The Use of Genetic Algorithms for the Development of Sensorimotor Control Systems

Philip Husbands and Inman Harvey and Dave Cliff and Geoffrey Miller

School of Cognitive and Computing Sciences
University of Sussex
Brighton BN1 9QH, England
E-mail: philh or inmanh or davec or geoffm @cogs.susx.ac.uk

Abstract

This paper provides a high-level review of current and recent work in the use of genetic algorithm based techniques to develop sensorimotor control systems for autonomous agents. It focuses on network-based controllers and genetic encoding issues associated with them. The paper closes with a discussion of the possibility of using artificial evolutionary techniques to help tackle more specifically scientific questions about natural sensorimotor systems.

Keywords

Genetic Algorithms – Neural Networks – Sensorimotor Control Systems

1 Introduction

Animals that we often think of as rather simple (e.g. arthropods[1]) in fact display a range of sophisticated adaptive behaviours, involving complex sensorimotor coordination [Young89]. These behaviours 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 [Ewert80]. 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 behaviour in the real world is no easy matter [Moravec83]. 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 '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 [Brooks91, Beer90].

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 [Cliff93, Beer92b]. A number of research projects are now actively exploring this possibility.

[1] That class of invertebrates including insects.

The artificial evolution approach maintains a population of viable genotypes (chromosomes), coding for control architectures. The genotypes are inter-bred 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 favoured 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 behaviour, as long as this behaviour is directly or implicitly included in the evaluation function.

This paper starts with a high-level review of current work in the use of genetic algorithm based techniques to develop sensorimotor control systems for autonomous agents. It goes on to argue that the most promising way of doing this involves the use of network-based controllers. This in turn means that issues to do with the genetic encoding of the networks become central to the endeavour. 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. There follows a discussion of the possibility of using artificial evolutionary techniques to help tackle more specifically scientific questions about natural sensorimotor systems.

2 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 [Barhen87]. A student of Brooks discussed some of the issues involved, with reference to subsumption architectures, in [Viola88]. De Garis [Garis92] proposed using GAs for building behavioural modules for artificial nervous systems, or 'artificial embryology'. However, it is only recently that more complete proposals have been made to use evolutionary approaches in robotics [Brooks92, Husbands92].

Brooks [Brooks92] outlines an approach based on Koza's Genetic Programming techniques [Koza92]. 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 (Behaviour Language) and further on down onto the actual robot harware.

Important work on an evolutionary approach to agent control using neural networks has been done by Beer and Gallagher [Beer92b]. 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 [Hopfield82], 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 [Beer92a] develops a dynamical systems perspective on control systems for autonomous

agents, influenced by early work in Cybernetics [ASHBY60]. In further developments of their evolutionary approach, Yamauchi and Beer [YAMAUCHI94] evolve networks which can control autonomous agents in tasks requiring sequential and learning behaviour.

Colombetti and Dorigo [COLOMBETTI92] 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 behaviour, plus a coordinating CS. Results are reported which have been generated in simulations, and then transferred to a real robot.

Parisi, Nolfi and Cecconi [PARISI92] investigated the relationship between learning and evolution in populations of back-propagation networks; these networks were the 'brains' of animats that received sensory input from a simple cellular world in which the task was to collect 'food'. In this paper they concluded that learning abilities can accelerate the evolutionary process, even when random learning tasks are used. A later analysis of this work [WILLIAMS93] demonstrated that this conclusion was not justified, and an alternative simpler explanation was given.

Koza successfully used the technique of Genetic Programming to develop subsumption architectures [BROOKS86] for simulated robots engaged in wall-following and box moving tasks [KOZA92].

Craig Reynolds [REYNOLDS94B] 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 [REYNOLDS94A].

Floreano and Mondada [FLOREANO94], 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.

