

General visual robot controller networks via artificial evolution*

Dave Cliff and Inman Harvey and Phil Husbands

School of Cognitive and Computing Sciences

University of Sussex, BRIGHTON BN1 9QH, U.K.

davec or inmanh or philh, all @cogs.susx.ac.uk

ABSTRACT

We discuss recent results from our ongoing research concerning the application of artificial evolution techniques (i.e. an extended form of genetic algorithm) to the problem of developing “neural” network controllers for visually guided robots. The robot is a small autonomous vehicle with extremely low-resolution vision, employing visual sensors which could readily be constructed from discrete analog components. In addition to visual sensing, the robot is equipped with a small number of mechanical tactile sensors. Activity from the sensors is fed to a recurrent dynamical artificial “neural” network, which acts as the robot controller, providing signals to motors governing the robot’s motion.

Rather than *designing* the control networks, we use a genetic algorithm which operates on encoded controller architectures. The controller architecture specifies the network connectivity, the number of “neural” processing units in the network, *and* factors governing the specification of the visual sensors. That is, the control network and the sensing morphology are evolved concurrently. A large number of network designs are randomly generated, and then simulated to evaluate their ability to produce useful behaviours in the robot. After all the designs have been evaluated, the encodings for the more successful architectures are “inter-bred” using techniques inspired by biological studies of evolution via mutation and recombination; thereby producing a new collection of network designs. If this process is repeated for a sufficient number of iterations, useful network architectures can emerge.

Prior to presentation of new results, this paper summarizes our rationale and past work, which has demonstrated that visually-guided control networks can arise without any explicit specification that visual processing should be employed: the evolutionary process opportunistically makes use of visual information if it is available.

*To appear in: D. Casasent (editor) *Proceedings of the Society of Photo-optical Instrumentation Engineers Conference 1993 (SPIE93), Session on Intelligent Robots and Computer Vision XII: Algorithms and Techniques*, SPIE Conference Volume 2055. Boston MA, 7–10 September 1993.

The new results discussed in this paper concern a move towards evolving *general* controllers: in our earlier work, each control architecture was evaluated in the same environment (modulo significant noise injected at a number of levels). Over a number of evolutionary iterations, the control architectures were seen to have adapted to be dependent on the particular visual characteristics of the environment. This paper describes results from recent experiments where the visual environment is varied significantly during the evolutionary process, with the aim of evolving controllers which can operate in a wide range of environments. The paper also includes discussion of our current work on transferring from simulation studies to real robots, constructed to eliminate any dependency on simulated sensing.

1 Introduction

Recently, there has been increasing activity within the artificial intelligence (AI) research community directed toward the development of complete, embodied, autonomous, agents: i.e. mobile robots. This represents a shift away from traditional assumptions in AI [2, 1], towards a new approach which attaches central importance to:

- Embodiment. It is claimed that interactions between an agent and its environment are of extreme importance in understanding natural intelligence and in developing artificial forms.
- The generation of adaptive behaviours in embodied agents. This is seen as the prime role of animal sensory-motor systems and of great importance in useful artificial systems.
- The bottom-up development of entire artificial ‘creatures’. The previous two items point to the study of complete behaviour generating systems for autonomous agents acting in realistically complex and uncertain environments. Because entire systems are involved, a gradual movement from the simple to the more sophisticated is deemed pragmatic. Many studies with these general characteristics have asked awkward questions about

the fundamental need for reasoning and representation within intelligent systems.

Further to this aim, we are attempting to develop highly automated techniques for the generation of specifications of ‘cognitive’ control architectures for simple visually guided robots, where ‘control architecture’ is taken to *include* specifications for the sensors (and, in principle, the actuators) of the robot. We view automation as necessary because the types of control architecture required are likely to be highly complex, with many (often indirect) interactions between constituent parts, and consequently the complexity of purely manual design of such architectures is likely to scale badly as further layers or modules are added to the architecture. The situation is analogous to (but *not* identical to) the attempted solution of complex combinatorial optimisation problems by hand: many years ago it was accepted that automatic aids were required in that field, which lead to developments in the computationally intensive area of Operations Research.

For reasons given in [10, 5], we believe that truly autonomous mobile robots will require visual processing capabilities, and that automatic techniques for the design of autonomous-agent cognitive control architectures should be based on the use of artificial evolution (i.e. a form of genetic algorithm), to develop parallel distributed processing systems (i.e. “neural networks”) which are capable of coordinating sensory-motor activity in autonomous agents so as to exhibit desired adaptive behaviours. Briefly, the rationale for our approach is as follows:

- For a mobile robot to achieve a high degree of autonomy, appropriate distal (as opposed to proximal) sensory information will be required, e.g. to enable successful navigation in unfamiliar complex cluttered dynamic environments. Vision is the primary sensory modality for the capture of distal sensory information used in animals, and studies of biological vision may well yield insight into the creation of artificial seeing systems. The passive nature of most forms of visual sensors also bring energy economies which are advantageous, as truly autonomous robots will have to carry their own power sources in a self-contained fashion.
- If visual sensing is to be used as the basis of sensory-motor control, high-bandwidth processing channels will eventually be required, and speed considerations require such processing to be massively parallel. Thus, parallel distributed processing (PDP) techniques are a necessity if complex adaptive behaviours are to be exhibited by robots operating in real time. The use of PDP techniques may also offer graceful degradation in the presence of noise and component failure. In accordance

with common terminology, we refer to our PDP controllers as ‘neural’ networks, but we appreciate that the link with biological nervous systems is more metaphorical than actual. However, in contrast with much of the current literature, we do not necessarily consider neural networks (biological *or* artificial) as purely computational devices. Interaction with dynamic environments forces consideration of the dynamics of the networks themselves, and we find it profitable to consider the robot’s sensory-motor controller and the environment within which the robot is situated as a pair of coupled dynamical systems. This issue is discussed in more detail later on in this paper.

