Incremental Evolution of Neural Network Architectures for Adaptive Behaviour

D. Cliff, I. Harvey, P. Husbands
CSRP 256, December 1992

Cognitive Science Research Paper

Serial No. CSRP 256

The University of Sussex
School of Cognitive and Computing Sciences
Falmer
BRIGHTON BN1 9QH
England, U.K.

Incremental Evolution of Neural Network Architectures for Adaptive Behaviour

Dave Cliff [1,2] and Inman Harvey[1] and Philip Husbands[1]
[1]School of Cognitive and Computing Sciences
[2]Neuroscience IRC, School of Biological Sciences
University of Sussex, Brighton BN1 9QH, U.K.
davec, inmanh, philh all @cogs.susx.ac.uk

Abstract

This paper describes aspects of our ongoing workolv(w)gSudynam(Sci..s)-.tivntrac(nt).(k)-mfh.(or.tracTda

with which the models can be simulated or built in available hardware is an important factor, and appropriate simplifications are made. In either case, it is important to note that the 'simplification' is made for *our* convenience: the ANN is easier to construct or understand. The problem with this approach is that in using simplified models, we may actually be making extended form of genetic algorithm, known as SAGA. Whereas most genetic algorithms are essentially performing optimisation in a fixed parameter space, SAGA allows for the *dimensionality* of the

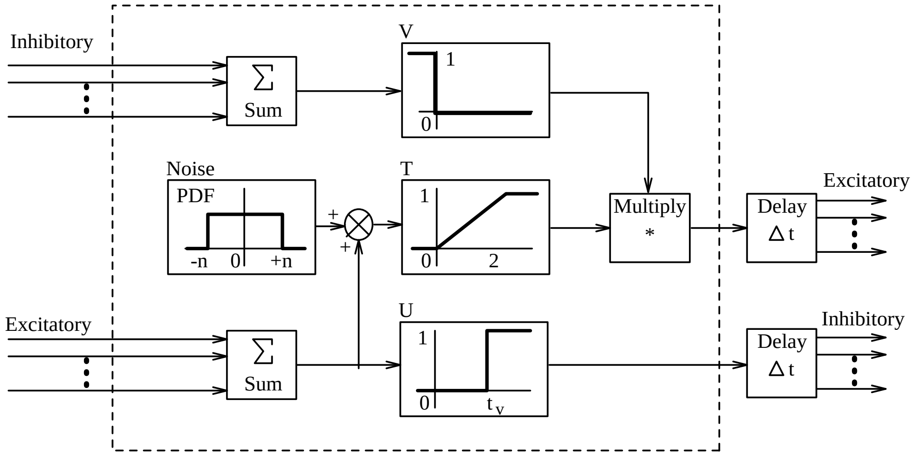


Figure 1: Schematic block diagram showing operations within a single model neuron. See text for further explanation.

the form:

$$\mathcal{U}(x) = \left\{ \right.$$

2.2 The Genetic Encoding

To enable the use of SAGA, the network architecture has to be encoded as a 'gene'. In our work to date, we have used a genetic encoding scheme

3 Evolving a Visually Guided Robot

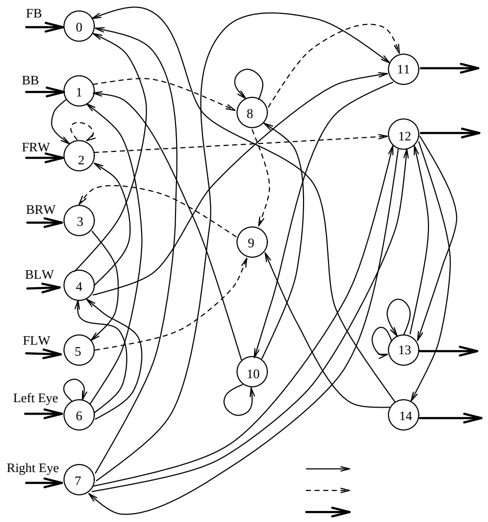
Here we briefly present some recent results. We attempted to evolve networks for a simple adaptive behaviour, which was for a simulated[1] visually guided robot to spend as much time as possible in the centre of a circular arena. The robots have two independent drive-wheels and a third free-wheel. The drive wheels may go at either full or half speed, either forwards or reverse, so the robot is capable of rotating on the spot, or travelling in wider-radius circles, or in straight lines, or stopping still.

