GasNets and CTRNNs – A Comparison in Terms of Evolvability

Sven Magg[1,2] Andrew Philippides[3]

[1]Department of Informatics, [3]Centre for Computational Neuroscience and Robotics
University of Sussex
S.Magg@herts.ac.uk

Abstract. In the last few years a lot of work has been done to discover why GasNets outperform other network types in terms of evolvability. In this work GasNets are again compared to CTRNNs on a shape discrimination task. This task is used as to solve it, or gain an advantage, a controller does not need timers or pattern generators. We show that GasNets are outperformed by CTRNNs in terms of evolvability on this task and possible reasons for the disadvantages of GasNets are investigated. It is shown that, on a simple task where there is no necessity for a timer or pattern generator, there may be other issues which are better tackled by CTRNNs.

1 Introduction

After GasNets, artificial neural networks inspired by gaseous signalling in biological neural systems, were introduced 1998 [4], they were used for evolution of controllers in many different tasks - from pattern generator tasks [12] to quadrupedal walking [5]. The findings in these experiments were that GasNets evolved faster than the same controller type without gas (e.g. [12] or [4]). Other studies also compared GasNet controllers to other controller types like continuous time recurrent neural networks (CTRNNs) or plastic neural networks (PNNs) [5][6]. In terms of evolvability (measured as the length of evolutionary runs till a successful controller was evolved or the robustness of evolved controllers), GasNets generally outperformed other network types or the GasNets without gas. After this higher performance in evolvability became evident, theoretical approaches were made to find the reason for this advantage. However, differences in fitness landscape properties between the GasNet and No-Gas classes which could explain the advantage for example, could not be found [12]. Other investigations focused on other properties of the GasNets, such as the coupling between the electrical and chemical signalling systems [7] and functional neutrality in evolution [14].

While some progress was made, none of these approaches found a definitive explanation of the improved evolvability of GasNets compared to other networks. This and other work did however lead to the current hypothesis that there are 3 possible reasons for improved evolvability [9]:

[2] Currently in the Adaptive Systems Research Group, University of Hertfordshire, Hatfield, UK

– modulatory effects
– different and separate temporal time scales of gas actions
– Flexible coupling of two different and interacting signalling systems through spatial embeddedness

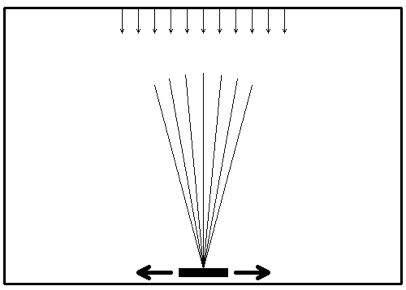
In many of the tasks where GasNet controllers were analyzed, it turned out that the network had at least one sub network which produced a cyclic pattern or timing signal [11]. These timers or pattern generators (PG) are easy to realize in GasNets and it was suggested that this ability to tune pattern generators to given environments can be an advantage of GasNets [12 chapter 7.3.4]. Moreover, in almost all tasks used, the ability to tune pattern generators is helpful. Walking is clearly a cyclic process where timers/PGs can help. Also in the triangle-square discrimination task [4] the controller made use of pattern generators within the network [11]. But what happens if the task doesn't need a PG? If there are solutions which can be found without timers/PGs, are GasNets still better to evolve?

The aim of this work is to answer these questions. To do this, a shape discrimination task is introduced, which was previously used with CTRNNs [2]. Timer or pattern generator sub networks should not lead to an evolvability advantage in this task because no cyclic behaviour is needed. Different GasNet controllers are compared to different CTRNN controllers in terms of evolvability which is judged by the fitness of evolved solutions and the length of time to evolve them (if the word performance is used in this work, then always in terms of evolvability).

The hypothesis to be proven is that GasNets are good to evolve pattern generators and gain advantages in tasks where they are useful, but perform worse on tasks where these abilities are almost useless compared to different types of neural networks.

2 Experimental Setup

2.1 Shape Discrimination Task



The shape discrimination task as used in this work was introduced by Randall Beer [2]. The robot has to discriminate between different falling objects and has to catch or avoid the objects, depending on the shape. The robot is represented by a line with a given length (30 Pixel) and is acting in a 2-dimensional, closed room (The room is 400 pixels wide and 275 pixels high, starting position is always (0,200)).