- Theoretical studies, and a growing body of empirical evidence, indicate that artificial evolution techniques can be highly effective in solving complex optimisation problems. While we do not fully ascribe to the view of evolution as optimisation (an animal is not a ‘solution’ to a ‘problem’ posed four billion years ago), we believe that, *if suitably extended*, genetic algorithms (GA’s) are the most promising development path. The particular form of genetic algorithm used in our work is described further below.

In order to evaluate our proposals, we have performed a number of exploratory experiments involving simulation studies of a simple visually guided mobile robot, evolved to exhibit elementary behaviours in a simple environment. The simulated robot is an accurate model of a real robot. The primary aim of these experiments was to apply our proposals to a realistic system which was sufficiently complex that it presented genuine challenges, but sufficiently simple that evaluation and analysis of the final evolved systems could be performed without unreasonable difficulty. Therefore it is important to appreciate that we only view the results in this paper as a promising beginning: the evolved controllers discussed here generate relatively mundane behaviours, but we are working *incrementally* towards more advanced behaviours, from the bottom up. As the results in this paper should demonstrate, despite the minimal nature of the experiments, the lessons learned are non-trivial.

The rest of this paper is organised as follows: Section 2 gives an overview of the robot simulator; Section 3 describes the form of neural network model used; Section 4 discusses the particular genetic algorithm employed; Section 5 summarises the experimental methods we used in evolving the particular controllers described here; Section 6 then gives a summary of some earlier results, followed by a discussion of a controller which operates in a variety of environments. Finally Section 7 discusses work currently in progress, where

special-purpose visuo-robotic equipment has been designed and built to avoid the need to simulate either sensors or actuators.

2 The Autonomous Robot Simulator

Our simulation studies are based on a careful simulation of a real robot, built in the School of Engineering at the University of Sussex. The body of the robot is cylindrical, with the cylinder axis oriented vertically. It has two independent drive wheels mounted left and right, and a trailing rear freewheel castor which gives tripod stability. In principle, the robot can travel in straight lines or in arcs of varying radii; the minimum radius is sufficiently small that the robot gives the appearance of spinning “on the spot”.

The robot does not have a fixed control architecture: there is elementary interfacing circuitry for its sensors and motors, but the interfaces can either be linked to custom-built control circuits, or via analogue/digital and digital/analogue converters to a notebook PC mounted on the top surface of the body, which can be programmed to simulate neural-network controllers. Its basic sensors consist of a number of one-bit tactile sensors mounted around the curved surface of its body. The tactile sensors are either “bumper-bars” over an arc of the robot’s circumference, or radially-oriented “whiskers”. The simulation uses fine-time-slice techniques to approximate the continuous nature of the real system. Standard Newtonian mechanics are used to simulate the motion of the robot, but noise is injected to prevent the motion from being wholly deterministic. Collisions of the robot or its tactile sensors with its surroundings are modelled as accurately as possible, using observations of the real system: the real robot’s mass to maximum speed ratio is such that collisions are inelastic, involving fairly simple rotational movements which depend on the speed and angle of incidence as well as the shape of the obstacle.

In our simulations, we modelled the robot configured with six tactile sensors: two bumpers (one front, one rear), and four radially symmetric whiskers oriented at angles of $\pm\pi/4$ and $\pm3\pi/4$ to the longitudinal midline of the robot. The real robot has been equipped with a CCD camera for certain applications, but in our work we decided to simulate very simple discrete photoreceptor units, which could reasonably be built using discrete components (photodiodes or LDR’s), with the addition of lenses to give particular acceptance-angles to each photoreceptor. The visual sampling of the photoreceptors was simulated using ray-tracing [6], with 4×4 regular supersampling on each pixel as an antialiasing measure, so each simulated photoreceptor’s output was always one of 17 discrete values in the range $[0, 1]$. The simulation assumed a constant and relatively triv-

ial lighting model, thereby avoiding the need to simulate nonlinear alterations in response dependent on wavelength, intensity, or temperature.

The aim was to start with the most minimal visual guidance capabilities, and proceed towards more complex systems. To this end, we worked with *just two* simulated photoreceptors (i.e. two pixels), mounted on the robot body at equal heights above the floor and symmetrically about the robot’s longitudinal midline. Given only two pixels, it was necessary to constrain the simulated robot’s environment and desired tasks in order for the experiments to remain tractable. For this reason, we experimented with a simulated environment where the robot was alone in a closed circular arena. The (planar) floor and ceiling of the arena were white, while the (cylindrical) walls were black. The ray-tracer used uniform diffuse lighting and all surfaces were “perfect” Lambertian, i.e. no surfaces exhibited specularities or irregularities in reflectance characteristics. For further details of the vision and environment simulation, see [5].

The results reported here all involved attempting to evolve a robot controller which would, from any starting point in the arena, guide the robot to the centre of the circular arena and keep it there.

3 The Neural Networks

We have experimented with a variety of types of ‘formal neurons’, which we refer to as *units*. In the work described here, the units draw some inspiration from biology, but were designed with the intention of being relatively easy to fabricate in hardware. Activity values in $[0, 1] \subset \mathbf{R}$ propagate along links between units, where each link has a particular attenuation factor (i.e. weight) and a particular propagation delay associated with it. Each neuron has separate inhibition and excitation channels. The excitatory output of a unit is determined by a compressive nonlinear function of the sum of excitatory inputs to the unit, to which uniformly-distributed noise (with zero mean) is added. However, if a unit receives *any* inhibitory input then its excitatory output is disabled. The inhibitory output of a unit is set by a step function of the sum of that unit’s excitatory inputs, and is not affected by any inhibitory input the unit receives.

