BRAIN AND COGNITION **34,** 130–159 (1997)
ARTICLE NO. BR970910

Artificial Evolution: A New Path for Artificial Intelligence?

P. Husbands, I. Harvey, D. Cliff, and G. Miller

School of Cognitive and Computing Sciences, University of Sussex, Brighton, United Kingdom

Recently there have been a number of proposals for the use of artificial evolution as a radically new approach to the development of control systems for autonomous robots. This paper explains the artificial evolution approach, using work at Sussex to illustrate it. The paper revolves around a case study on the concurrent evolution of control networks and visual sensor morphologies for a mobile robot. Wider intellectual issues surrounding the work are discussed, as is the use of more abstract evolutionary simulations as a new potentially useful tool in theoretical biology.

1. INTRODUCTION

This paper discusses and attempts to justify a particular approach to the development of autonomous agents with sensorimotor capabilities. The topic is treated from the standpoint of practical Artificial Intelligence, although potential wider implications are indicated. The method advocated is artificial evolution. Populations of agents are interbred under the influence of a task-based selection pressure. Starting from a population of random individuals, agents capable of performing the task well emerge.

The aim of this paper is to introduce some of the motivations underlying the use of artificial evolution and some of the key technical issues involved in implementing it. Little or no familiarity with the approach is assumed. In order to make much of the discussion more concrete, the central section of the paper (Section 5) is concerned with one particular experiment in evolutionary robotics.

Before discussing artificial evolution in some detail, the orientation of the work in relation to other areas of artificial intelligence will be dealt with. First, what do we mean by an autonomous agent? Simply this: a self-govern-

Address reprint requests to P. Husbands, School of Cognitive & Computing Sciences, University of Sussex, Brighton, BN1 9QH, United Kingdom. E-mail: philh or inmanh or davec or geoffm@cogs.susx.ac.uk.

0278-2626/97 $25.00

ing system that makes its own way in the world; there are no hidden leads or radio links connecting it to an operator. This is a rather large class, but, for reasons that will become clear shortly, we will restrict ourselves to those autonomous systems with sensorimotor capabilities.

The perspective taken on Artificial Intelligence (AI) is as follows. AI is regarded as the study of intelligence in natural and artificial systems through the construction of intelligence generating mechanisms. Along with Brooks and others (Brooks, 1991; Varela, Thompson, & Rosch, 1991) we regard the proper study of intelligence as an investigation into the interactions between autonomous agents and their environments. This involves the study of entire nervous systems; perception cannot be separated from action. Indeed, from our perspective nervous systems, natural and artificial, should be regarded as rather complex control systems. We agree with Brooks that the most sensible way to pursue such a study is through the construction of autonomous robots to act in complex, uncertain, dynamic environments. This follows from the belief that the greater part of intelligent behavior in animals (including humans) is bound up with sensorimotor coordination and ''every day'' survival skills. It is very hard to simulate in any detail an agent's sensory and motor couplings with its environment. Although more abstract computer models have their uses, as discussed later, they cannot reflect many of the messy real-time constraints imposed on embodied agents. To take these into account, it is frankly easier to use robots situated in the real world than it is to try and build some all encompassing super-simulation. Intelligence is a vague relative term used to label many adaptive behaviors (behaviors that tend to improve an organism's chance of survival—see (Ashby, 1952) for a good more precise definition). The old-fashioned human-centered notions of intelligence commonly used in the AI community until recently (Boden, 1977), with their stifling focus on abstract problem solving and deliberative thought, are here regarded as far too restrictive. Insects are intelligent. Humans are intelligent. The interesting questions are what mechanisms underly this intelligence and how can we build a robot that exhibits intelligence in this sense?

The kind of research described later is often placed under the ever wider umbrella of Artificial Life. However, most of this work fits neatly into the more specific field of Animat (artificial animals) research. However, since it is mainly concerned with developing mechanisms to generate intelligent behaviors, it can quite validly be regarded as a new approach to AI (Wilson, 1991).

Having provided some context, this paper continues with an argument for the use of artificial evolution in autonomous agent research, this is followed by a more detailed exposition of genetic algorithms, artificial evolution, and evolutionary robotics. The central section of the paper is concerned with a case study on the evolution of visually guided behaviors on a specialized mobile robot. The paper continues with a discussion of some of the more

advanced aspects of evolutionary robotics. We advocate the use of artificial neural network based robot controllers, which in turn means that issues to do with the genetic encoding of the networks become central to the endeavor. We outline a set of desirable properties for such encodings. A number of encoding schemes developed by other researchers are reviewed, and we present new methods of our own. Next a number of the most pertinent issues in evolutionary robotics are outlined. There follows a discussion of the possibility of using artificial evolutionary techniques to help tackle more specifically scientific questions about natural sensorimotor systems.

2. WHY EVOLVE?

Animals that we often think of as rather simple (e.g., arthropods—that class of invertebrates including insects) in fact display a range of sophisticated adaptive behaviors, involving complex sensorimotor coordination (Young, 1989). These behaviors are generated by remarkably few nerve cells, which might suggest that they are based on simple mechanisms. However, in general this does not appear to be the case (Ewert, 1980). Despite their size, the dynamics of arthropod nervous systems are intricate.

Under present technological constraints, control systems for autonomous robots will necessarily involve relatively small numbers of ''components,'' be they implemented in hardware or software. This suggests an analogy with arthropods: it is very likely that it will be necessary to develop complicated dynamical systems to control autonomous robots acting in uncertain complex environments.

Forty years of autonomous robotics research has taught us that generating the degree of sensorimotor coordination needed to sustain adaptive behavior in the real world is no easy matter (Moravec, 1983). We believe this is because the control systems needed will be of the complex dynamical systems variety, and these are inherently extremely difficult to design by traditional means. Indeed, the situation is even worse than is often expected; suitable sensor and actuator properties (including morphologies) are inextricably bound to the most appropriate ''internal'' dynamics of the control system and vice versa. Imposing the simplifying constraint of cleanly dividing the system's operation into a pipeline of sensing, internal processing, and acting now appears to be far too restrictive (Brooks, 1991; Beer, 1990).

To put it in slightly more abstract terms, we strongly suspect (along with Brooks, many biologists, and increasing numbers of cognitive scientists (Sloman, 1993)) that useful control systems to generate interesting behavior in autonomous robots will necessarily involve many emergent interactions between the constituent parts (even though there may be hierarchical functional decomposition within some of these parts). However, we go further by claiming that there is no evidence that humans are capable of designing systems with these characteristics using traditional analytical approaches. We are

very good at designing highly complex systems if we can divide them up into almost insulated subsystems with well defined interactions. However, when the number of interactions between modules increases exponentially with the addition of new modules, the design problem becomes intractable.

We, and a number of other authors, have suggested that the use of artificial evolution to fully, or partially, automate the design process may be a way forward (Cliff, Harvey, & Husbands, 1993; Beer & Gallagher, 1992). A number of research projects are now actively exploring this possibility. This new area is often referred to as evolutionary robotics.

The artificial evolution approach maintains a population of viable genotypes (chromosomes), coding for control architectures. The genotypes are interbred according to a selection pressure, much as in standard genetic algorithm work. This is controlled by a task-oriented evaluation function: the better the robot performs its task the more evolutionarily favored is its control architecture. 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 behavior, as long as this behavior is directly or implicitly included in the evaluation function.

3. GENETIC ALGORITHMS AND ARTIFICIAL EVOLUTION

Genetic algorithms (GAs) are adaptive search strategies based on a highly abstract model of biological evolution (Holland, 1975). They can be used as an optimization tool or as the basis of more general adaptive systems. The fundamental idea is as follows. A population of structures, representing candidate solutions to the problem at hand, is produced. Each member of the population is evaluated according to some fitness function. Fitness is equated with goodness of solution. Members of the population are selectively interbred in pairs to produce new candidate solutions. The fitter a member of the population the more likely it is to produce offspring. Genetic operators are used to facilitate the breeding; that is, operators which result in offspring inheriting properties from both parents (sexual reproduction). The offspring are evaluated and placed in the population, quite possibly replacing weaker members of the last generation. The process repeats to form the next generation. This form of selective breeding quickly results in those properties which promote greater fitness being transmitted throughout the population: better and better solutions appear. Normally some form of random mutation is also used to allow further variation. A simple form of this algorithm is as follows.

1. Create initial population of strings (genotypes). Each string of symbols (genes) is a candidate solution to the problem.
2. Assign a fitness value to each string in the population.
3. Pick a pair of (parent) strings for breeding. The fitter the string the more likely it is to be picked.

4. Put offspring produced in a temporary population.
5. Is the temporary population full? If yes, go to 3, else go to 6.
6. Replace the current population with the temporary population.
7. Has some stopping criteria being fulfilled? If yes, exit, if no, go to 2.

This population-based survival of the fittest scheme has been shown to act as a powerful problem solving method over a wide range of complex domains (Grefenstette, 1985, 1987; Schaffer, 1989; Belew & Booker, 1991; Davis, 1990).

The loose analogies between GAs and natural evolution should be clear. The structures encoding a solution to the problem (often strings of characters) can be thought of as the genotypes or artificial DNA. There will be some process for interpreting the structure as a solution: the phenotype. The interpretation is often implicitly embedded in the evaluation function and can be complex. When the encoding and the evaluation function are static (search space of fixed dimensions, a single well-defined evaluation function), we are in the realms of optimization. When they are not, the GA can be used to build adaptive systems; systems that are able to cope with a changing environment. The latter scenario is closer to the situation existing in natural evolution and is exploited in evolutionary robotics, as will be made clear in the next section.

