Temporall adaptive networks: Anal sis of GasNet robot controllers

Tom Smith[*1], **Phil Husbands**[2] **and Michael O'Shea**[1]
Centre for Computational Neuroscience and Robotics (CCNR)
[1]School of Biological Sciences, [2]School of Cognitive and Computing Sciences
University of Sussex, Brighton, UK
[*] toms@cogs.susx.ac.uk

A good performance, like a human life, is a temporal affair: a process in time.
- ■ortimer J. Adler

Abstract

There are immense problems in developing artificial nervous systems for autonomous machines operating in non-trivial environments. In particular, no principled methodology is in place to decide between solution classes and representations, and between methods by which solutions might be developed using hand-design or search techniques. In this paper we apply the techniques of dynamical systems theory to the analysis of successfully evolved robot control systems, in order to identify useful properties of the underlying control architecture. We investigate the suitability of two different neural network classes for a robotic visual discrimination task, through analysis of both successful controller behaviour and continued evolution of successful solutions in environments with modified characteristics. We argue that the temporally adaptable properties of the GasNet class identified through dynamical systems analysis, and found to be useful in order to re-evolve in modified environments, are crucial to the evolution of successful controllers for the original environment.

1 Introduction

Identification of control system classes capable of generating adaptive behaviour over time is a black art. ■any practitioners rely on systems that have "always worked in the past", others may use trial-and-error until success, but carry out no subsequent analysis of why that particular system actually worked. A major problem with such approaches is that it is extremely difficult to develop a more general understanding of the properties necessary for generating adaptive behaviour. In particular, is a particular solution class appropriate for a particular problem? Addressing this issue is crucial if we are to successfully apply techniques such as evolutionary computation to more complex adaptive behaviour problems than at present.

In this paper we develop an approach based on analysis of successfully evolved solutions. This allows us to identify properties of network classes that are potentially useful over a wider class of problems than the original task. We then develop a methodology for testing these properties, through analysis of the evolved solutions in modified environments.

We overview two classes of neural network, the "GasNet" and "NoGas", used as controllers in a visual shape discrimination problem, and give evidence that the GasNet class is more amenable to evolutionary search than the NoGas class. We then use the techniques of dynamical systems analysis to identify possible reasons for this increased evolutionary speed, and frame a number of hypotheses for the suitability of the GasNet class to robot control. In particular, we show how the properties of gas diffusion can be used to

filter out sensor input noise, produce simple pattern generation networks, and switch networks from one stable state to another. We hypothesise that these properties lead to GasNet solution spaces in which it is easier to find good controllers than the corresponding NoGas solution spaces.

We go on to compare the operation of two controllers, one GasNet solution and one NoGas solution, which utilise the same visual shape discrimination strategy. We argue that the GasNet controller is easier to tune to the

through oxidation), C_0 is a global constant, t_e is the last time at which emission was initiated, t_s is the last time at which emission ceased, and s is a genetically determined constant. It should be emphasised that r, C_0, s are determined uniquely for each node by the network genotype, and t_e, t_s will typically be different for each node in the network during operation, so the GasNet network class is heterogeneous in the sense that node properties are not the same across the network.

In other words, the gas concentration varies spatially as a Gaussian centred on the emitting node. The height of the Gaussian at any point within the circle of influence of the node linearly increases or decreases over time depending on whether or not the node is emitting gas, according to the function $T\ t)$ which saturates at a maximum of 1 and a minimum of 0. The total concentration at any point in the network is found by summing the concentrations from all emitting nodes. Figure 1 shows a possible GasNet, with node 3 emitting gas. Increased gas concentration is shown centred on the emitting node, extending out as far as node 2; thus node 3 can affect node 2 via the modulatory gas effect despite there being no direct synaptic connections.

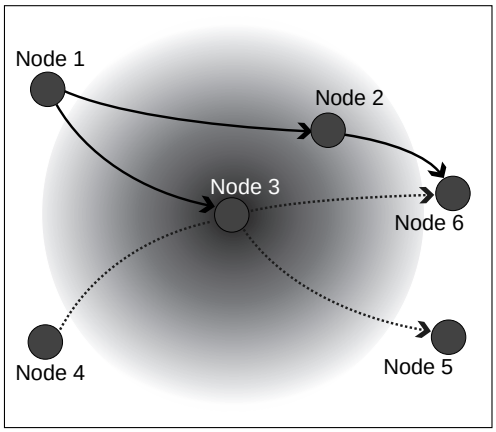


Figure 1: A hypothetical GasNet, with node 3 emitting gas. Increased gas concentration is shown centred on the emitting node, extending out as far as node 2; thus node 3 can affect node 2 via the modulatory gas effect despite there being no direct synaptic connections. All other nodes lie outside the radius of gas concentration, so are not affected by the gas concentration. However, nodes 5 and 6 are affected by synaptic output from node 3. Nodes 1 and 4 similarly affect node 3 through synaptic output.

It should be stressed that the model is a greatly simplified form of the real diffusion process, and certainly does not model the physics of diffusion accurately. Diffusion occurs instantaneously, with no spread of gas concentration over time. In other words nodes close to the emitting node are affected at the same time as nodes far from the emitting node. Also, individual node contributions are summed independently without regard to the local concentration gradient, thus concentration can flow from areas of low concentration to areas of high concentration. Finally, the two gases do not interact, so concentration of one does not affect the diffusion of the other. However, even such an abstract model can capture some of the properties of real diffusing gaseous modulation, and in the next section we describe the modulatory effects of gas concentration.