The network is connected to the robot’s sensors and actuators by identifying certain units as “input units” (which receive activity from sensors) and certain other units as “output units” (whose activities determine the motor settings). Units not involved in input or output roles are referred to as “hidden units”. All networks considered during evolution are well-formed, insofar as input units for the six tactile sensors and two photoreceptors are always present, and output units connect-

ing to the motors are also always present (each motor requires two output units: the units’ output values are in the range $[0,1]$ but the motors require control signals in the range $[-1,1]$).

While our networks are homogeneous, in the sense that only one type of unit is employed, there is no enforcement of regularities in connectivity: arbitrary connectivities may be employed, and the final evolved networks are invariably asymmetric, with high degrees of recurrency. In principle, evolution could be used to determine the individual thresholds associated with each unit and also for the delays and weights on each connection. However, in the experiments described below, the weights, delays, and thresholds for all units in all networks were all fixed at constant values (i.e. were not under evolutionary control, nor under the control of any learning algorithm). Nevertheless, the use of nonzero delays means that simple circuits can exhibit intrinsic dynamical behaviours; e.g. two units, appropriately connected, can act as a sawtooth oscillator. For further details of the neuron model, and supporting mathematical analysis of significant network structures, see [11].

4 The genetic algorithm

It is beyond the scope of this paper to give a full account of the genetic algorithm employed in our work, but a brief summary is given here. For a good general introduction to genetic algorithms, see [7].

We use Harvey’s SAGA artificial evolution techniques. The general principles underlying SAGA are common to most other genetic algorithms. Structures of interest (e.g. the controller architectures) are encoded as strings of characters drawn from some alphabet. Such a character-string is referred to as a *genotype*. The process starts with the creation of a set number of randomly-generated genotypes. Each individual genotype represents one possible controller architecture, and the collection of genotypes is referred to as the *population*. The process then enters an iterative loop, where each individual in the population is evaluated to determine how ‘fit’ its performance is, according to some objective *fitness function*. Once each individual has been assigned a fitness rating (usually a scalar real number), pairs of individuals are randomly selected (with replacement) to ‘breed’ a new population. In a direct analogue of Darwinian natural selection, the probability of an individual being selected for breeding depends on its fitness relative to the rest of the population, so fitter individuals have a greater chance of passing on their genetic material to subsequent generations. Each iteration cycle of evaluation and breeding is referred to as a *generation*. If parameters governing the fitness function and the breeding

operations are set appropriately, it is well documented that over a number of generations, fitness in the population can improve.

The significant difference between SAGA and more conventional genetic algorithms is that, whereas most genetic algorithms perform stochastic search in a fixed-dimensional parameter space, SAGA allows for the *dimensionality* of the current search-space to vary in the course of evolution. This is achieved by allowing the length of the genotypes to vary: the variation in length is governed by a random process, but if longer genotypes (which correspond with higher-dimensional search-spaces) are found to offer increased fitness, then they are retained by the natural selection process. Of course, in some situations it is feasible that shorter genotypes may confer increased fitness, so the dimensionality of the current search space may be reduced if that is appropriate. This feature of SAGA allows for truly *incremental* approaches to the evolution of control architectures to be explored: as long as the initial random population is of sufficient complexity to generate fitnesses which can be operated on by the selection mechanisms, SAGA may increase or decrease the dimensionality of the current search space in order to maximise fitness accordingly.

In the context of evolving neural networks, this has the implication that the number of links and units in the network need not be prespecified. So long as the initial networks are well-formed in the sense used above, SAGA is able to add or delete links and units in attempting to maximise fitness. For further details of SAGA techniques see e.g. [9, 8].

5 Experimental Methods

In attempting to evolve robot controller morphologies which would lead the robot to the centre of the circular arena from any starting point, we used the following fitness evaluation function:

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

Where $\mathbf{r}(t)$ is the 2-D vector from the robot’s position to the centre of the floor of the circular arena at time t , and $\forall t$ denotes the duration of the evaluation test (the sum is essentially a discrete approximation to a temporal integral). Put most simply, the more time the robots spend at or near the centre of the arena, the higher they are rated. The value s is a scale factor which ensures that the robots collect almost no score if they are near the walls of the arena. The maximum value of $|\mathbf{r}(t)|$ was 18.0 (the radius of the arena (20.0) minus the radius of the robot (2.0)), and $s = 16.0$. The robot’s height is 4.0.

The genotype for each individual controller was composed of two distinct ‘chromosomes’: one was relatively

short (16 bits) and fixed-length, while the other was long (initially about 1000 bits) and variable-length. The shorter chromosome coded for the positions and acceptance angles of the two photoreceptors, while the longer one coded for the control networks. Initially, all the controller networks were well-formed and had either one or two hidden units. In all individuals in the first generation, both chromosomes were randomly initialised.

Each individual controller was evaluated using \mathcal{E} for 100 timesteps, which implies $\mathcal{E} \in (0.0, 100.0]$. However, all the tests were conducted where the robot started at a random orientation and location, with the distribution of locations biased for positions distant from the centre (i.e. close to the walls), so the maximum possible score a controller could yield on any one trial was somewhere between 75 and 85, depending on the noisy interactions between the robot and its environment. Because of this variability in the score, we evaluated each controller 8 times, each with re-randomised positions and orientations, and then took its *worst* score as an indication of its fitness: this is a much more reliable method of generating *truly robust* solutions than taking the average or best score as fitness values.

We applied this evaluation method to eight separate populations, each of size 60, each over 100 generations. As was typical in most of our experiments, approximately 50% of the populations failed to evolve beyond trivial advances on the initial random controllers, while the remaining populations evolved close to optimal behavioural strategies. Under the \mathcal{E} evaluation function, the optimal behaviour is, from a random initial starting position, to move towards the centre of the arena as fast as possible, and when at the centre, stay there. As will be seen, such behaviours were exhibited by the evolved controllers examined in the next section.