There are many different implementations of this idea, varying markedly in their specifics, for further details see Holland (1975), Goldberg (1989) and Mitchell (1996). The breeding phase, where offspring are produced from parents, can makes use of a number of different ''genetic'' operators. Crossover is most often used. With this operator, sections of the parental genotype strings are exchanged to form new genotypes. Mutation is very common and involves randomly changing single genes (characters on the genotype string) to new legal values.

3.1 Evolutionary Robotics

The basic notion of Evolutionary Robotics is captured in Fig. 1. The evolutionary process, based on a genetic algorithm, involves evaluating, over many generations, whole populations of control systems specified by artificial genotypes. These are interbred using a Darwinian scheme in which the fittest individuals are most likely to produce offspring. Fitness is measured in terms of how good a robot's behavior is according to some evaluation criterion.

Just as natural evolution involves adaptations to existing species, we believe GAs should be used as a method for searching the space of possible adaptations of an existing robot, not as a search through the complete space of robots: successive adaptations over a long timescale can lead to long-term increases in complexity. For this reason, whereas most GAs operate on fixed-length genotypes, we believe it is necessary to work instead with variable-

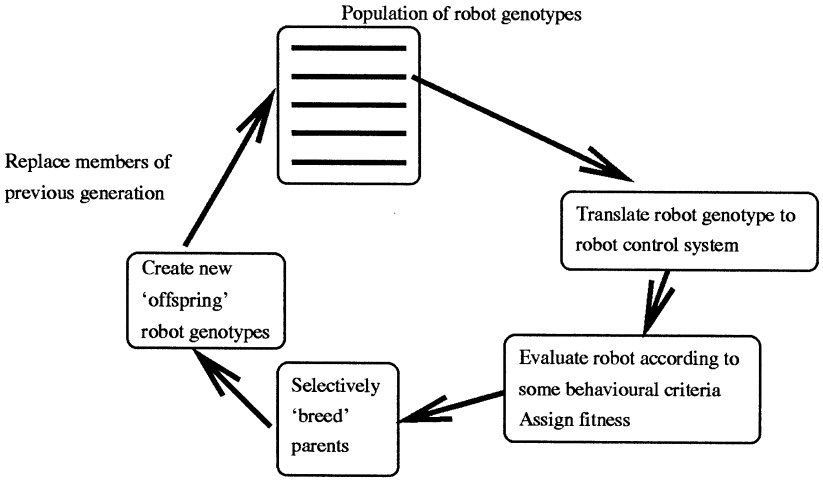

FIG. 1. The basic notion of evolutionary robotics.

length genotypes. This leads to an incremental approach. A series of gradually more demanding task-based evaluation schemes are used. In this way new capabilities are built on existing ones and the search space is always constrained enough to be manageable.

The basis for extending standard GAs to cope with this has been worked out by Harvey (1992b, 1994, 1992a), who describe the *Species Adaptation Genetic Algorithm* (SAGA). In SAGA, the population being evolved is always a fairly genetically converged species; and increases in genotype length (or other metric of expressive power), associated with increases in complexity, can happen only very gradually.

4. EVOLVE NETWORKS

Much of the work described in this paper uses artificial neural networks of some variety as the basic building blocks of the control systems being developed. We believe this is the most appropriate choice. For reasons given in Cliff et al. (1993), network search spaces are generally smoother than spaces of explicit control programs. Networks are naturally amenable to open-ended approaches and allow the researcher to work with very low-level primitives, thereby avoiding incorporating too many preconceptions about suitable control system properties. We advocate unrestricted recurrent real-time dynamical networks as one of the most general class of behavior generating systems. However, such systems are far too unconstrained, with a great many potential free parameters (such as neuron time constants and thresholds and connection delays and weights) to admit hand design. Therefore, this class of intrinsically very powerful systems can only really be explored with

the help of automated techniques, of which artificial evolution is a front runner.

5. AN EXAMPLE EXPERIMENT

This section makes much of the surrounding discussion more concrete by focusing on a particular experiment to evolve a network-based control system for a mobile robot engaged in visually guided tasks of increasing complexity.

There are many different ways of realizing each stage of the cycle shown in Fig. 1. A crucial decision is whether or not to use simulation at the evaluation stage, transferring the end results to the real world. Since an evolutionary approach potentially requires the evaluation of populations of robots over many generations, a natural first thought is that simulations will speed up the process, making it more feasible. Despite initial scepticism (Brooks, 1992), it has recently been shown that control systems evolved in carefully constructed simulations, with an appropriate treatment of noise, transfer extremely well to reality, generating almost identical behaviors in the real robot (Jakobi, Husbands, & Harvey, 1995; Thompson, 1995). However, both of these examples involved relatively simple robot–environment interaction dynamics. Once even low-bandwith vision is used, simulations become altogether more problematic. They become difficult and time consuming to construct and computationally very intensive to run. Hence evolving visually guided robots in the real world becomes a more attractive option. The case study described in this section revolves around a piece of robotic equipment specially designed to allow the real-world evolution of visually guided behaviors—the Sussex gantry robot.

5.0.1. Concurrent evolution of visual morphologies and control networks. Rather than imposing a fixed visual sampling morphology, we believe a more powerful approach is to allow the visual morphology to evolve along with the rest of the control system. Hence we genetically specify regions of the robot's visual field to be subsampled, these provide the only visual inputs to the control network. It would be desirable to have many aspects of the robot's morphology under genetic control, although this is not yet technically feasible.

5.0.2. The gantry robot. The gantry robot is shown in Fig. 3. The robot is cylindrical, some 150 mm in diameter. It is suspended from the gantry frame with stepper motors that allow translational movement in the X and Y directions, relative to a coordinate frame fixed to the gantry. The maximum X (and Y) speed is about 200 mm/sec. Such movements, together with appropriate rotation of the sensory apparatus, correspond to those which would be produced by left and right wheels. The visual sensory apparatus consists of a CCD camera pointing down at a mirror inclined at 45° to the vertical (see Fig. 4). The mirror can be rotated about a vertical axis so that its orienta-

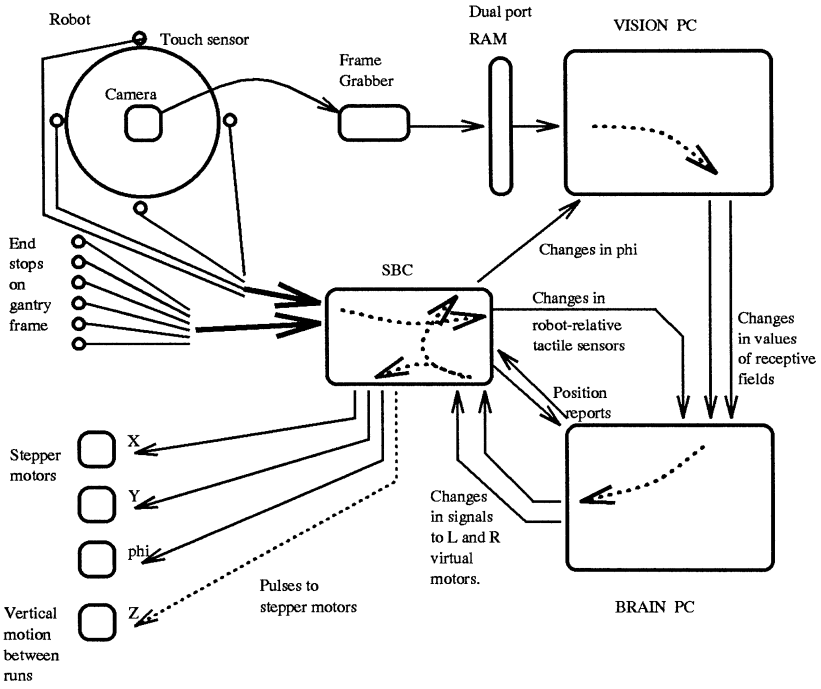

FIG. 2. The different roles of the Vision computer, the Brain computer and the SBC.

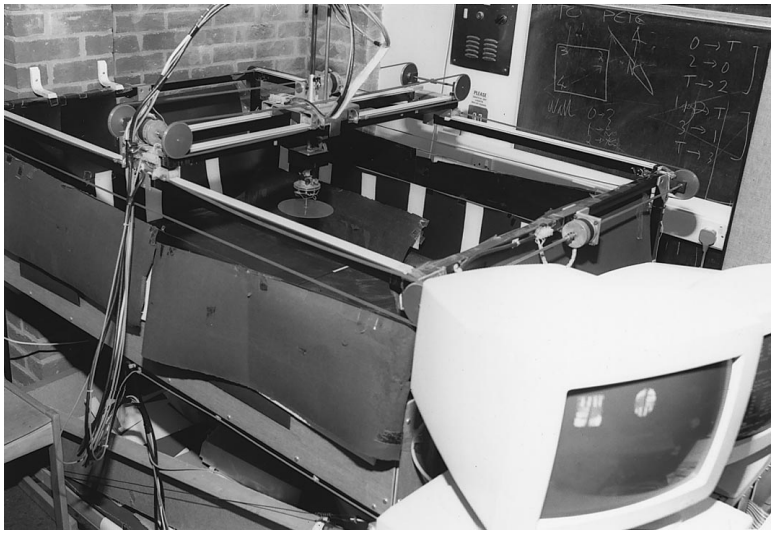

FIG. 3. The Gantry viewed from above. The horizontal girder moves along the side rails, and the robot is suspended from a platform which moves along this girder.

