Controlling Legged Robots with Coupled Artificial Biochemical Networks

Michael A. Lones[1,2], Andy M. Tyrrell[1,2], Susan Stepney[1,3] and Leo S. Caves[1,4]

[1]York Centre for Complex Systems Analysis (YCCSA)
[2]Department of Electronics, [3]Department of Computer Science, [4]Department of Biology
University of York, Heslington, York YO10 5DD, UK
{mal503,amt}@ohm.york.ac.uk, susan.stepney@cs.york.ac.uk, leo.caves@york.ac.uk

Abstract

Artificial biochemical networks (ABNs) are computational architectures motivated by the organisation of cells and tissues at a biochemical level. In previous work, we have shown how artificial biochemical networks can be used to control trajectories in discrete and continuous dynamical systems. In this work, we extend the approach to the control of a hybrid dynamical system: a legged robot. Taking inspiration from biological cells, in which complex behaviours come about through the interaction of different classes of biochemical network, we develop the notion of a coupled artificial biochemical network, in which an artificial genetic network controls the configuration of an artificial metabolic network. Using a higher-level robotic control task, we show how the coupled network finds solutions which can not be readily expressed using the artificial genetic network or artificial metabolic network alone. Our results also show the important role that non-linear maps can play as a natural source of complex dynamics.

Introduction

The structure and function of biological organisms emerges from the action and interaction of biochemical networks operating within cells. There are three main types of biochemical network: the *metabolic network*, comprising the protein-mediated chemical reactions that take place within the cell; the *signalling network*, composed of the protein-mediated responses to chemical messengers received by the cell; and the *genetic network*, which emerges from the regulatory interactions between genes.

From a computational perspective, biochemical networks are interesting for a number of reasons. This includes their ability to express complex behaviours, their compactness, their ability to adapt to changing environments, their robustness to environmental perturbation and—from the perspective of evolutionary computation—their evolvability. Such reasoning has motivated a host of computational models whose architectures are based upon the structure and function of biochemical networks. We refer to these collectively as *artificial biochemical networks*, or ABNs (Lones et al., 2010).

Perhaps best known of these are Boolean networks (Kauffman, 1969) and other kinds of *artificial genetic networks* (e.g. Reil, 1999; Banzhaf, 2003). By modelling the regulatory interactions which occur between genes, these models attempt to capture the dynamics of genetic networks, using these to generate complex, robust, behaviour. Another class of models, which includes P Systems (Păun, 2000) and artificial chemistries (e.g. Fontana, 1992; Banzhaf, 2004), can be categorised as *artificial metabolic networks*. These mimic the self-organising behaviour of biological metabolisms, and attempt to capture the manner in which complex behaviour can emerge from interactions between simple computational components. There has also been some work on *artificial signalling networks*, including early work on perceptron-like feed-forward networks (Bray, 1995) and more recent work on signalling-based classifier systems (Decraene et al., 2007).

ABNs have been used to implement a range of computational behaviours, including those required for robotic navigation (Ziegler and Banzhaf, 2001; Taylor, 2004), classification (Banzhaf and Lasarczyk, 2005), pole balancing (Nicolau et al., 2010) and image compression (Trefzer et al., 2010). In our research, we are interested in the ability of ABNs to control the kind of dynamics found in complex real world systems. In (Lones et al., 2010), we applied ABNs to the control of two numerical dynamical systems: the Lorenz equations, a continuous-time dissipative dynamical system; and Chirikov's standard map, a discrete-time conservative dynamical system. These both model complex dynamics found within real world systems, and also lie at opposite ends of the dynamical systems spectrum. In both cases, we were able to evolve ABNs capable of controlling trajectories in a prescribed manner.

However, many real world systems do not have purely continuous or discrete dynamics, but rather a hybrid of the two (Branicky, 2005). These often occur on different time scales, such that continuous state flow is occasionally interrupted by jump discontinuities caused by the occurrence of discrete events. Two common examples of this are physical systems with impact, such as a bouncing ball, and switched

systems, where a signal change causes a discrete change in behaviour. In this paper, we consider a problem which combines both of these: controlling the gait and direction of movement of a simulated legged robot.

Coupling between different classes of biochemical network plays an important part in the functioning of biological cells. The coupling between a genetic network and a metabolic network, in particular, is central to a cell's ability to both specialise and adapt to a changing environment. Taking inspiration from this biological behaviour, we investigate a hybrid ABN architecture, in which an artificial genetic network controls the expression of an artificial metabolic network. Results on the robot locomotion tasks suggest that such an architecture is particularly suited to problems that require reconfigurable dynamical behaviour.

The paper is structured as follows: We first introduce the ABN models used in this work. We then describe how these models are evolved. Finally, we introduce the robotic locomotion tasks to which they were applied, and present results and conclusions.

Artificial Biochemical Network Models

In this section, we describe the three ABN models used in this work: an artificial genetic network (AGN), an artificial metabolic network (AMN), and a hybrid ABN formed from the coupling of an AGN and an AMN. In addition to expressiveness and evolvability, our choice of models is also influenced by a desire for efficiency and simplicity. For this reason, the models use discrete-time rather than continuous-time updates (unlike, for instance, Banzhaf, 2003). Since continuous-time dynamical systems can often be reduced to discrete-time equivalents by taking Poincaré sections (Kantz and Schreiber, 2004), this arguably makes little difference in terms of expressiveness, but does considerably reduce execution time.