6 Results

Of the eight populations evolved under \mathcal{E} with a fixed wall-height of 15.0, the two populations which evolved to give the highest values of \mathcal{E} for their best individual will be considered. The best individual in the top-scoring population is referred to as C1 (controller-1), and the best individual in the second-highest-scoring population is referred to as C2 (controller-2).

As will be seen in the figures, the final evolved network controllers are fairly opaque tangles of connections, but qualitative analysis techniques can be used to eliminate some units and links from consideration. The qualitative techniques we have found most useful have been inspired by methods in the biological field of *neuroethology*. Neuroethology is the study of the neural mechanisms underlying the generation of behaviour in animals: see e.g. [3]. Given that both

real neural mechanisms in animals and artificial “neural” mechanisms in our simulator are both the result of Darwinian evolutionary processes, with the strong constraint of intermediate viability, it is not altogether surprising that similar analytic approaches are fruitful in both fields. Essentially, the qualitative analytic techniques involve identifying, from the network diagram, redundant units or links which may be vestigial, i.e. “evolutionary scaffolding” which contributed to fitness in earlier generations but have now fallen from use. Such redundant units and links can be deleted from the network (so long as tests are then made to ensure that the performance of the network is not altered by the deletion, i.e. to check that the deleted structure really *was* redundant). Following the application of these qualitative reduction techniques, particular processing *pathways* can be identified. Then, recordings of the activity levels of the units involved in a channel can be made while the system is running, and the roles of individual units and links in generating the behaviour is further clarified, to the point where a detailed understanding of the causal mechanisms is established. Key steps in this process for the analysis of C1 are presented in the following batch of figures. For full details of the C1 analysis see [4].

Figure 1 shows typical activity of the C1 controller, which is close to optimal; Figure 2 shows the full network for the C1 controller, note that although the individuals in the initial population had only one or two hidden units, the number of hidden units in this network has increased as a result of increases in genotype length. Figure 3 shows the results of elimination based on qualitative analysis. Figure 4 shows traces of activity values and observable variables for a typical run with the C1 controller; such information can be used to reduce the network to the fundamental visual pathways shown in Figure 5.

The C1 controller is clearly achieving the desired behaviour, and it is important to note here that the behaviour generated by C1 was evolved using *only* the \mathcal{E} function. All that \mathcal{E} specifies is a *desired behaviour*: it does *not* specify that visual input should be used to guide the behaviour. So, the results from C1 are an existence proof that visually guided behaviours can be evolved *without explicit reference to visual processing*: the fact that visual perception was made *available* to the evolutionary process was sufficient; C1 operates entirely on the basis of visual input (i.e. tactile sensing is not employed at all: the robot uses the visual signal to detect when it is on a collision course with the walls and takes evasive action before the tactile sensors contact the walls). Furthermore, the fact that tactile sensing becomes redundant in later generations means that the tactile-input units also become redundant, and as is clearly shown in Figure 5, the evolutionary process

can opportunistically allocate these redundant tactile processing units to act as “hidden” units (higher-order ‘interneurons’) for the visual processing requirements of the system. In this sense, the distinction between input, hidden, and output units which we impose on the networks in the first generation is blurred by evolution in subsequent generations.

On the basis of our qualitative analysis, we initially (and mistakenly!) concluded that both C1 and C2 were identifying the centre of the arena by monitoring the *absolute* light intensity of one or both of the photoreceptors: at the centre, photoreceptors will signal a particular intensity dependent on their acceptance-angles, and the particular height:radius ratio of the arena (which was 15:20) throughout the evolution of C1 and C2. This struck us as an overtly domain-dependent solution, and so we evolved another 8 populations of controllers, where on each trial the height:radius ratio of the arena was chosen randomly from the range 5:20 to 40:20, i.e. the heights were varied over almost one order of magnitude. Once again, the best individual from each of the two highest-scoring populations was selected for detailed analysis. By testing the variable-height controllers over the range of wall heights used during their evolution, we established that they gave responses relatively invariant to wall height, but slightly lower than the average responses of C1 and C2 in their fixed-height worlds. As a control, we then tested C1 and C2 over the same range of wall heights (i.e. [5,40]), and discovered that, while C1’s response rapidly deteriorated as the wall height was reduced below 15, C2 gave a fairly constant response which was *better* than the responses of the controllers evolved under varying-height conditions. This is an unexpected result: the C2 controller, while marginally less effective than C1 in an arena of height 15, was consistently effective across a range of wall heights which had not been used during its evolution.

The key issue here is the strong evolutionary pressure we exert for *robustness* (i.e. the presence of noise at a number of levels, and the fitness evaluation based on worst performance). Put most simply, in order to evolve a *robust* solution, the C2 controller has evolved as a *general* solution.

Typical behaviour for C2 is shown in Figure 6. Figure 7 shows the full C2 network, while Figure 8 shows the final reduced network. In order to further understand the behaviour of C2, it is necessary to go beyond purely qualitative analysis, and employ quantitative techniques. In order to do this, we treat the C2 controller as a dynamical system and apply the conventional approach of identifying a low-dimensional state-space which is convenient for analysis purposes. The state space we have found most useful is referred to as $r\phi$ space. Each photoreceptor has a separate $r\phi$

space: r is the Euclidian distance from the photoreceptor to the centre of the arena; ϕ is the angle made between the radial line from the centre of the arena to the photoreceptor and the forward optical axis of the photoreceptor. Wherever the robot is positioned in the arena, each photoreceptor will be occupying a particular point in $r\phi$ space and will also be signalling a particular intensity value. The intensity value signalled is dependent on the photoreceptor’s angle of acceptance, and on the height-radius ratio of the arena. As the radius is invariant in the experiments discussed here, only the height is significant. Figure 9 shows the $r\phi$ space for C2’s left photoreceptor when the height is 5, and Figure 10 shows the same space, when the height is 15. As can be seen, variations in wall-height have significant effects on the state-space. The state-spaces for the right photoreceptors are identical, modulo a translational shift along the ϕ axis (corresponding to the robot rotating on the spot so as to bring the right photoreceptor to the position previously occupied by the left photoreceptor).

