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 "interbred" 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)

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 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 sening 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 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 whicanthebtr7

(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

control, 1993. Also available as University of Sussex School of Cognitive and Computing Sciences Technical Report CSRP265.

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

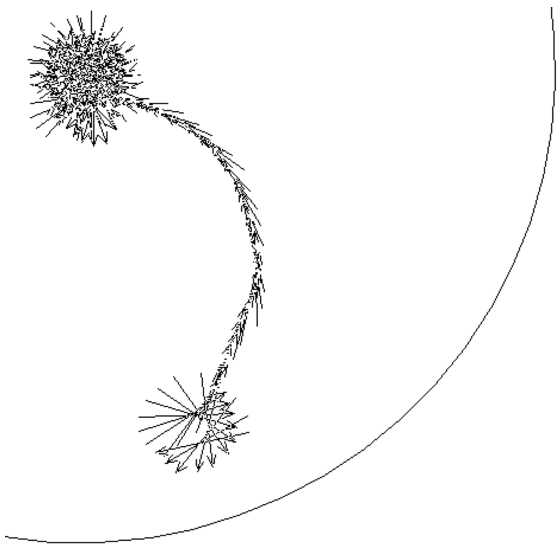
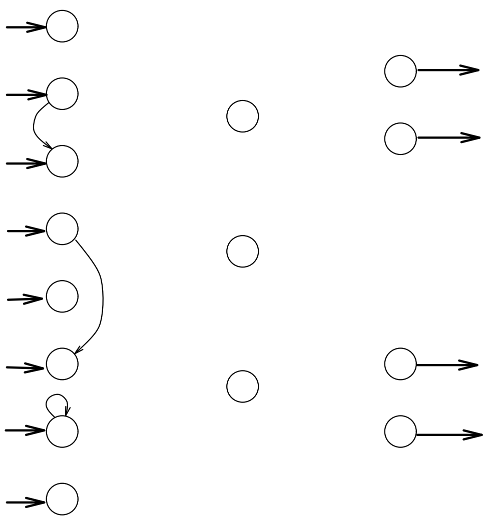
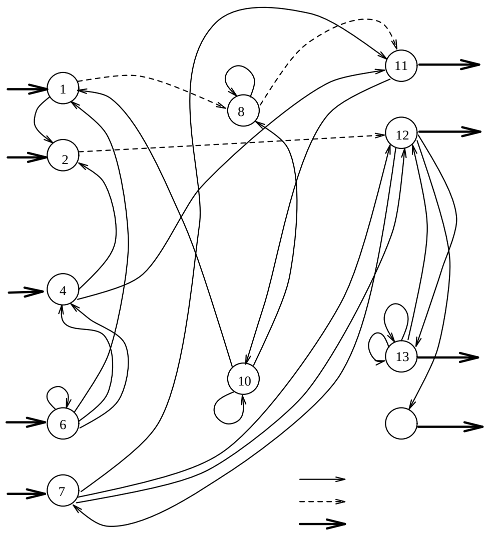
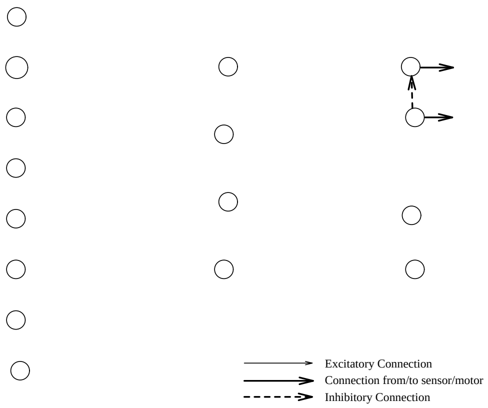


Figure 1: Typical behaviour of the C1 controller. The robot's position at each timestep is shown by an arrow; the midpoint of the arrow 'shaft' is the centre of the robot, and the length of the shaft is the same as the robot's diameter. The robot starts near the edge of the arena, moves to the centre, 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.





1
2
4
6
7
8
10
11
12
13


Excitatory Connection
Connection from/to sensor/motor
Inhibitory Connection