Artificial Genetic Network (AGN)

In general, the complex behaviour of biological genetic networks stems not from the complexity of their component parts, but from the complexity of their dynamics. Hence, a simple abstraction such as the Boolean network can display complex behaviour without the need to model biological details such as continuous-valued expression, asynchronous updates, continuous-time, and the presence of transcription factors. Nevertheless, there are advantages to using more complicated models, and in this work we use a continuous-valued generalisation of the Boolean network.

Continuous values have two main advantages. First, they make it easier to interface with external systems, since inputs and outputs do not need to be encoded in binary. Second, the size of the state space is not limited by the number of genes in the network. In a Boolean network, the number of possible states is 2^N, where N is the number of genes, meaning that small networks are always attracted to a limit cycle. When continuous values are used, the state space is infinite (within the limits of representation), meaning that small networks have the potential to exhibit more complex behaviours.

Formally, an AGN consists of an indexed set of genes, G. Each $g_i \in G$ has an expression level λ_i, an indexed set of regulatory inputs R_i, and a regulatory function f_i, which maps the expression levels of its regulatory inputs to its own expression level. The first time the AGN is executed, its expression levels are initialised from an indexed set of initial values, L_G. External inputs can be delivered to the network either by explicitly setting the expression levels of certain genes, or by introducing new regulatory inputs with fixed values. After iterating the network a specified number of times, t_G, outputs are captured from the final expression levels of designated genes.

Artificial Metabolic Network (AMN)

The artificial metabolic network complements the AGN described in the previous section. It is a simple artificial chemistry with continuous-valued chemicals and continuous-valued reactions. Formally, it consists of an indexed set of enzyme-analogous elements E which transform the concentrations of an indexed set of real-valued chemicals C. Each enzyme has a set of substrates $S_i \subseteq C$, a set of products $P_i \subseteq C$, and a reaction m_i which calculates the concentrations of its products based upon the concentrations of its substrates.

The first time the AMN is executed, its chemical concentrations are initialised from an indexed set of initial values, L_C. External inputs are delivered to the network by explicitly setting the concentrations of certain chemicals. At each time step, each enzyme e_i applies its reaction m_i to the current concentrations of its substrates S_i in order to determine the new concentrations of its products P_i. Where the same chemical is produced by multiple enzymes, i.e. when $\exists j, k : j \neq k \wedge c_i \in P_j \cap P_k$, the new concentration is the mean output value of all contributing enzymes: $c_i = \sum_{e_j \in E_{c_i}} {c_i}^{e_j} / |E_{c_i}|$ where E_{c_i} are enzymes for which $c_i \in P_i$ and ${c_i}^{e_j}$ is the output value of e_j for c_i. After iterating the network t_M times, outputs are captured from the final concentrations of designated chemicals.

Coupled Artificial Biochemical Network (CABN)

Biological biochemical networks interact with one another in a number of ways. Perhaps most significantly, the genetic network controls when and where proteins are expressed. This determines which enzymes are present in the metabolic network, and hence which reactions can take place within a cell. In effect, the genetic network is able to reconfigure the cell's processing machinery over the course of time. This behaviour occurs extensively in both single-celled and multicellular organisms. In the former, it allows the metabolism to be changed in order to react to the presence of different

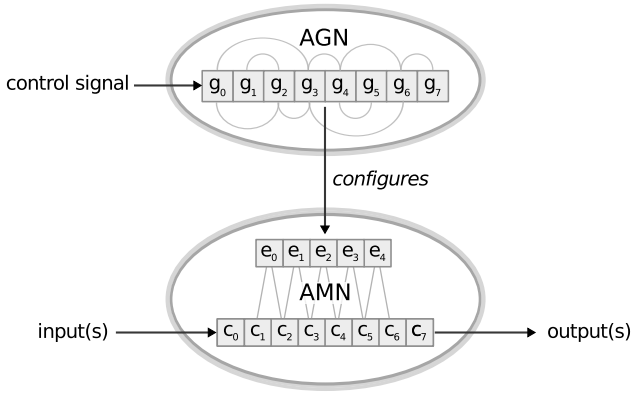


Figure 1: Coupled artificial biochemical network.

kinds of nutrients in the organism's environment. In the latter, it underlies the processes of cell specialisation and development which are fundamental to multi-cellular organisms.

In the coupled artificial biochemical network (CABN) model, we capture this idea of a genetic network controlling the expression of a metabolic network (See Fig. 1). Formally, a CABN comprises an AGN, an AMN, and an injective coupling function $\chi : G_C \rightarrow E$ where $G_C \subseteq G$ is the set of enzyme coding genes, i.e. each enzyme is coupled to a single gene, and some genes may not be enzyme coding (yet are still involved in regulating other genes). Coupling is carried out by giving each enzyme an expression level, ξ_i, and setting this to the expression level of the gene to which it is coupled, i.e. $\forall(g_i, e_j) \in \chi : \xi_j := \lambda_i$. This expression level then determines the relative influence of each enzyme when calculating the new concentration of a chemical:

$$c_i = \sum_{e_j \in E_{c_i}} \frac{\xi_i {c_i}^{e_j}}{\sum_{e_j \in E_{c_i}} \xi_i} \quad (1)$$

i.e. the new concentration is the mean of each enzyme's output value weighted by its relative expression level. This captures the idea that changes in the genetic network lead to changes in the balance between competing pathways in a metabolism.