Cliff, Harvey and Husbands have developed an evolutionary methodology which they have successfully applied to the development of control systems for both simulated agents and real robots [CLIFF93, HARVEY94]. The *incremental* evolution of arbitrarily recurrent neural networks is advocated, as is the concurrent evolution of the control network *and* aspects of the agent's morphology, particularly the positions and properties of sensors. In the simulation work, agents were evolved which, from any starting position and orientation, could reliably use vision to move to the centre of a circular room and stay there. This work involved the concurrent evolution of control networks and the position and properties of two visual sensors. The simple genetic encoding used (essentially a direct representation of the network wiring) was different from most others in that it was dynamic in length. This meant that arbitrary numbers of neurons and inter-neuron connections could be encoded. In [HUSBANDSNG] it is shown how these evolved control systems can be analysed in some detail. However, it was concluded that further use of simulation was highly problematic if more complex visual environments were to be used. To overcome this, a specialised piece of visuo-robotic equipment was developed that allows the evolution of control networks and visual morphologies without recourse to simulating the agent environment interactions. The equipment is fully described in [HARVEY94] and has been successfully used to evolve simple target approaching and following behaviours. It has also been used to develop agents which are able to distinguish between two differently shaped targets. In both the simulation work and the experiments with real robots, each individual was evaluated several times and the *worst* score obtained given as its fitness. This encouraged

robustness.

3 Why Evolve Networks?

Much of the work described in the previous section used 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 [CLIFF93], 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 behaviour 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 the front runner.

4 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 be made. In the simplest schemes the genotype is a direct description of the network wiring. 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[2]. 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 [GRUAU92] defines 7 properties of a 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 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.

[2] 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.

Kitano [KITANO90] 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 4 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 [GRUAU92] discusses Kitano's work, and also acknowledges earlier work by Wilson [WILSON87]. Gruau's Cellular Encoding (CE) 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.

5 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.

5.1 A language and compiler scheme

Experience with the primitive encoding we used in our early evolutionary robotics simulation studies [CLIFF93] 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 endeavour.

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. For a more complete description, including accounts

of its successful use in the evolution of autonomous agents for a variety of behavioral niches, see [CLIFF94].

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 conceptualised 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 localised 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 non-existent. 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: for further details, see [CLIFF94].

5.2 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[3]. Full details of this method can be found in [HUSBANDS94], which describes a class of developmental schemes. A representative member of this class is presented here.

In this force field scheme, 'neurons' are positioned across a 2D 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 Equation 1.

$$\frac{d^2\vec{R}_{ij}}{dt^2} = \sum_{k=0}^{N} \vec{F}_{ijk} - K\frac{d\vec{R}_{ij}}{dt} + \frac{G_{ij}}{l_{ij}^3 \left|\frac{d\vec{R}_{ij}}{dt}\right|}\frac{d\vec{R}_{ij}}{dt} \quad (1)$$

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 Equation 1 represents the vector sum of the attractive forces exerted on end_{ij} by all neurons. These forces are of the form given in Equation 2.

$$\vec{F}_{ijk} = \frac{GS_i AS_k \vec{r}_{ijk}}{|\vec{r}_{ijk}|^3} \quad (2)$$

Where $\vec{r}_{ijk}$ is the vector from end_{ij} to the centre of neuron k. GS_i and AS_k are genetically determined constants. The second term in Equation 1 is a 'viscous' resistive force to prevent dendrites sailing off into outer-space. 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 5. A particular encoding meeting these requirements, and specially developed for the force field method, is briefly outlined below. Further details can be found in [HUSBANDS94].

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'

[3]Recently [FLEISCHER94] has independently done related work, although their research is much more focused on biological modelling of cellular development and has not yet been incorporated into an evolutionary framework.

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' behaviour 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 localised on the string. This produces a form of modularity, where repeated patterns are formed by the re-expression of parts of the genotype. A more complex modular extension, involving several planes of neurons, is described in [HUSBANDS94].

The position of a neuron is described by genetically determined distance and direction from the last neuron to be layed down. The existence or otherwise 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 centre 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.

5.3 A cell-division method

In this proposal [HARVEY93], 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 intra-cellular 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 intra-cellular dynamics can be influenced by inter-cellular 'signals' received from neighbouring 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 neighbourhood 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 organisation; for instance, as a neural network with different cells being 'neurons' with specific characteristics, and with connections specified between them.

6 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. However, this field can potentially offer new tools and methods for investigating more specifically scientific topics. That is the focus of this section.

