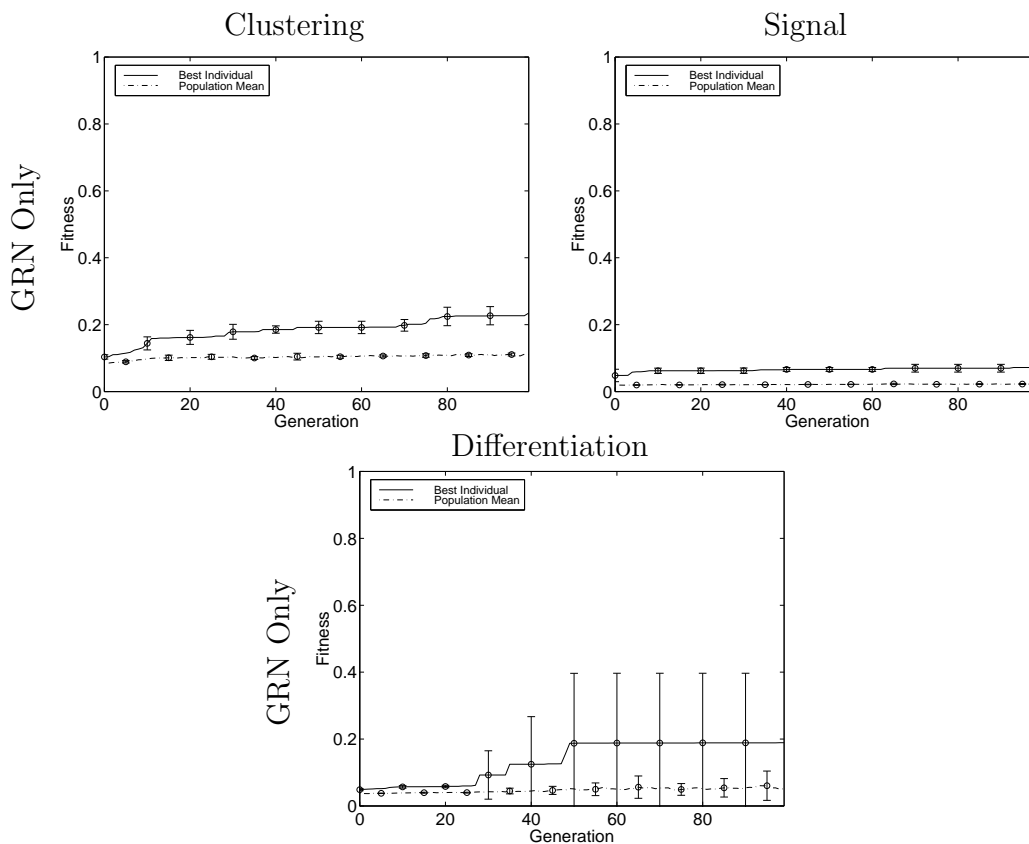


Figure 19: Mean and maximum fitness over time for runs with evolution of rate of protein dynamics.

6.3.3 Evolution of Rate of Protein Dynamics

Results from the runs in which the rates of protein dynamics were allowed to evolve are shown in Figure 19. Comparing these results to those where the rates were not allowed to evolve (left column of Figure 9), we see that the results with evolvable rates were actually *worse* than with fixed rates, for all tasks. This was somewhat unexpected, and further tests are underway to investigate this result. Looking at the evolved rates for each of the three parameters in the individual evolutionary runs for each task (not shown), it is apparent that there are no consistent values to which these parameters are evolving—the final values arrived at in each run are quite different. It may be that trying to evolve all three parameters concurrently and independently is not the best approach; further experiments are currently underway to test different approaches.

7 Conclusion

It has been demonstrated that by combining the GRN controller with the CAM controller it is possible to improve upon each individual system’s performance.

By adding the CAM controller (which already implements a robust self-organising behaviour) to the GRN controller, it removes the need for the GRN to interface directly with the robot’s sensors and actuators. This appears to make the evolution of effective controllers easier and improves the robustness of the controllers that are evolved. In theory this would also make it possible to transfer the same controller to any robot which implements the CAM model.

Additionally, adding the GRN controller to the CAM controller allows both temporal and spatial differentiation of A-CAM and protein expression across the population of robots. This allows more complex tasks to be performed with a population of (initially) purely homogeneous robots. As an example of temporal differentiation of expression, a GRN-CAM controller was evolved which caused the population of robots to adopt an initial spatial pattern, but then switch to a different pattern upon detection of an external signal (see Section 6.1 and Figure 13). As an example of spatial differentiation of expression, a GRN-CAM controller was evolved which in which some robots in the group developed one pattern of protein expression, while the other robots developed a different pattern of expression (see Section 6.1 and Figure 14). This could be used, for example, as the starting point for developing clusters in which sub-groups of robots perform specialised tasks.

The “smart mutation” mechanism described in Section 4.1 appears to provide a useful increase in performance for the evolved controllers (Section 6.3.1). The method could easily be applied to other applications that involve the evolution of genetic regulatory networks (e.g. [Bongard, 2002, Quick et al., 2003, Kumar and Bentley, 2003]).

Experiments with allowing the rates of the GRN dynamics to evolve in order to suit the particular task in hand (Section 5.5) did not produce successful results (Section 6.3.3). This was somewhat surprising, as it is reasonable to assume that there is an optimum combination of diffusion and attenuation rates for any given task. Further analysis is under way to investigate the reasons for this failure.