Each robot had six tactile sensors: two 'bumpers' (at front and back), and four radially symmetric wire 'whiskers'. The tactile sensors are primarily of use in detecting collisions with walls of the arena, and appropriately reorienting. The robot also has two directionally-sensitive photoreceptors, which allowed it to visually sense its environment (the walls of the simulated arena are dark, while the floor and ceiling are light).

Each individual robot was positioned at a randomly chosen point near the edge of the arena, in a random orientation. The robot then had a fixed finite 'lifetime', in which it had to get as close to the centre of the arena as it could, and then stay there. The robot's performance was evaluated by taking the gaussian function of a discrete temporal integral of its distance from the arena-centre during its lifetime: the higher the evaluation function, the more time the robot spent at or near the centre. As is demonstrated in [4], this is sufficient to evolve controllers for visually-guided behaviours: no explicit specification of visual processing is required.

We created a population of 60 robots with initially random genes, and evaluated each one over eight 'lifetimes'. At the end of the evaluation, we took the robot's *worst* score as a measure of its performance (best and average scores are too often deceptively high). When all 60 robots had been evaluated 8 times, the genes of the higher-scoring robots were 'inter-bred' using SAGA principles to create a new generation of 60 individuals. We repeated this process for 100 generations.

The typical behaviour of a robot controlled by an evolved network is that it finds its way to the centre of the circular arena, and then stays there by spinning on the spot. This is a perfectly acceptable strategy, given that the robots were evaluated only on the basis of how much time they spent at the centre, and not on the basis of how much energy they used. In this paper, we will consider two of the best evolved robot controller networks, referred to as C1 and C2. The architectures of the two networks are shown in Figures 2 and 3. Typical behaviours exhibited by the robots controlled by these networks are shown in Figures 4, 5, and corresponding time-plots of sensor inputs, internal section discusses our findings from studies of varying the amount of internal noise in the neuron model.



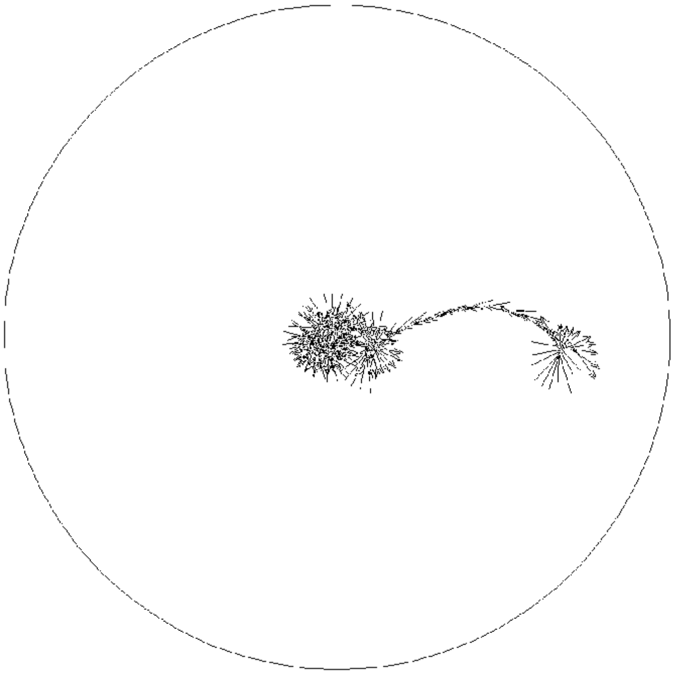


Figure 4: Typical behaviours of the robot controlled by evolved networks. The figure shows a top-down view of the circular arena; the robot'sca2(then2oh.n000.2(eha)1000.:000.(ofa)ToT60Td[isb)-0