Regulatory functions and enzyme reactions

Table 1 lists the mathematical functions from which regulatory functions (f) and enzyme reactions (m) are chosen.

Sigmoids model the switching behaviour of many non-linear biological systems, making them a good choice for approximating the behaviours of genetic and metabolic pathways. We use the logistic function, where s determines the slope and b the slope offset (or *bias*). For multiple inputs, $x = \sum_{j=0}^{n} i_j w_j$, where $i_0 ... i_n$ are inputs and $w_0 ... w_n \in [-1, 1]$ are corresponding input weights, with negative values indicating repression.

The remaining functions, all of which are discrete non-linear maps, are motivated by our earlier work (Lones et al., 2010) in which we found that the use of logistic maps within ABNs could lead to the evolution of more effective controllers. We hypothesised that this was due to evolution taking advantage of the complex dynamical behaviours displayed by non-linear discrete maps.

Table 1: Mathematical functions used within ABNs.

Logistic (Sigmoidal) function:

$f(x) = (1 + e^{-sx-b})^{-1}$, where $s \in [0, 20], b \in [-1, 1]$

Logistic map:

$x_{n+1} = rx(1 - x)$, where $r \in [0, 4]$

Arnold's cat map:

$(x_{n+1}, y_{n+1}) = ([2x_n + y_n] \bmod 1, [x_n + y_n] \bmod 1)$

Baker's map:

$$(x_{n+1}, y_{n+1}) = \begin{cases} (2x_n, y_n/2) & 0 \leq x_n \leq \frac{1}{2} \\ (2 - 2x_n, 1 - y_n/2) & \frac{1}{2} \leq x_n < 1 \end{cases}$$

Chirikov's standard map:

$p_{n+1} = (p_n + K \sin\theta_n) \bmod 2\pi, \quad K \in [0, 10]$
$\theta_{n+1} = (\theta_n + p_{n+1}) \bmod 2\pi$

In this work, we extend the approach by using four well-known discrete maps that capture the natural dynamics present in a range of biological and physical systems. *The logistic map* is a model of biological population growth. Depending on the value of parameter r, the system is attracted to either a fixed-point, cyclic or chaotic orbit (May, 1976). *Arnold's cat map* (Arnold and Avez, 1968) is a geometric transformation of the unit square with interesting periodic behaviour. *The baker's map* is an archetypal model of deterministic chaos, capturing the exponential sensitivity to initial conditions that results when kneading bread (Silva, 2008). *Chirikov's standard map* (Chirikov, 1969) captures the behaviour of dynamical systems with co-existing ordered and chaotic regimes. Its dynamics are ordered for low values of parameter K and become increasingly chaotic for higher values. The parameterised maps (the logistic map and Chirikov's map) can be used either with an evolved parameter value or with an extra input, whose current value is used to set the parameter. The latter is referred to as a tunable map, since its dynamics can be modified by the ABN during execution.

Evolving Artificial Biochemical Networks

Our ABNs are evolved using a standard generational evolutionary algorithm with tournament selection (size 4), uniform crossover (p=0.15), and point mutation (p=0.06). Crossover points always fall between gene or enzyme boundaries. Inputs and outputs (R_i, S_i and P_i) are represented by absolute references to indices. Function parameters (e.g. slopes, input weights) and initial values are rep-

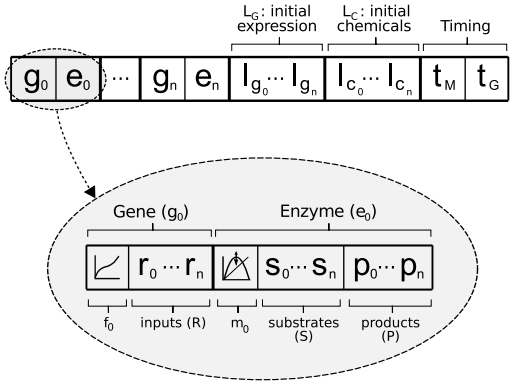


Figure 2: Genetic encoding of an artificial biochemical network.

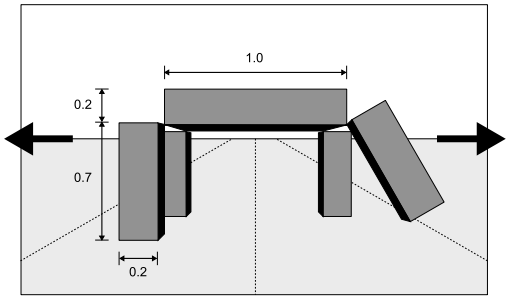


Figure 3: Quadruped robot simulated in Open Dynamics Environment. Arrows indicate the direction of movement along the x-axis plane.

resented as floating-point values and are mutated using a Gaussian distribution centred around the current value.

