EVOLUTIONARY ROBOTICS AT SUSSEX

I. Harvey, P. Husbands, D. Cliff, A. Thompson, N. Jakobi
School of Cognitive and Computing Sciences
University of Sussex, Brighton BN1 9QH, UK
inmanh, philh, davec, adrianth, nickja@cogs.susx.ac.uk

ABSTRACT

We give an overview of evolutionary robotics research at Sussex. We explain and justify our distinctive approaches to (artificial) evolution, and to the nature of robot control systems that are evolved. We illustrate by presenting results from research with evolved controllers for autonomous mobile robots; simulated robots, coevolved animats, real robots with software controllers or with a controller directly evolved in hardware.

KEYWORDS: Evolutionary Robotics, Artificial Evolution

WHY EVOLUTIONARY ROBOTICS?

When designing a control system for a robot, there are at least three major problems:
- It is not clear *how* a robot control system should be decomposed.
- Interactions between separate sub-systems are not limited to directly visible connecting links, but also include interactions mediated *via the environment*.
- As system complexity grows, the number of potential interactions between sub-parts of the system grows *exponentially*.

Classical approaches to robotics have often assumed a primary decomposition into Perception, Planning and Action modules. Many people now see this as a basic error [2]. Brooks acknowledges the latter two problems above in his subsumption architecture approach. This advocates slow and careful building up of a robot control system layer by layer, in an approach that is explicitly claimed to be inspired by natural evolution — though each new layer of behaviour is wired in by hand design.

An obvious alternative approach is to abandon hand design and explicitly use evolutionary techniques to incrementally evolve increasingly complex robot control systems. Unanticipated interactions between sub-systems need not directly bother an evolutionary process where the only benchmark is the behaviour of the whole system. Other individuals and groups have taken a similar evolutionary approach, such as [1][5][12]; here we concentrate on an overview of work at Sussex. We start with theoretical questions of what artificial evolutionary techniques and classes of control system are appropriate for evolutionary design. We discuss the relationship between robot simulations and reality, and the problem of evaluation within a noisy and uncertain environment. Sussex projects in this area are described, with both simulations and real robots, including hardware evolution.

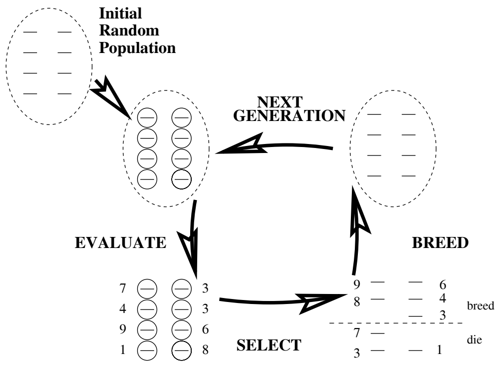

Figure 1: The GA Cycle.

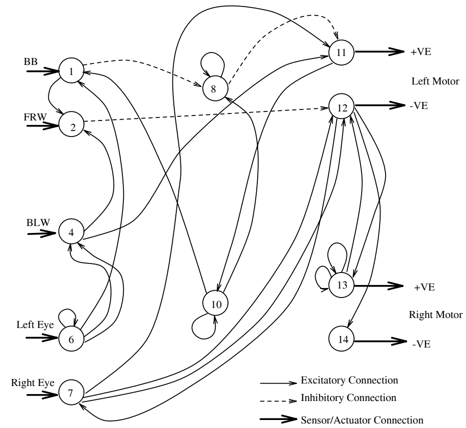

Figure 2: Network without redundant units.

ARTIFICIAL EVOLUTION FOR ROBOTS

Genetic Algorithms (GAs) are the most commonly used evolutionary algorithm for optimisation. Evolutionary Robotics (ER) typically needs adaptive improvement techniques [8] rather than optimisation techniques — a critical distinction.

Optimisation problems can be seen as search problems in some high-dimensional search space, of known size. Each dimension corresponds to a parameter that needs to be set, coded for on a small section of the genotype; in robotics, a genotype specifies the characteristics of a control system. Using a population of such genotypes (often initially random), each is evaluated on how good is the potential solution that it encodes. Fitter genotypes are preferentially selected to be parents of the next generation; offspring inherit genetic material from parents, and also undergo random mutations. This cycle of selection, reproduction with inheritance of genetic material, and variation, is repeated over many generations (Fig. 1).