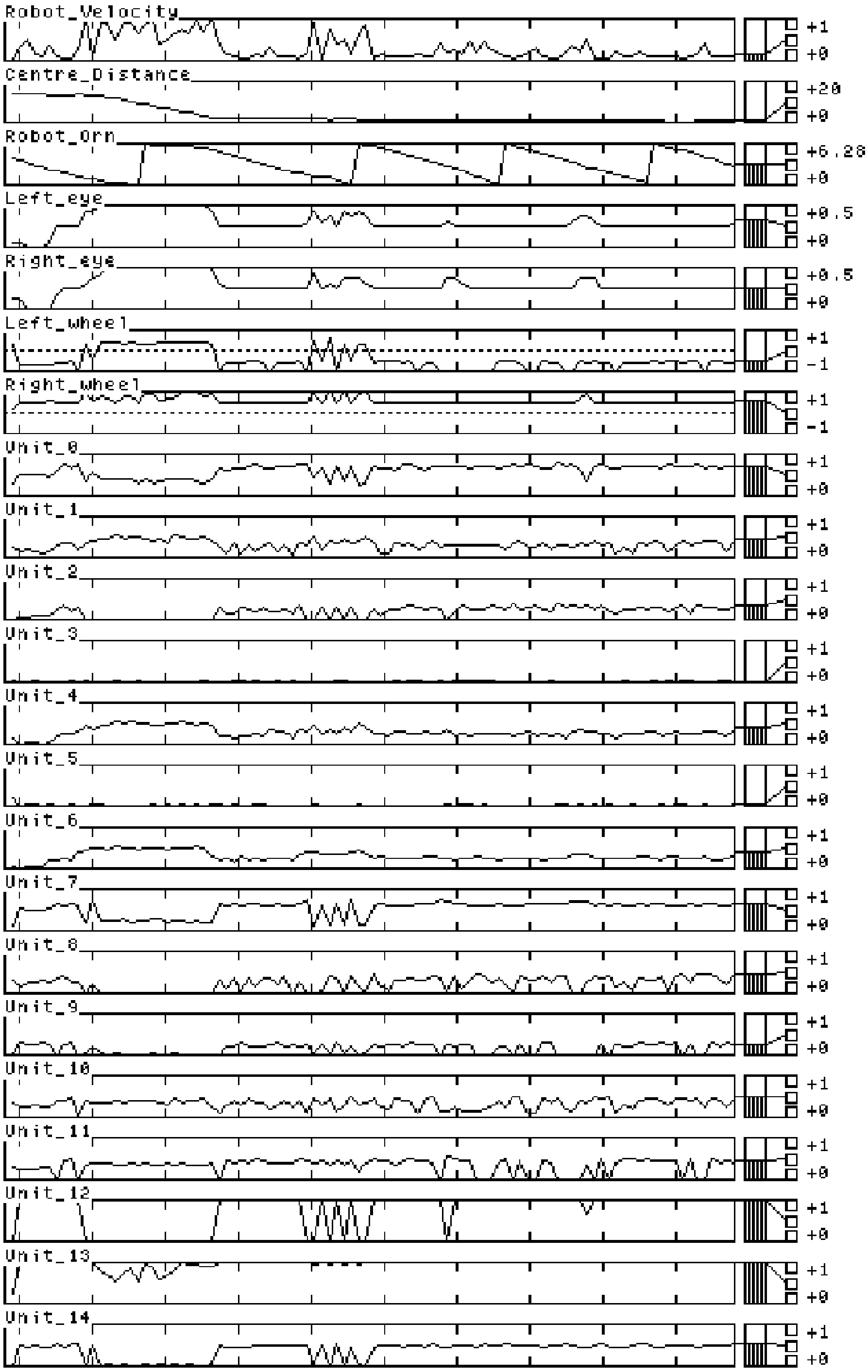


Figure 6: Time-plots of sensor, motor, and internal activation values for the C1 behaviour plotted in Figure 4. From top, graphs show: robot's velocity; distance of robot from centre of arena; visual input to left eye; visual input to right eye; output of motor for left wheel; output of motor for right wheel; excitatory output of the model neurons ("units") in the network.