2.2 Modulation by the gas4970n020TdngT11.3st laffect

$$f\ x) \quad = \quad \begin{cases} 0 & x \leq 0 \\ \lfloor x \rfloor & 0 < x < N \\ N-1 & \text{else} \end{cases} \tag{8}$$

where $\mathbf{P}[i]$ refers to the ith element of set $\mathbf{P}$, D_i^t

the phenotype value was obtained by normalising the genotype value to lie in the range $[0.0, 1.0]$[2] and multiplying by the relevant variable range. For nominal values, such as whether the node has a visual input or not, the phenotype value was calculated through the binary modulo division operator:

$$p = \begin{cases} g/99 & \text{continuous } p \\ g \bmod N_{nom} & \text{nominal } p \end{cases} \quad (9)$$

where p is the phenotype value, g the genotype value, N_{nom} the number of possible nominal phenotype values, and mod the binary modulo division operator, that is the remainder when g is integer divided by N_{nom}.

Each node in the network had either 19 or 25 variables associated with it, depending on which of two possible connectivity encoding schemes were used section 3.2). All variables were under evolutionary control, see figure 3. A single genotype thus consists of a string with length as some multiple of 19 or 25, coding for a variable number of network nodes.

$$\begin{array}{lrcl} & < genotype > & :: & < gene >)^* \\ \text{where} & < gene > & :: & < x >< y > \ < Connection >)^* < I_{on} >< I_r >< I_\theta >< I_{thr} > \\ & & & \quad < rec >< TE >< CE >< s >< R_e >< D^0 >< bias > \\ \text{Arcs:} & < Connection > & :: & < R_p >< \Theta_{1p} >< \Theta_{2p} >< R_n >< \Theta_{1n} >< \Theta_{2n} > \\ \text{Points:} & < Connection > & :: & < Pt_x >< Pt_y >< Pt_w >)^4 \end{array}$$

Figure 3: The genotype-to-phenotype mapping for the arcs and connections network schemes described in section 3.2.

The encoding shown in figure 3 was used to generate networks conceptualised to exist on a 2D Euclidean plane. $< x >$ and $< y >$ give the position of a network node on the plane. The next 6 or 12 numbers define the synaptic connectivity of the network; section 3.2 gives details of the *arc* and *point* schemes used to derive the connectivity. The rest of a gene is interpreted as follows. $< I_{on} >$ is a binary switch that determines whether or not a node has visual input. If it does, the following three variables encode the polar coordinates of a pixel in the camera image that the node will take input from, and a threshold below which input from that pixel is ignored visual input is normalised to lie in the range $[0.0, 1.0]$, this is the range of the threshold). See section 3.3 for details of the visual input to the network.

The value of $< rec >$ determines whether the node has an excitatory recurrent connection, an inhibitory recurrent connection or no recurrent connection to itself. $< TE >$ provides the circumstances under which the node will emit a gas. These are one of either: not at all; if the node electrical activity exceeds some threshold; if the concentration of gas 1 at the node site exceeds some threshold; or if the concentration of gas 2 at the node site exceeds some threshold the electrical and gas thresholds are set at 0.5 and 0.1 as described in section 2.1). $< CE >$ specifies the gas that the node can emit under the correct condition, either gas 1 or gas 2. $< s >$ is used to control the rate of gas build up/decay as described earlier by equation 3, its value ranges from 1 to 11. $< R_e >$ is the maximum radius of gas emission, this ranges from $10\% - 50\%$ of the plane dimension. $< D^0 >$ is the default value for the index used in equation 7 to determine the transfer parameter value K_i^t for each node. Finally, $< bias >$ is the bias term b_i in the node transfer function equation 1), restricted to the range $[-1.0, 1.0]$.

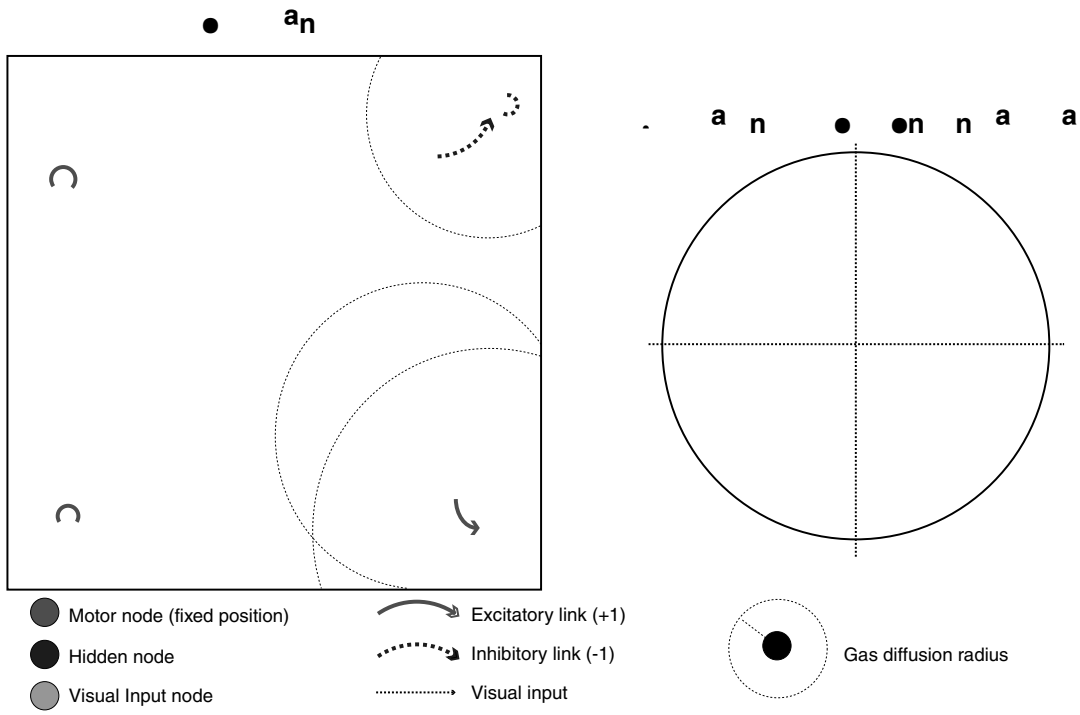
The encoding scheme used was the same for both the GasNet and NoGas classes, with the NoGas genotypes effectively encoded with a number of introns. For the NoGas controllers, certain of the genotype parameters were ignored completely. These parameters were $< TE >, < CE >, < s >, < R_e >$, encoding for the parameters of gas diffusion at each node.