The robot has seven sensor rays which are uniformly distributed over a fixed angle ($\Pi/6$) starting from the centre of the robot and facing straight up. The rays act as proximity sensors with a maximum sensing range of 220 Pixel and a maximum output value if the object has reached the robot.

Each robot has two motor neurons for horizontal motion. They define the speed and direction of the robot. The speed is given by $Output_{Neuron0}$-$Output_{Neuron1}$. Negative speed stands for motion to the left, positive values for motion to the right. The maximum speed is 2 Pixel/time step.

Two shapes are used for the task: Circles (radius = 30) and diamonds (side length = 30). Reference points are always the centres of the objects or the robot. The robot has to catch circles and avoid diamonds. The falling speed for all objects is always one Pixel per time step. This leads to a simulation time of 275 discrete time steps. The horizontal offset of the object can be +/- 50 Pixel from the robots starting point.

After the falling object has reached the floor ($x_{Obejct} = 0$) the fitness of the robot is evaluated by the following formula:

$$\sum_{i=1}^{24} p_i \Big/ 24 \quad \textbf{(1)}$$

where $p_i = d_i$ for diamonds and $p_i = 1\text{-}d_i$ for circular objects. d_i is the distance between centre of robot and objects and is clipped to a maximum distance and normalized to [0,1]. The maximum distance is 1.5 * ($radius_{Object}$ + $radius_{Robot}$) and has to be used, because it leads to a balance between avoidance distance for diamonds and accuracy in catching circles. 24 evaluation trials are performed for each robot with uniformly distributed dropping points and alternating object shapes.

2.2 Genetic Algorithm

In all experiments and for all network types, the same genetic algorithm is used. The competition is tournament based and the algorithm as follows:

The population is spatially distributed on a square plane and an individual randomly chosen. The tournament group consists of this individual and its 8 neighbours. The two fittest individuals are picked, recombined and the weakest individual in the group is then replaced by the mutated offspring.

As recombination, 1-point crossover is used with a randomly chosen crossing point between nodes (only whole nodes are transferred). For the shape discrimination task, the population size was always 324 and the genetic algorithm was running over a maximum of 200 pseudo generations or stopped if an individual with a fitness > 0.99 was found. One pseudo generation is equal to 324 selection and recombination processes.

Loci from the produced offspring are chosen for mutation with a fixed probability. This mutation rate is adjusted to the number of loci which can be mutated in the genotype (for some experiments, specific loci are locked) and therefore, always the same number of mutations are performed on average for each network type. If chosen, the locus is mutated using a Normal distribution with standard deviation of 1, which is scaled up (or down) to the range of the mutated value. The mean of the distribution is the original value of the locus. The maximum possible mutation is (upper limit – lower limit)/2. For example, the possibility to change a weight in a CTRNN ($\omega \in [-5,5]$) by less than 1.0 is 84%. The possibility for a change less than 2.0 is 98%.

2.3 Network Types and Characteristic Equations

2.3.1 CTRNN Network Types. All Continuous Time Recurrent Neural Networks used in this work have a fixed number of seven neurons. These neurons are divided into five inner neurons and two motor neurons. The inner neurons are fully connected. Each of them can have one non-weighted sensor input connection and is connected to the motor neurons. All connections between neurons have a weight value in each direction. The motor neurons are not connected to each other and can not receive input from sensors.

Three different CTRNN variations are used in this work, which are all based on this basic topology. Each network has a specific name in brackets for later references.

2.3.1.1 Standard CTRNN (C1). This is a conventional CTRNN [1] with the following characteristic equation:

$$y_i^{t+1} = y_i^t + \frac{\Delta t}{\tau_i}\left[-y_i^t + \sum_{j=1}^{N} \omega_{ji}\sigma(y_j^t + \theta_j) + I_i \right] \quad (2)$$

Where:

y_i^t is the activation of neuron i at time t

Δt is the time slice (Δt was 1.0 in all shape discrimination trials)

τ_i is the time constant of neuron i ($\tau \in [1,5]$)

ω_{ji} is the weight of the connection from node j to i ($\omega \in [-5,5]$)

θ_j is the bias term of neuron j ($\theta \in [-5,5]$)

I_i is the sensor input to the i'th neuron ($I \in [0,10]$, see experimental setup)

σ is the sigmoid function

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (3)$$

Biases, time constants, connection weights and the input source are under evolutionary control, i.e. for the input, the evolution can choose from which sensor ray the input is from.