We use a standardised genetic encoding for all ABN types (see Fig. 2). This represents the ABN as a sequence of genetic units, where each genetic unit has an optional regulatory region and an optional coding region. In a coupled network, the regulatory region encodes the gene and the coding region encodes the enzyme which it expresses. Where a gene does not express an enzyme (such as in an AGN), the coding region is empty. For an AMN, where there are no genes, the regulatory region is empty. The genetic encoding also includes the initial gene expression and chemical concentrations (where applicable) and timing information.

Controlling Legged Robot Locomotion

Legged robot locomotion is a challenging problem. In (Beer and Gallagher, 1992), the authors summarised the challenge by stating "A locomotion system must simultaneously solve the two tightly coupled problems of support and progression." In this paper, we address the locomotion of a simulated quadrupedal robot. There have been a number of previous attempts to evolve quadrupedal locomotion (e.g. Hornby et al., 2005; Kamio et al., 2003; Seo and Hyun, 2008; Clune et al., 2009). Since functional gaits can be generated by tapping sinusoidal functions at appropriate phase offsets, a common approach is to use genetic algorithms (Hornby et al., 2005) or genetic programming (Seo and Hyun, 2008) to generate sinusoid-based controllers. Another, potentially more robust, approach is to evolve neural networks (Beer and Gallagher, 1992; Clune et al., 2009).

Since our focus is upon using legged robot locomotion as a test bed for comparing the expressiveness of different ABN models, the robot (see Fig. 3) is purposely very simple in design, comprising a square top section with four legs connected by actuators at the corners. The actuators are limited to movement in the x-axis plane, with a maximum elevation of 60° from vertical. The robot was simulated using the Open Dynamics Engine (ODE) physics engine, with a step size of $\Delta_t = 0.05$s, friction of 200N, CFM (an ODE parameter) of 10^{-5}, and standard gravity. Actuators have a maximum angular velocity of 3m/s and a maximum torque of 150Nm. These values are sufficient to enable dynamic gaits, but not large enough to allow the body to be dragged by the front legs without the involvement of the rear legs. The ABN is executed every 10 simulation steps.

Generating Quadrupedal Gaits

The first task was to evolve ABNs capable of generating quadrupedal gaits, i.e. patterns of actuator movements which would cause the robot to move away form its starting position. The aim of this task was to determine whether the different ABN types and configurations were able to generate appropriate patterns of movement.

Experimental Settings A controller's fitness is the Euclidean distance moved by the robot within an evaluation period of 500 time steps. The population size is 200, with a generation limit of 100. ABNs have four inputs, corresponding to the actuator angles, and four outputs, which are used to set the torques of the actuators during the next 10 simulation steps. Note that the requirement to map angles to torques adds a degree of difficulty to this task. All inputs and outputs are linearly scaled to the interval $[0, 1]$. For AMNs and CABNs, inputs are delivered via initial chemical concentrations. For AGNs, inputs are delivered via initial gene expression levels.

Results Figure 4 compares the fitness distributions of evolved controllers. This shows that all three classes of ABN are capable of generating gaits which solve the movement task. It also indicates that there is no significant difference in the median performance of the AGN, AMN and CABN models. However, for all ABN models, the best controllers use Sigmoidal functions rather than non-linear maps. Solution length (i.e. network size) has relatively little impact. Examples of evolved behaviours are shown in Figure 5.

These results demonstrate that effective controllers can be expressed using any of the ABN models, although good controllers are more readily found when using Sigmoidal functions. It is interesting to note that there is no observable

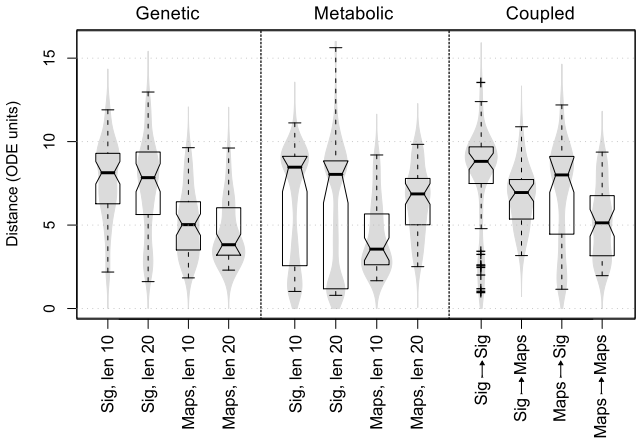
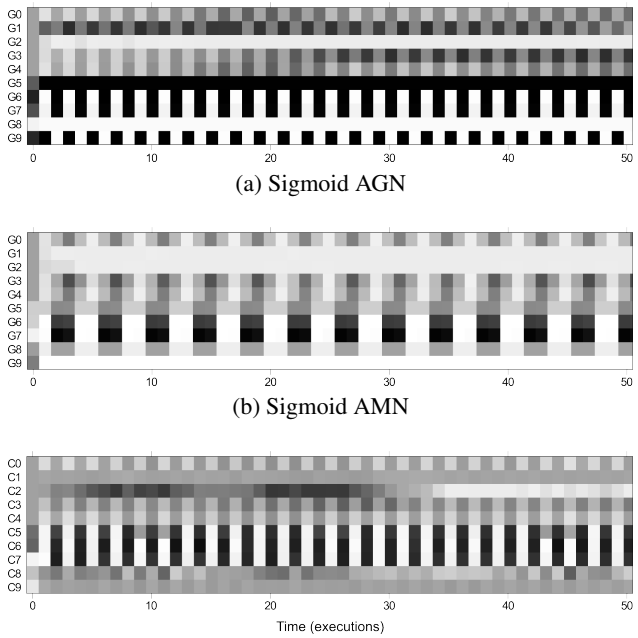