[2] This can be regarded as an approximation to a continuous $[0, 1]$ range; experiments show no significant difference between the two setups.

3.2 The network connectivity: Arcs and points

was also investigated whereby two of the points encoded +1 weighted connections, and two −1 weighted connections, but no significant differences were seen.

3.3 The network visual input and motor output



which the experimenter has access to certain global position information that is not passed directly to the evolutionary process. This position information may be used in the fitness function, for example to determine how close the robot approached to the triangle.

The minimal simulation of the gantry was developed by Jakobi 1998a,b). The base set of robot-environment interactions upon which behaviour could be reliably based, consisted of only two members. First, the way in which pixels of the camera image that are sampled from the walls of the arena but not from the floor or above the walls) return grey-scale values within certain intervals: over the range $[14, 15]$ for pixels that project onto either the triangle or the square, and over the range $[0, 13]$ for pixels that project onto the walls of the arena, but not onto either the triangle or the square. Second, the way in which the robot moves in response to motor signals.

To ensure that controllers were both base set robust and base set exclusive, in other words that controllers relied only on base set interactions and not on implementation aspects of the model, all other parameters were modelled unreliably. Over the possible ranges of pixel inputs, $[14, 15]$ for pixels that project onto either the triangle or the square, $[0, 13]$ for pixels that project onto the walls of the arena, and the entire $[0, 15]$ range for other cases, values were returned unreliably remember that the base set aspect is the range over which the pixel values are returned, *not* the way in which they are set over this range). This unreliability was set at the start of each trial, with possible effects varying pixel inputs as a function of time, or as a function of the orientation of the robot, or fixed for the entire evaluation at a random level set before each trial. The momentum of the robot was also made unreliable, with the momentum being fixed at the start of each trial. Similarly, small offsets were added to the wheel speeds, camera horizontal and vertical angles, and the positions of the shape vertices, with the offsets set randomly at the start of each trial. For further details see Jakobi 1998a,b).

4.2 Visual shape discrimination

Starting from an arbitrary position and orientation in a black-walled arena, the robot must navigate under extremely variable lighting conditions to one shape a white triangle) while ignoring a second shape a white square). Fitness over a singlesee

tdbkekd1.9929T64[0(2)0)4T7lstry)]TJthe7wheelonstartrando2of

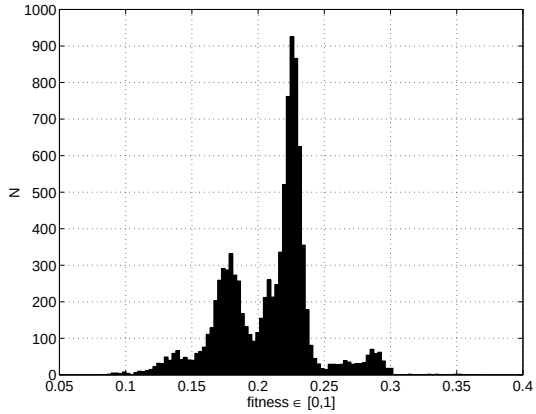


Figure 7: The fitness distribution of a single genotype evaluated 10

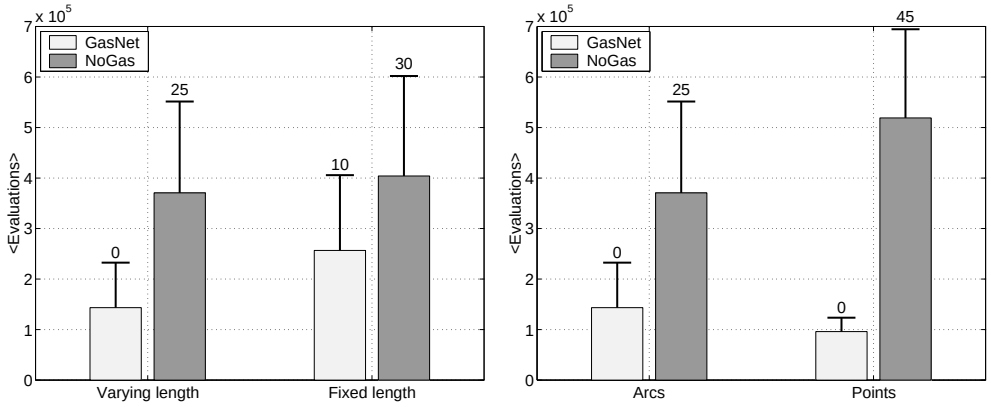
5 The evolutionary machinery

5.1 Mutation and recombination operators

Three mutation operators were applied to solutions with probability $\mu\%$ during the evolution and re-evolution experiments