2.3.1.2 CTRNN with no Temporal Dynamics (C2). This network has the same characteristic equation and topology as C1, but τ is set to 1.0 for all neurons and is not under evolutionary control. This effectively turns the neurons into reactive integrate-and-fire type neurons with no internal temporal dynamics.

2.3.1.3 CTRNN with Discrete Weights (C3). Same as C1, but only discrete weights are used. Connection weights in this network type can only be -5, 0 or 5.

2.3.2 GasNet Types. GasNets, introduced by Husbands et al. [4], have two different signalling mechanisms. They have electrical connections which can be compared to other ANN types and a gas signalling mechanism. The gases can be emitted by nodes and have modulatory effects on the transfer function of nodes in the vicinity of the emitting node. These gas connections work on different time scales than the electrical connections by build up and decay mechanisms. To model the gas diffusion the neurons of a GasNets are spatially distributed points on a square plane (in this work with side length 50 pixels). Electrical connections are based on neurons' positions in this plane with connections from a neuron being formed to all others within a genetically specified arc.

Detailed information on the GasNet model can be found in [12], [7]. The following chapters only repeat what is relevant for this work.

2.3.2.1 Standard GasNet (G1). This is a standard GasNet, already used in previous work [4-8,11-14] The characteristic equation is as follows:

$$y_i^t = \tanh\left[K_i^t\left(\sum_{j\in C_i}\omega_{ji}y_j^{t-1} + I_i^t\right) + \theta_i\right] \tag{4}$$

Where:

y_i^t is the output of neuron i at time t

K_i^t is the transfer function parameter

C is the set of nodes which have connections to node i

ω_{ji} is the weight of the electrical connection from node j to node i (ω ∈ [-1|1])

θ_j is the bias term of neuron j (θ ∈ [-1,1])

I_i is the sensor input to the i'th neuron (I ∈ [0,1], see experimental setup)

In this work two gas types are used. Gas 1 increases the transfer function parameter K and gas 2 decreases it. This is done, dependent on the gas concentration at a given node. The gas concentration at a given node j with a distance d to the emitting node i at time t, is given by the following equations:

$$C(d,t) = \begin{cases} e^{-(d/r)^2}\cdot T(t), d < r \\ 0 \qquad\qquad\qquad , else \end{cases} \tag{5}$$

$$T(t) = \begin{cases} H\left(\dfrac{t-t_e}{s}\right) & emitting \\ H\left(H\left(\dfrac{t_s-t_e}{s}\right) - H\left(\dfrac{t-t_s}{s}\right)\right) & not\ emitting \end{cases} \tag{6}$$

$$H(x) = \begin{cases} 0 & x \le 0 \\ x & 0 < x < 1 \\ 1 & else \end{cases} \tag{7}$$

where r is the radius of the gas cloud, s is the parameter that controls the build up/decay rate of the gas, t_e is the time node i started emitting and t_s the time node i stopped emitting. The parameters r and s are genetically determined for each node.

The parameter K_i^t for node i at time step t is then given by equations 8 to 11:

$$K_i^t = \mathrm{P}[D_i^t] \quad \mathbf{(8)}$$

$$\mathrm{P}=\{-4.0,-2.0,-1.0,-0.5,-0.25,-0.125,0.0,0.125,0.25,0.5,1.0,2.0,4.0\} \quad \mathbf{(9)}$$

$$D_i^t = f(D_i^0 + C_1^t(13 - D_i^0) - C_2^t D_i^0) \quad \mathbf{(10)}$$

$$f(x) = \begin{cases} 0 & x \le 0 \\ \lfloor x \rfloor & 0 < x < N \\ N-1 & else \end{cases} \quad \mathbf{(11)}$$

2.3.2.2 GasNet without Gas (G2). This network type was also used in previous work and is a standard GasNet but using no gas (NoGasNet). This means no neuron can emit gas and only electrical connections can be used to connect neurons.