Figure 4: Controlling legged robots using coupled and uncoupled ABNs with sigmoids (Sig) or discrete maps. Summary statistics for 50 runs are shown as notched box plots. Overlapping notches indicate when median values (thick horizontal bars) are not significantly different at the 95% confidence level. Kernel density estimates of underlying distributions are also shown (in grey), showing that some of the distributions are multimodal. The notation Fn1 $\mapsto$ Fn2 denotes a genetic network with Fn1 regulatory functions coupled to a metabolic network with Fn2 enzyme functions. Coupled networks comprise 10 genes (expressing up to 10 enzymes) and 10 chemicals. For uncoupled genetic and metabolic networks, results are shown for solution lengths of both 10 and 20 (genes, or enzymes and chemicals, respectively), to allow fair comparison with the coupled networks.



(a) Sigmoid AGN

(b) Sigmoid AMN

(c) Discrete-map AMN

Figure 5: Time series plots of ABNs generating quadrupedal gaits. Actuator angles are input via the first four gene expression levels (G0–G3) or chemical concentrations (C0–C3), and new torque settings are read from the last four (G6–G9, C6–C9). White represents 0, black represents 1, greyscales represent intermediate values.

penalty to using the structurally more complex coupled networks.

Higher Level Control of Locomotion

The second task introduced an extra level of difficulty, requiring the ABNs to control the robot's direction of movement in addition to its gait. The aim of the task was to test not only the ABNs' abilities to express suitable patterns of movement, but also their ability to switch between different patterns as required.

Objective function The robot is required to change direction by 180° when signalled to do so, whilst still moving as far as possible in the given direction. Controller fitness is measured over a sequence of epochs $< e_0, ..., e_{N-1} >$, each with a random duration between 300 and 600 time steps, with the required direction of movement reversing during subsequent epochs. The fitness function f is defined:

$$f = \frac{t_{\max} - t_{\min}}{N} \min\{\sum_{n \in \mathbb{N}_{\text{even}}, n<N} p(n), \sum_{n \in \mathbb{N}_{\text{odd}}, n<N} p(n)\} \quad (2)$$

where $t_{\max}$ and $t_{\min}$ are the maximum and minimum bounds on epoch duration and $p(n)$ is the progress made during epoch n, defined:

$$p(n) = \frac{d_n}{t_n}(2\frac{\eta_b(e_n, e_{n+1})}{\pi} - 1)(1 - \frac{\eta_w(e_n)}{\pi})\sigma_n \quad (3)$$

where d_n is the distance travelled during epoch n, t_n is the duration of epoch n, η_b is the difference in mean heading between two epochs, η_w is the difference in heading within an epoch (as measured during the first and last 50 time-steps of the epoch), and σ_n is a penalty for non-movement: equal to 1 if the robot has not moved for 100 subsequent ABN updates in epoch n, and 0 otherwise.

In effect, progress is the mean velocity in the required direction, with penalties for turning during an epoch and for non-movement. Assuming movement in a straight-line and no stopping, fitness is equivalent to the expected distance covered during an epoch in the forward or backward direction, whichever is shortest.

Experimental Settings A population of 500 is used for this task, to reflect its greater difficulty. In addition to the four actuator angles, the ABN also receives a direction input. This has the value 0 during even-numbered epochs and 1 during odd-numbered epochs. In addition to delivering this signal with the actuator angles, for AGNs and CABNs we also look at the effect of delivering the signal separately through the first regulatory input of one or more genes.

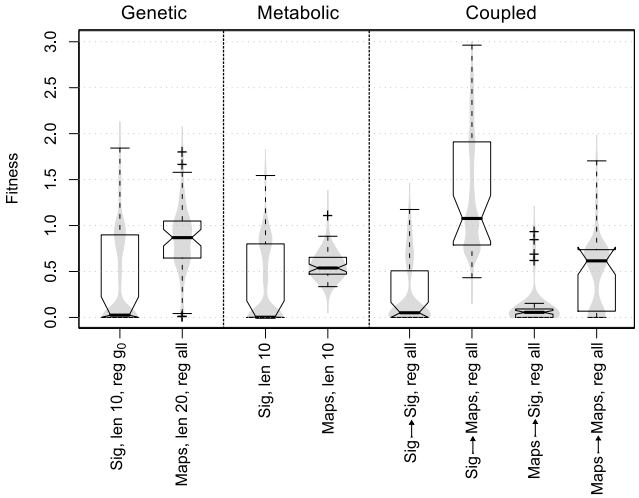


Figure 6: Controlling direction and movement of legged robots. For each function set (or pair of function sets in the case of the coupled network), results for the best-performing combination of solution size and (for genetic and coupled networks) regulatory signal destination are shown. For the latter, g_0 indicates that the control signal was delivered as a regulatory input to the first gene, all indicates that the control signal was delivered as a regulatory input to all genes.