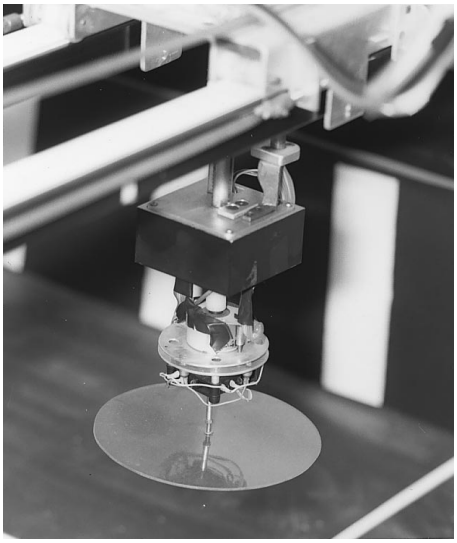

FIG. 4. The gantry robot. The camera inside the top box points down at the inclined mirror, which can be turned by the stepper motor beneath. The lower plastic disk is suspended from a joystick, to detect collisions with obstacles.

tion always corresponds to the direction the ''robot'' is facing. The visual inputs undergo some transformations en route to the control system, described later. The hardware is designed so that these transformations are done completely externally to the processing of the control system.

The control system for the robot is run off-board on a fast personal computer, the ''Brain PC.'' This computer receives any changes in visual input by interrupts from a second dedicated ''Vision PC.'' A third (single-board) computer, the SBC, sends interrupts to the Brain PC signaling tactile inputs resulting from the robot bumping into walls or physical obstacles. The only outputs of the control system are motor signals. These values are sent, via interrupts, to the SBC, which generates the appropriate stepper motor movements on the gantry.

The roles of the three computers are illustrated in Fig. 2. Continuous visual data is derived from the output of the small monochrome CCD camera. A purpose-built Frame Grabber transfers a 64×64 image at 50 Hz into a high-speed 2 K CMOS dual-port RAM, completely independently and asynchronously relative to any processing of the image by the Vision PC. The Brain PC runs the top-level genetic algorithm and during an individual evaluation, it is dedicated to running a genetically specified control system for a fixed period. At intervals during an evaluation, a signal is sent from the Brain PC to the SBC requesting the current position and orientation of the robot. These are used in keeping score according to the current fitness function. The Brain

PC receives signals, to be fed into the control system, representing sensory inputs from the Vision PC and the SBC. The visual signals are derived from averaging over genetically specified circular receptive patches in the camera's field of view.

This setup, with off-board computing and avoidance of tangled umbilicals, means that the apparatus can be run continuously for long periods of time—making artificial evolution feasible.

A top-level program automatically evaluates, in turn, each member of a population of control systems. A new population is produced by selective interbreeding and the cycle repeats. For full technical details of the system see Harvey et al. (1994).

5.0.3. The artificial neural networks. The artificial neurons used have 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. The noise was uniformly distributed in the real range $[-N, +N]$.

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

The networks' continuous nature was modeled by using very fine time slice techniques. In the experiments described in this paper the following neuron parameter setting were used: $N = 0.1$, $T_v = 0.75$, $T_1 = 0.0$, and $T_2 = 2.0$. The networks are hardwired in the sense that they do not undergo any architectural changes during their lifetime, they all had unit weights and time delays on their connections. These networks are just one of the class we are interested in investigating.

5.0.4. The genetic encoding. Two ''chromosomes'' per robot are used. One of these is a fixed length bit string encoding the position and size of three visual receptive patches as described above. Three eight-bit fields per patch are used to encode their radii and polar coordinates in the camera's circular field of view. The other chromosome is a variable-length character

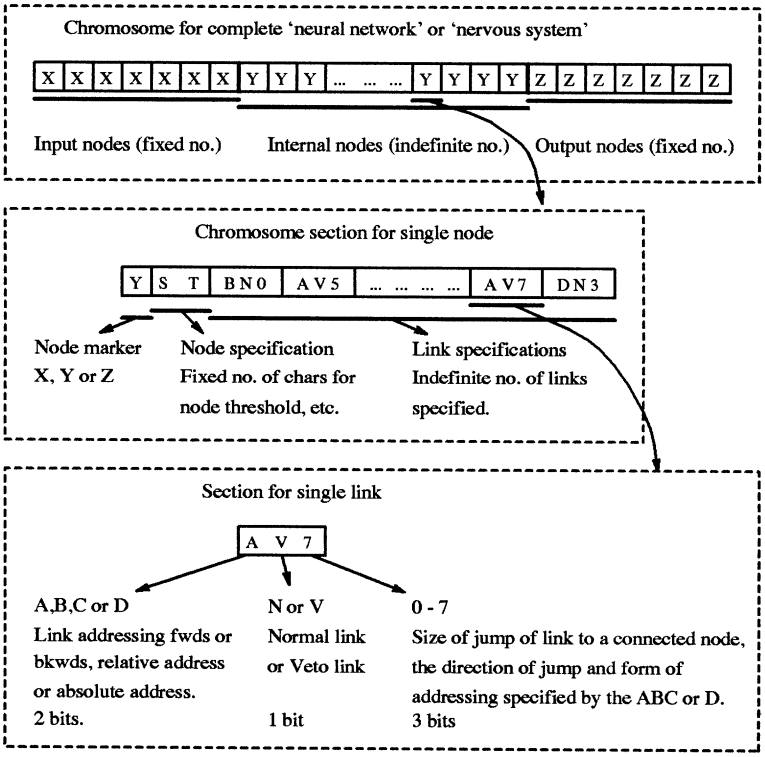

FIG. 5. The genetic encoding scheme.

string encoding the network topology. The genetic encoding used for the control network is illustrated in Fig. 5.

The network chromosome is interpreted sequentially. First the input units are coded for, each preceded by a marker. For each node, the first part of its gene can encode node properties such as threshold values; there then follows a variable number of character groups, each representing a connection from that node. Each group specifies whether it is an excitatory or veto connection, and then the target node is indicated by jump type and jump size. In a manner similar to that used in Harp and Samad (1992), the jump type allows for both relative and absolute addressing. Relative addressing is provided by jumps forward or backward along the genotype order; absolute addressing is relative to the start or end of the genotype. These modes of addressing mean that offspring produced by crossover will always be legal. There is one input node for each sensor (three visual, four tactile).

The internal nodes and output nodes are handled similarly with their own identifying genetic markers. Clearly this scheme allows for any number of internal nodes. The variable length of the resulting genotypes necessitates a

careful crossover operator which exchanges homologous segments. In keeping with SAGA principles, when a crossover between two parents can result in an offspring of different length, such changes in length (although allowed) are restricted to a minimum (Harvey, 1992a). There are four output neurons, two per motor. The outputs of each pair are differenced to give a signal in the range $[-1,1]$.

5.0.5. Experimental setup. In each of the experiments a population size of 30 was used with a genetic algorithm employing a linear rank-based selection method, ensuring the best individual in a population was twice as likely to breed as the median individual. Each generation took about 1.5 hr to evaluate. The most fit individual was always carried over to the next generation unchanged. A specialised crossover allowing small changes in length between offspring and parents was used (Cliff et al., 1993). Mutation rates were set at 1.0 bit per vision chromosome and 1.8 bits per network chromosome.

With the walls and floor of the gantry environment predominantly dark, initial tasks were navigating toward white paper targets. In keeping with the incremental evolutionary methodology, deliberately simple visual environments are used initially, as a basis to moving on to more complex ones. Illumination was provided by fluorescent lights in the ceiling above, with the gantry screened from significant daylight variations. However, the dark surfaces did not in practice provide uniform light intensities, neither over space nor over time. Even when the robot was stationary, individual pixel values would fluctuate by up to 13%.

