Analysing Recurrent Dynamical Networks Evolved for Robot Control

P. Husbands, I. Harvey, D. Cliff CSRP 265,
January 1993

CSRP 265, January 1993

Cognitive Science Research Paper

Serial No. CSRP 265

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

A version of this paper appears in:

Proc. 3rd. IEE Int. Conf. on Artificial Neural Nets, IEE Press 1993

ANALYSING RECURRENT DYNAMICAL NETWORKS EVOLVED FOR ROBOT CONTROL

Philip Husbands[1] and Inman Harvey[1] and Dave Cliff [1,2]
[1]School of Cognitive and Computing Sciences
[2]Neuroscience IRC, School of Biological Sciences
University of Sussex, BRIGHTON BN1 9QH, U.K.
Phone +44 273 606755, Fax +44 273 671320
davec or philh or inmanh, all @cogs.susx.ac.uk

Abstract

This paper shows how a mixture of qualitative and quantitative analysis can be used to understand a particular brand of arbitrarily recurrent continuous dynamical neural network used to generate robust behaviours in autonomous mobile robots. These networks have been evolved in an open-ended way using an extended form of genetic algorithm. After briefly covering the background to our research, properties of special frequently occurring subnetworks are analysed mathematically. Networks evolved to control simple robots with low resolution sensing are then analysed using a combination of knowledge of these mathematical properties and careful interpretation of time plots of sensor, neuron and motor activities.

1 Introduction

We have argued elsewhere (Husbands and Harvey (1), Harvey et al (2)), that continuous real-valued artificial neural networks with time delays between units and with unrestricted topologies are a powerful material for building autonomous mobile robot control systems. But their usefulness carries a high price tag: extreme design complexity. In (1) and (2) we explain the reasons behind our belief that the design by hand of such control systems becomes prohibitively difficult as increasingly complex or sophisticated robot behaviours are required, especially in complex, uncertain and dynamic environments. When dealing with such environments, visual sensing is likely to be highly advantageous, but adding visual guidance mechanisms compounds the design problem.

We have attempted to address these issues by developing a new methodology whereby robot network controllers are *evolved* rather than designed. An extended form of genetic algorithm (GA), as described in Harvey (3), allows a genuine form of artificial evolution, where new capabilities are built on older ones and there is a gradual increase in complexity of the control networks as the robot tasks are made incrementally more difficult. This open ended approach requires dynamic length 'genotypes' instead of the usual GA practice of working in predetermined fixed dimension search spaces.

This paper concentrates on qualitative and quantitative analyses of non-standard control networks developed by this method. The networks dealt with are more complex than those generally studied in the ANN community, but a detailed understanding of how they generate robust behaviours is important in the further development of our research. Later it is demonstrated that such an understanding can be provided by a mixture of mathematical analysis and careful interpretation of time plots of sensor, neuron and motor activities. First the evolutionary mechanisms and neuron model must be described.

2 Evolutionary Mechanisms

The artificial evolution approach maintains a population of viable genotypes (chromosomes), coding for control networks, which inter-breeds and mutates according to a selection pressure. This is controlled by a task-oriented evaluation function: the better the robot performs its task the more evolutionarily favoured is its control network. Rather than attempting to hand-design a system to perform a particular task or range of tasks well, the evolutionary approach allows their gradual emergence. There is no need for any assumptions about means to achieve a particular kind of behaviour, as long as this behaviour is directly or implicitly included in the evaluation function. The genotypes are strings of characters coding for neuron properties, the size and topology of the network (neither is restricted), and the properties of inter-neuron connections. The encoding has been designed so that genetic operations (crossover and mutation) on 'parents' always create 'children' coding for legal networks.

All experiments to date have been based on careful simulations of a physical robot constructed at Sussex. The robot is cylindrical in shape with two wheels towards the front and a trailing rear castor. It has front and back bumpers and four whiskers — two front and two back, each at 45° to the central axis of the robot, these tactile sensors give binary on/off signals. Primitive visual sensors were added for some simulation experiments, using ray-tracing techniques to accurately emulate the physics of real vision. The wheels have independent drives allowing turning on the spot and fairly unrestricted movement across a flat floor. The signals to the motor can be represented as a real value in the range $[-1.0, 1.0]$. This range is divided up into five more or less equal segments, and depending on which segment the signal falls into, the wheel will either remain stationary or rotate half/full speed forwards/backwards. The motor signal range is achieved by using two output neurons per motor, and subtracting one's output from the other's. An input unit is assigned to each sensor, but the number of hidden units is not specified. Starting from randomly wired networks, the evolutionary algorithm gradually produces robust noise tolerant controllers. For full details of all the above see (2).