2.3.2.3 GasNet with Weight Table (G3). This is a fully connected GasNet. All neurons are connected and evolution can set the weights of these connections. As weight values, -1, 0 and 1 are used, as in the original GasNet. This means, that the network initially is never totally connected, because on average 1/3rd of the connections has weight = 0. However it is much easier for evolution to connect two nodes by mutating only one locus than when using the spatial connectivity scheme of the standard GasNet.

2.3.2.4 GasNet with Weight Table but Real Weights (G4). Same network type as G3, but using real weights. The initial connection ratio is much higher in this net, because it is very unlikely to have connections with weight=0.0.

2.3.2.5 GasNet with Weight Table, Real Weights but no Gas (G5). Same network as G4 but again no gas is used.

2.3.2.6 GasNet without Spatial Distribution (G7). In this GasNet the G5 type is changed a little bit. Because the G5 type does not use gas any more and all connections are given by a weight table, the neurons don't have to be spatially distributed any more. So, in the G7 type, the x and y loci are set to 0.0 and locked for mutation.

3 Results

As can be seen in Table 1, the CTRNN controlled robots clearly outperform the GasNet controlled robots. The best robots evolved over all runs were controlled by C1 and C2 networks (standard CTRNN and CTRNN with $\tau = 1$). They had a fitness value of 0.990 on the evaluation trials and an average of around 0.995 on 100 random trials. The best of the best GasNet controlled robot had an average fitness of .991 and an average of 0.959 on 100 random trials. It should be pointed out that the genetic algorithm stopped if a best individual was found (with a fitness of 0.990) or if the limit of 200 generations was reached. This explains the maximum of 0.990 on the CTRNN runs and the average generation number of around 200 on the GasNet runs.

The C1- and C2-robots only once reached the maximum of 200 generations. The best individual in this run already had a fitness value of 0.949. Only one G1 run stopped before the maximum number of generations was reached (177 generations). Two G2 runs stopped before the maximum was reached (151 and 182 generations)

Table 1. Results of 20 evolutionary runs. The table shows the average fitness of the best individual and the average of the whole population after 20 runs, the average fitness of the best individual on 100 random trials and the average number of generations needed to evolve the best individual. The values in brackets specify the standard deviation

Network type:	Average of best individual:	Average fitness of population:	Average fitness on random trials:	Average Nr of generations:
C1	0.990 (0.010)	0.826 (0.030)	0.949 (0.043)	100.000 (42.7)
C2	0.990 (0.010)	0.798 (0.030)	0.953 (0.031)	108.000 (55.3)
G1	0.882 (0.099)	0.762 (0.075)	0.825 (0.119)	198.000 (5.1)
G2	0.775 (0.100)	0.683 (0.062)	0.723 (0.125)	196.000 (11.2)

Notice that the C2 variant, which has no internal temporal dynamics, performed as well as C1. This is evidence that different temporal scales are not needed in the nodes to complete the task and that the use of different temporal scales in the network does not lead to improved performance. Samples did not show evidence that the evolved solutions are different between both network types.

4 Why Are CTRNNs Better?

There are a lot of differences between CTRNNs and GasNets which could account for the CTRNN's better performance. In the following sections specific differences are highlighted and their influence on the advantage of CTRNNs evaluated.

4.1 Connection Scheme

The most eye-catching difference is the different connection scheme between both network types. CTRNN variants are almost fully connected whereas GasNets are only sparsely connected. This means that an evolving CTRNN only has to find a working set of weights, but an evolving GasNet also has to find the right sensor to motor connection mapping.

To find out if the connection scheme is accountable for a performance gain, a fully connected GasNet version (G3) was evaluated and compared to G1 and G2. For 7 neurons in a network, the gene length of G3 is the same as for G1. (In G1 seven loci are used to specify the connection scheme. In G3 the seven connection weights (-1, 0 or 1) are stored instead) so different mutation factors are no issue. No performance gain occurred by connecting all neurons in a GasNet (Table 2). In fact, the opposite occurred and the fully connected GasNet performs worse than the standard GasNet. This shows clearly that it is not the connection scheme on its own that leads to a better performance. It should be pointed out, however, that even in a G3 network the nets are never fully connected and the chance of a connection with weight = 0.0 is much higher than in a CTRNN with real weighted connections.