A GA optimisation approach typically starts with a population of random points crudely sampling the whole search space. Successive cycles focus the population of sample points towards fitter regions of the space. However, some domains — including much of evolutionary robotics — do not always fall into this convenient picture of a fixed-dimensional search space. Standard GA theory does not necessarily then apply.

SAGA — Species Adaptation Genetic Algorithms

In ER a genotype will specify the control system of a robot which is expected to produce appropriate behaviours. The number of components required may be unknown *a priori*; and when using incremental evolution, through successively more difficult tasks, the number of components needed will increase over time. Such incremental evolution calls for *GAs as adaptive improvers* rather than *GAs as optimisers*.

Species Adaptation Genetic Algorithms (SAGA) were developed for this purpose [6]. It was shown that progress through such a genotype space of increasing complexity will only be feasible through gradual increases in genotype length; this implies the evolution of a *species* — the population is largely genetically converged. With successive generations, selection is a force which tends to move such a population up hills on a fitness landscape, and keep it centred around a local optimum; whereas mutation explores outwards from the current population. For a given selection pressure, there is a maximum rate of mutation which simultaneously allows the population to retain a hold on its current hill-top, whilst maximising search along relatively high ridges in the landscape, potentially towards higher peaks. In SAGA, this means that rank-based selection should be used to maintain a constant selective pressure, and mutation rates should be of the order of 1 mutation per genotype [6].

What building blocks for a control system?

We must choose appropriate building blocks for evolution to work with. Primitives manipulated by the evolutionary process should be at the lowest level possible. Any high level semantic groupings inevitably incorporate the human designer's prejudices. We agree with Brooks [2] in dismissing the classical Perception, Planning, Action decomposition of robot control systems. Instead we see the robot as a whole — body, sensors, motors and 'nervous system' — as a dynamical system coupled (via sensors and motors) with a dynamic environment [1]. Hence the genotype should encode at the level of the primitives of a dynamical system.

One such system is a dynamic recurrent neural net (DRNN), with genetic specification of connections and of the timescales of internal feedback. These DRNNs can in principle simulate the temporal behaviour of any finite dynamical system, and are equivalent (with trivial transformations) to Brooks' subsumption architectures. We also deliberately introduce internal noise at the nodes of DRNNs, with two effects. First, it makes possible new types of feedback dynamics, such as self-bootstrapping feedback loops and oscillator loops. Second, it helps to make more smooth the fitness landscape on which the GA is operating.

ER IN SIMULATION

Experiments at Sussex have used a round two-wheeled mobile robot performing navigational tasks. Initial experiments [3] used simulations of such a robot with touch sensors and two visual inputs — simulated photoreceptors, with genetically specified fields of view. The robot task was to reach the centre of a circular arena, with white walls and black floor and ceiling. Grey-level visual inputs to each photoreceptor were calculated by ray-tracing. Robot motion was modelled carefully, including collisions and noisy motor properties, using measurements from a real robot.

The genetically specified DRNNs used had input nodes for each sensor, output nodes for each motor, and an arbitrary number of 'hidden' nodes. All nodes were noisy linear threshold devices. Connections were *excitatory* (weighted link joining the output of one unit to the input of another) or *veto* (an infinitely inhibitory link between two units). The task is set implicitly by the evaluation function, and robots were rated on the basis of how much time they spent at or near the centre of the arena; they always started near the perimeter, facing in a random direction.

Robots with successful evolved control systems make a smooth approach towards the centre of the arena, and circle there. Success was also achieved when the height of the wall was allowed to vary over one order of magnitude, each robot being given 10 trials with differing wall-heights across the full range; for robustness, the evaluation was based on the *worst* score it obtained across its trials.

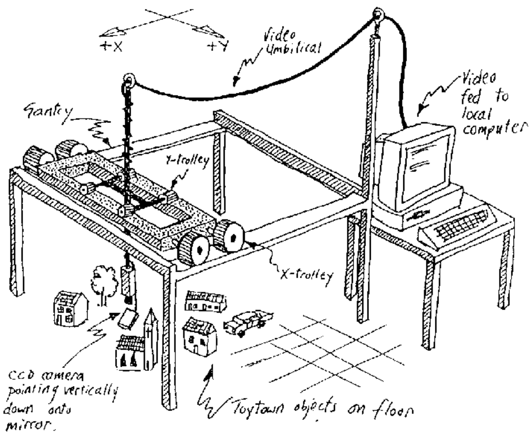

Figure 3: A cartoon sketch of the Gantry.