5.1. Results

5.1.1. Big target. In the first experiment, one long gantry wall was covered with white paper. The evaluation function ϵ_1, to be maximized, implicitly defines a target locating task, which we hoped would be achieved by visuomotor coordination

$$\epsilon_1 = \sum_{i=1}^{i=20} Y_i, \tag{2}$$

where Y_i are the perpendicular distances of the robot from the wall opposite that to which the target is attached, sampled at 20 fixed-time intervals throughout a robot trial which lasted a total of about 25 sec. The closer to the target the higher the score. For each robot architecture four trials were run, each starting in the same distant corner, but facing in four different partially random directions, to give a range of starts facing into obstacle walls as well as toward the target. As the final fitness of a robot control architecture was based on the *worst* of the four trials (to encourage robustness), and since in this case scores accumulated monotonically through a trial, this allowed later trials among the four to be prematurely terminated when they bettered previous trials. In addition, any control systems that had

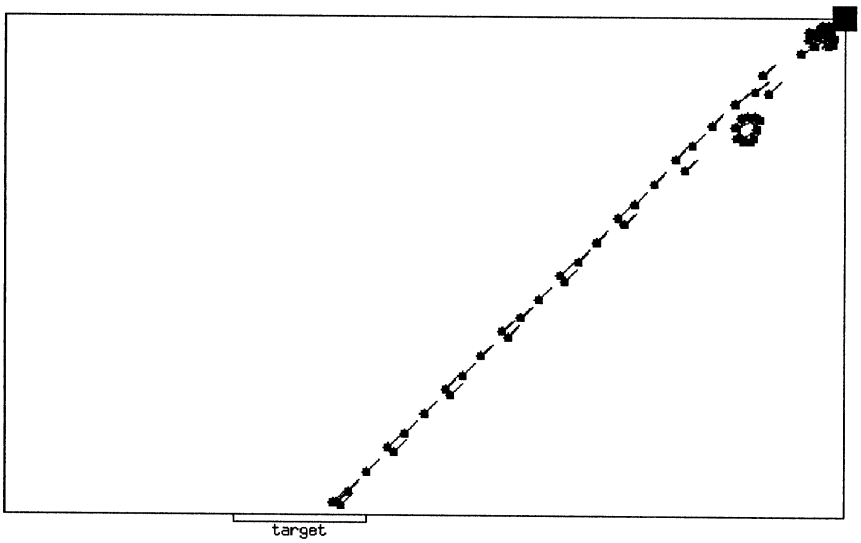

FIG. 6. Behavior of the best of a later generation evolved under second evaluation function. The dots and trailing lines show the front of the robot and its orientation. Coarsely sampled positions from each of four runs are shown, starting in different orientations from the top right corner.

not produced any movement by 1/3 of the way into a trial was aborted and given zero score.

The run was started from a converged population made entirely of clones of a single randomly generated individual picked out by us as displaying vaguely interesting behavior (but by no means able to do anything remotely like locate and approach the target). In two runs using this method very fit individuals appeared in less than 10 generations. From a start close to a corner, they would turn, avoiding contact with the walls by vision alone, then move straight toward the target, stopping when they reached it.

5.1.2. Small target. The experiment continued from the stage already reached, but now using a much narrower target placed about 2/3 of the way along the same wall the large target had been on, and away from the robot's starting corner (see Fig. 6), with evaluation ϵ_2

$$\epsilon_2 = \sum_{i=1}^{i=20} (-d_i), \tag{3}$$

where d_i is the distance of the robot from the center of the target at one of the sampled instances during an evaluation run. Again, the fitness of an individual was set to the worst evaluation score from four runs with starting conditions as in the first experiment. The initial population used was the 12th

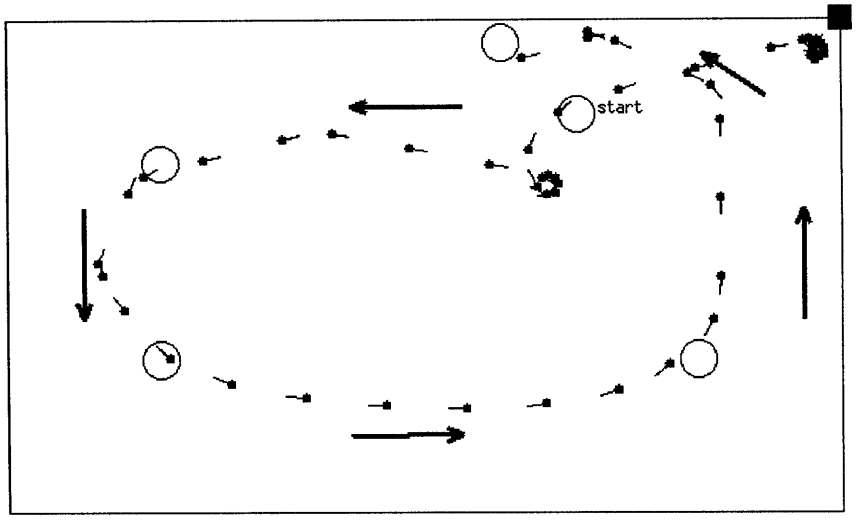

FIG. 7. Tracking behavior of the control system that generated the behavior shown in Fig. 6. The unfilled circles show the position of the target at a number of points on its path (starting position indicated). The arrows roughly indicate the path of the target.

generation from a run of the first experiment (i.e., we incrementally evolved on top of the existing behaviors).

Within six generations a network architecture and visual morphology had evolved displaying the behavior shown in Fig. 6. This control system was tested from widely varying random starting positions and orientations, with the target in different places, and with smaller and different-shaped targets. Its behavior was general enough to cope with all these conditions for which it had not explicitly been evolved. It was also able to cope well with moving targets as shown in Figs. 7 and 8.

5.1.3. Rectangles and triangles. The experiment continued with a distinguish-between-two-targets task. Two white paper targets were fixed to one of the gantry walls: one was a rectangle, the other was an isosceles triangle with the same base width and height as the rectangle. The robot was started at four positions and orientations near the opposite wall such that it was not biased toward either of the two targets. The evaluation function ϵ_3, to be maximized, was

$$\epsilon_3 = \sum_{i=1}^{i=20} [\beta(D_{1_i} - d_{1_i}) - \sigma\,(D_{2_i}, d_{2_i})], \tag{4}$$

where D_1 is the distance of target 1 (in this case the triangle) from the gantry origin; d_1 is the distance of the robot from target 1; and D_2 and d_2 are the corresponding distances for target 2 (in this case the rectangle). These are

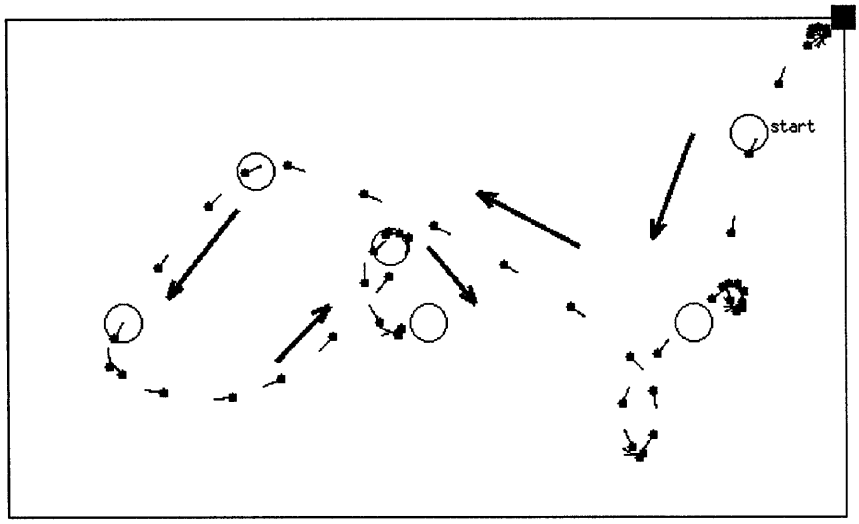

FIG. 8. Further tracking behavior of the control system that generated the behavior shown in Fig. 7.

sampled at regular intervals, as before. The value of β is $(D_1 - d_1)$ unless d_1 is less than some threshold, in which case it is $3 \times (D_1 - d_1)$. The value of σ (a penalty function) is zero unless d_2 is less than the same threshold, in which case it is $I - (D_2 - d_2)$, where I is the distance between the targets; I is more than double the threshold distance. High fitnesses are achieved for approaching the triangle but ignoring the rectangle. It was hoped that this experiment might demonstrate the efficacy of concurrently evolving the visual sampling morphology along with the control networks.

After about 15 generations of a run using as an initial population the last generation of the incremental small target experiment, fit individuals emerged capable of approaching the triangle, but not the rectangle, from each of the four widely spaced starting positions and orientations. The behavior generated by the fittest of these control systems is shown in Fig. 9. When started from many different positions and orientations near the far wall, and with the targets in different positions relative to each other, this controller repeatedly exhibited very similar behaviors to those shown.

The active part of the evolved network that generated this behavior is shown in Fig. 10. The evolved visual morphology for this control system is shown in the inset. Only receptive fields 1 and 2 were used by the controller.

Detailed analyses of this evolved system can be found in Harvey, Husbands, & Cliff (1994) and Husbands (1996). To crudely summarize, unless there is a difference in the visual inputs for receptive fields 1 and 2, the robot makes rotational movements. When there is a difference it moves in a straight

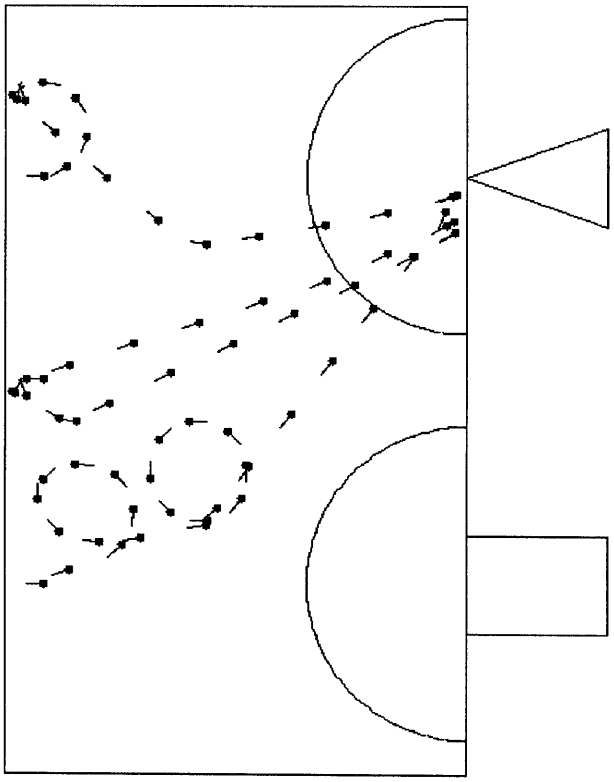

FIG. 9. Behavior of a fit individual in the two target environment. The rectangle and triangle indicate the positions of the targets. The semicircles mark the ''penalty'' (near rectangle) and ''bonus score'' (near triangle) zones associated with the fitness function. In these four runs the robot was started directly facing each of the two targets, and twice from a position midway between the two targets: once facing into the wall and once facing out.

line. The visual sensor layout and network dynamics have evolved such that it fixates on the sloping edge of the triangle and moves toward it.