3 Neuron Model

The neuron model employed to date has been designed for its usefulness in control applications rather than for biological plausibility or ease of analysis. Figure 1 represents the operation of a neuron. There are separate channels for excitation and inhibition. Real values in the range [0,1] propagate along excitatory links subject to delays associated with the links. The inhibitory (or veto) channel mechanism works as follows. If the sum of excitatory inputs exceeds a threshold, T_v, the value 1.0 is propagated along any inhibitory output links the unit may have, otherwise a value of 0.0 is propagated. Veto links also have associated delays. Any unit that receives a non zero inhibitory input has its excitatory output reduced to zero (i.e. is vetoed). In the absence of inhibitory input, excitatory outputs are produced by summing all excitatory inputs, adding a quantity of noise, and passing the resulting sum through a simple linear threshold function, $F(x)$, given below. Noise was added to provide further potentially interesting and useful dynamics and to give an indication of the properties of a physical implementation of a unit which would be likely to include naturally occurring noise. The noise was uniformly distributed in the real range [-N,+N].

$$F(x) = \begin{cases} 0, & \text{if } x \leq T_1 \\ \frac{x-T_1}{T_2-T_1}, & \text{if } T_1 < x < T_2 \\ 1, & \text{if } x \geq T_2. \end{cases} \tag{1}$$

In all our work to date the networks have been simulated in software. Their continuous nature is modelled by using a very fine time slice approach. At each time step of the robot kinematics simulation sensor readings are fed into the network, the network is run through a number of cycles (with a variance to counter distorting periodic effects), and then the network outputs are converted into motor signals which allows the calculation of the new position of the robot. Time delays on the connections refer to network cycles, hence a unit delay is much smaller than the duration of a single robot step. In the work described in this paper unit delays are used throughout, although we have also evolved networks where much longer time delays (under genetic control) were possible, Cliff et al (7). The values of the other parameters mentioned above used in our experiments were N=0.1, T_v=0.75, T_1=0.0 and T_2=2.0, giving a slope of 0.5 to F.

4 Generator Units

Before going on to analyse particular controller networks, a commonly occurring subnetwork configuration will be looked at in detail. This will enable a deeper understanding of the full networks described later.

Nearly all the successful evolved networks we have studied involve one or more neurons with three or more excitatory feedback loops. This arrangement allows internal noise to rapidly build up the output from the unit to the upper threshold level without requiring any external input to the network. The unit then acts as a source of constant high output (unless vetoed) which typically contributes to the motor signals giving the basic behaviour of the robot. We

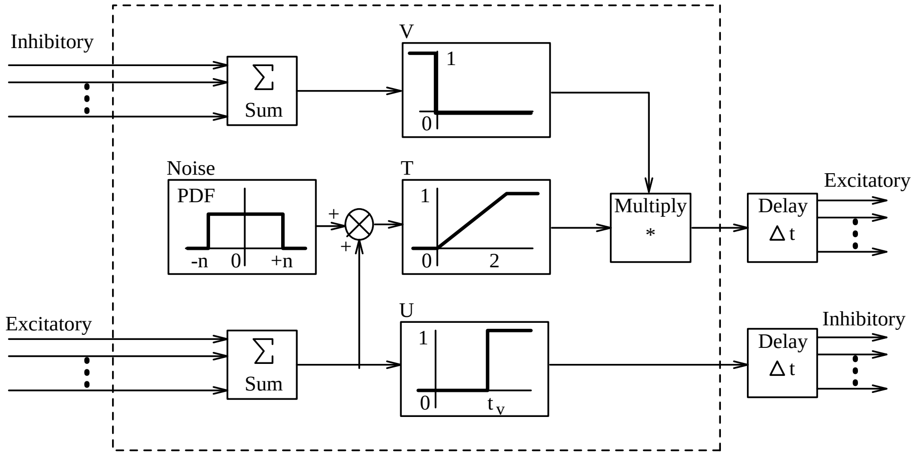

Figure 1: Block diagram of neuron operation.

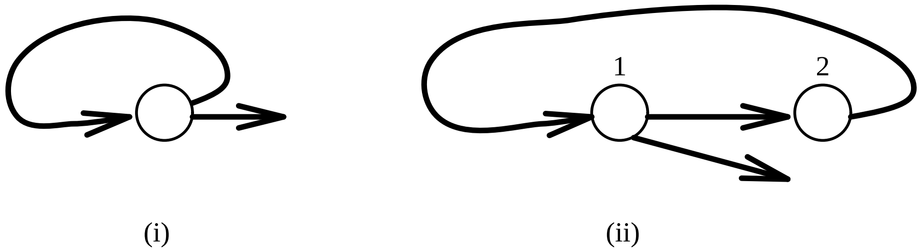

Figure 2: Generator feedback loops.

will refer to such neurons as *generator units* and will now analyse their properties in detail. These are not entirely obvious since internal noise is uniformly distributed with mean zero. In the following all connections have unit time delays, which is the case for all the experiments described in this paper.