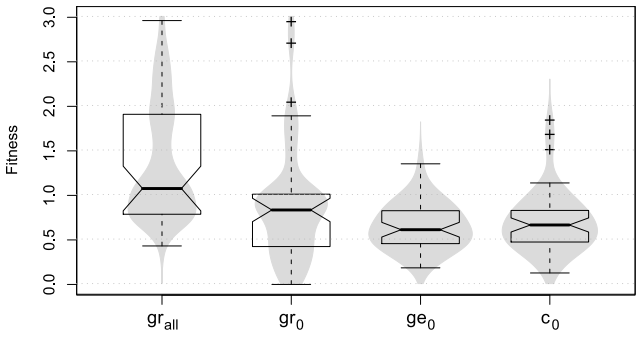


Figure 7: Comparing the effect of delivering the direction signal to different locations within the Sig $\mapsto$ Maps coupled network. gr_{all} indicates a regulatory input to all genes, gr_0 is a regulatory input to the first gene, ge_0 is the initial expression of the first gene, and c_0 is the concentration of the first chemical.

Results Well-behaved controllers (i.e. those which correctly respond to the direction signal and produce effective gaits) generally have a fitness greater than about 1.5: those with lower fitnesses tend to have periodic or inconsistent behaviours.

Figure 6 compares the fitness distributions of evolved controllers, suggesting that most combinations of ABN model and function set choice do not lead to well-behaved controllers. In fact, the majority of evolved Sigmoidal AGN and AMN were only capable of movement in one direction, giving them a median fitness of zero. Discrete-map AGNs and AMNs achieved higher fitness, but generally did not respond to the direction signal, displaying a range of periodic and aperiodic behaviours.

Table 2: Occurrence of discrete maps within final solutions from all Sig $\mapsto$ Maps CABN runs where fitness is greater than 1.5.

Maps	In solutions	Mean occurrences per solution
Baker's map	100%	2.3
Tunable standard map	78%	1.6
Standard map	78%	1.6
Tunable logistic map	72%	1.2
Arnold's cat map	61%	1.5
Logistic map	50%	1.7

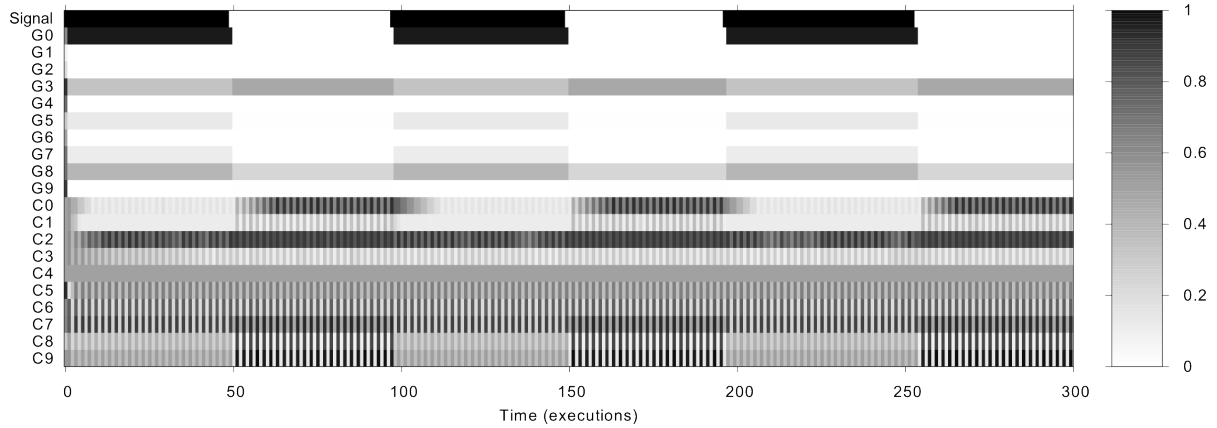
Notably, only coupled networks comprising a Sigmoidal AGN and a discrete-map AMN (denoted Sig $\mapsto$ Maps) were able to consistently generate competent controllers[1], and only when the direction signal was delivered as a regulatory input to each gene. Figure 7 shows the effect of delivering this signal to other locations within the Sig $\mapsto$ Maps coupled networks; showing that delivering the direction signal via a gene's initial expression or a chemical's initial concentration was generally ineffective.

Figure 8 shows some representative examples of how these Sig $\mapsto$ Maps networks control gait and respond to the direction signal. In most evolved networks, the AMN is responsible for generating appropriate patterns of actuator movements and the AGN is responsible for switching between different patterns by regulating the influence of different enzymes. It is interesting to note that their behaviour over time resemble the dynamics of biological biochemical networks, in that a slow-changing genetic network controls a fast changing metabolic network. This may also explain why Sigmoidal functions, which are more amenable to producing slow-changing dynamics, play a productive role within coupled controllers but not within the stand-alone AMN and AGN controllers.