The case study described above has been included to provide a concrete focus to the issues discussed in this paper. However, this is only one experiment of many, making use of one particular type of network, genetic encoding, and experimental setup. The rest of this paper introduces other aspect of such research.

6. GENETIC ENCODINGS AND DEVELOPMENTAL SCHEMES

Once the decision to evolve network-based systems has been taken, the question of how to encode the networks on an artificial genotype becomes crucially important. Without a suitable encoding scheme little progress can

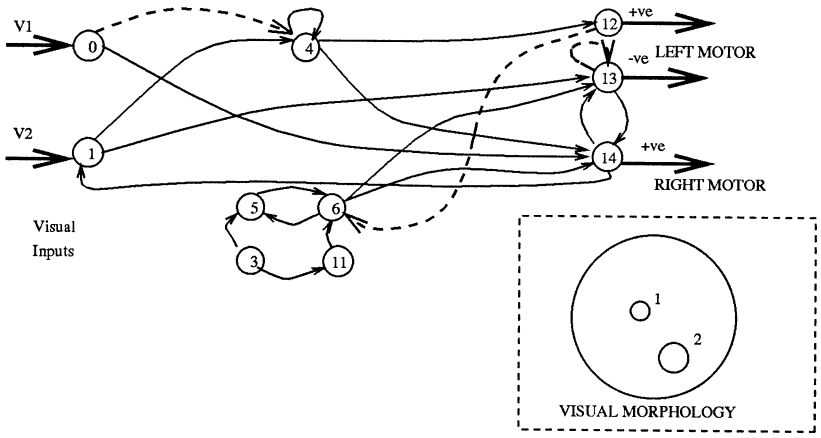

FIG. 10. Active part of the control system that generated fit behavior for the rectangle and triangle experiment. Visual morphology shown in the inset.

be made. In the simplest schemes the genotype is a direct description of the network wiring. An example of that kind of scheme is the genetic encoding used with the gantry robot and described in the previous section. Such encodings will necessarily be restrictive. Much more powerful approaches, allowing complete open-endedness and modularity through the repeated use of genotype sections, must involve a more complex interpretive process.[1] This can be thought of as being loosely analogous to the developmental processes that occur in nature to produce a phenotype from a genotype. Since we regard encoding issues as being central to evolutionary development of control systems, this and the following section concentrate on this area.

Gruau (1992) defines seven properties of genetic encoding of neural networks that should be considered. These include: *completeness,* any NN architecture should be capable of being encoded; *compactness,* one encoding scheme is more compact than the second if for any NN architecture the first genetic encoding is shorter than that given by the second; *closure,* implies that any genotype encodes some architecture; *modularity,* a genetic encoding would be modular if parts of the genotype specify subnetworks of the complete network, and other parts specify the connections between such subnetworks, this decomposition could be recursive. We endorse all these considerations, especially modularity which would seem necessary for sensorimotor systems employing vision. Additional points are: *smooth interaction with*

[1] In this context modularity refers to a developmental process analogous to the use of subroutines in programs. For instance, the left limbs and right limbs of animals will not be independently ''coded for'' in DNA, but rather generated by the same genetic information expressed more than once.

genetic operators, the encoding should allow relatively smooth movements around the search space; the encoding should *not presuppose the dimensionality of the search space,* incremental evolution requires an open-ended approach in which the dimensionality of the search space cannot be specified in advance, the encoding should allow variable numbers of neurons and connections; the encoding must allow *specification of sensory and motor properties* as well as that of a control network.

Kitano (1990) developed an early method for encoding networks which took into account some of the issues raised above. Although his technique was not specifically developed for sensorimotor systems, it can be applied to them. The genotype was used to encode a graph generation grammar. Kitano's system allows a linear genotype to operate on a square matrix of characters, initially 1×1. Each act of rewriting expands the matrix into four quadrants each the same size as the previous matrix, with the contents of each quadrant specified by the genotype. At the end of a succession of n such rewriting steps, a network connection matrix of size $2^n \times 2^n$ is produced. In this way scalability and modularity start to be implemented in a compact genetic encoding of large regular networks.

Gruau (1992) discusses Kitano's work and also acknowledges earlier work by Wilson (1987). Gruau's cellular encoding is a form of ''neural network machine language,'' which he claims has all the above desirable properties. This is a form of rewriting grammar, where the rewriting is considered as a form of developmental process involving ''rewriting neurons'' or ''cells.'' Rewriting operators include PAR which divides a cell into two cells that inherit the same input and output links as their parent, CLIP which can cut links, WAIT which delays rewriting operations so as to change the order in which later operations are carried out. Further operators SPLIT and CLONE allow for the desirable property of modularity to be achieved. In total 13 operators are used. Although it has not yet been done, he proposes using his method for the development of sensorimotor control systems.

7. DEVELOPMENTAL SCHEMES FOR SENSORIMOTOR SYSTEMS

This section outlines three schemes recently developed at Sussex for encoding network-based sensorimotor control systems. They take into account the issues listed earlier and are specifically aimed at encoding whole control systems: that is, control networks along with sensor and motor morphologies.

7.1. A Language and Compiler Scheme

Experience with the primitive encoding we used in our early evolutionary robotics simulation studies (Cliff et al., 1993) lead us to develop a language- and compiler-type genetic encoding scheme which is tailored to the demands of evolving sensory–motor coordination morphologies and in particular to encoding repeated structures as are commonly found necessary in dealing

with visual sensory processing. As with genetic programming, the genome is a program, which is expressed as a 1D string—although at the conceptual level a higher-dimensional space offers a more appropriate descriptive framework. The encoding scheme is essentially a new programming language, called ''NCAGE'' (from ''Network Control Architecture Genetic Encoding''); NCAGE allows for specifying sensory—motor controller morphologies based on ''neural network'' parallel-distributed processing architectures. The artificial genomes are interpreted as NCAGE programs which are ''compiled'' in a ''morphogenesis'' process to create controller structures. It is important to note that we do not consider the DNA-encoded genomes of biological systems as programs and neither do we consider biological morphogenesis as comparable to compiling or executing a computer program. The notions of ''genome-as-program'' and ''morphogenesis-as-compilation'' used here are nothing more than metaphors invoked in the exposition of what is at present essentially an engineering endeavor.

It is beyond the scope of this paper to fully describe this new encoding scheme: a brief description of its key features is given below.

> The NCAGE language draws on elementary vector-graphics programming facilities found in many graphics languages (and in platform-specific vector graphic extensions to general programming languages). It thus bears superficial similarities to turtle-graphics languages such as LOGO. Essentially, the genome is interpreted as a sequence of subroutine specifications and subroutine calls. Subroutines may call other subroutines including themselves. Subroutine calls are made from ''positions'' in a high-dimensional space (typically conceptualized as one of a number of distinct but superpositioned Euclidian 2 spaces or 3 spaces). Calls may reposition a ''cursor'' (cf. turtle) or may place one or more ''neurons'' of differing types at a position specified relative to the current cursor position.

The encoding scheme is modular, has varying resolution of numeric values, is robust with respect to crossover and mutation, allows for recursively repeatable structures (with systematic variations where necessary), and has inbuilt symmetry and symmetry-breaking facilities. Structure specifications are largely independent of their position on the genome, and so a transposition operator can be used to good effect. An inversion operator is also used, but because the genome is read left-to-right, inversion does not preserve structure specifications and is used primarily as an operator for achieving extremely high mutation rates within a localized sequence on the genome, while preserving the statistical distribution of characters on the inverted portion of the genome.

Because the encoding has to satisfy requirements imposed by the genetic operators, NCAGE differs significantly from traditional computer programming languages. The most marked difference is that portions of the genome may be interpreted ''junk'' or ''silent'' code: while many programming languages allow for the specification of subroutines which are never called, most will generate terminal error conditions when the subroutines are partially

complete or nonexistent. The NCAGE interpreter does not generate an error when it encounters calls to unspecified subroutines (such calls are simply ignored), and sequences of instructions which cannot be parsed as generating useful structures are likewise ignored.

The genomes are expressed in a three-character alphabet, although in principle a binary alphabet could be employed at the cost of proportionately longer strings. Under the three-character scheme, two characters are used as bits for data, and the third is a ''stop'' control character used for terminating specifications at varying levels in the genome interpretation process. Theoretically, any sufficiently long random string over the chosen alphabet will be interpretable as a specification of a controller architecture. However, practical considerations entail that some structure (i.e., high-order correlations) are introduced in the generation of initial random genomes, to reduce their length. Experience with the encoding indicates that the inclusion of junk code on the genome increases the robustness of the encodings with respect to the genetic operators employed.