Consider the single neuron feeding back onto itself as shown in Figure 2(i). The input at time t, I_t, is related to the output at time $t-1$, O_{t-1}, by $I_t = F(O_{t-1} + n_t)$, where F is the transfer function of equation 1 and n_t is the noise injected by the unit at time t. Ignoring the threshold terms for the moment, and concentrating on the linear part of the function, we can write:

$$O_t = k(O_{t-1} + n_t) \tag{2}$$

where k is the slope of the linear part of F. Hence,

$$O_0 = kn_0 \tag{3}$$

$$O_1 = k(kn_0 + n_1) \tag{4}$$

$$\cdots O_t = k^{t+1}n_0 + k^t n_1 + \cdots + kn_t \tag{5}$$

$$= \sum_{i=0}^{t} k^{t+1-i} n_i = \sum_{i=0}^{t} a_i \tag{6}$$

Now the noise terms, n_i, can be considered as independent random variables uniformly distributed over the real range [-N,+N], giving them probability distribution function $P(x) =$

$1/2N$. Using elementary probability theory, it can be shown that they each have mean $\mu_{n_i} = 0$ and variance $\sigma^2_{n_i} = N^2/3$ (Meyer (4)). The linear relationships between the a_is and n_is means that the a_is can be regarded as independent random variables with mean $\mu_{a_i} = 0$ and variance $\sigma^2_{a_i} = k^{t+1-i}N^2/3$. In other words the output of the unit can be regarded as the sum of a series of independent random variables. To return to the temporarily ignored threshold conditions of F, because O_t can never become negative (lower threshold condition), we can use a version of the central limit theorem to guarantee that the distribution of the random variable Z given below,

$$Z = \frac{O_t - \sum_{i=0}^{t} \mu_{a_i}}{\sqrt{\sum_{i=0}^{t} \sigma^2_{a_i}}} \tag{7}$$

is closely approximated by the *positive* half of the standardized normal distribution $\mathcal{N}(0,1)$. Now, $\sum_{i=0}^{t} \mu_{a_i} = 0$ and $\sum_{i=0}^{t} \sigma^2_{a_i} = \sum_{i=0}^{t} k^{t+1-i}\frac{N^2}{3} = \frac{N^2}{3} \times \frac{k-k^{t+2}}{1-k}$, hence:

$$Z = \frac{O_t}{\frac{N}{\sqrt{3}} \times \sqrt{\frac{k-k^{t+2}}{1-k}}} \tag{8}$$

This result can be used to estimate the expected time for O_t to reach the upper threshold value of 1.0. Clearly this depends on the value of k. The larger k, the shorter the time. From Equation 2 and using the fact that the maximum negative amount of noise is -N, once the upper threshold has been reached it can only be maintained if $k(1-N) > 1$, i.e. $k > 1/(1-N)$. Setting N=0.1, the value used in all our experiments to date, then $k > 1.11$. Given that we have used a fixed slope of 0.5 for the neuron transfer function, k can be *effectively* altered by changing the number of feedback loops. With three loops $k = 1.5$, and the probability of O_t reaching 1.0 by $t = 40$ is given by $2(1 - \Phi(0.0025))$, from the formula for Z, and where Φ is the cumulative distribution function of $\mathcal{N}(0,1)$. The value of this expression is greater than 0.999.

Hence we can conclude that with three feedback loops, and in the absence of other inputs, a neuron will easily reach the saturation condition within 40 network cycles, the minimum possible per *single* robot step, and therefore will act as a generator unit. This is exactly what is observed as will be seen in the next section.

Surprisingly, the two unit mutual feedback loop shown in Figure 2(ii) has properties equivalent to the single unit loop. Using the same notation as above, $O_{1_t} = k(O_{2_{t-1}} + n_{1_t})$, with a symmetric expression holding for O_{2_t}. Hence,

$$O_{1_t} = k^{t+1}n_{1_0} + k^t n_{2_1} + \cdots + k^2 n_{2_{t-1}} + k n_{1_t} \tag{9}$$

Since n_1 and n_2 have exactly the same distribution this expression is identical to that in Equation 5 and the same analysis holds. Indeed, the interplay between the delays on the connections and the noise injected at each unit, means that a loop containing any number of units will have the same behaviour; any of its neurons can act as generator units.

