

Evolving Robot Behaviours with Diffusing Gas Networks

Phil Husbands

School of Cognitive and Computing Sciences
and
Centre for Computational Neuroscience and Robotics
University of Sussex, Brighton, UK
philh@cogs.susx.ac.uk

Abstract. This paper introduces a new type of artificial nervous system and shows that it is possible to use evolutionary computing techniques to find robot controllers based on them. The controllers are built from networks inspired by the modulatory effects of freely diffusing gases, especially nitric oxide, in real neuronal networks. Using Jakobi's radical minimal simulations, successful behaviours have been consistently evolved in far fewer evaluations than were needed when using more conventional connectionist style networks. Indeed the reduction is by a factor of roughly one order of magnitude.

1 Introduction

In evolutionary robotics, the predominant class of systems for generating behaviours is that of artificial neural networks (ANNs). These networks can be envisaged as simple nodes connected together by directional wires along which signals flow. The nodes perform an input output mapping that is usually some sort of sigmoid function [7]. Occasionally a simple differential equation is used instead, providing the possibility of richer dynamics [1]. Some have used feed-forward architectures, others have explored more free form arbitrarily recurrent networks. The original inspiration for all these styles of network is the neuroscience of the 1940s and 1950s. They abstract something of the electrical properties and behaviours of real neuronal networks. However, two obvious questions raise their heads. Do we have any reason to believe that these kinds of systems are capable of generating adaptive behaviours in autonomous robots of a kind that is much more sophisticated than we can manage today? Even if they are, will it be possible to find the networks in question, through evolutionary search or some other technique?

As has been pointed out by various people (e.g. [3]), advances in neuroscience have made it clear that the propagation of action potentials, and the changing of synaptic connection strengths, is only a very small part of the story of the brain (e.g [16]). This in turn means that connectionist style networks, and even recurrent dynamical ones, are generally very different kinds of systems from those that generate sophisticated adaptive behaviours in animals. Indeed, it may be

better to think of natural nervous systems as some kind of ever changing chemical machine [16]. Although our picture of biological neural networks changes every few years, advances over the past decade or so can provide a rich source of inspiration in devising alternative styles of artificial network. Among others, Brooks and colleagues went some way down that path by using ideas gleaned from some of the properties of the lobster hormonal system [2].

As far as the author of this paper is concerned, current understandings of nervous systems seem to suggest that a useful abstract way to think of them is in terms of several interacting classes of dynamical processes, each with distinct characteristics (e.g. electrical, short-range chemical, long-range chemical)¹. These processes can have very different spatial and temporal properties and may be heavily intertwined, modulating each other in complex ways. For instance, Hebbian style changes in synaptic efficacy can be thought of as one of these processes that can be turned on and off and localised by other processes, such as the change in local concentration of a particular chemical. At the detailed level, there will be complex cascades of chemical reactions involved, but it is suggested that it may be possible to pull out more abstract systems that describe the gross dynamical principles underlying the behaviour of nervous systems. It is hoped that versions of these abstract systems, based on interacting dynamical processes, can be developed that are sufficient to underpin adaptive machines more advanced than those available today. It is very difficult to see how such systems can be investigated and developed except through some form of artificial evolution. It is likely that, at least in principle, for any of these systems a functionally equivalent recurrent dynamical network exists. However, my bet is that in general it will be horrendously convoluted and almost impossible to find. It is hoped that the kinds of systems just sketched will be more amenable to evolution than standard ANNs and will be far easier to scale up. We shall see.

This paper describes some very initial investigations into the kind of systems outlined in the previous paragraph. The main inspiration has been the recent discovery that freely diffusing nitric oxide (NO) is synthesised in, and emitted by, nerve cells in many parts of nervous systems. As it diffuses, it acts as a neurotransmitter and is implicated in the modulation of various properties of nerve cells and synapses [4].

After describing the networks inspired by this biological phenomenon, so called GasNets, a number of successful evolutionary robotics experiments using this new style of network are discussed. The paper closes with some conclusions, including the observation that, using GasNets, successful behaviours have been consistently evolved in far fewer evaluations than were needed when using more conventional connectionist style networks.

¹ ANNs incorporating Hebbian connection strength changes can be thought of as a simple example of this kind of system. I am proposing exploring far richer systems.

2 GasNets

In this section a new form of artificial nervous system inspired by two aspects of biological neuronal networks is described in detail. These two aspects are: the heterogeneity of intrinsic nerve cell properties found in much of the nervous system of invertebrates; and the modulation of these properties by diffusing NO, emitted from within the nervous system itself. Many of the known and postulated effects of NO are not included, but will be the inspiration for future developments of this style of network.