If the $r\phi$ photoreceptor spaces for the left and right photoreceptors are superimposed, the space can be partitioned into distinct areas where the left and right visual inputs take on particular combinations of values, as illustrated in Figure 11. Using purely theoretical techniques, based on steady-state analysis of primary feedback loops in the C2 network, it is possible to calculate a vector field over the partitioned combined state-spaces which indicates the likeliest move the robot will make for any particular input. The predicted field is shown in Figure 12: empirically derived state-space trajectories are illustrated in Figure 13; as can be seen, there is close accordance between the predicted and empirical state spaces, which indicates that our analysis is accurate. The striking feature of the C2 state-space fields is that, at any wall height, there is always at least one point attractor corresponding to near-optimal behaviour. So we can demonstrate both theoretically and empirically that the C2 controller operates successfully in circular arenas of any height. For further details of the analysis, see [12].

7 Working with a Real Robot

The computational costs of performing accurate simulations were such that, using C-code optimised for speed, each run of one population of size 60 over 100 generations, with 8×100 timesteps per individual took a little over 24 hours on a single-user SUN SPARC 2 workstation. This is significantly slower than real-time, but further computational savings could only be made by decreasing the realism of the simulation, which is a step we are not prepared to take. For this reason, we are now working with a purpose-built cartesian

(gantry) robot which has four degrees of freedom, three translational (x, y, z) and one rotational (pan-angle for a CCD camera mounting). Visual sensing is performed by a custom-built frame-grabber, which feeds images at 50Hz to a 66MHz 486DX2 “front end” PC, which is responsible for handling low-level visual, tactile, and motor processing. The “front-end” machine feeds visual and tactile sensory information to a 33MHz 486DX PC, where the genetic algorithm runs and the control networks are simulated; this machine sends control signals back to the front-end machine, thereby completing the feedback loop. For a schematic diagram of the original design of this robot, see [5]. This robotic equipment allows for the study of the evolution of advanced visually mediated behaviours, such as visually guided navigation. This equipment is in the final test phases at the moment, and will very shortly be used to apply our methods, so far tested only in simulation, to real sensory-motor hardware.

8 Conclusion

We have summarised our rationale for applying artificial evolution techniques to the problem of designing control architectures which generate visually guided behaviours in autonomous mobile robots. Despite the mundane nature of the evolved behaviours, our approach of treating the sensors (and actuators) as part of the control architecture, and evolving them concurrently with the the ‘neural’ processing networks, combined with the use of an evolutionary scheme which does not involve predetermined bounds on the dimensionality of the search space, appears to have yielded designs of genuine interest. We have demonstrated that SAGA-based artificial evolution can opportunistically identify the appropriate sensor modalities for the task at hand, and reconfigure the processing networks if necessary in order to take advantage of vestigial or redundant aspects of the network. Furthermore we have demonstrated that *understanding* the evolved networks is a task requiring both qualitative and quantitative analysis, and that evolution can produce solutions whose true sophistication may only be revealed when they are tested in environments or domains which differ significantly from those used during evolutionary development. While the results from these simulation studies are very promising, the final test of our ideas and methods (working with real hardware) is essential, and is currently underway.

9 REFERENCES

- [1] R. A. Brooks. Intelligence without reason. In *Proceedings IJCAI-91*, 1991.
- [2] R. A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47:139–159, 1991.
- [3] J. M. Camhi. *Neuroethology: Nerve Cells and the Natural Behaviour of Animals*. Sinauer Associates Inc., Sunderland, Mass., 1984.
- [4] D. T. Cliff, I. Harvey, and P. Husbands. Visual sensory-motor networks without design: Evolving visually guided robots, 1993. *Proc. Mechanical Computer Systems that Perceive and Act (MCPA93)*.
- [5] 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)*, pages 374–383. MIT Press Bradford Books, Cambridge, MA, 1993. Also available as University of Sussex School of Cognitive and Computing Sciences Technical Report CSRP220.
- [6] A. S. Glassner, editor. *An Introduction to Ray Tracing*. Academic Press, London, 1989.
- [7] D. E. Goldberg. *Genetic Algorithms*. Addison Wesley, 1989.
- [8] I. Harvey. The SAGA cross: the mechanics of crossover for variable-length genetic algorithms. In R. Männer and B. Manderick, editors, *Parallel Problem Solving from Nature, 2*, pages 269–278. North-Holland, 1992. Also available as University of Sussex School of Cognitive and Computing Sciences Technical Report CSRP223.
- [9] I. Harvey. Species adaptation genetic algorithms: A basis for a continuing SAGA. In F.J. Varela and P. Bourguine, editors, *Towards a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life (ECAL91)*, pages 346–354. M.I.T. Press — Bradford Books, Cambridge MA, 1992. Also available as University of Sussex School of Cognitive and Computing Sciences Technical Report CSRP221.
- [10] I. Harvey, P. Husbands, and D. Cliff. Issues in evolutionary robotics. In J.-A. Meyer, H. Roitblat, and S. Wilson, editors, *Proceedings of the Second International Conference on Simulation of Adaptive Behaviour (SAB92)*, pages 364–373. M.I.T. Press — Bradford Books, Cambridge MA, 1993. Also available as University of Sussex School of Cognitive and Computing Sciences Technical Report CSRP219.
- [11] P. Husbands, I. Harvey, and D. T. Cliff. Analysing recurrent dynamical networks evolved for robot

control, 1993. Also available as University of Sussex School of Cognitive and Computing Sciences Technical Report CSR265.