5.3 Speed of evolution results

The evolution of solutions based on the GasNet class consistently produces successful robot control solutions in significantly fewer evaluations than required by the evolution of solutions based on the NoGas class. This result holds over a number of different evolutionary algorithms, with a number of different



(a) Variable and fixed length genotypes

(b) Arcs and points connectivity schemes

Figure 10: The mean number of evaluations required for **(a)** variable and fixed length genotypes, and **(b)** Arcs and points connectivity schemes. Data averaged over twenty runs of the distributed evolutionary algorithm. The error bars represent 95% confidence limits for the mean, the number above the bar gives the percentage of runs failing to finish in $1,000,000$ evaluations.

distribution of the search space surrounding solutions to build up a description of the fitness landscape. However, applying such measures to the GasNet and NoGas search spaces shows no measurable differences in ruggedness, modality and neutrality between the landscapes underlying the two classes Smith et al., 2002b).

through approaching a different attractor state. Such analysis has been carried out to understand the behaviour of a variety of evolved robot controllers, most especially in the work by Beer and co-workers see for example Beer, 1990; Beer and Gallagher, 1992; Beer, 1995; Chiel et al., 1999; Calvatti and Beer, 2001). However, in this paper we are interested in analysing the controllers operation in terms of how easy or difficult such controllers may be to evolve, especially when using different robot control classes. In the next two sections, we apply the basic techniques to two example dynamical systems; a predator-prey population model, and an evolved pattern generation GasNet controller.

7.2 An example dynamical system: Predator-prey populations

The classic biological dynamical system is the predator-prey or host-parasite) population model famously studied by both Lotka 1925) and Volterra 1926). In this model we are interested in how the populations of the two species vary over time, especially with respect to initial conditions of the state variables.

Consider a population x. Over time, the population increases exponentially in size through breeding, but with a self-limiting factor dependent on the current population size, for example due to overcrowding or limited food resources. Thus we derive the differential equation for the rate of change of the population:

$$\frac{dx}{dt} = a.x\ 1 - b.x) \tag{11}$$

Now consider a prey population x, and a predator population y. Both populations change over time as given by equation 11, although we neglect the self-limiting factor for the prey population as we assume the predators never let the prey population reach such a level. However, there is an additional population interaction term when both predator and prey are present: the probability of a predator-prey encounter is proportional to the product of *both* the predator and prey populations. Thus the predator population increases, and the prey population decreases, with rate proportional to the product of the two populations. From these premises we derive the differential equations, or dynamical laws, governing the rate of change of the two populations:

$$\frac{dx}{dt} = a.x\ 1 - y) \tag{12}$$

$$\frac{dy}{dt} = b.y\ 1 - c.y + x) \tag{13}$$

the Volterra equations for the predator-prey system. In the rest of this paper, we consider discrete time-step neural networks in which the activity of a network node is derived directly, rather than through differential equations. Thus we use the discrete form of the above equations Sandefur, 1990, gives a good introduction to discrete dynamical systems):

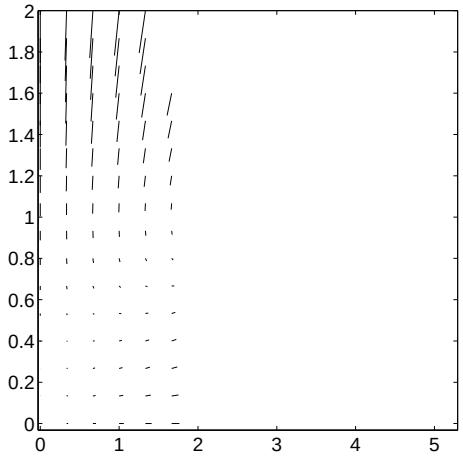
$$\frac{x^{t+1} - x^t}{\Delta t} = a.x^t\ 1 - y^t) \tag{14}$$

$$\frac{y^{t+1} - y^t}{\Delta t} = b.y^t\ 1 - c.y^t + x^t) \tag{15}$$

the change in x and y from time-step $t \rightarrow t+1$. Without loss of generality, we can take the time-step $\Delta t = 1$. Thus we have our dynamical equations for the predator-prey system $\mathbf{A}^{t+1} = \mathbf{F}\ \mathbf{A}^t)$:

$$\begin{pmatrix} x^{t+1} \\ y^{t+1} \end{pmatrix} = \begin{pmatrix} a.x^t\ 1 + 1/a - y^t) \\ b.y^t\ 1 + 1/b - c.y^t + x^t) \end{pmatrix} \tag{16}$$

Now, we are interested in the behaviour of the system over time, so we need to find the equilibrium values. When the system is in equilibrium, the state variables do not change over time: $d\mathbf{A}/dt = 0$, or $\mathbf{A}^{t+1} = \mathbf{A}^t$. Solving for our system, we find three equilibrium values at $0, 0)$, $0, 1/c)$ and $c - 1, 1)$.