These are abstract systems founded on some of the general principles, rather than details, of biological networks. They are not models of real nervous systems. The abstractions chosen for this initial investigation into this style of network are: heterogeneity in terms of transfer functions at the nodes in the network, modulation of intrinsic properties in terms of changing these functions as the network runs. One last general point – these networks should not be thought of as computational devices. They could be thought of as mathematical systems, but I prefer to regard them much more as simulations of physical devices.

2.1 The 4 gas GasNet

The networks used in the experiments described later consist of units connected together by excitatory links, with a weight of +1, and inhibitory links, with a weight of -1. The output, O_j , of a node j is a function of the normalised sum of its inputs, S_j , as described by equation 1. In addition to this underlying network in which positive and negative ‘signals’ flow between units, an abstract process loosely analogous to the diffusion of gaseous modulators is at play. Some units can emit ‘gases’ which diffuse and are capable of modulating the behaviour of other units by changing their transfer functions in ways described in detail later. This form of modulation allows a kind of plasticity in the network in which the intrinsic properties of units are changing as the network operates. The networks function in a 2D plane; their geometric layout is a crucial element in the way in which the ‘gases’ diffuse and affect the properties of network nodes. This aspect of the networks is described in more detail later.

$$O_j = f(\overline{S}_j) \quad (1)$$

Where,

$$\overline{S}_j = \frac{(\sum_{p \in P_j} O_p - \sum_{n \in N_j} O_n + \sum_{k \in SEN_j} I_k)}{(np_j + nn_j + ns_j)} + R \quad (2)$$

In equation 2, P_j is the set of network elements with excitatory connections to element j . Likewise, N_j is the set of elements with inhibitory link to j , and SEN_j is the set of sensors connected to j . R is the default activation of a node ($= 0.05$). np_j , nn_j and ns_j are, respectively, the number of positive, negative and sensor connections to element j , i.e.:

$$np_j = |P_j| \quad (3)$$

$$nn_j = |N_j| \quad (4)$$

$$ns_j = |SEN_j| \quad (5)$$

Normalizing by dividing by the number of inputs keeps the summed input in the range [-1,1]. The transfer function, f , is defined in equation 6, its output range is [-1,1] given the restriction on the input range.

$$f(x) = \begin{cases} 0, & \text{if } x < 0 \text{ and } (a < 0 \text{ or } b < 0) \\ (x^a + x^b)/2, & \text{otherwise} \end{cases} \quad (6)$$

Where,

$$a, b \in PP = \{0.1, 0.2, 0.3 \dots 0.8, 1, 2, 3 \dots 9, 10\} \quad (7)$$

Overlays of plots of this function for many of the possible combinations of a and b are shown in figure 1. As can be seen, a wide range of output responses to a given input are possible, depending on the values of the parameters a and b . As will be seen later, default values of a and b for each node are set genetically, but are changed by diffusing gases as the network runs. It is genetically determined whether or not a node will emit one of four gases, and under what circumstances emission will occur (either when the ‘electrical’ activation of the node exceeds a threshold, or the concentration of one of the gases, genetically determined, in the vicinity of the node exceeds a threshold).

Gas Diffusion in the Networks A very abstract model of gas diffusion is used. For an emitting node, the concentration of gas at distance d from the node is given by equation 8. Here r is the genetically determined radius of influence of the node, so that concentration falls to zero for $d > r$. $TC(t)$ is a linear function that models build up and decay of concentration after the node has started/stopped emitting. The slope of this function is individually genetically determined for each emitting node, C_0 is a global constant.

$$C(d, t) = \begin{cases} C_0 \times e^{\frac{-2d}{r}} \times TC(t), & d < r \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

$$TC(t) = \begin{cases} H\left(\frac{(t-t_e)}{k}\right), & \text{emitting} \\ H\left(H\left(\frac{(t_s-t_e)}{k}\right) - H\left(\frac{(t-t_s)}{k}\right)\right), & \text{not emitting} \end{cases} \quad (9)$$

Where, t_e is the time at which emission was last turned on, t_s is the time at which emission was last turned off, k is genetically determined for each node and:

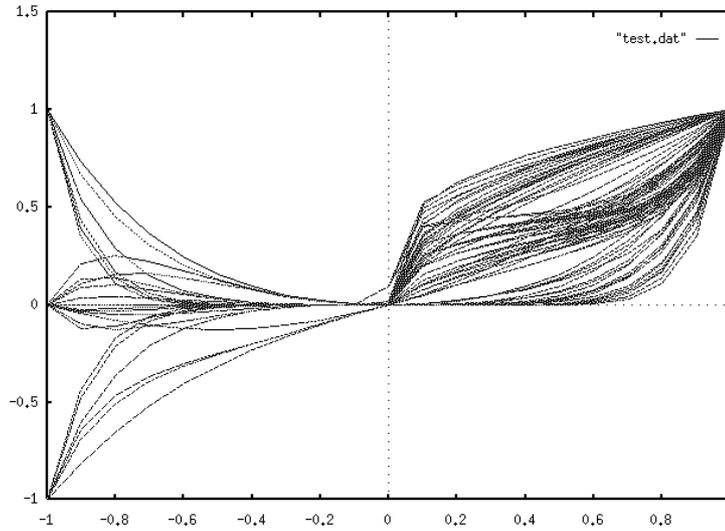


Fig. 1. The transfer function used for the 4 gas GasNets. Input on the X axis, output on the Y. Overlays of the function (see equation 6) for many combinations of a and b are shown.