7.2. A Force Field Development Scheme

A second, contrasting, scheme makes use of a highly implicit dynamical process governed by a system of ordinary differential equations, the parameters of which are encoded on the genotype. This process describes the growth of a network-based sensorimotor control system. Again, in no way is this scheme intended to be a model of any biological process. It was developed simply as a method having the properties we believe are desirable for artificial evolution.

In this force field scheme, ''neurons'' are positioned across a two-dimensional plane where they exert attractive forces on the free ends of ''dendrites'' growing out from themselves and other neurons. The ends of the dendrites move under the influence of the force field created by the distribution of neurons. If a dendrite end moves to be within a small distance, ϵ, from a neuron it stops growing and connects to that neuron (which may be its parent). Dendrites do not affect each other and may cross in arbitrary ways. The equations of motion of the dendrite ends are given by ordinary differential equations of the form shown in Eq. (5).

$$\frac{d^2\vec{R}_{ij}}{dt^2} = \sum_{k=0}^{N} \vec{F}_{ijk} - K\frac{d\vec{R}_{ij}}{dt} + \frac{G_{ij}}{l_{ij}^3\left|\dfrac{d\vec{R}_{ij}}{dt}\right|}\frac{d\vec{R}_{ij}}{dt}, \tag{5}$$

where $\vec{R}_{ij}$ is the position vector of the end of the jth dendrite of the ith neuron (henceforth referred to as end_{ij}). The first term on the RHS of Eq. (5) represents the vector sum of the attractive forces exerted on end_{ij} by all neurons. These forces are of the form given in Eq. (6).

$$\vec{F}_{ijk} = \frac{GS_i AS_k \, \vec{r}_{ijk}}{|\vec{r}_{ijk}|^3}, \tag{6}$$

where $\vec{r}_{ijk}$ is the vector from end_{ij} to the center of neuron k. GS_i and AS_k are genetically determined constants. The second term in Eq. (5) is a ''viscous'' resistive force to prevent dendrites sailing off into outerspace. The third term provides a force in the direction of motion of the dendrite end and is inversely proportional to the cube of l_{ij}, the current length of the dendrite. This force drops off very rapidly but encourages dendrites to, at least initially, escape from the influence of their parent neuron. G_{ij} is a genetically encoded constant. In the computational implementation of the scheme, the differential equations were approximately integrated using the Euler method with an adaptive time step. One feature of this method is that the lengths of the resulting dendrite paths can be translated into time delays or weights for use in the operation of the network.

A genotype to be used with this scheme must encode the parameters of the equations, along with the positions of the neurons and the initial directions of growth of the dendrites. In principle, a large number of different encodings would suffice. However, as already discussed, it is preferable to use an encoding exhibiting the desirable properties outlined in Section 7. A particular encoding meeting these requirements, and specially developed for the force field method, is briefly outlined below.

In this method a bit string is used as a neutral encoding medium. That is, any bit string can be interpreted as a control system (although it may be an empty one). The core of the interpreting algorithm is as follows. The string is scanned from left to right until a particular type of marker (short bit pattern) is found. If the marker bounds the start of a valid string section, sequences of bits are read and turned into ''neuron'' parameter values for use with the force field development scheme. As with the previously described language and compiler model, each of these read operations counts the number of 1s in a sequence and uses that number to map to the parameter value. The algorithm rewinds back to the start-section marker and then searches forward to the next occurrence of a second type of marker. This signals a new ''mode'' of interpretation in which dendrite properties are determined. This is repeated until yet another form of marker is encountered. The algorithm then moves back to the first marker and searches forward to the next occurrence of a start-section marker. The whole process then repeats. This ''looping back'' behavior means the algorithm can potentially reuse parts of the string many times. This results in the encoding of relatively large parts of the networks being localized on the string. This produces a form of modularity, where repeated patterns are formed by the reexpression of parts of the genotype.

The position of a neuron is described by genetically determined distance and direction from the last neuron to be laid down. The existence or other-

wise of a particular marker determines whether or not a neuron acts as a visual receptor. If it is a visual receptor, its position on the plane is mapped onto a position within the robot's receptive field. In the scheme currently being used, two special motor neurons are placed near the center of the plane. The network then develops around them. This is convenient for a two motor system, but many other ways of handling motor neurons can be incorporated into the method.

7.3 A Cell Division Method

In this proposal, a naive model is used of the development of a multicellular organism by cell division from a single initial cell. Every cell contains the same DNA, the genotype, which acts as a constraint on an intracellular dynamics of transcription and translation of ''enzymes'' which themselves initiate or repress the production of further enzymes. The genotype and also the enzymes are bit strings.

Within one cell, any initial enzymes are template-matched against the genotype; wherever matches occur, transcription from the genotype starts and produces further enzymes. The ensuing intracellular dynamics can be influenced by intercellular ''signals'' received from neighboring cells. The production of particular enzymes initiates cell-splitting; other particular enzymes, when they are first produced, signal the completion of the developmental process.

In this way, from an initial single cell with a genotype, development produces a number of cells that can be considered as positioned in some space with neighborhood relations. Although all containing the same DNA, the cells can be differentiated into different classes by the particular distinctive internal dynamics of their enzyme production process. Thus at this stage the whole group of cells can be interpreted as a structure with organization; for instance, as a neural network with different cells being ''neurons'' with specific characteristics, and with connections specified between them.

8. SOME RELATED WORK ON EVOLUTIONARY DEVELOPMENT OF SENSORIMOTOR CONTROL SYSTEMS

This section provides a brief high-level review of research into the use of genetic algorithm based techniques for the development of sensorimotor control systems for autonomous agents.

In a traditional autonomous robotics context, mention is made of a proposed evolutionary approach in Barhen, Dress, & Jorgensen (1987). A student of Brooks discussed some of the issues involved, with reference to subsumption architectures, in Viola (1988). De Garis (1992) proposed using GAs for building behavioral modules for artificial nervous systems, or ''artificial embryology.'' However, it is only recently that more complete propos-

als have been made to use evolutionary approaches in robotics (Brooks, 1992; Husbands & Harvey, 1992).

Brooks (1992) outlines an approach based on Koza's genetic programming techniques (Koza, 1992). He acknowledged that time constraints would probably necessitate the use of simulations. However, he stressed the dangers of using simulated worlds rather than real worlds. He proposed that by evolving the control program incrementally the search space can be kept small at any time. He noted that symmetries or repeated structures should be exploited so that only a single module needs to be evolved, which is then repeatedly used. Brooks proposed a high-level language, GEN, which could be evolved, and then compiled down into BL (Behavior Language) and further on down onto the actual robot hardware.

Important work on an evolutionary approach to agent control using neural networks has been done by Beer and Gallagher (1992). They explore the evolution of continuous-time recurrent neural networks as a mechanism for adaptive agent control, using as example tasks chemotaxis, and locomotion-control for a six-legged insect-like agent. The networks are based on the continuous Hopfield model (Hopfield, 1982), but allow arbitrary recurrent connections. They used a standard genetic algorithm to determine neuron time constants and thresholds and connection weights. A fixed number of network parameters are encoded in a straightforward way on bitstring ''genotypes.'' They report success in their objectives; in the case of locomotion control, controllers were evolved that in practice generated a tripod gait (front and back legs on one side in phase with the middle leg on the opposite side). This was achieved both with and without the use of sensors which measured the angular position of each leg.

Beer and Gallagher (1992) develop a dynamical systems perspective on control systems for autonomous agents, influenced by early work in Cybernetics (Ashby, 1952). In further developments of their evolutionary approach, Yamauchi and Beer (1994) evolve networks which can control autonomous agents in tasks requiring sequential and learning behavior.

Colombetti and Dorigo (1992) use Classifier Systems (CSs) for robot control. In this work the ALECSYS implementation is used to build a hierarchical architecture of CSs—one for each desired behavior, plus a coordinating CS. Results are reported which have been generated in simulations and then transferred to a real robot.

Parisi, Nolfi, and Cecconi (1992) investigated the relationship between learning and evolution in populations of back-propagation networks; these networks were the ''brains'' of animals that received sensory input from a simple cellular world in which the task was to collect ''food.'' This work made use of abstract computer models rather than real robots.

Koza successfully used the technique of genetic programming to develop subsumption architectures (Brooks, 1986) for simulated robots engaged in wall-following and box-moving tasks (Koza, 1992).

Craig Reynolds (1993) uses genetic programming to create control programs which enable a simple simulated moving vehicle to avoid collisions. He comments that these solutions are brittle, vulnerable to any slight changes or to noise. In further work where the fitness-testing includes noise, he reports that the brittleness problem is overcome, and only compact robust solutions survive (Reynolds, 1994).

Floreano and Mondada (1994) were able to run a GA on a real robot in real time, rather than a simulation. The GA set the weights and thresholds in a simple recurrent network where every sensory input was connected to both motor outputs. The task was to traverse a circular corridor while avoiding obstacles, and this work demonstrates that with well-designed equipment it is possible to avoid the problems associated with simulations.