2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
1
2
3
4
5

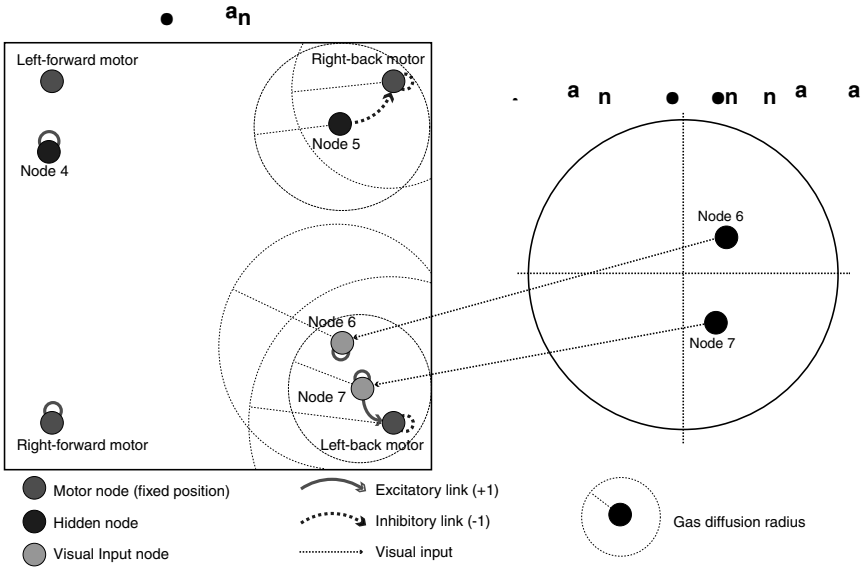
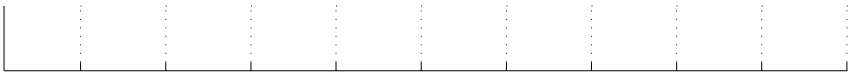


Figure 13: *Open-loop* GasNet visual discrimination network. Gas diffusion radii are shown only where diffusion occurs. The node plane is shown with x, y positions of each node, the connections between each node (indicating whether etof



gas 2. Consideration of the concentration of this negative gas in the region of the motor node shows that K_2 will decrease from 4 to 0.25, seen in

to identify a

tanh $-0.5y + 0.48$) and $y_1 = \tanh\ 0.16$) respectively, while node 4 stabilises at a constant negative value, $y_4 = \tanh\ 0.25y - 0.28$) but is unused by the network. Thus both forward motor nodes are continually on, and behaviour is governed by the two subnetworks acting on the back motor nodes. In the next section we analyse the switching subnetwork.

8.1 Stable state switching

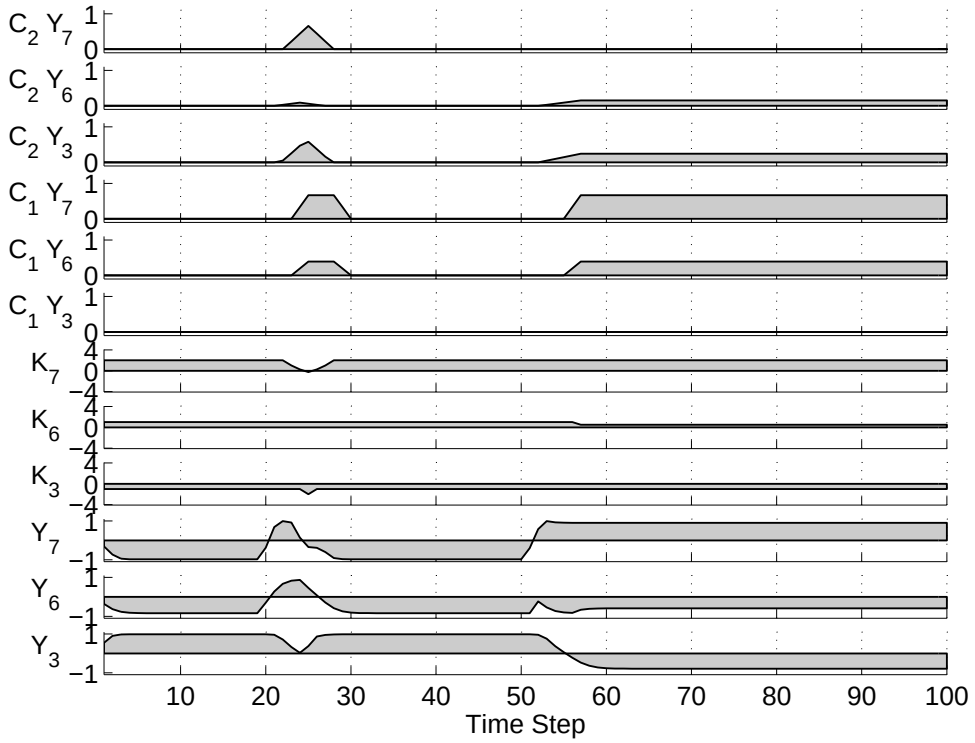


Figure 17: For nodes 3 (the left-back motor node), 6 and 7, involved in the 'switch' subnetwork described fully in section 8.1, the figure shows data over a run of 100 time-steps for node output $Y \in [-1, 1]$, node transfer parameter $K \in [-4, 4]$, positive and negative gas concentrations $C_1, C_2 \in [0, 1]$ at the node site. Area between the output and time axis is shaded for clarity.

In the open-loop controller, the only nodes receiving external visual input are in the subnetwork involved in the triangle discrimination network, consisting of the left-back motor node 3, and nodes 6 and 7. The subnetwork, shown in the bottom-right corner of the node plane figure 13), regulates the left-back motor node through electrical synapse and gas diffusion effects. Both nodes 6 and 7 receive recurrent and visual input, while the motor node 3 receives recurrent input, plus an input from node 7. Figure 17 shows the output Y, transfer function parameter K, and gas concentrations C_1, C_2 for the three nodes. The three node subnetwork produces a dynamic system which can produce a permanent switch from one stable state to another, when a specific combination of high external sensory input is received. Note that we are treating the subnetwork as a non-autonomous dynamical system, receiving external input from the environment. External visual input is received by nodes 6 and 7, but only when the visual input level is above the genetically specified node input thresholds. We analyse the subnetwork behaviour for the cases when inputs are below threshold, and when inputs are above visual input threshold levels.