$$H(x) = \begin{cases} x, & x < 1 \\ 0, & x \leq 0 \\ 1, & \text{otherwise.} \end{cases} \quad (10)$$

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 is linearly increased or decreased depending on whether the node is emitting or not. Note $TC(t)$ saturates at a maximum of 1 and a minimum of 0.

Modulation by the Gases The values of a and b in the network unit transfer function (see equation 6) are changed (or *modulated*) by the presence of gases at the site of a unit. This modulation is described by equations 12–17 and happens continually as the network runs. This provides a form of plasticity very different from that found in more traditional artificial neural networks. At every time step the value of a for node $_i$, a^i , is updated according to equation 11.

$$a^i = PP[a_{index}^i] \quad (11)$$

Where,

$$a_{index}^i = S(N, a_{ni}^i) \quad (12)$$

Here a_{index}^i is node $_i$'s index into the set PP shown in equation 7. N is the number of elements in PP . At each time step a_{ni}^i is updated according to equation 13. The linear (thresholded) function S is described by equation 14.

$$a_{ni}^i = a_{def.index}^i + \frac{C_1}{C_0 \times K} \times (N - a_{def.index}^i) - \frac{C_2}{C_0 \times K} \times (a_{def.index}^i) \quad (13)$$

Where $a_{def.index}^i$ is the genetically set default value for a_{index}^i , C_1 is the concentration of gas 1 at the site of node $_i$, C_2 is the concentration of gas 2 at the site of node $_i$ and C_0 and K are global constants. So, a_{index}^i increases in direct proportion to the concentration of gas 1, and decreases linearly with respect to the concentration of gas 2. In this way the value of a for node $_i$ is changed by the presence of gases 1 and 2 at the node's site.

$$S(N, x) = \begin{cases} x, & 0 \leq x \leq N \\ 0, & x < 0 \\ N, & x > N \end{cases} \quad (14)$$

Similarly the value of b at node $_i$, b^i , is changed by the presence of gases 3 and 4. This is described by equations 15- 17.

$$b^i = PP[b_{index}^i] \quad (15)$$

$$b_{index}^i = S(N, b_{ni}^i) \quad (16)$$

$$b_{ni}^i = b_{def.index}^i + \frac{C_3}{C_0 \times K} \times (N - b_{def.index}^i) - \frac{C_4}{C_0 \times K} \times (b_{def.index}^i) \quad (17)$$

2.2 The 2 gas GasNet

This paper will concentrate on early results found using the 4 gas style GasNet. However, we are currently investigating slightly simpler versions using 1 or 2 gases. For instance, in one of our 2 gas networks the transfer function at each node is of the form shown in equation 18². Again the weights are restricted to be either +1 or -1. The value of b is genetically set at each node as is the default value of k . The two gases raise and lower the value of k in a similar manner to the way a and b are changed in the 4 gas model.

$$O_j = \tanh([k \times \sum_i O_i w_{ij}] + b) \quad (18)$$

² Thanks to Andy Philippides for suggesting this form.

3 Network encoding

The basic genotype used to encode 4 gas GasNets consisted of an array of real numbers. Each node in the network had 15 real values associated with it. That is:

$\langle \text{genotype} \rangle :: (\langle \text{gene} \rangle)^*$

$\langle \text{gene} \rangle :: \langle x \rangle \langle y \rangle \langle R_p \rangle \langle \Theta_{1p} \rangle \langle \Theta_{2p} \rangle \langle R_n \rangle \langle \Theta_{1n} \rangle \langle \Theta_{2n} \rangle$
 $\langle \text{rec} \rangle \langle TE \rangle \langle CE \rangle \langle k \rangle \langle R_e \rangle \langle a_{def.ind} \rangle \langle b_{def.ind} \rangle$

This encoding was used to generate networks conceptualized to exist on a 2D Euclidean plane. x and y give the position of a network node on the plane. The next six numbers define two segments of circles, centred on the node. These segments are used to determine the connectivity of the network. R_p gives the radius of the ‘positive’ segment, Θ_{1p} its angular extent and Θ_{2p} its orientation. R_n , Θ_{1n} and Θ_{2n} define a ‘negative’ segment. The radii range from zero to half the plane dimension, the angles range from zero to 2π . The segments are illustrated in figure 2. Any node that falls within a positive segment has an excitatory (+1) link made to it from the segment’s parent node. Any node that falls within a negative segment has an inhibitory (-1) link made to it from the segment’s parent node. Developing networks on a 2D plane has a number of advantages as discussed in [11]. The idea of using a plane has been previously explored by at least Husbands [8], Nolfi [15] and Jakobi [10]. Of course, in this case the geometry of the network is crucial to the workings of the gas diffusion, so the networks *have* to exist on a plane. This particular encoding is directly inspired by Jakobi’s [11].