- [12] P. Husbands, I. Harvey, and D. T. Cliff. Circle in the round: State space attractors for sighted robots, 1993. Submitted.

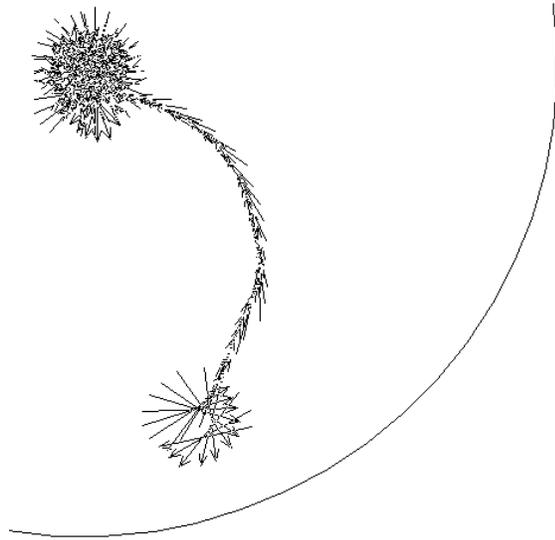


Figure 1: Typical behaviour of the C1 controller. The robot's position at each timestep is shown by an arrow; the midpoint of the arrow 'shaft' is the centre of the robot, and the length of the shaft is the same as the robot's diameter. The robot starts near the edge of the arena, moves to the centre, and then spins on the spot. The 'tip' of the arrow shows the 'front' of the robot, which is not necessarily the direction of travel: although in this case the robot is moving forwards, it can travel in reverse.

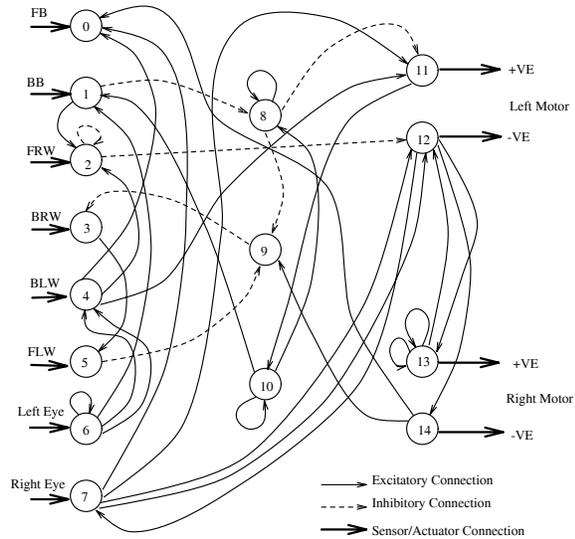


Figure 2: Full C1 control network. The left-hand column are units originally designated as input units: FB=Front Bumper; BB=Back Bumper; FRW=Front Right Whisker; BRW=Back Right Whisker; BLW=Back Left Whisker; FLW=Front Left Whisker. Right-hand column shows output units, which are paired and differenced to give two motor signals in the range [-1,1] from four 'neuron' outputs in the range [0,1]. Centre column shows 'hidden units'.

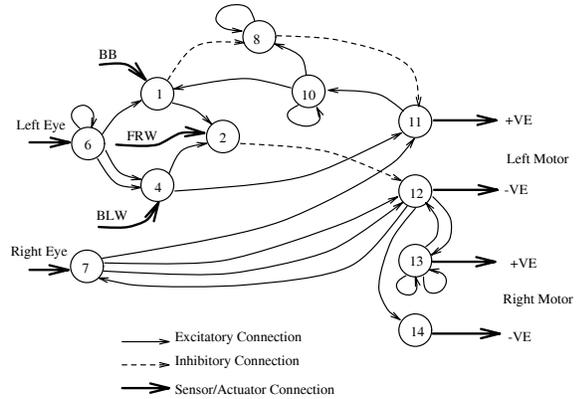
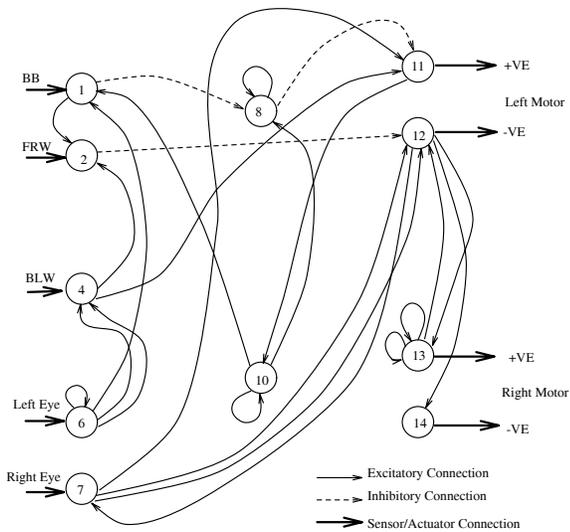


Figure 3: Network with redundant and non-visual units deleted; see text for further details.

Figure 5: Final C1 network. Note that unit 2, originally a tactile input unit, is now a second-order visual processing unit.