8.1.1 Inputs below threshold

Both nodes 6 and 7 have the same high visual input threshold, with only intensities above 0.84 having any effect. So we can investigate the case when input is below this, where the equations simplify to $y_6^t = \tanh\ y_6^{t-1} - 0.38)$ and $y_7^t = \tanh\ 2y_7^{t-1} - 0.32)$ in the absence of gas, $K_3^t = -1, K_6^t = 1$ and $K_7^t = 2)$. The stable solution to these equations is $y_3 \approx 1.0; y_6 \approx -0.8; y_7 \approx -1.0$. Note that y_7 has 3 stable fixed points, but applying fixed point stability analysis section 7) shows that from an initial position of $y_7 = 0.0$, the $y_7 \approx -1.0$ solution is reached. However, the other y_7 solutions are crucial when input is above threshold, a situation that is analysed in the sections below. Both nodes 6 and 7 emit negative gas when output activity is high, but this is not the case for the stable point. In the presence of negative gas, node 3 emits positive gas - again this is not the case for the stable point.

Thus we have the general picture when no visual input is received above the threshold level of 0.84. Both visual input nodes 6 and 7 are highly inhibited, and the left-back motor node 3 is highly excited. No nodes are emitting gas, and gas concentrations are zero in the neighbourhood of each node. Thus the left motor is inhibited, and the robot circles counter-clockwise, due to the right motor being on for seven in eight time-steps remember that the spiking subnetwork on the left-back motor node only turns off the motor one in eight time-steps, see section 7.3). So what happens when inputs are above threshold?

8.1.2 Inputs above threshold

The following analysis assumes inputs take their maximum value of 1.0, but is qualitatively the same for all values above the visual input thresholds of 0.84. In the presence of high input to both nodes again in the absence of high gas concentrations), the equations simplify to $y_6^t = \tanh\ y_6^{t-1} + 0.62)$ and $y_7^t = \tanh\ 2y_7^{t-1} + 1.68)$; the stable solution is $y_3 \approx -0.8; y_6 \approx 0.9; y_7 \approx 1.0$. Note how all the node output activities have reversed; the previously inhibited nodes 6 and 7 are now excited, while the previously excited left-back motor node is now inhibited. The input threshold has produced an on/off 'switch'. The immediate effect of high visual input to node 7 is to turn off the left-back motor node through the inhibitory connection, thus turning on the left motor, so the robot goes in a straight line.

However, the picture is complicated by the emission of gas from the subnetwork nodes. Both nodes 6 and 7 emit negative gas when highly active, and node 3 emits positive gas in the presence of high negative gas concentrations. Three different scenarios are investigated: where both inputs go high at the same time, and where either input goes high first.

In the model of gas diffusion used, gas concentration builds up according to equations 2 to 4, reaching a maximum concentration $C = C_0 e^{-(d/r)^2}$. The node 6 characteristics ensure negative gas spreads out very quickly over a large area: the concentration of negative gas at node 7 due to node 6 emission quickly affects the transfer function on the very next time-step). The small distance between nodes 6 and 7, and the high value of the radius of gas emission r for node 6, produce a gas concentration that drops K_7 from 2 to -0.25. Now $y_7^t = \tanh\ -0.25y_7^{t-1} - 0.57)$ has a stable negative solution $-0.43)$ even with high positivev

the behaviour. Even with high input, node 6 cannot now produce output sufficient to emit gas so cannot inhibit node 7. Now, the three solutions to the node 7 equation with no input $y_7^t = \tanh\ 2y_7^{t-1} - 0.32))$ mentioned previously come into play. From an initial condition of $y_7 \approx 1.0$, even with no external input, there is a stable solution at $y_7 \approx 0$. the network is now in a highly stable state with node 7 output at near maximum with or without external input, node 6 inhibited due to negative gas emitted by node 7, and the left-back motor inhibited due to node 7 synapse output. The overall effect is to switch the network into a permanent open-loop behaviour where further external input is irrelevant. Due to the inhibition of the left-back motor node, the left motor is on and the robot continues straight-ahead. So under what conditions does node 7 receive bright visual input four or more time-steps before node 6?

8.1.3 Visual input positions, success and failure modes

Figure 13 shows that the visual inputs to nodes 6 and 7 are vertically aligned in the visual field, with 7 directly below 6. Scanning across the square will cause both nodes to receive bright input at roughly the same time, th node 7 will be inhibited by node 6, and the robot will continue rotating. However, scanning across the triangle will cause node 7 to receive bright visual input significantly before node 6, th inhibiting node 6. This in turn will cause the network to switch into the permanent open-loop state, and the robot will continue straight-ahead. Table 1 summarises the behaviour of the robot as determined by the switching subnetwork.

Node 6 input	Node 7 input	Node receiving first bright input	Left-back motor node	Robot motion
dark	dark	-	excited	ccw rotation
bright	dark	6	excited	ccw rotation
bright	bright	6	excited	ccw rotation
bright	bright	same time	excited	ccw rotation
dark	bright	7	inhibited	straight-forward
bright	bright	7	inhibited	straight-forward

Table 1: Summary of 'switch' subnetwork behaviour, showing the robot motion based on the visual

8.2 Open-loop GasNet controller summary

The overall behaviour of the robot controller can be summarised as follows. In the absence of bright visual input, the robot rotates counter-clockwise, with the right motor permanently excited, and the left motor inhibited by the switching subnetwork. This behaviour continues, until the robot scans across a bright object, such that the lower half of the visual field receives bright input significantly before the upper half. This permanently switches off the left-back motor node, exciting the left motor and causing the robot to move straight-forward. Now the effect of the spiking subnetwork is seen; once every eight time-steps the right motor is turned off, thus the robot moves in a slow clockwise arc back towards the triangle, which it has rotated past. So we have explained in full the behaviour seen in the two example evaluations, shown in figure 16.