The rest of a gene is interpreted as follows. The value of $rec \in \{0, 1, 2\}$ determines whether the node has no recurrent connection to itself, an excitatory recurrent connection or an inhibitory recurrent connection, respectively. $TE \in \{-1, 0, 1, 2, 3, 4\}$ provides the circumstances under which the node will emit a gas. These are: not at all, if its ‘electrical’ activity exceeds a threshold, or if the concentration of the referenced gas (1,2,3 or 4) at the node site exceeds a threshold. $CE \in \{1, 2, 3, 4\}$ gives the gas the node can emit. k is a real in the range $[1,15]$ and is used to control the rate of gas build up/decay as described earlier by equation 9. R_e is the maximum radius of gas emission, this ranges from 2 to half the plane dimension. $a_{def.ind}$ and $b_{def.ind}$ are the default values for the a and b indices as used in equation 11 to determine the default values of a and b for the node.

This basic encoding can be used to search for network topologies and geometries with a fixed number of nodes, or a dynamic length version can be used to evolve networks in a more open-ended way [6]. Of course, the use of the segments to determine the connectivity means the number of connections, and hence the basic architecture, is never fixed. To say nothing of the additional dynamic properties introduced by the diffusing gases.

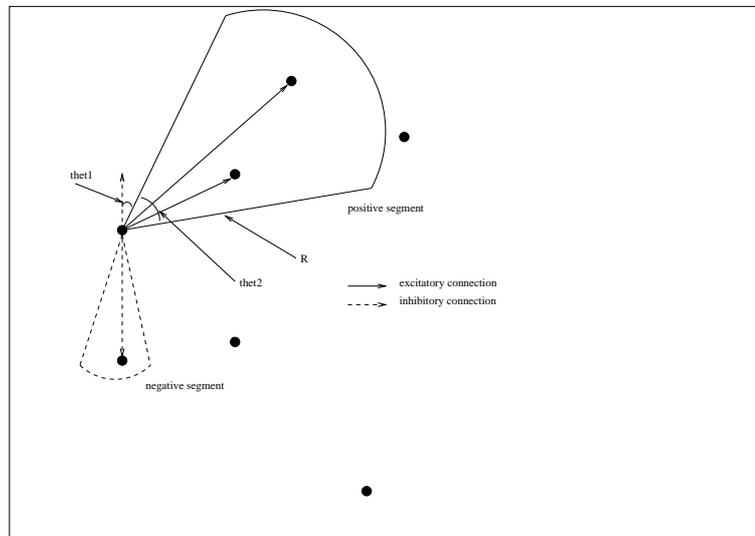


Fig. 2. Positive and negative segments define the connectivity of the network. The network develops and functions on a 2D plane.

4 Experiments

In order to start exploring the properties of GasNets, particularly their evolvability and suitability as control systems, it was decided to rerun some of Jakobi's recent experiments [10, 11], substituting the new type of network for his more conventional connectionist ones. It would then be possible to compare the kinds of solutions found, how quickly they evolved, and so on. A major reason for choosing Jakobi's work was the fact that he had evolved the behaviours in question using his innovative and radical minimal simulations [11, 10, 12]. Control systems evolved using these ultra-lean ultra-fast simulations transfer perfectly to reality. Because of their speed, many evolutionary runs can be performed allowing the kind of exploration desired in this case. Jakobi's original minimal simulation code was used for the experiments described here.

4.1 Khepera with state: the T-maze

The first behaviour attempted was that required to perform Jakobi's T-maze task. This is illustrated in figure 3. A Khepera robot [13], making use of 6 IR proximity sensors and 2 ambient light sensors, moves along a corridor. A light shines from either the left or the right, chosen randomly. Once the Khepera reaches the T-junction it should turn in the direction the light shone from.