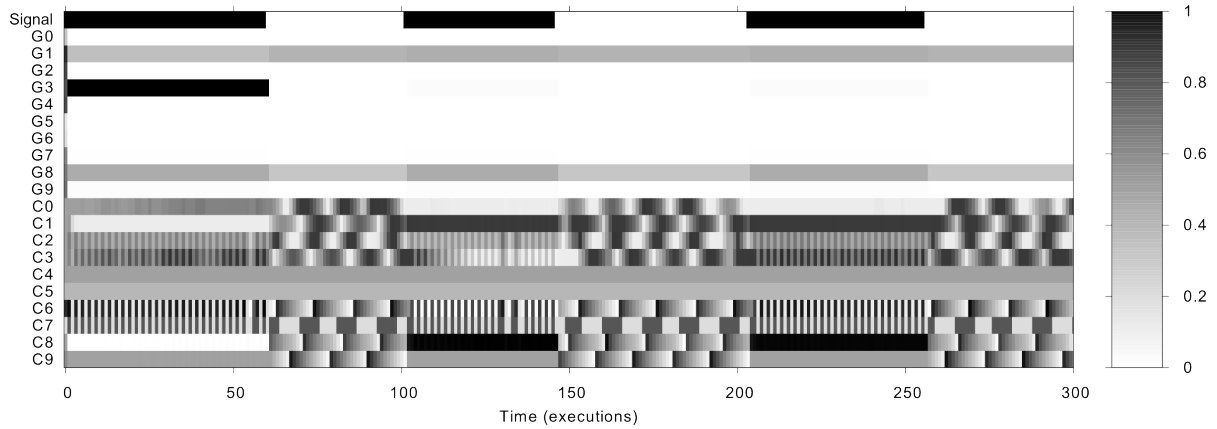
We can hypothesise that there are two reasons why discrete maps are useful for this task. First, they can individually carry out behaviours which would require a number of interconnected Sigmoids to implement—to use a biological analogy, they are the equivalent of a whole biochemical pathway. Arguably, this entails that certain pattern generators can be evolved more readily than in a Sigmoidal network, and using fewer genes. Second, all the discrete maps we use have chaotic phases. When in this phase, their dynamics are highly sensitive to small perturbations, meaning that relatively small changes in gene expression can lead to rapid switching between different attractor states—precisely the behaviour we are looking for in many control tasks.

Table 2 lists the relative occurrence of the different dis-

[1] 18 of the 50 runs generated solutions with fitness greater than 1.5, compared to only a handful for all the other ABNs.



(a) In this example, the AMN generates a single cyclic pattern (C5) which is then scaled and propagated to the outputs (C6–C9). The scaling for each output (and hence the direction of the resulting gait) is determined by the current gene expression pattern.



(b) In this second example, the AMN generates two different cyclic patterns (bunny hopping and a four-legged wading movement), which the AGN switches between in response to changes in the direction signal.

Figure 8: Time series plots of Sig $\mapsto$ Maps coupled ABNs controlling the direction and gait of a legged robot. The Signal input specifies the required direction of movement. GO–G9 are the expression levels of the genes in the AGN. C0–C9 are the concentration levels of the chemicals in the AMN.

crete maps in the final solutions of successful runs. All of the maps are used by evolution, with most of them appearing in the majority of solutions. The baker's map, in particular, appears in all of the successful controllers, and usually occurs multiple times in these solutions. Since the baker's map is a model of deterministic chaos, this supports our hypothesis that chaotic dynamics are useful. The standard map is also well-represented in evolved solutions, perhaps reflecting its relatively high degree of expressiveness and configurability. It is also notable that the tunable versions of the logistic and Chirikov's maps are often used.

Conclusions

In this paper, we have shown that artificial biochemical networks can be evolved to control the locomotion of a simulated legged robot. We used two artificial biochemical network models—an artificial genetic network and an artificial metabolic network—and looked at how these models can be used both individually and when coupled together.

For a simple movement task, where the robot was required to move as far as possible from its starting position, both individual and coupled networks could be evolved to generate suitable gaits. However, for a harder task, where the robot was required to reverse its direction of movement when given a signal, only coupled networks could be evolved to express suitable behaviours. Analysis of the resulting controllers suggests there is a clear separation of effort, with the artificial metabolic network generating patterns of actuator movements and the artificial genetic network switching between different patterns as appropriate.

We found that non-linear discrete maps play an important role in solving the harder of the two problems. When used as functional elements within artificial biochemical networks, these maps provide a useful source of configurable pre-packaged dynamics. Of the maps used in this study, the chaotic baker's map occurred most within evolved solutions.

This finding supports the idea that the inherent instability of chaotic maps makes them useful for rapidly switching between different behaviours.

We also found that the destination of the direction signal has a large effect upon the ability of the networks to solve the harder task. This may reflect the important role that signal recruitment plays within the evolution of biological biochemical networks. Rather than pre-specifying the destination of signals, as we have done in this work, in future work we will look at whether an artificial signalling network can be used to deliver signals to appropriate parts of the genetic and metabolic networks.

Acknowledgements

This research is funded by the EPSRC (ref: EP/F060041/1). The authors would also like to acknowledge the support of the White Rose Grid in providing computational resources.