5 Experimental Setup

We started our research program by successfully evolving networks to generate obstacle avoidance behaviours in cluttered environments (1), but recently we have concentrated on uncluttered minimalist environments to allow a very close focus on the neural basis of simple robot behaviours. However, these environments are still noisy as are the (realistically) simulated robot kinematics. The results included in this paper are derived from experiments in which the robot moves in a cylindrical arena (see Figure 4). Populations of size 100 were evolved for 100 generations under the following evaluation function:

$$\mathcal{E} = \sum_{\forall t} \exp(-s|\mathbf{r}(t)|^2) \tag{10}$$

Where $\mathbf{r}(t)$ is the 2D vector from the robot's position at time t to the centre of the arena, and $\forall t$ denotes the duration of the evaluation run. $\mathcal{E}$ drops off very rapidly the more time the robots spend near the walls. On each evaluation run the robot was started in a random position close to the wall. This implicit evaluation function was chosen to study, with different sensor configurations, how well artificial evolution produced networks to keep the robots as far away from the walls as possible for as long as possible.

6 Network Controller Analysis

Figure 3 shows a typical network controller that evolved after about 20 generations under fitness function $\mathcal{E}$, and with unit time delays on all links. In this case the robot only had the tactile sensors mentioned earlier. Figure 4 shows the behaviour generated by this network. After eliminating those units with no output links and by using time plots of sensor, neuron and motor activities to further eliminate units which play no role in this behaviour, it is a straightforward matter to arrive at the reduced network shown in Figure 5 as the essential behaviour generator. Its operation can be readily understood. In the absence of sensory input the right motor is jammed on full forwards due to high activity in unit 12 which is a generator unit as explained above. The left motor maintains half speed forwards by virtue of an input of value 1.0 from unit 12. The neuron transfer function halves this signal, and noise can never be sufficient to shift the signal out of the half forwards range. Hence the basic behaviour is to move in a wide circle. This is a reasonable strategy to score on the $\mathcal{E}$ fitness function. It can be seen that the robot moves sharply away from the wall on contact, which again is a useful strategy given the fitness function. For the behaviour shown, triggering of the back bumper or front right whisker produces the tight turn away from the wall. For the sake of simplicity only those parts of the circuitry involving these sensors are shown; other similar circuits are present for other sensors to handle collisions from different angles. If either of the two sensors mentioned are activated unit 1 inhibits unit 10 while activity is propagated to unit 11. Hence the left motor switches from half forwards to full reverse. At the same time, input to unit 9 goes high causing the inhibition of unit 12. This switches the right motor off and the tight turn is executed. Once there is no further sensory input the network returns to its previous mode with unit 12 acting as a generator and circling behaviour is resumed.

Figure 6 shows the behaviour generated by the fittest network in the population after about

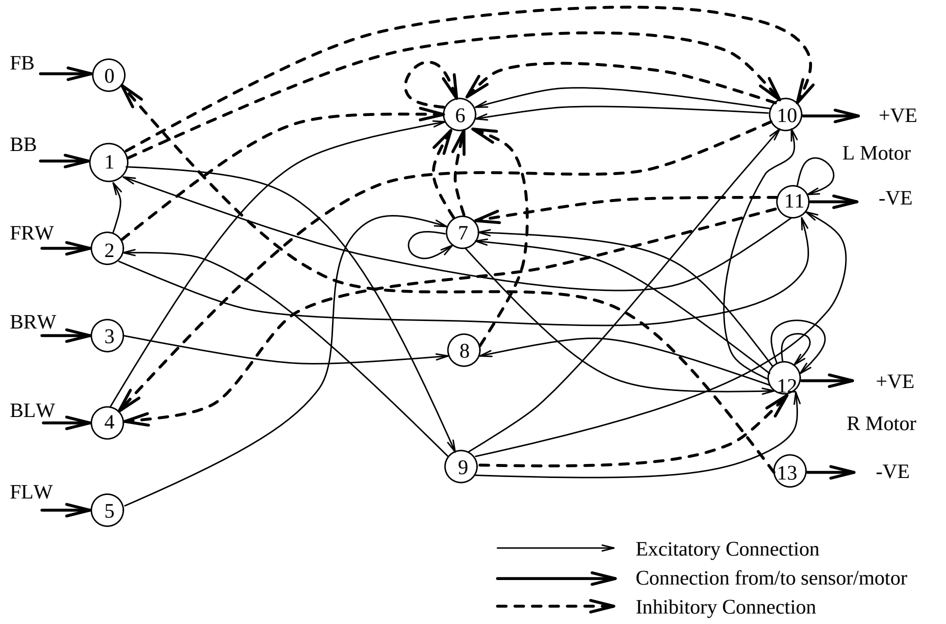

Figure 3: Fittest network appearing in early generations. Dashed connections are veto, solid lines are excitatory links. BB=back bumper, FRW=front right whisker and similar for other sensory inputs.