For this experiment, following Jakobi [10], a fixed number of network nodes was used. Also, bilateral symmetry was imposed on the control network. 14 nodes

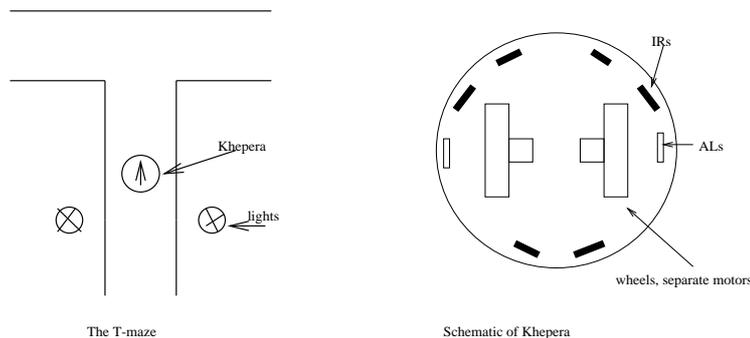


Fig. 3. Jakobi's T-maze task. At the junction the robot must turn in whichever direction the light was shining from. Schematic of Khepera, showing sensor positions, at the right of the figure.

were used. One for each of the sensors, 2 for each motor (motor output was calculated as forward-motor-node output - backward-motor-node output), and 2 others. Bilateral symmetry was achieved by encoding for 7 nodes. Their (x,y) coordinates were constrained to lie in the left half of the network plane. Each of the nodes was reflected in the line $x=(\text{half plane width})$, creating 7 new nodes. The positive and negative segments of the original nodes were also reflected in the half way line. All other genetically encoded properties were inherited intact. On developing the connectivity of the network, as determined by the segments, a symmetrical network is formed. The left hand side is connected to the Khepera's left hand sensors and motor, the right hand side is connected to the Khepera's right hand sensors and motor. One predetermined node for each sensor and motor.

A distributed, or diffusion, GA of the kind described in [9] was used. It employed a population of 225 spread out over a 15x15 grid with overlapping Gaussian local neighbourhoods, in which local selection rules operated. Standard one point crossover was used with a probability of 0.9. Mutation operated as follows. With a rate of one mutation per genotype, any value to be mutated was changed by a random real in the range $\pm 10\%$ of its total range 80% of the time and by a random value in the range $\pm 40\%$ the remaining 20% of the time.

Each evaluation consisted of 12 trials. The starting orientation of the robot, the corridor width and length and the position of the lights were all randomly varied between trials. Other aspects of the simulation were randomly varied within and across trials in keeping with the minimal simulation methodology [10]. The fitness function (taken from [10]) is shown in equation 19.

$$f = d_1 + d_2 + \text{bonus} \quad (19)$$

Where d_1 is the distance moved down the first corridor, d_2 is the distance moved down the second corridor and *bonus* is 100 if the robot turns in the correct

direction at the junction, 0 if it doesn't. A trial is aborted if the robot touches a wall.

Jakobi had originally used binary networks in which the connectivity, weights on the connections and node thresholds were genetically encoded [10]. He had been able consistently evolve robust successful controllers in 1,000 generations using a similar GA to that described above, with a population of 100. This may seem rather a large number of evaluations, but it must be appreciated that evolving in a minimal simulation is in many senses harder than in reality, because of the extreme use of noise. Hence it would be expected that a greater number of evaluations would be required than in reality, although this is heavily offset by the speed at which the simulations run. The resulting controllers are extremely robust, capable of successful behaviour in a wide range of conditions.

To date more than 15 runs have been completed using the GasNets. In each run success was achieved in less than 100 psuedo generations³, in several cases in less than 50. *In other words, the number of evaluations needed was decreased by a factor of one order of magnitude.*

Interestingly, some of the successful controllers made use of the gases and some didn't. All were very heterogeneous as far as the transfer functions of the nodes were concerned. In the runs where the final successful controllers didn't use gases, they *had* been used extensively during the evolution of the final population. When the gases were removed all together, it was not possible to evolve successful controllers within 500 generations. Since the gases are the agents of a form of plasticity in the networks, these observations suggest that the Baldwin effect [14] is at play, albeit a rather different form than has been observed before. This will need further investigation.

Many different successful behaviour generating mechanisms were observed. The evolved controller shown in figure 4 represents a class of mechanisms, making use of the gases, of which several examples were seen.

Briefly, the network shown in figure 4 works as follows. The motor transfer functions are such that the default behaviour, in the absence of sensory input, is straight line forward motion. The left most and right most IR sensors (see figure 3) are connected directly to the right and left motors, respectively. This arrangement, coupled with the particular motor transfer functions, provides a basic Braitenburg style obstacle avoidance behaviour. This is what enables the robot to travel along the corridors without crashing. The brief chemical activity initiated at nodes 1 and 8 by the IR stimulus, and the ensuing transfer function modulation, reinforces the Braitenburg behaviour. Nodes 3 and 10 are connected to the left and right ambient light sensors respectively. When either is stimulated by the light shining from the side of the first corridor it feeds into the nodes 0 and 7 feedback loop. The transfer functions are such that only a continuous stimuli ramps up the activity levels and initiates the release of gas 2 from whichever of nodes 0 and 7 is being directly fed the active AL signal. The evolved time constants on the gas build up and decay processes mean that only the continuous AL stimuli as the robot moves through the light zone (as opposed

³ One psuedo generation occurs every N offspring events, where N = population size.