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, among other things, 'psychological selection' [MILLER93A, MILLER93B] and 'sensory exploitation' [RYAN90]. 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, open-ended method for simulating the evolutionary origins and effects of sensorimotor systems, because such systems can be modelled 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 modelled. 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.

7 Conclusions

This paper has given a high-level review of current and recent work in the use of genetic algorithm based techniques for the development of sensorimotor control systems for autonomous agents. It has focused on genetic encoding issues as being particularly important, and has presented work in this area. The paper closed with a discussion of the wider scientific applicability of artificial evolutionary techniques.

References

[Ashby60] W. Ross Ashby. *Design for a Brain.* Chapman, 1960.

[Barhen87] J. Barhen, W.B. Dress, and C.C. Jorgensen. Applications of concurrent neuromorphic algorithms for autonomous robots. In R. Eckmiller and C.v.d. Malsburg, editors, *Neural Computers*, pages 321–333. Springer-Verlag, 1987.

[Beer90] R. D. Beer. *Intelligence as Adaptive Behaviour: An Experiment in Computational Neuroethology.* Academic Press, 1990.

[Beer92a] R.D. Beer. A dynamical systems perspective on autonomous agents. Technical Report CES-92-11, Case Western Reserve University, Cleveland, Ohio, 1992.

[Beer92b] R.D. Beer and J.C. Gallagher. Evolving dynamic neural networks for adaptive behavior. *Adaptive Behavior*, 1(1):91–122, 1992.

[Brooks86] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE J. Rob. Autom.*, 2:14–23, 1986.

[Brooks91] R.A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47:139–159, 1991.

[Brooks92] Rodney A. Brooks. Artificial life and real robots. In F. J. Varela and P. Bourgine, editors, *Proceedings of the First European Conference on Artificial Life*, pages 3–10. MIT Press/Bradford Books, Cambridge, MA, 1992.

[Cliff93] D. Cliff, I. Harvey, and P. Husbands. Explorations in evolutionary robotics. *Adaptive Behavior*, 2(1):73–110, 1993.

[Cliff94] D. Cliff. Ncage: Network control architecture genetic encoding. Technical Report CSRP-325, School of Cognitive and Computing Sciences, University of Sussex, 1994.

[Colombetti92] M. Colombetti and M. Dorigo. Learning to control an autonomouys robot by distributed genetic algorithms. In J.-A. Meyer, H. Roitblat, and S. Wilson, editors, *From Animals to Animats 2, Proc. of 2nd Intl. Conf. on Simulation of Adaptive Behavior, SAB'92*, pages 305–312. MIT Press/Bradford Books, 1992.

[EWERT80] J.-P. Ewert. *Neuroethology.* Springer-Verlag, 1980.

[FLEISCHER94] K. Fleischer. A simulation testbed for the study of multicellular development: The multiple mechanisms of morphogenesis. In C. Langton, editor, *Artificial Life III*, pages 389–416. Santa Fe Institute Studies in the Sciences of Complexity, Proceedings Vol. XVI, Addison-Wesley, Redwood City CA, 1994.

[FLOREANO94] D. Floreano and F. Mondada. Automatic creation of an autonomous agent: Genetic evolution of a neural-network driven robot. In D. Cliff, P. Husbands, J.-A. Meyer, and S. Wilson, editors, *From Animals to Animats 3, Proc. of 3rd Intl. Conf. on Simulation of Adaptive Behavior, SAB'94.* MIT Press/Bradford Books, 1994.

[GARIS92] Hugo de Garis. The genetic programming of steerable behaviors in GenNets. In F. J. Varela and P. Bourgine, editors, *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, pages 272–281. MIT Press/Bradford Books, Cambridge, MA, 1992.

[GRUAU92] F. Gruau. Cellular encoding of genetic neural networks. Technical Report 92-21, Laboratoire de l'Informatique du Parallelisme, Ecole Normale Superieure de Lyon, 1992.

[HARVEY93] I. Harvey. *The Artificial Evolution of Adaptive Behaviour.* PhD thesis, University of Sussex, 1993.