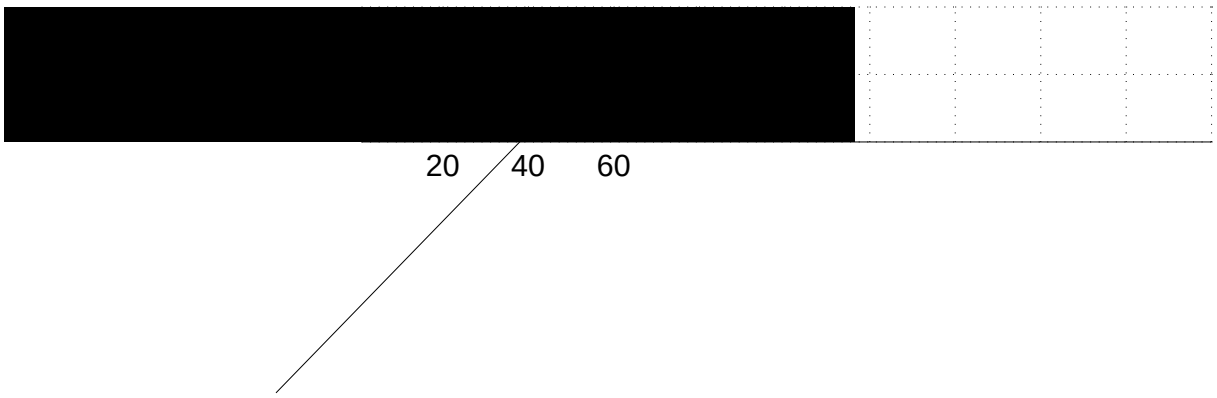
The two subnetworks analysed are crucial to the understanding of the robot controller triangle discrimination, in conjunction with the robot-environment coupling. The primary robot-environment coupling is the permanent switch mechanism; scanning across the square will produce no change in the robot motion beyond a slight slowing of the turn. By contrast, scanning across the triangle will lock the robot into

to

the timing mechanism is sufficiently long, which is the case when scanning across the square, but not when scanning across the triangle. Thus the controller approaches the triangle, but rotates past the square. In the next sections we describe the methods by which the GasNet and NoGas solutions produce such a timing mechanism.

9.1 The GasNet "timer"

With the GasNet class, it is simple to produce a timing mechanism that retains activity for some time after the initial input has been received. A single node receiving visual input, and with the property that gas emission occurs when the node output activity is high, will start emitting gas when bright input is received. The gas concentration built-up during emission will take some time to decay once bright input is no longer received, with this decay time being a function of the genetically specified rate of gas concentration build-up. Remember from equations 2 to 4 section 2.1) that gas concentration decays in a Gaussian fashion with distance from the emitting node, but increases linearly over time during emission, and decreases linearly over time once emission stops. As the time taken for gas concentration to build up to a maximum is specified by the genotype, the mechanism is simple to tune to the characteristics of the environment.



As proposed in section 8.3, we hypothesise that the

	Double speed	Quarter speed

re-evolution, however there is evidence of faster evolution from the median numbers of generations. Thus from our sample of GasNet controllers, we also see evidence of significantly faster re-evolution to modified environments; the GasNets are more tunable.

In the next section, we investigate the hypothesis further, using a more abstract fitness evaluation. We evolve GasNet and NoGas networks for a central pattern generation problem, in which the output of a single node is evaluated against a test pattern. We then use successfully evolved networks to seed the initial populations for a re-evolution environment where fitness is evaluated against a different test pattern.

11 Evolving central pattern generator networks

In this section, we test the GasNet and NoGas classes further in a central pattern generation CPG) experiment. We evolve fully connected GasNet and NoGas networks, with output tested against some required test pattern, then re-evolve successful controllers against different test patterns. We argue that the increased evolutionary speed seen for the GasNet class over the NoGas class on both the original evolution and the re-evolution experiments lends support to the hypothesis that the GasNet class is more tunable to the characteristics of the environment, which in this case correspond to the desired pattern output.

11.1 The central pattern generation network

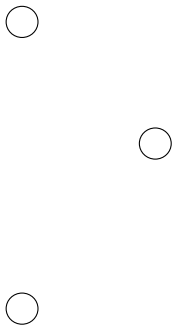
The networks used in the CPG experiment consisted of four fully-connected nodes, including recurrent connections other size networks were also investigated, with comparable results). Connection weights between the four nodes were genetically specified, and were constrained to lie in the range $\{-1, 1\}$. Each node received a genetically specified fixed bias input, and the same tanh input-output transfer function was used as in previous experiments:

$$O_i^t = \tanh\left[K_i^t\left(\sum_{j\in C_i} w_{ji}O_j^{t-1}\right) + b_i\right] \quad (37)$$

where O_i^t is the ith node activity at time-point t, K_i^t the transfer function parameter, fixed for the NoGas networks, but able to vary for the GasNet networks, w_{ji} the connection weight from node j to node i, and b_i the bias input to node i.