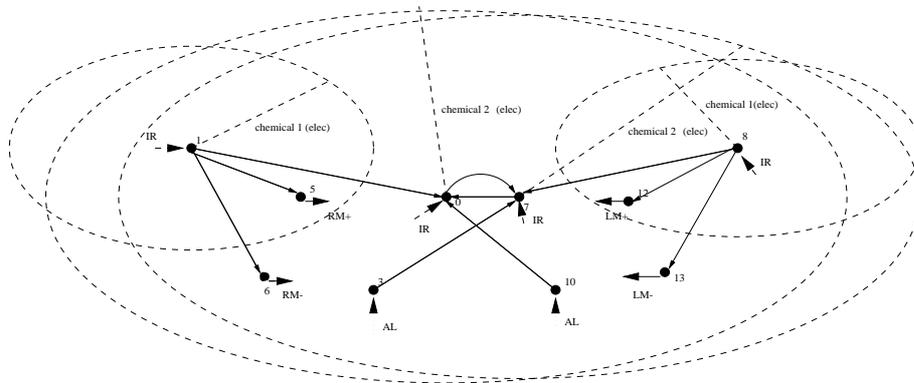


Fig. 4. An evolved successful control network for the T-maze task. Network connectivity plus the radius of influence of emitting nodes is shown. These are depicted as ellipses as the X scale is twice the Y scale. See text for details.

to the very brief IR stimuli from nodes 1 and 8 ensured by the successful obstacle avoidance subnetwork) triggers this part of the circuit. If the left AL sensor is active it triggers release of gas 2 from node 7. Because node 7 is closer to the left motor nodes (12 and 13) than the right motor nodes (5 and 6) it has a stronger modulatory effect on their transfer functions. The overall effect is to reduce left motor output with respect to right. By the time the robot comes to the junction it inevitably turns to the left. In a similar way, if the right AL sensor had been on, the robot would have turned to the right. The underlying Braitenburgh obstacle avoider still works under this modulation. The genetically determined slow decay of gas 2 is important to the successful operation of this behaviour. Obviously the geometric layout of the network is crucial. Unfortunately space does not allow a more detailed analysis here. Figure 5 shows the transfer functions for the network. Note that they cover a very wide range of response types. This was found in every successful controller examined, including those that did not use gases. This, along with the observed role that the differences in transfer functions plays in successful controllers, suggests transfer function heterogeneity can be a very useful thing in sensorimotor systems. All of this is moving some way from the kind of positions held in mainstream cognitive science as to what a behaviour generating mechanism should look like. It adds more grist to the mills of left-field philosophers of cognitive science such as Wheeler [17].

4.2 Khepera with more state, the double T-maze

The second behaviour attempted was that required to perform the double T-maze task. This is illustrated in figure 6.

This time there is a sequence of 2 lights. The robot must turn at a pair of junctions in accordance with the directions from which the two lights are shining

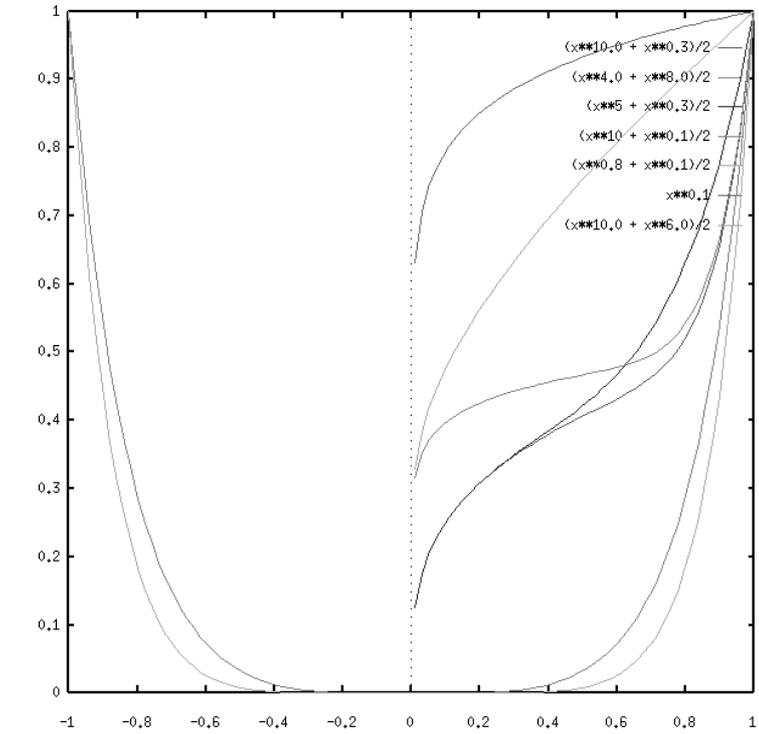


Fig. 5. The transfer functions at the nodes of the controller shown in figure 4

(e.g. right,left). The experimental setup was essentially the same as for the single T-maze, with a slightly modified fitness function.

$$f = d_1 + d_2 + d_3 + \textit{bonus}_1 + \textit{bonus}_2 \quad (20)$$

Jakobi was not able to evolve successful controllers for this task using his binary networks. However, with the GasNets, for 5 runs completed to date, success was achieved in 3 runs by 350 generations; in 1 run success was achieved by 700 generations; and in 1 run success was not achieved by 1000 generations. At the time of writing none of these runs has yet been analysed.