Table 2. Evolution of GasNet G3 compared to standard GasNet G1 and NoGasNet G2

Network type:	Average of best individual:	Average fitness of population:	Average fitness on random trials:	Average Nr of generations:
G1	0.882 (0.099)	0.762 (0.075)	0.825 (0.119)	198.000 (5.1)
G2	0.775 (0.100)	0.683 (0.062)	0.723 (0.125)	196.000 (11.2)
G3	0.743 (0.048)	0.591 (0.022)	0.699 (0.062)	200.000 (0.0)

4.2 Real Weights

Apart from the connection scheme, CTRNNs also use real weights for connections while GasNets only use inhibitory (weight = -1) or excitatory (weight = 1) connections. To find out if this influences evolution, a CTRNN variant (C3) with discrete weights (-5, 0 or 5) and GasNet variants (G4 and G5) with weight tables and real weights (with and without gas) were evaluated and compared to the standard CTRNN and GasNet versions.

Table 3. Comparison of evolution results for networks with varying weight type

Network type:	Average of best individual:	Average fitness of population:	Average fitness on random trials	Average Nr of generations:
C1	0.990 (0.010)	0.826 (0.030)	0.949 (0.043)	100.000 (42.7)
C2	0.990 (0.010)	0.798 (0.030)	0.953 (0.031)	108.000 (55.3)
G5	0.973 (0.041)	0.703 (0.035)	0.911 (0.075)	160.000 (36.4)
G4	0.954 (0.048)	0.766 (0.049)	0.920 (0.063)	185.000 (26.0)
G1	0.882 (0.099)	0.762 (0.075)	0.825 (0.119)	198.000 (5.1)
C3	0.874 (0.051)	0.631 (0.019)	0.835 (0.062)	200.000 (0.0)
G2	0.775 (0.100)	0.683 (0.062)	0.723 (0.125)	196.000 (11.2)

As one can see in Table 3, if the connections in a fully connected GasNet are combined with real weights, the results of the best individual almost reach the CTRNN results. Real weights for connections seem to be crucial for successfully evolved controllers in this task. As a crosscheck the network C3 was used which performed worse than all network types with real weights. Possible reasons for this are that it is easier for evolution to fine tune connections or to use the same neuron output as input with different strengths in different neurons.

As it turned out in the GasNet experiments, a successfully evolved solution doesn't need a lot of connections. It seems reasonable that if there are only a few connections, it is crucial to be able to fine tune them. Real weights provide this opportunity. More experiments have to be done to prove this and to find the specific reasons for the necessity of real weights. However, even with full connection scheme and real weights, the GasNets are still outperformed by both CTRNN variants in terms of evolution time. Thus, real weights are important, but not the only reason.

4.3 Neutrality

After examining many GasNet runs, we noticed that the best individual often does not change over long periods. Moreover, the fitness of the best individual plotted over generations shows long, flat regions: much longer than CTRNN runs. In [14], Tom Smith et al. show that GasNets have high functional neutrality, i.e. "many distinct neural network structures will produce the same functional mapping from sensory input to motor output" [14]. Perhaps this neutrality causes the long and flat regions?

To answer this question, a different kind of mutation operator is used. Every time a genotype is mutated, the corresponding phenotype is compared to the phenotype corresponding to the original genotype. If no change in the phenotype can be detected (i.e. phenotype has same electrical and gas connections), then the genotype is mutated again. This procedure continues till the mutation affects the phenotype. Although the mutation operator prevents neutral mutations, the picture does not change significantly. There are still long periods without change which leads to the conclusion that evolutionary search got stuck, but not for neutrality reasons. Also the overall results do not change (Table 4).

However, type G7 was also used which suggests that neutrality is not unimportant. Types G7 and G5 nn are functionally the same since changes to the coordinates in G5 are functionally neutral. Thus, in G7 mutation cannot change these coordinates and so less neutral mutations are made.