References

Arnold, V. and Avez, A. (1968). *Ergodic problems in classical mechanics*. Benjamin, New York.

Banzhaf, W. (2003). Artificial regulatory networks and genetic programming. In Riolo, R. L. and Worzel, B., editors, *Genetic Programming Theory and Practice*, chapter 4, pages 43–62. Kluwer.

Banzhaf, W. (2004). Artificial chemistries—towards constructive dynamical systems. *Solid State Phenomena*, 97/98:43–50.

Banzhaf, W. and Lasarczyk, C. (2005). Genetic programming of an algorithmic chemistry. In Koza, J., O'Reilly, U.-M., Yu, T., Riolo, R., and Worzel, B., editors, *Genetic Programming Theory and Practice II*, pages 175–190. Springer US.

Beer, R. and Gallagher, J. (1992). Evolving dynamical neural networks for adaptive behavior. *Adaptive Behavior*, 1(1):91–122.

Branicky, M. S. (2005). Introduction to hybrid systems. In Hristu-Varsakelis, D. and Levine, W., editors, *Handbook of Networked and Embedded Control Systems*. Birkhauser.

Bray, D. (1995). Protein molecules as computational elements in living cells. *Nature*, 376:307–312.

Chirikov, B. V. (1969). Research concerning the theory of nonlinear resonance and stochasticity. Technical report, Institute of Nuclear Physics, Novosibirsk.

Clune, J., Beckmann, B. E., Ofria, C., and Pennock, R. T. (2009). Evolving coordinated quadruped gaits with the HyperNEAT generative encoding. In Tyrrell, A. et al., editors, *Proc. 2009 Congress on Evolutionary Computation (CEC 2009)*. IEEE.

Decraene, J., Mitchell, G. G., and McMullin, B. (2007). Evolving artificial cell signaling networks: Perspectives and methods. In Dressler, F. and Carreras, I., editors, *Advances in Biologically Inspired Information Systems*, pages 167–186. Springer.

Fontana, W. (1992). Algorithmic chemistry. In Langton, C. G., Taylor, C., Farmer, J. D., and Rasmussen, S., editors, *Artificial Life II*, pages 159–210. Addison-Wesley.

Hornby, G., Takamura, S., Yamamoto, T., and Fujita, M. (2005). Autonomous evolution of dynamic gaits with two quadruped robots. *IEEE Transactions on Robotics*, 21(3):402–410.

Kamio, S., Mitsuhashi, H., and Iba, H. (2003). Integration of genetic programming and reinforcement learning for real robots. In Cantú-Paz, E. et al., editors, *Proc. 2003 Genetic and Evolutionary Computation Conference (GECCO'03)*, volume 2723 of *LNCS*, pages 470–482, Chicago. Springer-Verlag.

Kantz, H. and Schreiber, T. (2004). *Nonlinear Time Series Analysis*. Cambridge University Press, 2nd edition.

Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *J Theor Biol*, 22(3):437–467.

Lones, M. A., Tyrrell, A. M., Stepney, S., and Caves, L. S. (2010). Controlling complex dynamics with artificial biochemical networks. In Esparcia-Alczar, A. I. et al., editors, *Proc. 2010 European Conference on Genetic Programming (EuroGP 2010)*, volume 6021 of *Lecture Notes in Computer Science*, pages 159–170. Springer Berlin / Heidelberg.

May, R. M. (1976). Simple mathematical models with very complicated dynamics. *Nature*, 261:459–467.

Nicolau, M., Schoenauer, M., and Banzhaf, W. (2010). Evolving genes to balance a pole. In Esparcia-Alczar et al., editors, *Proc. 2010 European Conference on Genetic Programming (EuroGP 2010)*, volume 6021 of *Lecture Notes in Computer Science*, pages 196–207. Springer Berlin / Heidelberg.

Păun, Gh. (2000). Computing with membranes. *Journal of Computer and System Sciences*, 61(1):108–143.

Reil, T. (1999). Dynamics of gene expression in an artificial genome - implications for biological and artificial ontogeny. In *Proc. 5th European Conference on Artificial Life (ECAL'99)*, volume 1674 of *Lecture Notes in Artificial Intelligence*, pages 457–466. Springer-Verlag.

Seo, K. and Hyun, S. (2008). Genetic programming based automatic gait generation for quadruped robots. In Keijzer, M. et al., editors, *Proc. 2008 Genetic and Evolutionary Computation Conference (GECCO'08)*, pages 293–294, Atlanta, GA, USA. ACM.

Silva, C. E. (2008). *Invitation to ergodic theory*. AMS.

Taylor, T. (2004). A genetic regulatory network-inspired real-time controller for a group of underwater robots. In Groen, F. et al., editors, *Intelligent Autonomous Systems 8 (Proceedings of IAS8)*, pages 403–412, Amsterdam. IOS Press.

Trefzer, M. A., Kuyucu, T., Miller, J. F., and Tyrrell, A. M. (2010). Image compression of natural images using artificial gene regulatory networks. In *Proc. 2010 Genetic and Evolutionary Computation Conference (GECCO'10)*, Portland, Oregon. ACM.

Ziegler, J. and Banzhaf, W. (2001). Evolving control metabolisms for a robot. *Artificial Life*, 7:171–190.