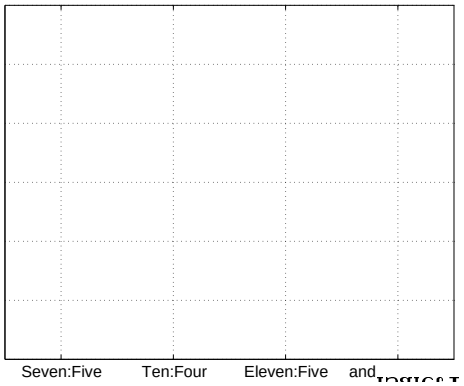
Gas diffusion and modulation occurred exactly as described for the robot visual discrimination problem in section 2. Each node had a set x, y position in the gas diffusion plane, and was able to emit one of two gases, respectively increasing or decreasing the transfer function parameter K_i^t of nearby nodes. However, it should be emphasised that in the CPG networks the electrical architecture, in other words the pattern of synaptic connections between the nodes, was not specified arbitrarily in the gas diffusion plane, but specified directly on the genotype in terms of the weights between nodes. Figure 21 shows the network setup.

The NoGas genotype consisted of 18 integers in the range $[0, 99]$, encoding the 4 node biases b_i, 4 node transfer parameters K_i, and 10 connection weights w_{ji}. The GasNet genotype consisted of the NoGas genotype plus an extra six parameters per node for the gas diffusion parameters the type of gas emitted $< CE >$, the conditions under which gas emission occurs $< TE >$, the radius of gas emission r, the gas build-up parameter s, and the x, y co-ordinates of the node in the 2D gas plane), thus the GasNet genotype consists of 42 integers in the range $[0, 99]$.



Pattern	Statistic	GasNet	NoGas
Seven:Five	■ean σ)	16390 34060)	33530 45910)
	■edian	2500	3260
Ten:Four	■ean σ)	18000 36080)	36000 46230)
	■edian	2120	5010
Eleven:Five	■ean σ)	22410 39100)	43830 48060)
	■edian	3070	5470
Eleven:Seven	■ean σ)	13750 32130)	31610 45080)
	■edian	1460	2980

Table 4: The number of evaluations required to evolve successful GasNet and NoGas networks, for the four experiments where fitness is evaluated over the four different test patterns: **Ten:Four**, **Eleven:Five**, **Eleven:Seven**, and **Seven:Five**.



The different time-scales over which these two mechanisms operate was seen to be crucial to this pattern generation.

In the final section, we conclude with discussion of *temporally adaptive* networks.

13 Discussion: Temporally adaptive networks

One feature common to many of the neural network classes used for generating adaptive behaviour, is the incorporation of *time*. Few evolutionary robotics practitioners rely on feedforward networks consisting of nodes that retain no activity over time, with most using 0Td[adaptiv)1998.7 e)]TJynTd[no)-2000.02 de0.028 no)-200

References

Beer, R. 1990). *Intelligence as Adaptive Behaviour: An Experiment in Computational Neuroethology.* Academic Press, Cambridge, Massachusetts.

Beer, R. 1995). On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behaviour*, 3 4):469–509.

Beer, R., Chiel, H., and Gallagher, J. 1999). Evolution and analysis of model CPGs for walking II. general principles and individual variability. *Journal of Computational Neuroscience*, 7 2):119–147.

Beer, R. D. and Gallagher, J. C. 1992). Evolving dynamical neural networks for adaptive behaviour. *Adaptive Behaviour*, 1:94–110.

Calvatti, A. and Beer, R. 2001). Analysis of a distributed model of leg coordination I. individual coordination mechanisms. *Biological Cybernetics*, 82:197–206.

Chiel, H., Beer, R., and Gallagher, J. 1999). Evolution and analysis of model CPGs for walking I. Dynamical Modules. *Journal of Computational Neuroscience*, 7 2):99–118.

Devaney, R. 1989). *Chaos, Fractals and Dynamics: Computer Experiments in Mathematics.* Addison-Wesley, Redwood, California.

Gallagher, J. C. and Beer, R. D. 1999). Evolution and analysis of dynamical neural networks for agents integrating vision, locomotion, and short-term memory. In Banzhaf, of theGECCO1999

Sandefur, J. 1990). *Discrete Dynamical Systems: Theory and Applications*. Oxford University Press, Oxford, UK.

Smith, T., Husbands, P., Layzell, P., and O'Shea, ■. 2002a). Fitness landscapes and evolvability. *Evolutionary Computation*, 10 1):1–34.

Smith, T., Husbands, P., and O'Shea, ■. 2001a). Neutral networks and evolvability with complex genotype-phenotype mapping. In Kelemen, J. and Sosík, P., editors, *Advances in Artificial Life, 6th European Conference (ECAL'2001)*, pages 272–281. Springer, Berlin.

Smith, T., Husbands, P., and O'Shea, ■. 2001b). Not measuring evolvability: Initial exploration of an evolutionary robotics search space. In *Proceedings of the 2001 Congress on Evolutionary Computation (CEC'2001)*, pages 9–16. IEEE Press, Piscataway, New Jersey.

Smith, T., Husbands, P., and O'Shea, ■. 2002b). Local evolvability of statistically neutral GasNet robot controllers. *Biosystems*. In press.

Smith, T. ■. C. 2002). *The Evolvability of Artificial Neural Networks for Robot Control*. PhD thesis, School of Biological Sciences, University of Sussex, UK. Submitted.

Volterra, V. 1926). Fluctuations in the abundance of a species considered mathematically. *Nature*, 118:558–560. Reprinted in Real, L. and Brown, J.H. 1991). *Foundations of Ecology: Classic Papers With Commentaries*. University of Chicago Press, Chicago, Illinois.