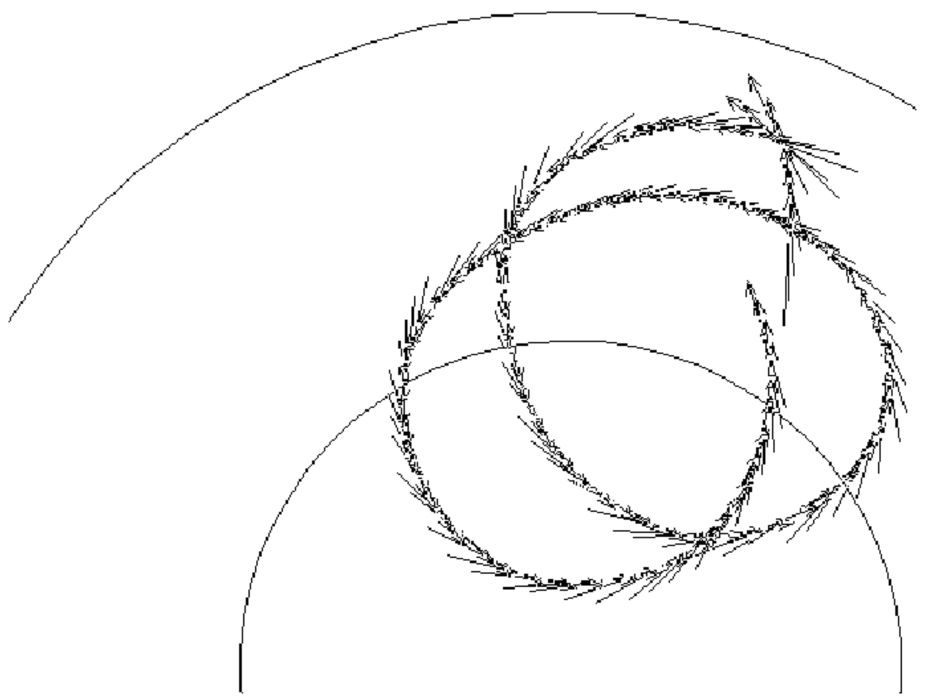

Figure 4: Behaviour generated over a finite time by network controller shown in Figure 3. Arrows indicate robot orientation. Arrow length = robot diameter. The inner circle indicates the higher scoring region under $\mathcal{E}$.

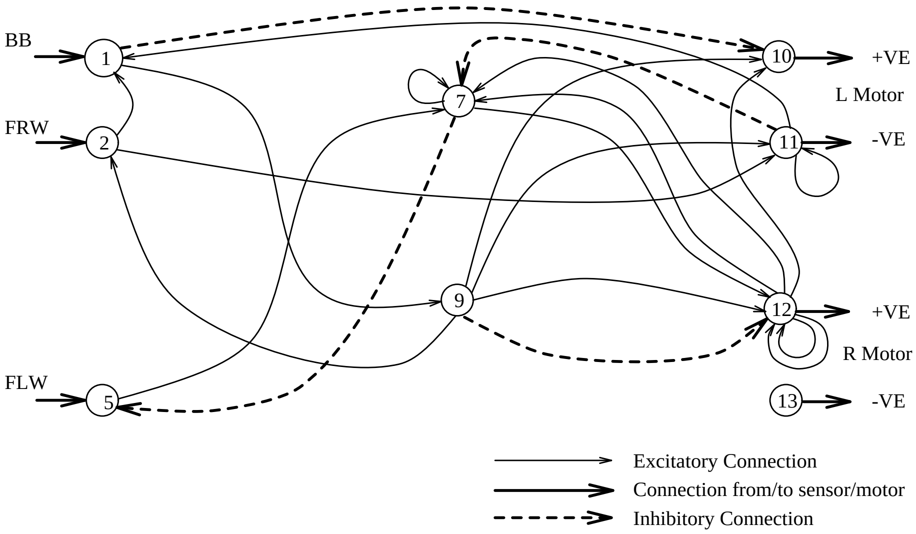

Figure 5: Reduced network which is essentially responsible for generating the behaviour shown.

70 generations. It can be seen that the path is wider than for the earlier behaviours and is able to reliably score higher on $\mathcal{E}$. The path is achieved by oscillating between straight line motion and the previous circling behaviour. This is caused by the oscillating right motor signal as can be seen from the activity plots shown in Figure 7.

Analysis of the full network reveals that the circuits for turning away from the walls on contact are still present but the reduced network responsible for the wide path behaviour is now as shown in Figure 8. Unit 12 is no longer a generator unit but unit 8 essentially is, being involved in two mutual feedback loops with unit 12 and one with unit 10, as well as having inputs from other parts of the network.

High input from unit 8 keeps the output of unit 10 at about 0.5, clamping the left motor on half speed forwards. Remembering that a signal is halved (modulo noise) each time it passes through a unit, the input from 5 to 12 is about 0.25 and from 6 about 0.1. Add these to the input of 1.0 from 8, and the output from unit 12 should be about 0.67 with finite amounts of noise keeping the the value on a random walk between strict upper and lower limits of 0.79 and 0.55 (it does not have sufficient feedback to overcome these). The motor signal boundary value between half and full forwards is 0.67. Hence evolution has produced a tightly tuned oscillator by making use of internal noise. Quite a remarkable result and a difficult design to have achieved by hand.