9. SOME COMMON OBJECTIONS

One common objection to the use of artificial evolution is the amount of time it is likely to take to evolve anything useful. This is difficult to answer. However, work done so far has shown that it is possible to evolve simple control systems in simulation in a matter of 2 or 3 hr (Jakobi et al., 1995) and in the real world in about 1 day (Harvey et al., 1994). It is too early to say how things will scale up as more complex tasks are used.

Another complaint is that the entire morphology of the robot, as well as its control system, should be evolved. This is a valid criticism. Successful adaptive behavior depends on harmonious relationships between body morphology and nervous system dynamics. However, some progress is being made in this direction in the work of Harvey, Husbands, and others where the visual morphology of a real robot is concurrently evolved along with a control network (Harvey et al., 1994). As described earlier, this is done by allowing the subsampling pattern (position and size of receptive fields) of a video camera image to be under evolutionary control (see Section 5).

Sometimes it is stated that as more complex tasks are investigated, it will become extremely difficult to design evaluation functions. Opinion is divided over this issue. One of the implicit assumption of the field is that it is generally much easier to produce a criteria for deciding how well a robot achieved a task than it is to specify how the task should be achieved. Evaluation functions can be very implicit. For instance, tasks such as exploration and foraging can be set up as straight survival tests. Only those robots that maintain viability for sufficiently long get a chance to breed. Maintaining viability will involve finding and exploiting energy sources. However, issues relating to evaluation, both explicit and implicit, are likely to become increasingly important as attempts are made to evolve more complex behaviors.

Finally, a common assumption is that the evolved systems will be impossible to understand. There are two answers to this. The first is that there is no evidence to suggest that the systems will be impenetrable. The second is that

even if they are, so what. One of the aims of this kind of work is to develop mechanism to generate certain sorts of behaviors and then to gain insight by analyzing these mechanisms. Both Beer and Gallagher (1992) and Husbands et al. (1995) have used dynamical systems approaches to analyzing evolved controllers. They have both demonstrated ways to choose a state space at the appropriate level to allow a clear picture of the internal mechanisms. Husbands et al. also show how to incorporate an understanding of agent–environment coupling into the analysis. So far these kinds of analysis have been for relatively simple systems. It is possible that as things become more complex, state spaces at a tractable level of abstraction will not be found. However, if the robot works, and is robust and reliable, an important aspect of the research would still be successful.

10. EVOLUTIONARY SIMULATIONS AS SCIENCE: TRACING THE ORIGINS AND EFFECTS OF SENSORIMOTOR SYSTEMS IN NATURE

While much of the work mentioned so far is biologically informed and inspired, most of it has a strong engineering characteristic. In other words, the primary goal is to develop working control systems for autonomous mobile robots and then to understand their underlying mechanisms. However, this field can potentially offer new tools and methods for investigating more specifically scientific topics. That is the focus of this section, where the use of artificial evolution in the context of more abstract computer studies will be discussed.

Very little is known about the evolutionary origins and effects of basic sensorimotor systems in nature. Brains and behaviors do not fossilize well, so normal paleontological methods cannot generally be used to trace the evolution of sensorimotor systems. Behavioral ecologists can construct optimality or game theoretic models of how behavioral strategies evolve, but these models are usually too abstract to explain the evolution of specific sensorimotor systems in specific species. Even experimental studies in fast-breeding species cannot study sensorimotor evolution for more than a few dozen generations. Neuroethologists can derive phylogenies and probable selective pressures by comparing sensorimotor adaptations across species, but cannot test evolutionary hypotheses very directly. Because of these methodological problems, evolutionary computer simulations are our only real hope for understanding the long-term adaptation of sensorimotor systems to habitats and econiches and the long-term coevolution of sensorimotor systems interacting within and between species.

This gap in our scientific understanding of sensorimotor evolution is important because (1) sensorimotor control is the essence of ‘‘adaptive agency,’’ and the evolution of sensorimotor control is fundamental to the success of all animal species, and (2) sensorimotor systems, once evolved,

can in turn exert strong selection pressures on other organisms, resulting in the evolution of camouflage, warning coloration, mimicry, lures, protean behavior, sexual displays, communication, and many other forms of adaptive display. This second phenomenon has received increasing attention in the last few years and has been termed ''psychological selection'' (Miller, 1993; Miller & Freyd, 1993), ''sensory drive'' (Endler, 1992), ''sensory exploitation'' (Ryan, 1990), ''signal selection'' (Zahavi, 1991), and ''the influence of receiver psychology on the evolution of animal signals'' (Guilford & Dawkins, 1991). In such cases of ''sensory exploitation,'' where behavioral adaptations in one animal evolve to exploit particular sensory biases in other animals, we clearly cannot understand the co-evolution without simulating the relevant sensorimotor systems in some detail.

Genetic algorithms offer a general, openended method for simulating the evolutionary origins and effects of sensorimotor systems, because such systems can be modeled at almost any size and any level of description, from detailed neural network designs (as we have used in our evolutionary robotics work), up to abstract parameters of behavioral strategies, and because such systems can be left to evolve in any simulatable habitat or ecosystem. Since different scientific problems require simulations at quite different scales and levels of description, we must be explicit about our research goals and careful about finding the right simulation methods for those goals. For example, studying the phylogeny of visual circuits in a particular genus of beetle might require evolving quite detailed neural networks under particular ecological conditions, but the studying the general influence of visual associative learning on the evolution of warning coloration might require much more general models of vision in predators and coloration in prey. In general, engineering research needs more detailed, lower-level simulations of sensorimotor systems than almost any scientific research would require, because sensorimotor systems for autonomous robots must actually work, whereas sensorimotor models of animals need only fit the neuroethological data.

Even if one's scientific goal is to understand neural development, learning, perception, or the mechanisms of dynamic behavior, rather than evolution itself, there is still considerable benefit to parameterizing one's model of the phenomenon in a way that allows alternative models to evolve through GA methods. Simulated evolution can be used to test the plausibility, robustness, and evolutionary stability of models of development and behavior just as real evolution tested the actual mechanisms of development and behavior that are being modeled. Human imagination is poor at envisioning alternatives to one's cherished model of some behavioral phenomenon; simulated evolution can act as a constructive critic that generates alternative hypotheses which can then be tested by observation and experimentation.

In the future, we envision a more integrated science of sensorimotor evolution that combines data and methods from cladistics, experimental psychology, neuroethology, behavioral ecology, population genetics, and computer

simulation. Evolutionary simulation is unusually exciting, colorful, and fast as an empirical research method, but ideally, it will be absorbed into the scientific mainstream as just one means among many for studying natural evolutionary processes.

11. CONCLUSIONS

Using artificial evolution as a new approach to AI has been discussed. More specifically it has been advocated for the development of network-based controllers for autonomous robots, autonomous robotics being seen as the most appropriate vehicle for AI research. It has been argued that it has at least the following advantages.

- It allows the exploration of areas of design space that are not amenable to traditional rational analytic hand-design methods.
- It allows the concurrent exploration of control architectures and sensor morphologies in a principled way.
- It enables us to work with very low-level primitives and helps to throw away as many preconceptions as possible about how behaviors should be generated.

Some discussion of the role of artificial evolution in more abstract studies of potential benefit to theoretical biology has also been given.

REFERENCES

Ashby, W. R. 1952. *Design for a brain.* London: Chapman & Hall.

Barhen, J., Dress, W. B., & Jorgensen, C. C. 1987. Applications of concurrent neuromorphic algorithms for autonomous robots. In R. Eckmiller and C.v.d. Malsburg (Eds.), *Neural computers.* New York/Berlin: Springer–Verlag. Pp. 321–333.

Beer, R. 1990. *Intelligence as adaptive behaviour: An experiment in computational neuro-ethology.* New York: Academic Press.

Beer, R., & Gallagher, J. C. 1992. Evolving dynamic neural networks for adaptive behavior. *Adaptive Behavior,* **1**(1), 91–122.

Belew, R., & Booker, L. 1991. *Proceedings of the 4th International Conference on Genetic Algorithms.* Morgan Kaufmann.

Boden, M. 1977. *AI and natural man.* Brighton: Harvester Press.

Brooks, R. A. 1986. A robust layered control system for a mobile robot. *IEEE J. Rob. Autom.,* **2,** 14–23.

Brooks, R. A. 1991. Intelligence without representation. *Artificial Intelligence,* **47,** 139–159.

Brooks, R. A. 1992. Artificial life and real robots. In F. J. Varela & P. Bourgine (Eds.), *Proceedings of the First European Conference on Artificial Life.* Cambridge, MA: MIT Press/Bradford Books. Pp. 3–10.

Cliff, D., Harvey, I., & Husbands, P. 1993. Explorations in evolutionary robotics. *Adaptive Behavior,* **2**(1), 73–110.

Colombetti, M., & Dorigo, M. 1992. Learning to control an autonomous robot by distributed genetic algorithms. In J.-A. Meyer, H. Roitblat, S. Wilson (Eds.), *from animals to animats 2, Proc. of 2nd Intl. Conf. on Simulation of Adaptive Behavior, SAB'92.* Cambridge, MA: MIT Press/Bradford Books. Pp. 305–312.