Table 4. Results from GasNet variants compared to the same variants using the "noNeutrality"-mutation operator (nn) and the results of the GasNet variant G7

Network type:	Average of best individual:	Average fitness of population:	Average fitness on random trials	Average Nr. of generations:
G5 nn	0.986 (0.024)	0.702 (0.044)	0.914 (0.048)	143.000 (49.3)
G7	0.984 (0.037)	0.722 (0.043)	0.913 (0.065)	119.000 (53.1)
G5	0.973 (0.041)	0.703 (0.035)	0.911 (0.075)	160.000 (36.4)
G4	0.954 (0.048)	0.766 (0.049)	0.920 (0.063)	185.000 (26.0)
G4 nn	0.948 (0.064)	0.710 (0.043)	0.894 (0.089)	163.000 (44.1)
G1	0.882 (0.099)	0.762 (0.075)	0.825 (0.119)	198.000 (5.1)
G1 nn	0.865 (0.057)	0.738 (0.050)	0.794 (0.093)	197.000 (9.4)

The time to evolve a successful G7 individual was significantly less than other GasNet types and is close to the result of the CTRNN results.

4.4 Fine Tuning Ability

The fitness value attained is strongly dependent on the exact position of the robot. If a robot is able to distinguish between circles and diamonds but cannot reach the exact position of the object at the end, the fitness value can be the same or even worse as the fitness of a robot that fails to distinguish shapes a few times, but has reached the exact position in all other trials. Exact positioning is rewarded. In 100 random trials (50% circles), the fitness of a robot that misses the exact position by one pixel every time while catching circles is 0.989 ((0.978 * 50 + 50) / 100). The average fitness of a robot that fails to distinguish a shape once, but positions exactly in 99 other trials is 0.990 (assuming that they all successfully avoid diamonds).

To find out if GasNets are likely to evolve controllers which can correctly distinguish shapes, but fail in finding the exact position (fine tuning), a different fitness function is used:

$$f = \begin{cases} 0.0 & , d < \min \\ (d - \min)/(\max - \min) & , \min < d < \max \\ 1.0 & , d > \max \end{cases} \tag{12}$$

$$fitness = \begin{cases} f & , diamond \\ 1 - f & , circular\ object \end{cases} \tag{13}$$

Where d is the distance between the centres of robot and object, min = $\text{radius}_{\text{Object}}$ and max = 1.5 * ($\text{radius}_{\text{Object}}$ + $\text{radius}_{\text{Robot}}$). This means that the robot does not have to find the exact position but only has to reach the area under the object.

Table 5. Results with new fitness (nf) compared to standard GasNet (G1) and NoGasNet (G2)

Network type:	Average of best individual:	Average fitness of population:	Average fitness on random trials	Average Nr. of generations:
G1	0.882 (0.099)	0.762 (0.075)	0.825 (0.119)	198.000 (5.1)
G1 nf	0.845 (0.096)	0.720 (0.049)	0.741 (0.150)	186.000 (37.5)
G2	0.775 (0.100)	0.683 (0.062)	0.723 (0.125)	196.000 (11.2)

The results in Table 5 show there is no difference between G1 and G1nf with new fitness. Hence, the problem of the GasNet evolution does not seem to be fine tuning of parameters governing the robot's final position.

5 Discussion

This work set out to give evidence that GasNets are outperformed by other neural network types if the solution to a given task does not need timer or pattern generator sub networks. To prove this, a task was chosen where the ability to use different time scales in the network gives no advantage. Samples over evolved controllers from both types showed, that successful solutions used active scanning and no examined GasNet had a pattern generator sub net. As shown in section 3, CTRNNs with or without evolvable time constants perform the same, which shows that timing is not necessarily needed for a successful solution. CTRNNs solve this task easily, while GasNets perform much worse. Further evidence that it is timing that is important is that GasNets have been shown to outperform other network types on tasks where pattern generation is needed [4] [5]. However, in a comparison with CTRNNs on a simple pattern generation task, while GasNets were superior to CTRNNs on one pattern, the converse was true for a second pattern [16]. It is possible that these differences are due to the range of temporal dynamics available to the two types of networks, but further work is needed to investigate this fully.

Different reasons for the disadvantage of GasNet controllers were examined. It was shown, that connectivity and fine tuning issues have no big impact on the results of the evolutionary runs. A fully connected GasNet with original GasNet weights (-1,0,1) using a weight table does not lead to a measurable performance gain. The crucial issue seems to be to have real weighted connections. As soon as a GasNet has real weighted connections, its performance is much better, while a CTRNN with discrete weights performs much worse. While the reason for the necessity of real value weights in this task is not known it is possible that the difference between the standard GasNet and its no-gas counterpart can be explained by the need for evolution to use different connection types where real weights are not available to enrich the connection scheme.