6.1 Evolving visually guided robots

Cliff et al (5) describes in detail experiments in which primitive visual sensors were concurrently evolved with the control networks. Figure 9 shows the behaviour of a robot with two very simple 'eyes' evolved under the same evaluation function $\mathcal{E}$. As well as the network, the position and angle of acceptance of the eyes is under evolutionary control. The walls of the arena are black while the floor and ceiling are both white. This results in different visual in-

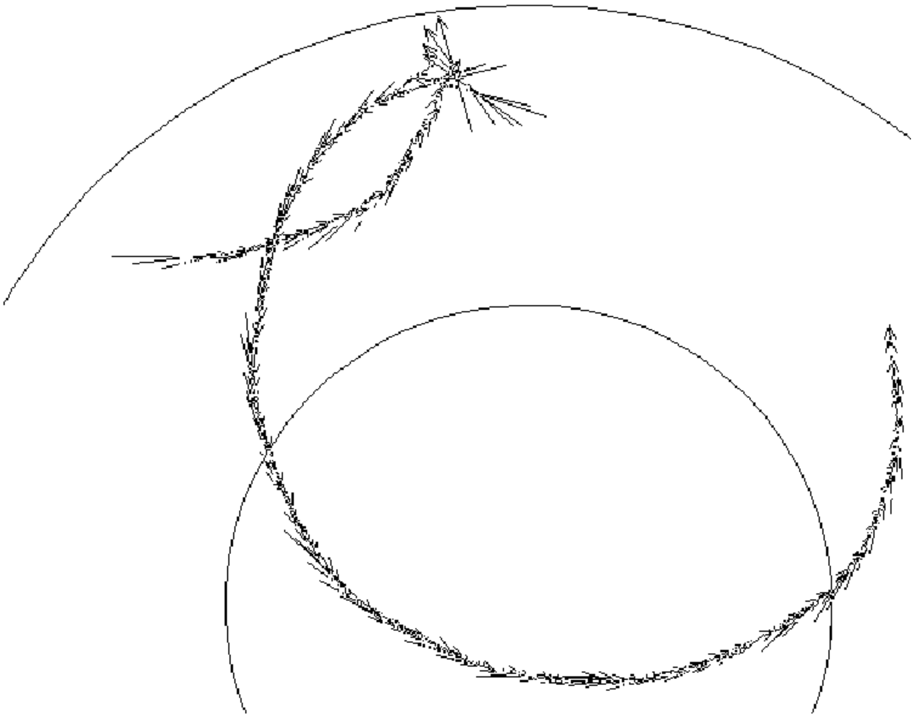

Figure 6: Fittest behaviour found in later generations. Display format same as in Figure 4

puts for different positions in the arena. Clearly the control system has evolved to make use of regularities in the environment to score very highly on the difficult implicit fitness function $\mathcal{E}$. This is a significant result: even though $\mathcal{E}$ does not explicitly involve monitoring visual input, the robot evolves to perform visually guided behaviours. The robot starts off near the wall where visual input is very low, motor signals are then controlled by visual input to move the robot in an almost straight line to the centre where it circles for the rest of the evaluation run. This is almost optimal behaviour and is possible because there is a correlation between visual input and position that evolution has been able to exploit. A detailed analysis of networks with visual input can be found in Cliff et al (6).

As the complexity of the environment increases, accurately simulating even low-resolution vision requires sufficiently large computational resources that, eventually, simulation has to be abandoned in favour of using real robots. We are currently constructing specialised visuo-robotic equipment which entirely eliminates the need for simulation: see (5) for further details.

7 Conclusions

This paper has outlined a framework in which we are using artificial evolution to develop arbitrarily recurrent continuous dynamical neural networks to generate robust behaviour in autonomous mobile robots. It has been shown how evolved networks, although at first sight rather complex, can be understood with a mixture of qualitative and quantitative analysis.