4.3 Visually guided behaviours

In this section a very brief mention of some on-going joint work with Tom Smith will be made. We are in the process of investigating the use of GasNets to control a visually guided robot. The Sussex gantry robot [5] is best thought of as a two wheeled device with a fixed forward pointing video camera. We have been concurrently evolving network controllers and the robot's visual morphology (the

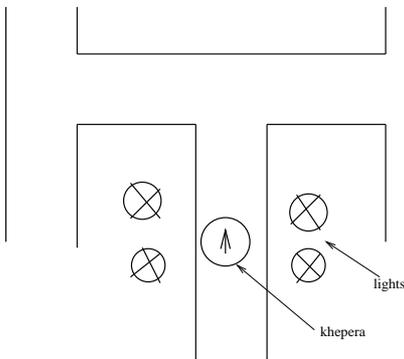


Fig. 6. The double T-maze.

genetically specified number and positions of pixels from the camera’s image that provide the only sensory input to the robot). We are using another of Jakobi’s minimal simulations [10] to evolve a target discriminating behaviour (move to a triangle while ignoring a rectangle) under very noisy lighting conditions.

Jakobi reported needing 6,000 generations of his GA to reliably evolve robust controllers. We have found successful controllers in less than 800 generations using the 4 gas GasNets and the same GA and encoding described earlier. This time the GA was allowed to find the appropriate number of network nodes by using gene insertion and deletion operators that allowed the genotype length to vary. A successful controller (complete with evolved visual morphology) is shown in figure 7. It is structurally much less complex than Jakobi’s evolved controllers. Indeed it seems remarkably simple considering the very noisy nature of the lighting and the relative complexity of the task. This work will be reported on in detail elsewhere and is mentioned here to demonstrate that GasNets are not merely restricted to good performance on one type of task.

5 Conclusions and Discussion

This paper has introduced a new type of artificial nervous system and has shown that it is possible to use evolutionary computing techniques to find robot controllers based on these systems. This has been demonstrated for a range of behaviours involving two different robots. These successes were achieved using very similar setups to those employed by Jakobi when he previously evolved controllers for the same tasks. The main (although admittedly not only) difference between our experiments and his were the style of network used. He used fairly standard connectionist type networks while we used GasNets, as described earlier in this paper. We found that we were able to consistently evolve successful controllers in far fewer evaluations than him. Indeed the reduction was by a factor of roughly an order of magnitude.

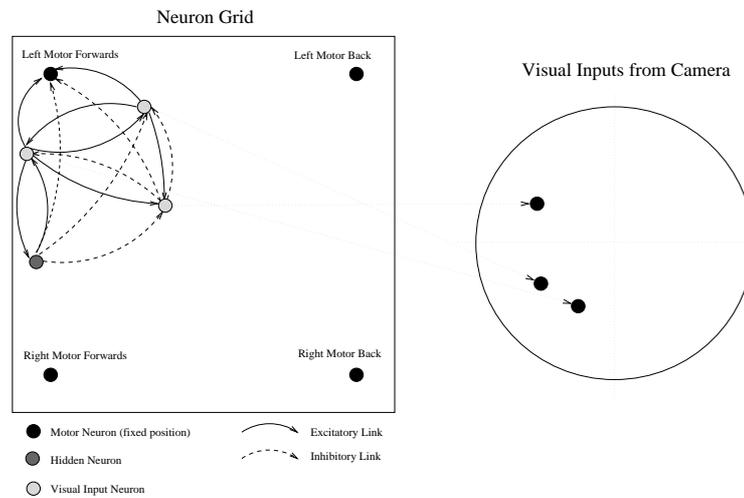


Fig. 7. Successful evolved GasNet for a visually guided behaviour. See text for details.

This suggests that the space of possible behaviours open to being generated by GasNets is somehow ‘thicker’ than for Jakobi’s more conventional networks. It is easier to find successful controllers in this space; it is rich with useful network dynamics and mechanisms.

This in turn suggests that networks involving a number of interacting dynamical processes with distinct properties and characteristics, may well be a very powerful building block for evolutionary robotics. The inspiration for these networks were, of course, real nervous systems. Recent advances in neuroscience mean that there is a rich seam of inspiration to mine. In abstracting principles from biological systems in order to build adaptive machines, it is hoped that a mutually beneficial interface between neuroscience and AI will flourish and prove to be profoundly important in both fields. The initial studies reported here are the very beginning of a new line of research at Sussex. There is much more of this story to come.