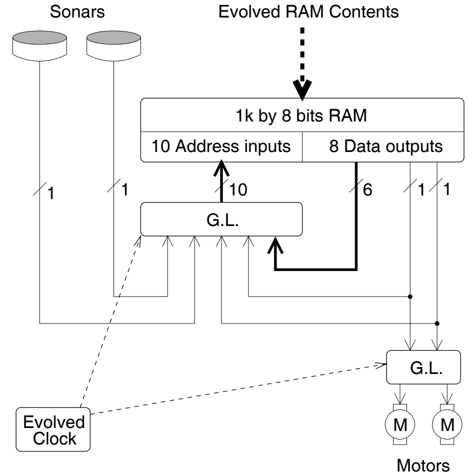

Figure 4: The evolved DSM.

Analysis of an evolved network starts with identification of redundant units and connections. Since early stages of evolution allowed visual signals to prevent the robot from bumping into the walls, the touch-sensors are unused, and their nodes can be recruited as extra 'hidden' nodes. The results of eliminating redundant nodes from a successful network are shown in Fig. 2. An analysis of the attractors of the dynamics of the robot with such a control system has been made [9].

Coevolution

At Sussex further work in simulation has involved exploring the dynamics of co-evolution in pursuit-evasion contests [4]. One species of pursuing animats have their fitnesses determined by the current strategies of another species of evaders, and *vice versa*. Such a coevolutionary 'Arms Race' may have implications for incremental evolution of robots, as a method of automatically increasing task complexity whilst taking humans out of the loop.

THE GANTRY

Ray-tracing in simulation is computationally expensive. For dynamic real-world domains with noisy lighting conditions it is necessary to use a real robot. Evolution requires the evaluation of many trials, which should be automated. We developed a specialised piece of visuo-robotic equipment for this — the gantry-robot.

The robot is cylindrical, 150mm in diameter, and moves in a real environment. Instead of using wheels, the robot is suspended from the gantry-frame with stepper motors that allow translational movement in X and Y directions (Fig. 3), corresponding to that which would be produced by left and right wheels. The visual input is from a CCD camera pointing down at a mirror inclined at 45^{o}, which can be rotated about a vertical axis so as to 'see' along the direction the 'robot' is facing. The CCD image is subsampled into 3 or more genetically specified virtual photoreceptors, or receptive fields — we are using minimal bandwidth vision.

We used the same networks and genetic encoding schemes as before. Tasks were navigating towards white paper targets, in a predominantly dark arena. Using an incremental evolutionary methodology, simple visual environments were used initially, moving on to more complex ones in this sequence of tasks [7]:

(1) Forward movement **(2)** Movement towards large target
(3) Movement towards small target **(4)** Distinguishing triangle from rectangle

An initial random population of 30 needed about 10 generations to achieve success at each stage, which each had appropriate evaluation functions. Control systems capable of reaching the small target were found to generalise to following a moving target of similar size. For the final task, two white targets were fixed to one wall, one triangular and one rectangular. The robots were given trials with differing start positions, not biased towards either target. The evaluation function added a bonus for getting close to the triangle, but subtracted a penalty for nearing the rectangle.

The successful networks were of a similar complexity to that of Fig. 2. The networks evolved such that robots rotated on the spot when visual inputs were both low or both high; but moved in a straight line when only one was high. The visual morphology evolved such that the visual inputs changed in unison when crossing a vertical dark/light edge, and only differed significantly at an oblique edge. Thus the control system was an 'oblique dark/light boundary detector' rather than a 'triangle detector'. In the context, it performed the required task of detecting the triangle, and rejecting the square.

EVOLVABLE HARDWARE

The robot control systems for the experiments above, though conceptualised as dynamical systems, have been implemented in software. They can also be implemented directly in hardware [13], using *intrinsic* hardware evolution, where each genetically specified piece of hardware is tested for real *in situ*. The low-level physics of the hardware can be utilised, and the components can behave at their natural timescales, without the necessity of global clocking or other design constraints.

Thompson used artificial evolution to design a hardware controller, a *Dynamic State Machine* (DSM), for a mobile robot using sonars to avoid walls in a corridor. Success was achieved with a DSM of just 32 bits of RAM and 3 flip-flops (excluding clock generation) which took sonar echo pulses directly, without pre-processing, and output appropriate pulses direct to the motors. The genetic specification of the DSM (Fig. 4) determined whether each signal was synchronised by a clock; and if so, the frequency of that clock. The DSM was intimately coupled to the real-time dynamics of its sensorimotor environment.