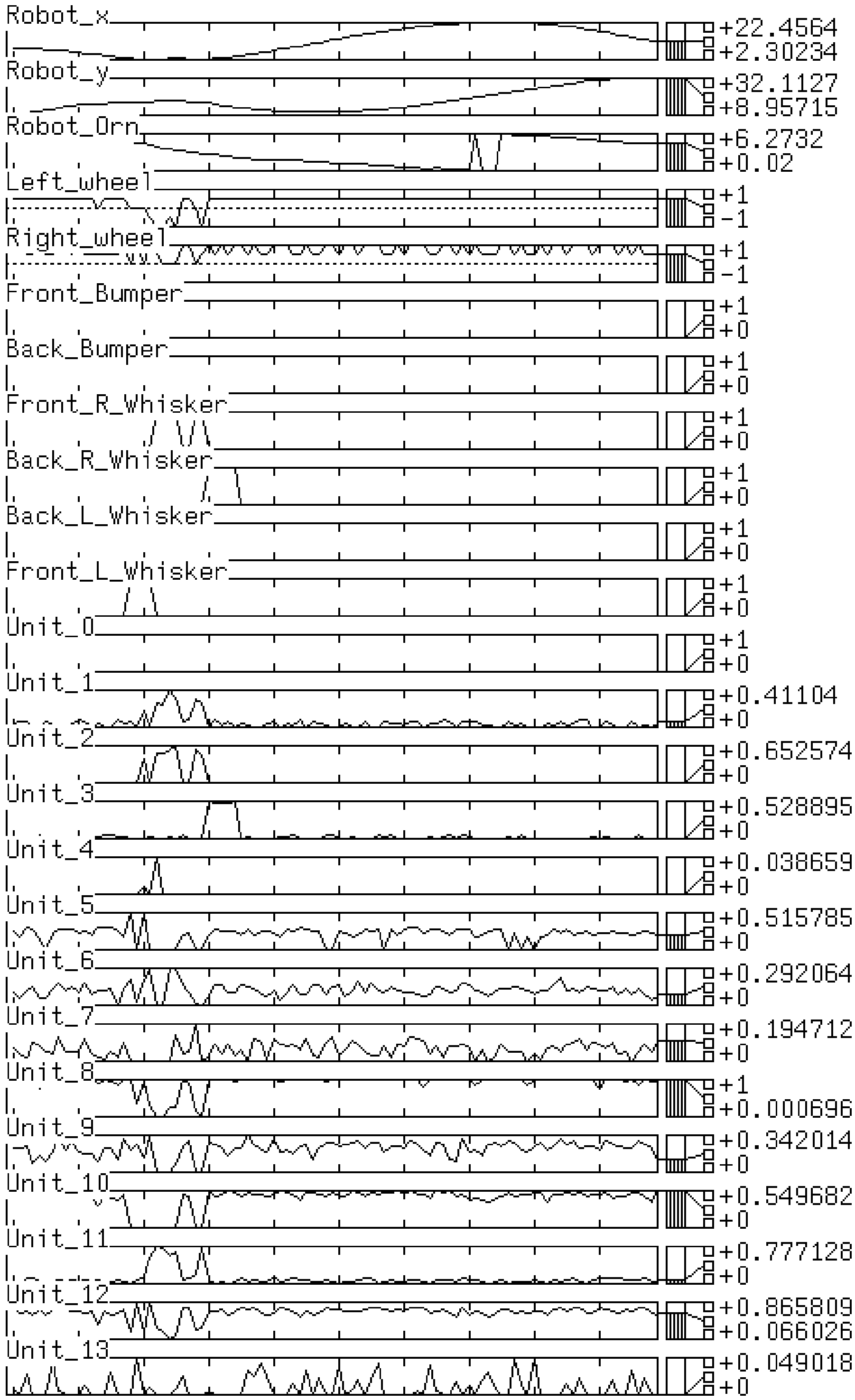

Figure 7: Activity plots for behaviour shown in figure 6

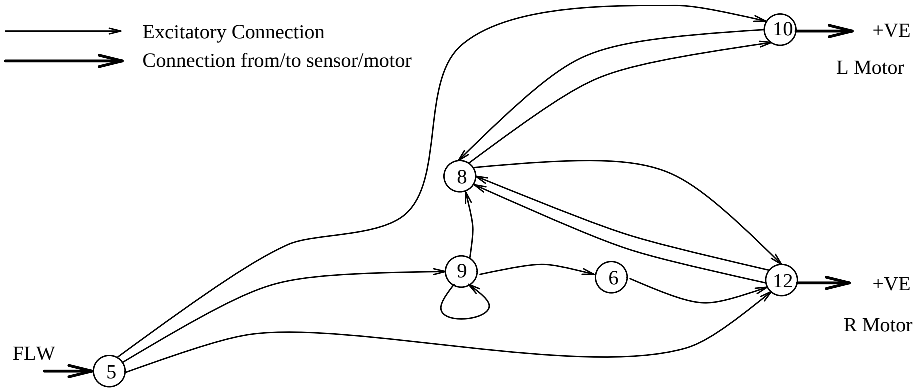

Figure 8: Reduced network responsible for generating behaviour shown in Figure 6

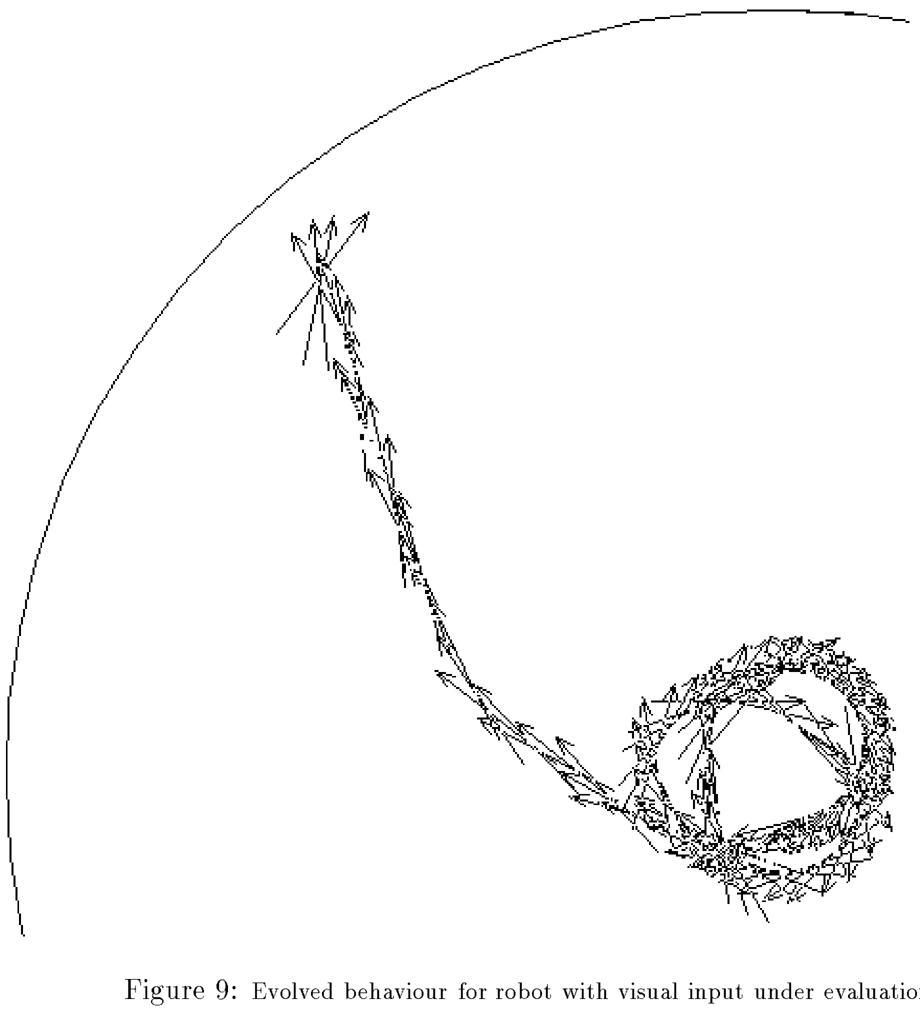

Figure 9: Evolved behaviour for robot with visual input under evaluation function $\mathcal{E}$.

References

1. Husbands, P. and Harvey, I., 1992, *Proc. 3rd Annual Conf. on Artificial Intelligence, Simulation and Planning in High Autonomy Systems*, IEEE Computer Society Press, 1992, 139-146.

2. Harvey, I., Husbands, P. and Cliff, D., 1993, Issues in evolutionary robotics. *Proc. 2nd Int. Conf. Simulation Adaptive Behaviour*, Meyer, J., Roitblat, H. and Wilson, S. (Eds.), MIT Press, 1993.

3. Harvey, I., 1992, Species Adaptation Genetic Algorithms: A Basis for a Continuing SAGA,*Toward a Practice of Autonomous Systems, Proc. 1st Euro. Conf. Artificial Life*, F.J. Varela and P. Bourgine (Eds.), MIT Press, 1992, 346-354.

4. Meyer, P., 1970, *Introductory probability and statistical applications*, Addison Wesley.

5. Cliff, D., Husbands, P. and Harvey, I., 1993, Evolving visually guided robots. *Proc. 2nd Int. Conf. Simulation Adaptive Behaviour*, Meyer, J., Roitblat, H. and Wilson, S. (Eds.), MIT Press, 1993.

6. Cliff, D., Husbands, P. and Harvey, I., 1993, Analysis of evolved sensory-motor controllers.*CSRP-264*, COGS, University of Sussex.
Also to appear in Proceedings of Second European Conference on Artificial Life (ECAL93), Brussels, May 24–26 1993.

7. Cliff, D., Husbands, P. and Harvey, I., 1993, Evolving recurrent dynamical networks for robot control. *Proc. of ANNGA93, Int. Conf. on Artificial Neural Networks and Genetic Algorithms*, Innsbruck, Springer Verlag (in press).