Acknowledgements

Many thanks to Nick Jakobi for his minimal simulation code. Without his innovative techniques this study would not have been possible. This exploratory study would not have been possible without it. Many thanks to Mick O’Shea (Co-Director with me of the Sussex CCNR) and other colleagues from the School of Biological Sciences for introducing me to some of the more exotic avenues of contemporary neuroscience, especially the role of NO. Thanks to Tom Smith for figure 7 and for collaboration on the studies very briefly mentioned in section 4.3. Most of this work was done while I was at UniVap, Sao Jose dos Campos, SP, Brazil. I gratefully acknowledge the sponsorship of the British Council

(SPA/126/881) and of FAPESP (proj. 96/7200-8) and pay tribute to Pedro Paulo Balbi de Oliveira's unstinting hospitality.

References

1. R.D. Beer and J.C. Gallagher. Evolving dynamic neural networks for adaptive behavior. *Adaptive Behavior*, 1(1):91–122, 1992.
2. R. A. Brooks. Challenges for complete creature architectures. In J.-A. Meyer and S.W. Wilson, editors, *From Animals to Animats: Proceedings of The First International Conference on Simulation of Adaptive Behavior*, pages 434–433. MIT Press/Bradford Books, Cambridge, MA, 1991.
3. R. A. Brooks. Coherent behavior from many adaptive processes. In D. Cliff, P. Husbands, J.-A. Meyer, and S.W. Wilson, editors, *From Animals to Animats 3: Proceedings of The Third International Conference on Simulation of Adaptive Behavior*, pages 22–29. MIT Press/Bradford Books, Cambridge, MA, 1994.
4. J. Garthwaite. Glutamate, nitric oxide and cell-cell signalling in the nervous system. *Trends in Neuroscience*, 14:60–67, 1991.
5. I. Harvey, P. Husbands, and D. Cliff. Seeing the light: Artificial evolution, real vision. In D. Cliff, P. Husbands, J.-A. Meyer, and S. Wilson, editors, *From Animals to Animats 3, Proc. of 3rd Intl. Conf. on Simulation of Adaptive Behavior, SAB'94*, pages 392–401. MIT Press/Bradford Books, 1994.
6. Inman Harvey. Evolutionary robotics and SAGA: the case for hill crawling and tournament selection. In C. Langton, editor, *Artificial Life III*, pages 299–326. Santa Fe Institute Studies in the Sciences of Complexity, Proceedings Vol. XVI, Addison-Wesley, Redwood City CA, 1994.
7. G.E. Hinton, J.L. McClelland, and D.E. Rumelhart. Distributed representations. In D.E. Rumelhart, J.L. McClelland, and the PDP Research Group, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, pages 77–109. MIT Press, 1986.
8. P. Husbands, I. Harvey, D. Cliff, and G. Miller. The use of genetic algorithms for the development of sensorimotor control systems. In P. Gaussier and J-D. Nicoud, editors, *Proceedings of From Perception to Action Conference*, pages 110–121. IEEE Computer Society Press, 1994.
9. Philip Husbands. Distributed coevolutionary genetic algorithms for multi-criteria and multi-constraint optimisation. In T. Fogarty, editor, *Evolutionary Computing, AISB Workshop Selected Papers*, pages 150–165. Springer-Verlag, Lecture Notes in Computer Science Vol. 865, 1994.
10. N. Jakobi. Evolutionary robotics and the radical envelope of noise hypothesis. *Adaptive Behavior*, 6(2):325–368, 1998.
11. N. Jakobi. *Minimal Simulations for Evolutionary Robotics*. PhD thesis, University of Sussex, 1998.
12. N. Jakobi. Running across the reality gap: Octopod locomotion evolved in a minimal simulation. In P. Husbands and J.-A. Meyer, editors, *EvoRobot98*, page (this vol.). Springer-Verlag, 1998.
13. K-Team. Khepera users manual. EPFL, Lausanne, June 1993.
14. G. Mayley. Landscapes, learning costs and genetic assimilation. *Evolutionary Computation*, 4(3), 1996.

15. S. Nolfi D. Parisi. Evolving artificial neural networks that develop in time. In F. Moran, A. Moreno, and J.J. Merelo, editors, *Advances in Artificial Life: Proceedings of the third European conference on Artificial Life.*, Berlin, 1995. Springer-Verlag.
16. D. Purves and George Augustine. *Neuroscience*. Sinauer, 1997.
17. M. Wheeler. From robots to rothko: The bringing forth of worlds. In M. Boden, editor, *The Philosophy of Artificial Life*, pages 209–236. OUP, Oxford, 1996.