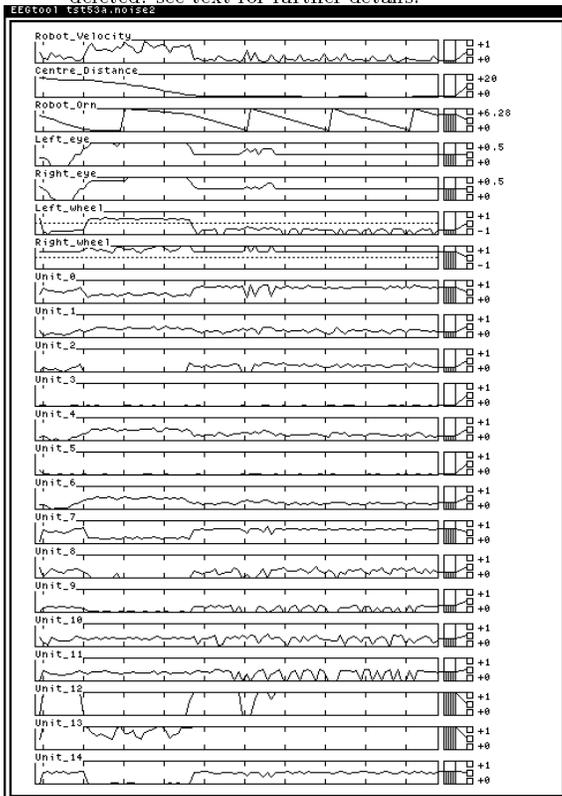


Figure 4: Record of observables and activity levels for the activity illustrated in Figure 1. Horizontal axis is time. From top: robot's velocity; robot's orientation; visual input to left photoreceptor; visual input to right photoreceptor; output of left wheel; output of right wheel; activity levels in the control network units 0 to 14.

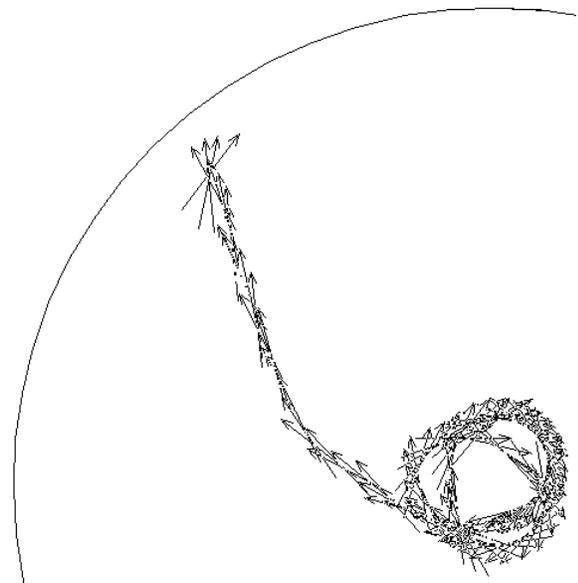


Figure 6: Typical behaviour of the C2 controller, with noise. Display format as for Figure 1. The robot starts near the edge of the arena, moves to the centre, and then around the centre in a low-radius turn. As can be seen, the C2 controller drives the robot in reverse (backwards).

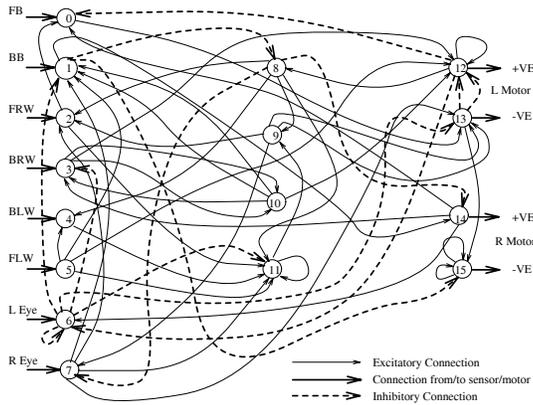


Figure 7: Full C2 control network. Display format as for Figure 2.

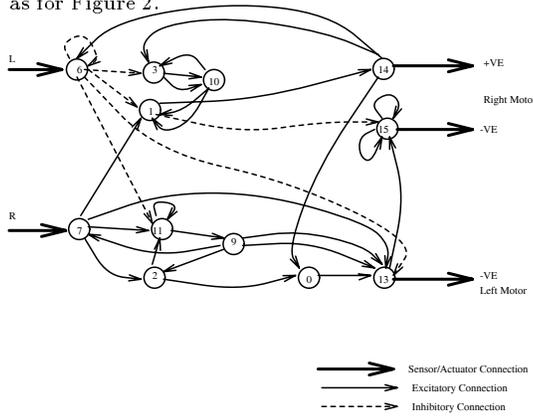


Figure 8: C2 visual guidance pathways. Note that, for the sake of clarity, the positions of the left and right motor outputs have been interchanged.

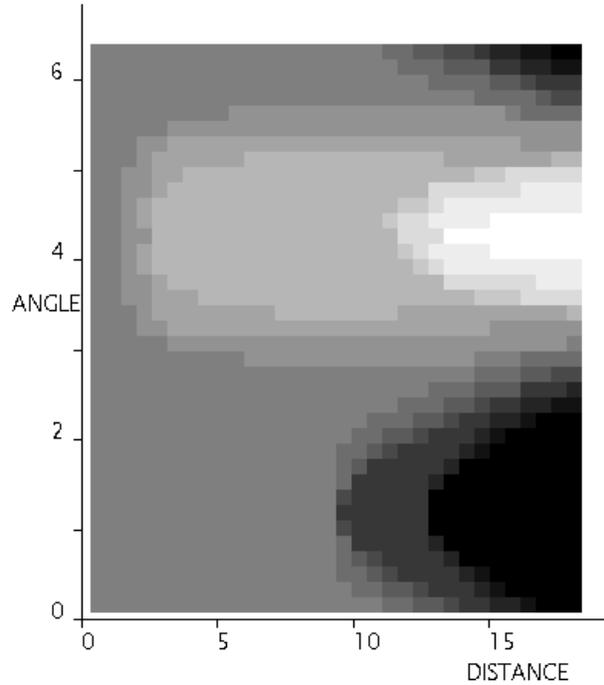


Figure 9: C2 Left photoreceptor $r\phi$ space when wall-height=5; intensity levels shown 0...1 (black-white). 'ANGLE' axis is ϕ ; 'DISTANCE' axis is r .