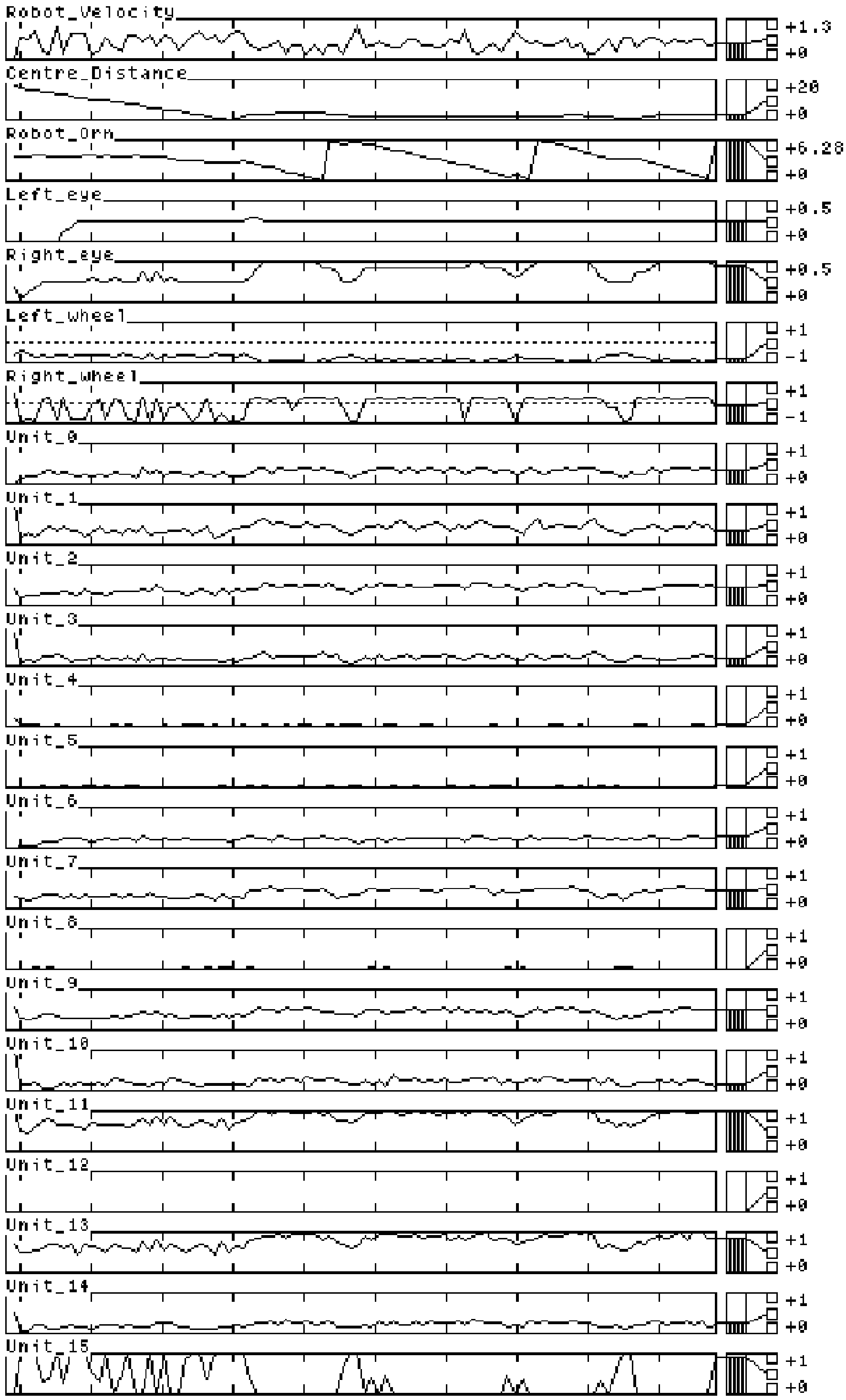
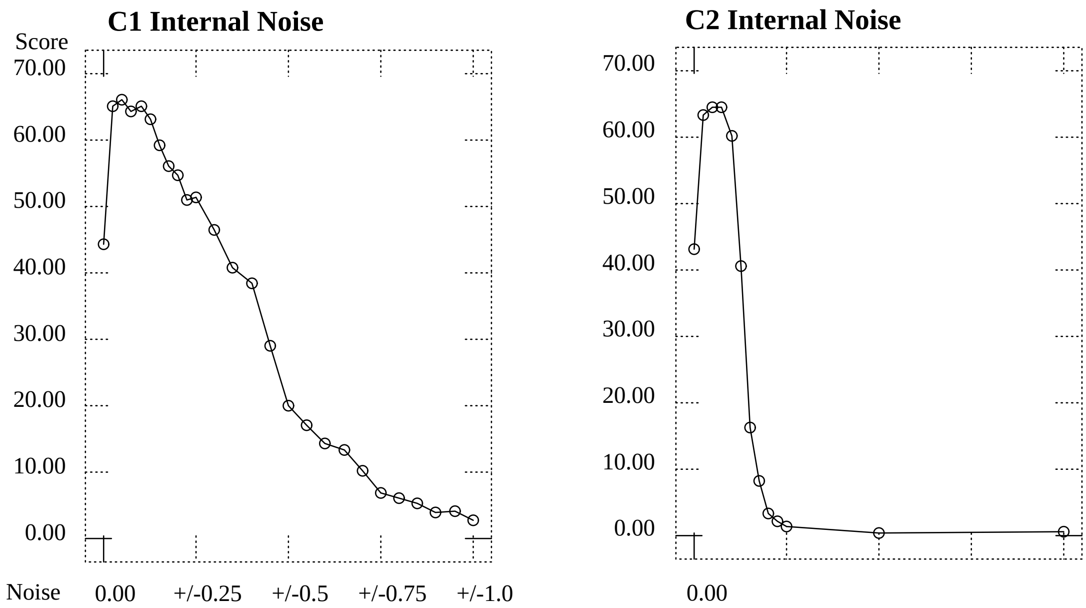