[HARVEY94] I. Harvey, P. Husbands, and D. Cliff. Seeing the light: Artificial evolution, real vision. In D. Cliff, P. Husbands, J.-A. Meyer, and S. Wilson, editors, *From Animals to Animats 3, Proc. of 3rd Intl. Conf. on Simulation of Adaptive Behavior, SAB'94.* MIT Press/Bradford Books, 1994.

[HOPFIELD82] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79:2554–2558, 1982.

[HUSBANDS92] P. Husbands and I. Harvey. Evolution versus design: Controlling autonomous robots. In *Integrating Perception, Planning and Action, Proceedings of 3rd Annual Conference on Artificial Intelligence, Simulation and Planning*, pages 139–146. IEEE Press, 1992.

[HUSBANDS94] P. Husbands. A force field development scheme for use with genetic encodings of network-based sensorimotor control systems. Technical Report CSRP-326, School of Cognitive and Computing Sciences, University of Sussex, 1994.

[HUSBANDSNG] P. Husbands, I. Harvey, and D. Cliff. Circle in the round: state space attractors for evolved sighted robots. *Robotics and Autonomous Systems*, Forthcoming.

[KITANO90] H. Kitano. Designing neural networks using genetic algorithms with graph generation system. *Complex Systems*, 4:461–476, 1990.

[KOZA92] J. Koza. *Genetic Programming: On the programming of computers by means of natural selection.* MIT Press, 1992.

[MILLER93A] G. F. Miller. *Evolution of the human brain through runaway sexual selection: The mind as a protean courtship device.* PhD thesis, Department of Psychology, Stanford University, 1993.

[MILLER93B] G. F. Miller and J. J. Freyd. Dynamic mental representations of animate motion: The interplay among evolutionary, cognitive, and behavioral dynamics. Technical Report CSRP-290, School of Cognitive and Computing Sciences, University of Sussex, 1993.

[MORAVEC83] H.P. Moravec. The Stanford Cart and the CMU Rover. In *Proc. of IEEE*, volume 71, pages 872–884, 1983.

[PARISI92] D. Parisi, S. Nolfi, and F. Cecconi. Learning, behavior, and evolution. In F. J. Varela and P. Bourgine, editors, *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, pages 207–216. MIT Press/Bradford Books, Cambridge, MA, 1992.

[REYNOLDS94A] C.W. Reynolds. Evolution of corridor following behavior in a noisy world. In D. Cliff, P. Husbands, J.-A. Meyer, and S. Wilson, editors, *From Animals to Animats 3, Proc. of 3rd Intl. Conf. on Simulation of Adaptive Behavior, SAB'94.* MIT Press/Bradford Books, 1994.

[REYNOLDS94B] C.W. Reynolds. An evolved, vision-based model of obstacle avoidance behavior. In C. Langton, editor, *Artificial Life III.* Santa Fe Institute Studies in the Sciences of Complexity, Proceedings Vol. XVI, Addison-Wesley, Redwood City CA, 1994.

[RYAN90] M. J. Ryan. Sexual selection, sensory systems, and sensory exploitation. *Oxford Surveys of Evol. Biology*, 7:156–195, 1990.

[VIOLA88] P. Viola. Mobile robot evolution. Bachelors thesis, M.I.T., 1988.

[WILLIAMS93] B.V. Williams and D.G. Bounds. Learning and evolution in populations of backprop networks. In *Proceedings of the Second European Conference on Artificial Life*, pages 1139–1149. Preprints, 1993.

[WILSON87] S. Wilson. The genetic algorithm and biological development. In J. J. Grefenstette, editor, *Genetic Algorithms snd their Applications: Proceedings of the Second Intl. Conf. on Genetic Algorithms*, pages 247–251, Hillsdale NJ, 1987. Lawrence Erlbaum Associates.

[YAMAUCHI94] B. Yamauchi and R. Beer. Integrating reactive, sequential, and learning behavior using dynamical neural networks. In D. Cliff, P. Husbands, J.-A. Meyer, and S. Wilson, editors, *From Animals to Animats 3, Proc. of 3rd Intl. Conf. on Simulation of Adaptive Behavior, SAB'94.* MIT Press/Bradford Books, 1994.

[YOUNG89] D. Young. *Nerve cells and animal behaviour.* Cambridge, 1989.