Some possibilities for future work are to allow proteins in the combined GRN-CAM controller to control the expression of A-CAMs in localised areas of the membrane or to diffuse externally between the robots as they do when the GRN controller is used by itself. Both of these additions would greatly increase the range and complexity of tasks that can be performed with this system.

Finally, the current simulation can be modified to allow three dimensional movement. This would then make it possible to investigate the transfer of the evolved controllers to the real hardware. The results of the robustness tests indicate that the transfer of controllers evolved in simulation onto the real hardware should be feasible without a great change in the resulting behaviour.

Appendix: Experiment Details

Genetic Algorithm Parameters: Population Size= 60, Generations= 100, Elite Group Size= 1, Tournament Size= 3, Crossover Probability= 0.6, Mutation Probability= 0.002, Initial Genome Size= 1000. *CAM Parameters:* Repulsion Factor $m = 2$, Damping Coefficient $d = 0.4$, Environmental Perturbations: Max Random Forces= 0.01N, Max Random Torque= 0.001Nm. *Protein Dynamics Parameters:* Attenuation constant $k = 0.05$, Internal diffusion constant $d = 0.05$, External diffusion constant $e = 0.025$, External signal attenuation constant $k_T = 0.0009$, External Signal Fitness Function Parameters: $k_v = 0.25$, $k_f = 0.8$.

Acknowledgements

This work was conducted under the HYDRA project (EU grant IST 2001 33060, <http://www.hydra-robot.com>). Facilities were provided by the Institute of Perception, Action and Behaviour in the School of Informatics, University of Edinburgh.

References

- Tamio Arai, Enrico Pagello, and Lynne E. Parker. Guest editorial: Advances in multirobot systems. *IEEE Transactions on Robotics and Automation*, 18(5): 655–661, 2002.
- 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.
- 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.
- Bradley Efron and Gail Gong. A leisurely look at the bootstrap, the jackknife, and cross-validation. *The American Statistician*, 37(1):36–48, February 1983.
- Jakob Fredslund and Maja J. Matarić. A general, local algorithm for robot formations. *IEEE Trans.on Rob. and Auto., Special Issue on Multi-Robot Systems*, 18(5):837–846, 2002.
- E. Peter Greenberg. Bacterial communication: Tiny teamwork. *Nature*, 424:134, 2003.
- Owen Holland and Chris Melhuish. Stigmergy, self-organisation, and sorting in collective robotics. *Artificial Life*, 5(2):173–202, 1999.
- Sanjeev Kumar and Peter Bentley. Biologically inspired evolutionary development. In *Proceedings of the International Conference on Evolvable Systems (ICES 2003)*, 2003.
- Jose C.M. Mombach and James A. Glazier. Single cell motion in aggregates of embryonic cells. *Physical Review Letters*, 76(16):3032–3035, April 1996.
- Julien Nembrini, Alan Winfield, and Chris Melhuish. Minimalist coherent swarming of wireless networked autonomous mobile robots. In *Proceedings of SAB*, pages 373–382, 2002.
- Esben H. Østergaard, David J. Christensen, Peter Eggenberger, Tim Taylor, Peter Ottery, and Henrik H. Lund. Hydra: From cellular biology to shape-changing artefacts. In *Proceedings of the 15th International Conference on Artificial Neural Networks (ICANN 2005)*, pages 275–281. Springer, 2005.
- Peter Ottery. *Using Differential Adhesion to Control Self-Assembly and Self-Repair of Collections of Modular Mobile Robots*. PhD thesis, School of Informatics, University of Edinburgh, 2006.

- Peter Ottery and John Hallam. Steps toward self-reconfigurable robot systems by modelling cellular adhesion mechanisms. In F. Groen, N. Amato, A. Bonarini, E. Yoshida, and B. Kröse, editors, *Proceedings of IAS8*, pages 720–728, Amsterdam, 2004. IOS Press.
- 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 Seventh European Conference on Artificial Life (ECAL'2003)*, volume 2801 of *Lecture Notes in Artificial Intelligence*. Springer Verlag, 2003.
- M. Quinn, L. Smith, G. Mayley, and P. Husbands. Evolving controllers for a homogeneous system of physical robots: Structured cooperation with minimal sensors. *Philosophical Transactions of the Royal Society of London, Series A: Mathematical, Physical and Engineering Sciences*, 361:2321–2344, 2003.
- 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 (ECAL'99)*. Springer, 1999.
- Jean Paul Rieu, Naoki Kataoka, and Yasuji Sawada. Quantitative analysis of cell motion during sorting in two-dimensional aggregates of dissociated hydra cell. *Physical Review E*, 57(1):924–931, January 1998.
- Malcolm S. Steinberg. Reconstruction of tissues by dissociated cells. *Science*, 141 (3579):401–408, August 1963.
- Tim Taylor. A genetic regulatory network-inspired real-time controller for a group of underwater robots. In F. Groen, N. Amato, A. Bonarini, E. Yoshida, and B. Kröse, editors, *Proceedings of IAS8*, pages 404–412, Amsterdam, 2004. IOS Press.
- Vito Trianni, Stefano Nolfi, and Marco Dorigo. Hole avoidance: Experiments in coordinated motion on rough terrain. In F. Groen, N. Amato, A. Bonarini, E. Yoshida, and B. Kröse, editors, *Proceedings of IAS8*, pages 29–36, Amsterdam, 2004. IOS Press.
- Jens Wawerla, Gaurav S. Sukhatme, and Maja J. Matarić. Collective construction with multiple robots. In *Proceedings of IROS*, pages 2696–2701, 2002.