Figure 7: Time-plots of sensor, motor, and internal activation values for the C2 behaviour plotted in Figure 5. Display format as for Figure 6.

4 Varying the Noise

As was mentioned above, each robot is monitored for the same fixed amount of time, during which its fitness value is calculated. For further details of this process, see [3]. For the purposes of this discussion, it is sufficient to note that, if the robot spent all its time at the centre, it would receive a score of 100. But, because each robot's randomly chosen initial position is always some distance from the centre, this maximum score can never be reached: an optimum controller would score about 85 points. A robot which never moved would score less than one.

For the C1 and C2 controllers, after 100 generations of evolution, both networks managed an *average* score of around 65 (peak scores were nearer 80). These are the scores obtained with internal noise $n = 0.1$ injected in the model neurons. Both networks were then tested with different values of n, varying from $n = 0$ (i.e. no noise) to $n = 1.0$ (i.e. noise uniformly distributed in the range $[-1.0, 1.0]$). For each value of n, the network was evaluated 80 times, and the average score taken. Results from these tests are shown in Figure 8.



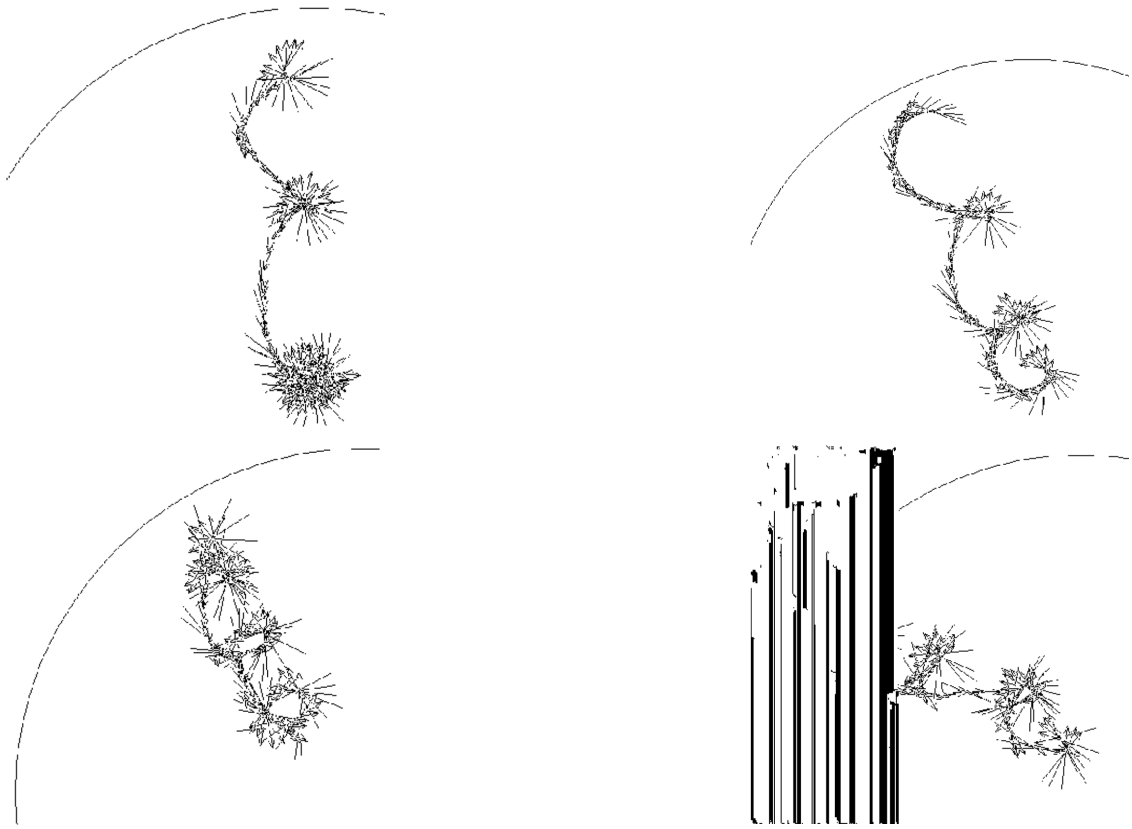


Figure 9: Typical behaviour resulting from using C1 with varying levels of internal noise. Display format as for Figure 4. At the top, on the left noise ± 0.2, on the right ± 0.4. Below, on left ± 0.6, on right ± 0.8.

4.1 Discussion

Further examination of the results indicates that the drop in performance when noise is eliminated is due to the recurrent dynamical nature of the networks: the recurrency implies that the network architectures contain feedback loops at a number of levels. That is, it is common to see a unit with connection(s) to itself, or two mutually excitatory units, or cycles of excitatory links incorporating several units. In such cases, low levels of internal noise may build up over time by a process of accumulation through feedback loops. However, because the noise distribution is centred on zero, it is also possible that these high levels of activity could then works, the drop in *average* performance was due to an increase in the number of near-zero scores: peak scores were still high, but under certain conditions the absence of noise allowed the activity in the network to fall to such an extent that the robot was rendered immobile. Put more formally, the noise helps the state trajectory of the controller system from becoming trapped on attractors which correspond to inactivity or unproductive behaviours. In this sense, it is realistic to describe the networks as *using* noise to produce useful behaviours.