Davis, L. 1990. *The handbook of genetic algorithms.* Princeton, NJ: Van Nostrand–Reinhold.

de Garis, H. 1992. The genetic programming of steerable behaviors in genNets. Toward a

practice of autonomous systems. In F. J. Varela & P. Bourgine (Eds.), *Proceedings of the First European Conference on Artificial Life.* Cambridge, MA: MIT Press/Bradford Books. Pp. 272–281.

Endler, J. A. 1992. Signals, signal conditions, and the direction of evolution. *American Naturalist,* **139,** 125–153.

Ewert, J.-P. 1980. *Neuroethology.* New York/Berlin: Springer-Verlag.

Floreano, D., & F. Mondada. 1994. Automatic creation of an autonomous agent: Genetic evolution of a neural-network driven robot. In D. Cliff, P. Husbands, J.-A. Meyer, & S. Wilson (Eds.), *From animals to animats 3, Proc. of 3rd Intl. Conf. on Simulation of Adaptive Behavior, SAB'94.* Cambridge, MA: MIT Press/Bradford Books.

Goldberg, D. E. 1989. *Genetic Algorithms in search, optimization and machine learning,* Reading, MA: Addison–Wesley.

Gruau, F. 1992. *Cellular encoding of genetic neural networks.* TR.92-21, Laboratoire de l'Informatique du Parallelisme, Ecole Normale Superieure de Lyon.

Grefenstette, J. 1985. *Proceedings of an International Conference on GAs.* Hillsdale, NJ: Erlbaum.

Grefenstette, J. 1987. *Proceedings of the 2nd International Conference on GAs.* Hillsdale, NJ: Erlbaum.

Guilford, T., & Dawkins, M. S. 1991. Receiver psychology and the evolution of animal signals. *Animal Behavior,* **42,** 1–14.

Harp, S. A., & Samad, T. 1992. Genetic synthesis of neural network architecture. In L. Davis (Ed.), *Handbook of genetic algorithms.* Princeton, NJ: Van Nostrand–Reinhold. Pp. 202–221.

Harvey, I. 1992a. The SAGA cross: The mechanics of crossover for variable-length genetic algorithms. In R. Manner & B. Manderick (Eds.), *Proc. PPSN 2.* Amsterdam: North Holland. Pp. 269–278.

Harvey, I. 1992b. Species adaptation genetic algorithms: The basis for a continuing SAGA. In F.J. Varela & P. Bourgine (Eds.), *Proceedings of the First European Conference on Artificial Life.* Cambridge, MA: MIT Press/Bradford Books. Pp. 346–354.

Harvey, I. 1994. Evolutionary robotics and SAGA: The case for hill crawling and tournament selection. In C. Langton (Ed.), *Artificial life III.* Redwood City, CA: Addison–Wesley. Pp. 299–326. Santa Fe Institute Studies in the Sciences of Complexity. Proceedings Vol. XVI.

Harvey, I., Husbands, P., & Cliff, D. 1994. Seeing the light: Artificial evolution, real vision. In D. Cliff, P. Husbands, J.-A. Meyer, & S. Wilson (Eds.), *From animals to animats 3, Proc. of 3rd Intl. Conf. on Simulation of Adaptive Behavior, SAB'94.* Cambridge; MA: MIT Press/Bradford Books. Pp. 392–401.

Holland, J. 1975. *Adaptation in natural and artificial systems.* Ann Arbor, MI: University of Michigan Press.

Hopfield, J. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences,* **79,** 2554–2558.

Husbands, P. 1996. Rectangles, robots, triangles and transients. In M. Sugisaka (Ed.). *Proceedings of International Symposium on Artificial Life and Robotics, Oita, Japan.* Pp. 252–255.

Husbands, P., & Harvey, I. 1992. Evolution versus design: Controlling autonomous robots, integrating perception, planning and action. *Proceedings of 3rd Annual Conference on Artificial Intelligence, Simulation and Planning.* New York: IEEE Press. Pp. 139–146.

Husbands, P., Harvey, I., Cliff, D., & Miller, G. 1994. The use of genetic algorithms for the development of sensorimotor control systems. In P. Gaussier, & J.-D. Nicoud (Eds.), *Proceedings of from Perception to Action Conference.* Washington, DC: IEEE Comput. Soc. Pp. 110–121.

Husbands, P., Harvey, I., & Cliff, D. 1995. Circle in the round: State space attractors for evolved sighted robots. *Robotics and Autonomous Systems,* **15,** 83–106.

Jakobi, N., Husbands, P., & Harvey, I. 1995. Noise and the reality gap: The use of simulation in evolutionary robotics. In F. Moran, A. Moreno, J. J. Merelo, & P. Chacon (Eds.). *Advances in Artificial Life: Proc. 3rd European Conference on Artificial Life.* Berlin: Springer–Verlag. Lecture Notes in Artificial Intelligence. Vol. 929. Pp. 704–720.

Kitano, H. 1990. Designing neural networks using genetic algorithms with graph generation system. *Complex Systems,* **4,** 461–476.

Koza, J. 1992. *Genetic programming: On the programming of computers by means of natural selection.* Cambridge, MA: MIT Press.

Miller, G. F. 1993. *Evolution of the human brain through runaway sexual selection: The mind as a protean courtship device.* Ph.D. thesis, Stanford University Psychology Department.

Miller, G. F., & Cliff, D. 1994. Protean behavior in dynamic games: Arguments for the co-evolution of pursuit-evasion tactics in simulated robots. *From animals to animats 3, Proceedings of the Third International Conference on Simulation of Adaptive Behavior.* Cambridge, MA: MIT Press.

Miller, G. F., & Freyd, J. J. 1993. *Dynamic mental representations of animate motion: The interplay among evolutionary, cognitive, and behavioral dynamics.* Cognitive Science Research Paper CSRP-290; University of Sussex.

Mitchell, M. 1996. *An introduction to genetic algorithms.* Cambridge, MA: MIT Press/Bradford Books.

Moravec, H. 1983. The stanford cart and the CMU rover. *Proceedings of the IEEE,* **71,** 872–884.

Parisi, D., Nolfi, S., & Cecconi, F. 1992. Learning, behavior, and evolution. Toward a practice of autonomous systems. In F. J. Varela & P. Bourgine (Eds.), *Proceedings of the First European Conference on Artificial Life.* Cambridge, MA: MIT Press/Bradford Books. Pp. 207–216.

Reynolds, C. 1993. *An evolved, vision-based model of obstacle avoidance behavior. Artificial life III.* C. Langton (Ed.), Santa Fe Institute Studies in the Sciences of Complexity, Proceedings Vol. XVI. Reading, MA: Addison–Wesley.

Reynolds, C. 1994. Evolution of corridor following behavior in a noisy world. In D. Cliff, P. Husbands, J.-A. Meyer, & S. Wilson (Eds.), *From animals to animats 3, Proc. of 3rd Intl. Conf. on Simulation of Adaptive Behavior, SAB'94.* Cambridge, MA: MIT Press/Bradford Books.

Ryan, M. J. 1990. Sexual selection, sensory systems, and sensory exploitation. *Oxford Surveys of Evol. Biology,* **7,** 156–195.

Schaffer, J. 1989. *Proceedings of the 3rd International Conference on GAs.* Los Altos, CA: Kaufmann.

Sloman, A. 1993. The mind as a control system. Inc. Cambridge: Hookway & D. Peterson (Eds.), *Philosophy and the cognitive sciences.* CUP.

Smithers, T. 1994. On why better robots make it harder. In D. Cliff, P. Husbands, J.-A. Meyer, & S. Wilson (Eds.), *From animals to animats 3, Proc. of 3rd Intl. Conf. on Simulation of Adaptive Behavior, SAB'94.* Cambridge, MA: MIT Press/Bradford Books. Pp. 54–72.

Thompson, A. 1995. Evolving electronic robot controllers that exploit hardware resources. In F. Moran, A. Moreno, J. J. Merelo, & P. Chacon (Eds.), *Advances in Artificial Life: Proc. 3rd European Conference on Artificial Life.* Berlin: Springer–Verlag. Lecture Notes in Artificial Intelligence. Vol. 929. Pp. 640–656.

Varela, F., Thompson, E., & Rosch, E. 1991. *The embodied mind.* Cambridge, MA: MIT Press.

Viola, P. 1988. *Mobile robot evolution.* Bachelors Thesis, MIT.

Wilson, S. 1987. The genetic algorithm and biological development. In J. J. Grefenstette (Ed.), *Genetic algorithms and their applications: Proceedings of the Second Intl. Conf. on Genetic Algorithms.* Hillsdale, NJ: Erlbaum. Pp. 247–251.

Wilson, S. 1991. The animat path to AI. In J.-A. Meyer & S. Wilson (Eds.), *From Animals to Animats.* Cambridge, MA: MIT Press.

Yamauchi, B., & Beer, R. 1994. Integrating reactive, sequential, and learning behavior using dynamical neural networks. In D. Cliff, P. Husbands, J.-A. Meyer & S. Wilson (Eds.), *From animals to animats 3, Proc. of 3rd Intl. Conf. on Simulation of Adaptive Behavior, SAB'94*. Cambridge, MA: MIT Press/Bradford Books. Pp. 382–391.

Young, D. 1989. *Nerve cells and animal behaviour*. Cambridge: CUP.

Zahavi, A. 1991. On the definition of sexual selection, Fisher's model, and the evolution of waste and of signals in general. *Animal Behaviour*, **42**(3), 501–503.