In very recent work, to be published, Thompson has applied these techniques to a Field Programmable Gate Array (FPGA) from the forthcoming Xilinx XC6200 family. Circuits on an unclocked FPGA can be evolved to generate desired output frequencies over a wide range, from 10Hz to 1MHz.

EVOLUTION WITH *KHEPERA*

When using simulations it is an important to decide just how realistic the model should be, and how noise should be handled. Jakobi [11] built a simulator, *Khepsim*, for the *Khepera* robot from EPFL in Lausanne. This was based on a spatially continuous, two dimensional model of the underlying real world physics, using a profile derived from the motors and sensors of a real *Khepera*.

IR and ambient light values were calculated by ray-tracing. Runs were performed in simulation with different noise levels — zero, observed noise, double observed noise — and tested on a real robot, for obstacle-avoiding and light-seeking tasks. It was concluded that simulated noise levels should be similar to real levels for systems evolved in simulation to transfer properly. If there is a significant difference in noise levels (too high *or* too low), then whole different classes of behaviours become available which, while acquiring high fitness scores in simulation, fail to work in reality.

SUMMARY

We have discussed the use of SAGA for incremental evolution through a space of dynamical robot control systems. Other relevant aspects are also being researched at Sussex*, such as artificial morphogenesis [10], the design of fitness functions to 'shape' evolution towards desired goals, interactions between learning and evolution.

Evolutionary Robotics is a research area in its infancy; the tests for all newborn AI philosophies are whether they can grow up into the real world, and scale up with increasing complexity. In the evolutionary experiments at Sussex we have started to demonstrate the possibilities in simulation, on real robots, and directly in silicon.

REFERENCES

1. R. Beer and J. Gallagher "Evolving dynamic neural networks for adaptive behavior". *Adap. Beh.* 1(1):91–122, 1992.
2. R. Brooks "A robust layered control system for a mobile robot". *IEEE J. Rob. Autom.*, 2:14-23, 1986.
3. D. Cliff, I. Harvey, and P. Husbands. "Explorations in evolutionary robotics". *Adap. Beh.*, 2(1):71–104, 1993.
4. D. Cliff and G. Miller. "Tracking the Red Queen" In F. Morán et. al., eds., *Advances in Artificial Life: Proceedings of the 3rd ECAL*, pp. 200–218. Springer-Verlag, 1995.
5. D. Floreano and F. Mondada. "Automatic creation of an autonomous agent", In D. Cliff et. al., eds., *From Animals to Animats 3*, MIT Press/Bradford Books, 1994.
6. I. Harvey. "Evolutionary robotics and SAGA: the case for hill crawling and tournament selection". In C. Langton, ed., *Artificial Life III*, pp. 299–326. Addison Wesley, 1993.
7. I. Harvey, P. Husbands, and D. Cliff. "Seeing the light: Artificial evolution, real vision". In D. Cliff et. al. eds., *From Animals to Animats 3*, MIT Press/Bradford Books, 1994.
8. J. Holland. *Adaptation in Natural and Artificial Systems.* Univ. Mich. Press, Ann Arbor, 1975.
9. P. Husbands, I. Harvey, D. Cliff. "Circle in the round", *J. Rob. and Aut. Sys.* 15:83-106, 1995.
10. P. Husbands, I. Harvey, D. Cliff, and G. Miller. "The use of genetic algorithms for the development of sensorimotor control systems". In P. Gaussier and J.-D. Nicoud, eds., *From Perception to Action*, pages 110–121, Los Alamitos, CA, 1994. IEEE Computer Society Press.
11. N. Jakobi, P. Husbands, and I. Harvey. "Noise and the reality gap". In F. Morán et. al., eds., *Advances in Artificial Life: Proceedings of the 3rd ECAL*, pp. 704–720. Springer-Verlag, 1995.
12. S. Nolfi, D. Floreano, O. Miglino, and F. Mondada. "How to evolve autonomous robots". In R. Brooks and P. Maes, eds., *Artificial Life IV*, pages 190–197. MIT Press/Bradford Books, 1994.
13. A. Thompson, I. Harvey, and P. Husbands. "Unconstrained evolution and hard consequences". In E. Sanchez and M. Tomassini, eds., *Towards Evolvable Hardware.* Springer-Verlag, 1996.

* More information is available via WWW on http://www.cogs.susx.ac.uk/lab/adapt/