It was also shown that the time evolution needs to find a reasonable good individual decreases significantly for GasNets if loci with a high possibility of neutral mutations are taken out of evolutionary control. This is not surprising but can be still be outweighed by other issues for more complex tasks (e.g better robustness) and therefore worth accepting.

The results support the initial hypothesis, that while GasNets are good to evolve timers and pattern generators, they have disadvantages if other issues are more important. No successfully evolved GasNet controller that was analyzed during this work was using a timer/PG sub network or used a technique where timer/PG sub networks are useful. This work therefore provides evidence that for simple tasks which do not require timers or pattern generators, other issues which are not suited to GasNet type networks become more important. More research on the dynamics of GasNets is thus needed to further classify the type of tasks where they outperform other network types.

6 References

1. Beer, R.D. (1995): On the dynamics of small continuous-time recurrent neural networks. Adaptive Behavior 3(4):471-511.
2. Beer, R.D. (1996): Toward the evolution of dynamical neural networks for minimally cognitive behavior. In P. Maes, M. Mataric, J. Meyer, J. Pollack and S. Wilson (Eds.), "From animals to animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior" (pp. 421-429). MIT Press.
3. Funahashi, K. (1989): On the approximate realization of continuous mappings by neural networks. Neural Networks, 2, 183—192.
4. Husbands, P., Smith, T., Jakobi, N., and O'Shea, M. (1998): Better living through chemistry: Evolving GasNets for robot control. Connection Science, 10(3-4):185-210.
5. McHale, G. and Husbands, P. (2004): GasNets and other Evolvable Neural Networks applied to Bipedal Locomotion. From Animals to Animats 8: Proceedings of the Eigth International Conference on Simulation of Adaptive Behavior (SAB 2004)
6. McHale G., Husbands P. (2004): Quadrupedal Locomotion: GasNets, CTRNNs and Hybrid CTRNN/PNNs Compared. In Jordan Pollack, Mark Bedau, Phil Husbands, Takashi Ikegami and Richard A. Watson (Eds), "Artificial Life IX: Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems" , 106-113, MIT Press, 2004.
7. Philippides, A., Husbands P., Smith, T. and O'Shea, M.. (2005): Flexible couplings: diffusing neuromodulators and adaptive robotics. Artificial Life , 11(1&2), 139-160, 2005.
8. Philippides, A.O., Husbands, P., Smith, T. and O'Shea, M. (2002): Fast and Loose: Biologically Inspired Couplings. In Standish, R.K., Bedau, M.A. and Abbass, H.A. , editors, "Artificial life VIII. Proceedings of the 8th international conference on artificial life", pages 292-301. MIT Press.
9. Philippides, A. Personal communication
10. Slocum, A.C., Downey, D.C. and Beer, R.D. (2000): Further experiments in the evolution of minimally cognitive behavior: From perceiving affordances to selective attention. In J. Meyer, A. Berthoz, D. Floreano, H. Roitblat and S. Wilson (Eds.), "From Animals to Animats 6: Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior" (pp. 430-439). MIT Press.
11. Smith, T., Husbands, P., Philippides, A., and O'Shea, M. (2003): Temporally adaptive networks: analysis of GasNet robot control networks. In R. K. Standish, M. A. Bedau, and H. A. Abbass, Eds. "Proceedings of the Eighth international Conference on Artificial Life" MIT Press, Cambridge, MA, 274-282.
12. Smith, T. (2002): The evolvability of artificial neural networks for robot control. PhD Thesis. School of Biological Sciences, University of Sussex.
13. Smith, T.M.C. and Philippides, A. (2000): Nitric Oxide Signalling in Real and Artificial Neural Networks. British Telecom Technology Journal, 18(4):140-149.
14. Smith, T.M.C., Philippides, A., Husbands, P. and O'Shea, M. (2002): Neutrality and Ruggedness in Robot Landscapes. In "Congress on Evolutionary Computation: CEC2002", pages 1348-1353. IEEE Press.
15. Yamauchi, B. and Beer, R.D. (1994): Sequential behavior and learning in evolved dynamical neural networks. Adaptive Behavior 2(3):219-246.
16. Magg, S. (2005): CTRNNs and GasNets : A comparison in terms of evolvability. Master Thesis, Department of Informatics, University of Sussex