This is a significant issue: if networks evolve to take advantage of internal noise, then it is important to ensure that the internal noise used in simulation (i.e. during evolution) closely matches the true noise levels that will be found when the evolved controller is put into use. In our current work, this is something of a non-issue because the real robot can be controlled by evolved networks simulated in the same manner as was employed in evolution, using the robot's on-board microcomputer. However, if our methods are used to develop control networks which will be realised in hardware, with each model neuron implemented as an electronic circuit, then it is essential that a fairly precise characterisation of the tolerances and internal noise distributions of the model neuron circuit should be incorporated in the evolution simulation.

5 Conclusion

Our work is motivated by concerns that prior network models may have been over-simplistic, and have not paid sufficient attention to the generation of adaptive behaviour. We have demonstrated that, using a neuron model with elementary dynamics, recurrent networks can exhibit rich dynamical activity that is not unduly hampered by noise, and can be used for evolving controller networks that generate adaptive behaviour. We have presented results which indicate that the networks use noise to avoid the effects 'unproductive' attractors can have on the state trajectory of the controller network. The evolved networks have a distinctive appearance, in that they do not resemble networks designed by humans. As far as we know, we are the only research group who have successfully employed truly incremental evolution in creating dynamic recurrent networks for the generation of adaptive behaviour. We expect that our techniques will, as time progresses, become standard practice.

References

[1] R. D. Beer and J. C. Gallagher. Evolving dynamical neural networks for adaptive behaviour. *Adaptive Behaviour*, 1(1):91–122, 1991.

[2] D. T. Cliff. Computational neuroethology: A provisional manifesto. In J.-A. Meyer and S. W. Wilson, editors, *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior (SAB90)*, pages 29–39, Cambridge MA, 1991. M.I.T. Press — Bradford Books. Also available as University of Sussex School of Cognitive and Computing Sciences Technical Report CSRP162.

[3] D. T. Cliff, P. Husbands, and I. Harvey. Analysis of evolved sensory motor controllers. Technical Report CSRP 264, University of Sussex School of Cognitive and Computing Sciences, 1992.

[4] D. T. Cliff, P. Husbands, and I. Harvey. Evolving visually guided robots. In J.-A. Meyer, H. Roitblat, and S. Wilson, editors, *Proceedings of the Second International Conference on Simulation of Adaptive Behaviour (SAB92)*. MIT Press Bradford Books, Cambridge, MA, 1993. In Press. Also available as University of Sussex School of Cognitive and Computing Sciences Technical Report CSRP220.

[5] H. de Garis. Genetic programming: Building artificial nervous systems using genetically programmed neural network modules. In B. W. Porter and R. J. MooneyroTr000.2(MIT)-1