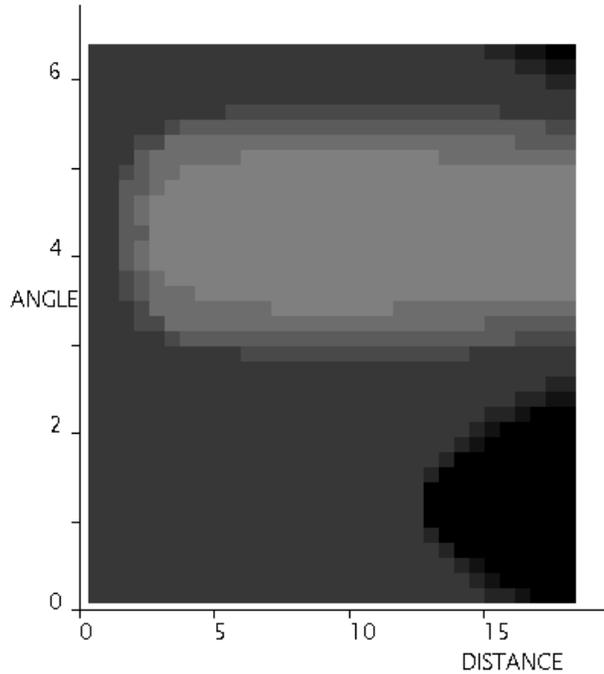


Figure 10: C2 left photoreceptor $r\phi$ space when wall-height=15.

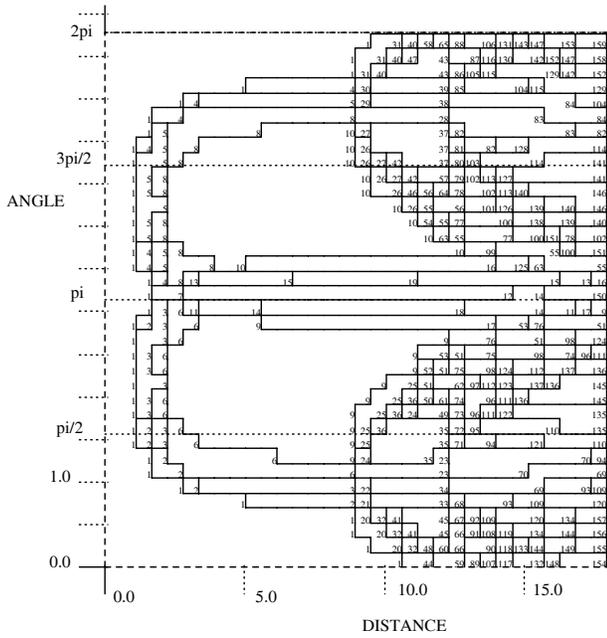


Figure 11: Distinct visual regions in $r\phi$ space for left/right photoreceptor pairs when wall-height=5.

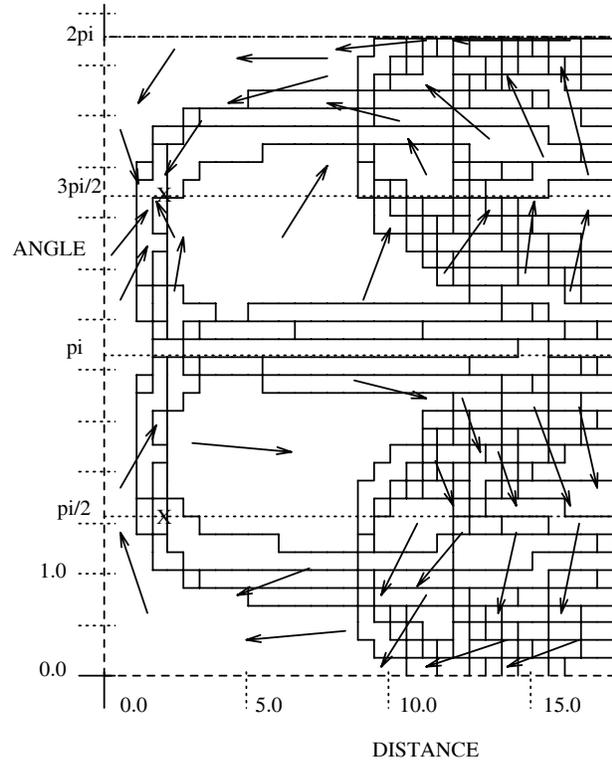


Figure 12: Theoretically constructed phase portrait in $r\phi$ space for C2 at wall-height=5. Two attractors are present (marked by X): the one at $\phi = 3\pi/2$ is dominant.

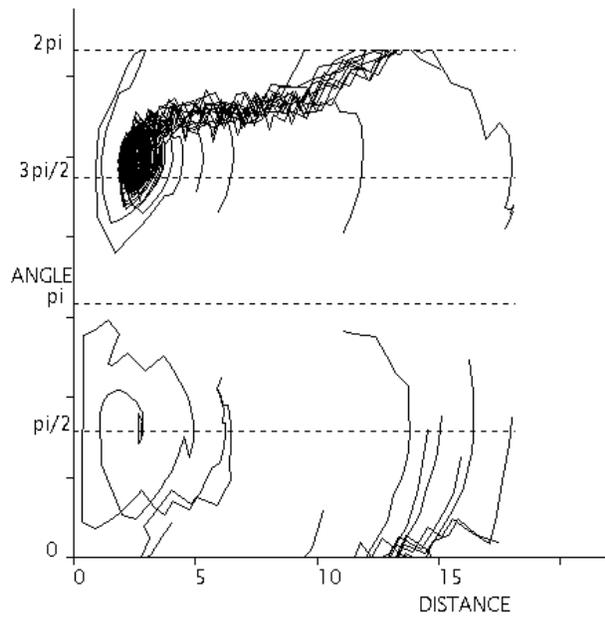


Figure 13: C2 trajectories translated into $r\phi$ space for wall-height=5, as can be seen the theoretical and empirical trajectories are in good agreement.